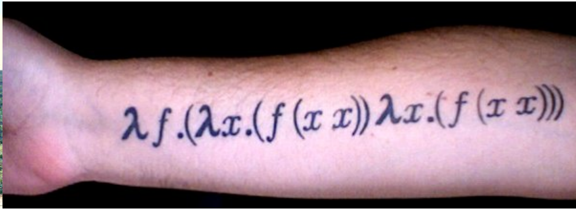
The Lambda Calculus

Stephen A. Edwards

Columbia University

Spring 2023



The Lambda Calculus

$$
\begin{array}{lcll}
x & ::= & \text{a variable: a character or string} & \\
e & ::= & x & \text{Variable} \\
 & & (\lambda x \;.\; e) & \text{Function Abstraction} \\
 & & (e\ e) & \text{Function Application}
\end{array}
$$

The Lambda Calculus

x	$::=$	a variable: a character or string	
e	$::=$	x	Variable
		$(\lambda x \;.\; e)$	Function Abstraction
		$(e\;e)$	Function Application

x	$::=$	a variable: a character or string	
e	$::=$	x	Variable
		$\lambda x \;.\; e$	Function Abstraction
		$e\;e$	Function Application
		(e)	Parentheses

Application binds left-to-right, more strongly than abstraction

$$\lambda x\,.\,\lambda y\,.\,f\;g\;x \quad \text{means} \quad (\lambda x\,.\,(\lambda y\,.\,((f\;g)\;x)))$$

NOTATIONS

DE

LOGIQUE MATHÉMATIQUE

PAR

G. PEANO

Professeur d'Analyse infinitésimale à l'Université de Turin.

INTRODUCTION

AU

FORMULAIRE DE MATHÉMATIQUE

publié par la

« Rivista di Matematica »

TURIN

—

1894

Peano, 1894

For reducing parenthetical clutter

$a\,.\,bc$, $ab\,.\,c$, $ab\,.\,cd$

signifient $a\,(bc)$, $(ab)\,c$, $(ab)\,(cd)$,

et la formule

$ab\,.\,cd :\, e\,.\,fg \,.\!\because\, hk\,.\,l$

est équivalente à la

$\{[(ab)\,(cd)]\,[e\,(fg)]\}\,[(hk)\,l]$,

qu'on peut aussi écrire avec les *vinculums:*

$\underline{\underline{\underline{ab}\;\underline{cd}}\;\underline{e\;\underline{fg}}}\;\underline{\underline{hk}\;l}.$

Vinculum: A line indicating grouping, e.g., $\sqrt{3 + 6}$

Like Haskell's $ operator: `a b $ c d = (a b)(c d)`

PRINCIPIA MATHEMATICA

BY

ALFRED NORTH WHITEHEAD, Sc.D., F.R.S.
Fellow and late Lecturer of Trinity College, Cambridge

AND

BERTRAND RUSSELL, M.A., F.R.S.
Lecturer and late Fellow of Trinity College, Cambridge

VOLUME I

Cambridge
at the University Press
1910

Whitehead and Russell, 1910

"$p \vee q \,.\supset.\, q \vee p$" means the proposition "'p or q' implies 'q or p.'" When we *assert* this proposition, instead of merely considering it, we write

$$\text{"}\vdash : p \vee q \,.\supset.\, q \vee p\text{,"}$$

where the two dots after the assertion-sign show that what is asserted is the whole of what follows the assertion-sign, since there are not as many as two dots anywhere else. If we had written "$p : \vee : q \,.\supset.\, q \vee p$," that would mean the proposition "either p is true, or q implies 'q or p.'" If we wished to assert this, we should have to put three dots after the assertion-sign. If we

apart from some determination given to x and y, they retain in that context their ambiguous differentiation. Thus "x is hurt" is an ambiguous "value" of a propositional function. When we wish to speak of the propositional function corresponding to "x is hurt," we shall write "$\hat{x}$ is hurt." Thus "$\hat{x}$ is hurt" is the propositional function and "x is hurt" is an ambiguous value of that function. Accordingly though "x is hurt" and "y is hurt" *occurring in the same context* can be distinguished, "$\hat{x}$ is hurt" and "$\hat{y}$ is hurt" convey no distinction of meaning at all. More generally, ϕx is an

$$\hat{x} + 1$$

"The function that takes x and adds one to it"
Caret: "this is a function with this argument"

A SET OF POSTULATES FOR THE FOUNDATION OF LOGIC.

BY ALONZO CHURCH.

If **F** is a function and **A** is a value of the independent variable for which the function is defined, then {**F**}(**A**) represents the value taken on by the function **F** when the independent variable takes on the value **A**. The usual notation is **F**(**A**). We introduce the braces on account of the possibility that **F** might be a combination of several symbols, but, in the case that **F** is a single symbol, we shall often use the notation **F**(**A**) as an abbreviation for the fuller expression.

Application

If **M** is any formula containing the variable **x**, then λ**x**[**M**] is a symbol for the function whose values are those given by the formula.

Abstraction

II. *If* **J** *is true, if* **M** *and* **N** *are well-formed, if the variable* **x** *occurs in* **M**, *and if the bound variables in* **M** *are distinct both from the variable* **x** *and from the free variables in* **N**, *then* **K**, *the result of substituting* S_{N}^{x} **M**| *for a particular occurrence of* {λ**x** . **M**}(**N**) *in* **J**, *is also true.*

β-Reduction

Church, 1932

The name *lambda* comes from the mathematician Alonzo Church's notation for functions (Church 1941). Lisp usually prefers expressive names over terse Greek letters, but lambda is an exception. A better name would be `make-function`. Lambda derives from the notation in Russell and Whitehead's *Principia Mathematica*, which used a caret over bound variables: $\hat{x}(x + x)$. Church wanted a one-dimensional string, so he moved the caret in front: $\hat{}x(x+x)$. The caret looked funny with nothing below it, so Church switched to the closest thing, an uppercase lambda, $\Lambda x(x + x)$. The Λ was easily confused with other symbols, so eventually the lowercase lambda was substituted: $\lambda x(x + x)$. John McCarthy was a student of Church's at Princeton, so when McCarthy invented Lisp in 1958, he adopted the lambda notation.

Peter Norvig, *Paradigms of Artificial Intelligence Programming*, 1992

Beta-Conversion

For variable x and lambda expressions M and N,

$$(\lambda x \,.\, M)\, N \;\longrightarrow\; M\,[x := N]$$

"Substitute N for x in M in a capture-avoiding manner" (sometimes $[N/x]$ or $[x \longrightarrow N]$)

$$(\lambda x \,.\, x)\, 1 \;\longrightarrow\; 1$$

Beta-Conversion

For variable x and lambda expressions M and N,

$$(\lambda x\,.\,M)\,N \;\longrightarrow\; M\,[x := N]$$

"Substitute N for x in M in a capture-avoiding manner" (sometimes $[N/x]$ or $[x \longrightarrow N]$)

$$(\lambda x\,.\,x)\,1 \;\longrightarrow\; 1$$

$$(\lambda x\,.\;+\;x\,x)\,2 \;\longrightarrow\; +\,2\,2$$

Beta-Conversion

For variable x and lambda expressions M and N,

$$(\lambda x\,.\,M)\,N \;\longrightarrow\; M\,[x := N]$$

"Substitute N for x in M in a capture-avoiding manner" (sometimes $[N/x]$ or $[x \longrightarrow N]$)

$$\begin{aligned}
(\lambda x\,.\,x)\,1 &\longrightarrow 1\\
(\lambda x\,.\,+\;x\,x)\,2 &\longrightarrow +\,2\,2\\
(\lambda x\,.\,\lambda y\,.\,+\;(+\,x\,y)\,x)\,1\,2 &= ((\lambda x\,.\,(\lambda y\,.\,+\;(+\,x\,y)\,x))\,1)\,2\\
&\longrightarrow (\lambda y\,.\,+\;(+\,1\,y)\,1)\;2\\
&\longrightarrow +\,(+\,1\,2)\,1
\end{aligned}$$

"Currying," after Haskell Curry (1958); Church (1932) credited it to Schönfinkel (1924), who may have taken it from Frege (1893)

Beta-Conversion

For variable x and lambda expressions M and N,

$$(\lambda x\,.\,M)\,N \;\longrightarrow\; M\,[x := N]$$

"Substitute N for x in M in a capture-avoiding manner" (sometimes $[N/x]$ or $[x \longrightarrow N]$)

$$\begin{aligned}(\lambda x\,.\,x\,3)\,(\lambda y\,.\,+\;y\,y) &\longrightarrow (\lambda y\,.\,+\;y\,y)\,3\\ &\longrightarrow +\,3\,3\end{aligned}$$

Beta-Conversion

For variable x and lambda expressions M and N,

$$(\lambda x \,.\, M)\, N \;\longrightarrow\; M\,[x := N]$$

"Substitute N for x in M in a capture-avoiding manner" (sometimes $[N/x]$ or $[x \longrightarrow N]$)

$$\begin{aligned}(\lambda x \,.\, x\, 3)\,(\lambda y \,.\, +\; y\; y) &\longrightarrow (\lambda y \,.\, +\; y\; y)\, 3 \\ &\longrightarrow +\, 3\, 3 \\ (\lambda x \,.\, (\lambda x \,.\, x)\, x\,)\, 1 &\longrightarrow (\lambda x \,.\, x\,)\, 1\end{aligned}$$

Beta-Conversion

For variable x and lambda expressions M and N,

$$(\lambda x\,.\,M)\,N \;\longrightarrow\; M\,[x := N]$$

"Substitute N for x in M in a capture-avoiding manner" (sometimes $[N/x]$ or $[x \longrightarrow N]$)

$$\begin{aligned}
(\lambda x\,.\,x\,3)\,(\lambda y\,.\,+\;y\;y) &\longrightarrow (\lambda y\,.\,+\;y\;y)\,3 \\
&\longrightarrow +\,3\,3 \\
(\lambda x\,.\,(\lambda x\,.\,x)\,x\,)\,1 &\longrightarrow (\lambda x\,.\,x\,)\,1 \\
(\lambda x\,.\,\lambda y\,.\,x\;y\,)\,y &\not\longrightarrow (\lambda y\,.\,y\;y)
\end{aligned}$$

Beta-Conversion

For variable x and lambda expressions M and N,

$$(\lambda x\,.\,M)\,N \;\longrightarrow\; M\,[x := N]$$

"Substitute N for x in M in a capture-avoiding manner" (sometimes $[N/x]$ or $[x \longrightarrow N]$)

$$\begin{aligned}
(\lambda x\,.\,x\,3)\,(\lambda y\,.\,+\;y\;y) &\longrightarrow (\lambda y\,.\,+\;y\;y)\,3\\
&\longrightarrow +\,3\,3\\
(\lambda x\,.\,(\lambda x\,.\,x)\,x\,)\,1 &\longrightarrow (\lambda x\,.\,x\,)\,1\\
(\lambda x\,.\,\lambda y\,.\,x\;y\,)\;y &= (\lambda x\,.\,\lambda z\,.\,x\;z\,)\;y\\
&\longrightarrow (\lambda z\,.\,y\;z)
\end{aligned}$$

Free Variables and Substitution

For variables $x \neq y$ and lambda terms M, M_1, M_2, N,

$$
\begin{aligned}
\mathrm{FV}(x) &= \{\, x \,\} \\
\mathrm{FV}(\lambda x\,.\,M) &= \mathrm{FV}(M) \setminus \{x\} \\
\mathrm{FV}(M_1\ M_2) &= \mathrm{FV}(M_1) \cup \mathrm{FV}(M_2)
\end{aligned}
$$

Free Variables: All those outside the scope of a λ

$$
\begin{aligned}
\mathrm{FV}(\, z\,) &= \{\, z \,\} \\
\mathrm{FV}(\, \lambda x\,.\,x\,) &= \{\,\} \\
\mathrm{FV}(\, \lambda x.\, x\, y\, z\,) &= \{\, y,\, z \,\} \\
\mathrm{FV}(\, \lambda x.\, x\,(\lambda y\,.\,y)\, z\,) &= \{\, z \,\} \\
\mathrm{FV}(\, \lambda x.\, x\,(\lambda y\,.\,y)\, y\,) &= \{\, y \,\}
\end{aligned}
$$

Free Variables and Substitution

For variables $x \neq y$ and lambda terms M, M_1, M_2, N,

$$\mathrm{FV}(x) = \{ x \} \qquad\qquad x\,[x := N] = N$$

$$\mathrm{FV}(\lambda x\,.\,M) = \mathrm{FV}(M) \setminus \{x\}$$

$$\mathrm{FV}(M_1\ M_2) = \mathrm{FV}(M_1) \cup \mathrm{FV}(M_2)$$

Base case: just replace the variable

$$x[\,x := y\,] = y$$

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

Free Variables and Substitution

For variables $x \neq y$ and lambda terms M, M_1, M_2, N,

$$\begin{aligned} \mathrm{FV}(x) &= \{ x \} \\ \mathrm{FV}(\lambda x . M) &= \mathrm{FV}(M) \setminus \{x\} \\ \mathrm{FV}(M_1\ M_2) &= \mathrm{FV}(M_1) \cup \mathrm{FV}(M_2) \end{aligned}$$

$$\begin{aligned} x[x := N] &= N \\ y[x := N] &= y \end{aligned}$$

Base case: leave other variables alone

$$\begin{aligned} z[\, x := y \,] &= z \\ z[\, x := (\lambda x . x\ y) \,] &= z \end{aligned}$$

Free Variables and Substitution

For variables $x \neq y$ and lambda terms M, M_1, M_2, N,

$$\begin{aligned} \mathrm{FV}(x) &= \{ x \} & x[x := N] &= N \\ \mathrm{FV}(\lambda x \,.\, M) &= \mathrm{FV}(M) \setminus \{x\} & y[x := N] &= y \\ \mathrm{FV}(M_1\ M_2) &= \mathrm{FV}(M_1) \cup \mathrm{FV}(M_2) & (M_1\ M_2)[x := N] &= (M_1[x := N]\ M_2[x := N]) \end{aligned}$$

Simple recursion on application: replace in both

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

Free Variables and Substitution

For variables $x \neq y$ and lambda terms M, M_1, M_2, N,

$$\begin{aligned} \mathrm{FV}(x) &= \{\, x \,\} \\ \mathrm{FV}(\lambda x\,.\,M) &= \mathrm{FV}(M) \setminus \{x\} \\ \mathrm{FV}(M_1\ M_2) &= \mathrm{FV}(M_1) \cup \mathrm{FV}(M_2) \end{aligned}$$

$$\begin{aligned} x\,[x := N] &= N \\ y\,[x := N] &= y \\ (M_1\ M_2)[x := N] &= (M_1[x := N]\, M_2[x := N]) \\ (\lambda x\,.\,M)[x := N] &= (\lambda x\,.\,M) \end{aligned}$$

Stop at a lambda term that binds the variable

$$\begin{aligned} (\lambda x\,.\,x\,(\lambda w\,.\,x)\,z)[\,x := (\lambda z\,.\,z\,z)\,] &= (\lambda x\,.\,x\,(\lambda w\,.\,x)\,z) \\ (z\,(\lambda z\,.\,z))[\,z := (\lambda x\,.\,x)\,] &= ((\lambda x\,.\,x)\,(\lambda z\,.\,z)) \end{aligned}$$

Free Variables and Substitution: Free Variables in N are the danger

For variables $x \neq y$ and lambda terms M, M_1, M_2, N,

$$\begin{aligned} \mathrm{FV}(x) &= \{x\} \\ \mathrm{FV}(\lambda x\,.\,M) &= \mathrm{FV}(M) \setminus \{x\} \\ \mathrm{FV}(M_1\ M_2) &= \mathrm{FV}(M_1) \cup \mathrm{FV}(M_2) \end{aligned}$$

$$\begin{aligned} x\,[x := N] &= N \\ y\,[x := N] &= y \\ (M_1\ M_2)[x := N] &= (M_1[x := N]\ M_2[x := N]) \\ (\lambda x\,.\,M)[x := N] &= (\lambda x\,.\,M) \\ (\lambda y\,.\,M)[x := N] &= (\lambda y\,.\,M[x := N]) \text{ if } y \notin \mathrm{FV}(N) \end{aligned}$$

Substitute in a lambda term when you won't accidentally bind a free variable

$$\begin{aligned} (\lambda y\,.\,x\ y\ z)[\,x := (\lambda z\,.\,z)\,] &= (\lambda y\,.\,(\lambda z\,.\,z)\ y\ z) \\ (\lambda y\,.\,x)[\,x := y\,] &\neq (\lambda y\,.\,y) \qquad \text{(can't be done)} \\ (\lambda y\,.\,x)[\,x := (\lambda z\,.\,z\ y)\,] &\neq (\lambda y\,.\,(\lambda z\,.\,z\ y)) \qquad \text{(can't be done)} \\ (\lambda x\,.\,z\ x)\,(z\,(\lambda z\,.\,z))[\,z := (\lambda x\,.\,x)\,] &= (\lambda x\,.\,(\lambda x\,.\,x)\ x)\,((\lambda x\,.\,x)\,(\lambda z\,.\,z)) \end{aligned}$$

Alpha-Conversion

For variables x, y and lambda expression M,

$$\lambda x \,.\, M \;=\; \lambda y \,.\, M\,[x := y] \text{ if } y \notin \mathrm{FV}(M)$$

"Rename the argument x to y provided y does not appear free in M"
You need to pick a "fresh" variable y

$$(\lambda x \,.\, x) \;=\; (\lambda y \,.\, y)$$

$$(\lambda x \,.\, y) \;\neq\; (\lambda y \,.\, y)$$

$$(\lambda x \,.\, x\,(\lambda x \,.\, x)\,(\lambda z \,.\, x\,(\lambda x \,.\, x))) \;=\; (\lambda y \,.\, y\,(\lambda x \,.\, x)\,(\lambda z \,.\, y\,(\lambda x \,.\, x))$$

Beta-Abstraction

$$M' = (\lambda\, x\, .\, M)\, N \text{ if } M' = M[\, x := N\,]$$

$$\begin{aligned} +\,1\,2 &= (\lambda x\,.\; +\; x\,2)\,1 \\ x\,(\lambda y\,.\,y)\,z &= (\lambda w\,.\,x\,w\,z)\,(\lambda y\,.\,y) \end{aligned}$$

Beta-Abstraction

$$M' = (\lambda\, x\, .\, M)\, N \text{ if } M' = M[\, x := N\,]$$

$$+\,1\,2 \;=\; (\lambda x\, .\; +\; x\, 2)\, 1$$
$$x\, (\lambda y\, .\, y)\, z \;=\; (\lambda w\, .\, x\, w\, z)\, (\lambda y\, .\, y)$$

η-conversion

$$\lambda x\, .\, M\, x = M \text{ if } x \notin \mathrm{FV}(M)$$

$$\lambda x\, .\; +\; 1\, x \;=\; +\, 1$$
$$\lambda x\, .\, (\lambda y\, .\, y\, y)\, x \;=\; \lambda y\, .\, y\, y$$
$$\lambda x\, .\, (\lambda y\, .\, x\, y)\, x \;\neq\; (\lambda y\, .\, x\, y)$$

Single-Step β-Reduction

$$\frac{}{(\lambda x \,.\, M)\, N \longrightarrow_\beta M\,[\,x := N\,]}\ \text{reduce} \qquad \frac{M \longrightarrow_\beta M'}{\lambda x \,.\, M \longrightarrow_\beta \lambda x \,.\, M'}\ \text{body}$$

$$\frac{M_1 \longrightarrow_\beta M_1'}{M_1\, M_2 \longrightarrow_\beta M_1'\, M_2}\ \text{func} \qquad \frac{M_2 \longrightarrow_\beta M_2'}{M_1\, M_2 \longrightarrow_\beta M_1\, M_2'}\ \text{arg}$$

Variables: lambda variable x lambda expressions M, M', M'', N, M_1, M_2, M_1', M_2'

Judgments: $M \longrightarrow_\beta M'$ means M β-reduces to M' in one step

Can apply or reduce applied function or its argument

$$\underline{(\lambda x \,.\, (\lambda z \,.\, z\, z)\, x\, 3)\, ((\lambda y \,.\, y\, y)\, (\lambda w \,.\, w))} \longrightarrow_\beta (\lambda z \,.\, z\, z)\, ((\lambda y \,.\, y\, y)\, (\lambda w \,.\, w))\, 3$$

Single-Step β-Reduction

$$\frac{}{(\lambda x\,.\,M)\,N \longrightarrow_\beta M\,[\,x := N\,]}\text{ reduce} \qquad \frac{M \longrightarrow_\beta M'}{\lambda x\,.\,M \longrightarrow_\beta \lambda x\,.\,M'}\text{ body}$$

$$\frac{M_1 \longrightarrow_\beta M_1'}{M_1\,M_2 \longrightarrow_\beta M_1'\,M_2}\text{ func} \qquad \frac{M_2 \longrightarrow_\beta M_2'}{M_1\,M_2 \longrightarrow_\beta M_1\,M_2'}\text{ arg}$$

Variables: lambda variable x lambda expressions M, M', M'', N, M_1, M_2, M_1', M_2'
Judgments: $M \longrightarrow_\beta M'$ means M β-reduces to M' in one step

Can apply or reduce applied function or its argument

$$\underline{(\lambda x\,.\,(\lambda z\,.\,z\,z)\,x\,3)\,((\lambda y\,.\,y\,y)\,(\lambda w\,.\,w))} \longrightarrow_\beta (\lambda z\,.\,z\,z)\,((\lambda y\,.\,y\,y)\,(\lambda w\,.\,w))\,3$$

$$(\lambda x\,.\,\underline{(\lambda z\,.\,z\,z)\,x}\,3)\,((\lambda y\,.\,y\,y)\,(\lambda w\,.\,w)) \longrightarrow_\beta (\lambda x\,.\,(x\,x)\,3)\,((\lambda y\,.\,y\,y)\,(\lambda w\,.\,w))$$

Single-Step β-Reduction

$$\frac{}{(\lambda x\,.\,M)\,N \longrightarrow_\beta M\,[\,x := N\,]}\ \text{reduce} \qquad \frac{M \longrightarrow_\beta M'}{\lambda x\,.\,M \longrightarrow_\beta \lambda x\,.\,M'}\ \text{body}$$

$$\frac{M_1 \longrightarrow_\beta M_1'}{M_1\,M_2 \longrightarrow_\beta M_1'\,M_2}\ \text{func} \qquad \frac{M_2 \longrightarrow_\beta M_2'}{M_1\,M_2 \longrightarrow_\beta M_1\,M_2'}\ \text{arg}$$

Variables: lambda variable x lambda expressions M, M', M'', N, M_1, M_2, M_1', M_2'

Judgments: $M \longrightarrow_\beta M'$ means M β-reduces to M' in one step

Can apply or reduce applied function or its argument

$$\underline{(\lambda x\,.\,(\lambda z\,.\,z\,z)\,x\,3)\,((\lambda y\,.\,y\,y)\,(\lambda w\,.\,w))} \longrightarrow_\beta (\lambda z\,.\,z\,z)\,((\lambda y\,.\,y\,y)\,(\lambda w\,.\,w))\,3$$

$$(\lambda x\,.\,\underline{(\lambda z\,.\,z\,z)\,x}\,3)\,((\lambda y\,.\,y\,y)\,(\lambda w\,.\,w)) \longrightarrow_\beta (\lambda x\,.\,(x\,x)\,3)\,((\lambda y\,.\,y\,y)\,(\lambda w\,.\,w))$$

$$(\lambda x\,.\,(\lambda z\,.\,z\,z)\,x\,3)\,(\underline{(\lambda y\,.\,y\,y)\,(\lambda w\,.\,w)}) \longrightarrow_\beta (\lambda x\,.\,(\lambda z\,.\,z\,z)\,x\,3)\,((\lambda w\,.\,w)\,(\lambda w\,.\,w))$$

Single-Step β-Reduction

$$\frac{}{(\lambda x \,.\, M)\, N \longrightarrow_{\beta} M\,[\,x := N\,]}\ \text{reduce} \qquad \frac{M \longrightarrow_{\beta} M'}{\lambda x \,.\, M \longrightarrow_{\beta} \lambda x \,.\, M'}\ \text{body}$$

$$\frac{M_1 \longrightarrow_{\beta} M_1'}{M_1\, M_2 \longrightarrow_{\beta} M_1'\, M_2}\ \text{func} \qquad \frac{M_2 \longrightarrow_{\beta} M_2'}{M_1\, M_2 \longrightarrow_{\beta} M_1\, M_2'}\ \text{arg}$$

Multi-step β-Reduction

$$\frac{}{M \longrightarrow_{\beta}^{*} M}\ \text{do-nothing}$$

$$\frac{M \longrightarrow_{\beta} M'}{M \longrightarrow_{\beta}^{*} M'}\ \text{one-step} \qquad \frac{M \longrightarrow_{\beta}^{*} M' \qquad M' \longrightarrow_{\beta}^{*} M''}{M \longrightarrow_{\beta}^{*} M''}\ \text{multi-step}$$

Normal Form

A lambda expression M is *in normal form* if there is no M' such that $M \longrightarrow_\beta M'$.

M' is *a normal form of* lambda expression M if $M \longrightarrow^*_\beta M'$ and M' is in normal form.

$\lambda x \,.\, x$	In normal form	
$x\,(\lambda z \,.\, z\,z)$	In normal form	
$(\lambda x \,.\, x)\,z$	Not in normal form	$(\lambda x \,.\, x)\,z \longrightarrow^*_\beta z$
$(\lambda z \,.\, z\,z)\,(\lambda z \,.\, z\,z)$	Not in normal form	$(\lambda z \,.\, z\,z)\,(\lambda z \,.\, z\,z) \longrightarrow_\beta (\lambda z \,.\, z\,z)\,(\lambda z \,.\, z\,z)$

Church Booleans

$$\begin{aligned} \text{true} &= \lambda x\,.\,\lambda y\,.\,x \\ \text{false} &= \lambda x\,.\,\lambda y\,.\,y \end{aligned}$$

$$\begin{aligned} \text{and} &= \lambda p\,.\,\lambda q\,.\,p\;q\;p \\ \text{or} &= \lambda p\,.\,\lambda q\,.\,p\;p\;q \\ \text{not} &= \lambda p\,.\,\lambda x\,.\,\lambda y.\,p\;y\;x \end{aligned}$$

$$\begin{aligned} \text{and true false} &= (\lambda p\,.\,\lambda q\,.\,p\;q\;p)\;\text{true false} \\ &\longrightarrow_\beta \text{true false true} \\ &= (\lambda x\,.\,\lambda y\,.\,x)\;\text{false true} \\ &\longrightarrow_\beta^* \text{false} \end{aligned}$$

$$\begin{aligned} \text{not true} &= (\lambda p\,.\,\lambda x\,.\,\lambda y.\,p\;y\;x)\;\text{true} \\ &\longrightarrow_\beta \lambda x\,.\,\lambda y\,.\,\text{true}\;y\;x \\ &= \lambda x\,.\,\lambda y\,.\,(\lambda x\,.\,\lambda y\,.\,x)\;y\;x \\ &\longrightarrow_\beta^* \lambda x\,.\,\lambda y\,.\,y \\ &= \text{false} \end{aligned}$$

Pairs

$$\begin{aligned}
\text{pair} &= \lambda x\,.\,\lambda y\,.\,\lambda f\,.\,f\;x\;y \\
\text{fst} &= \lambda p\,.\,p\,(\lambda x\,.\,\lambda y\,.\,x) \\
\text{snd} &= \lambda p\,.\,p\,(\lambda x\,.\,\lambda y\,.\,y)
\end{aligned}$$

$$\begin{aligned}
\text{fst}\,(\text{pair}\;a\;b) &= \text{fst}\,((\lambda x\,.\,\lambda y\,.\,\lambda f\,.\,f\;x\;y)\;a\;b) \\
&\longrightarrow^*_\beta \text{fst}\,(\lambda f\,.\,f\;a\;b) \\
&= (\lambda p\,.\,p\,(\lambda x\,.\,\lambda y\,.\,x))\,(\lambda f\,.\,f\;a\;b) \\
&\longrightarrow_\beta (\lambda f\,.\,f\;a\;b)\,(\lambda x\,.\,\lambda y\,.\,x) \\
&\longrightarrow_\beta (\lambda x\,.\,\lambda y\,.\,x)\;a\;b \\
&\longrightarrow^*_\beta a
\end{aligned}$$

$$\begin{aligned}
\text{snd}\,(\text{pair}\;a\;b) &= \text{snd}\,((\lambda x\,.\,\lambda y\,.\,\lambda f\,.\,f\;x\;y)\;a\;b) \\
&\longrightarrow^*_\beta \text{snd}\,(\lambda f\,.\,f\;a\;b) \\
&= (\lambda p\,.\,p\,(\lambda x\,.\,\lambda y\,.\,y))\,(\lambda f\,.\,f\;a\;b) \\
&\longrightarrow_\beta (\lambda f\,.\,f\;a\;b)\,(\lambda x\,.\,\lambda y\,.\,y) \\
&\longrightarrow_\beta (\lambda x\,.\,\lambda y\,.\,y)\;a\;b \\
&\longrightarrow^*_\beta b
\end{aligned}$$

Church Numerals

$$\begin{aligned}0 &= \lambda f\,.\,\lambda x\,.\,x\\ 1 &= \lambda f\,.\,\lambda x\,.\,f\;x\\ 2 &= \lambda f\,.\,\lambda x\,.\,f\,(f\;x)\\ 3 &= \lambda f\,.\,\lambda x\,.\,f\,(f\,(f\;x))\end{aligned}$$

$$\mathrm{succ} = \lambda n\,.\,\lambda f\,.\,\lambda x\,.\,f\,(\,n\;f\;x\,)$$

$$\begin{aligned}\mathrm{succ}\,0 &= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,f\,(n\;f\;x))\;0\\ &\longrightarrow_\beta\; \lambda f\,.\,\lambda x\,.\,f\,(0\;f\;x)\\ &= \lambda f\,.\,\lambda x\,.\,f\,(\,(\lambda f\,.\,\lambda x\,.\,x)\;f\;x)\\ &\longrightarrow_\beta^*\; \lambda f\,.\,\lambda x\,.\,f\;x\\ &= 1\end{aligned}$$

Church Numerals

$$
\begin{aligned}
0 &= \lambda f\,.\,\lambda x\,.\,x \\
1 &= \lambda f\,.\,\lambda x\,.\,f\;x \\
2 &= \lambda f\,.\,\lambda x\,.\,f\,(f\;x) \\
3 &= \lambda f\,.\,\lambda x\,.\,f\,(f\,(f\;x))
\end{aligned}
$$

$$
\mathrm{succ} = \lambda n\,.\,\lambda f\,.\,\lambda x\,.\,f\,(\,n\,f\;x\,)
$$

$$
\begin{aligned}
\mathrm{succ}\,2 &= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,f\,(n\,f\;x))\,2 \\
&\longrightarrow_{\beta} \lambda f\,.\,\lambda x\,.\,f\,(2\,f\;x) \\
&= \lambda f\,.\,\lambda x\,.\,f\,((\lambda f\,.\,\lambda x\,.\,f\,(f\;x))\,f\;x) \\
&\longrightarrow_{\beta}^{*} \lambda f\,.\,\lambda x\,.\,f\,(f\,(f\;x)) \\
&= 3
\end{aligned}
$$

Church Numerals

$$
\begin{aligned}
0 &= \lambda f\,.\,\lambda x\,.\,x \\
1 &= \lambda f\,.\,\lambda x\,.\,f\,x \\
2 &= \lambda f\,.\,\lambda x\,.\,f\,(f\,x) \\
3 &= \lambda f\,.\,\lambda x\,.\,f\,(f\,(f\,x))
\end{aligned}
$$

$$
\begin{aligned}
\text{succ} &= \lambda n\,.\,\lambda f\,.\,\lambda x\,.\,f\,(\,n\,f\,x\,) \\
\text{plus} &= \lambda m\,.\,\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,m\,f\,(\,n\,f\,x\,)
\end{aligned}
$$

$$
\begin{aligned}
\text{plus}\,3\,2 &= (\lambda m\,.\,\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,m\,f\,(\,n\,f\,x\,))\,3\,2 \\
&\longrightarrow_\beta^* \lambda f\,.\lambda x\,.\,3\,f\,(\,2\,f\,x) \\
&\longrightarrow_\beta^* \lambda f\,.\lambda x\,.\,f\,(f\,(f\,(2\,f\,x))) \\
&\longrightarrow_\beta^* \lambda f\,.\lambda x\,.\,f\,(f\,(f\,(f\,(\,f\,x)))) \\
&= 5
\end{aligned}
$$

Church Numerals

$$
\begin{aligned}
0 &= \lambda f\,.\,\lambda x\,.\,x \\
1 &= \lambda f\,.\,\lambda x\,.\,f\,x \\
2 &= \lambda f\,.\,\lambda x\,.\,f\,(f\,x) \\
3 &= \lambda f\,.\,\lambda x\,.\,f\,(f\,(f\,x))
\end{aligned}
$$

$$
\begin{aligned}
\text{succ} &= \lambda n\,.\,\lambda f\,.\,\lambda x\,.\,f\,(\,n\,f\,x\,) \\
\text{plus} &= \lambda m\,.\,\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,m\,f\,(\,n\,f\,x\,) \\
\text{mult} &= \lambda m\,.\,\lambda n\,.\,\lambda f\,.\,m\,(\,n\,f\,)
\end{aligned}
$$

$$
\begin{aligned}
\text{mult}\,2\,3 &= (\lambda m\,.\lambda n\,.\lambda f\,.m\,(n\,f))\,2\,3 \\
&\longrightarrow^{*}_{\beta} \lambda f\,.\,2\,(3\,f) \\
&\longrightarrow^{*}_{\beta} \lambda f\,.\,2\,(\lambda x\,.\,f(f(f\,x))) \\
&=_{\alpha} \lambda f\,.\,2\,(\lambda y\,.\,f(f(f\,y))) \\
&\longrightarrow^{*}_{\beta} \lambda f\,.\,\lambda x\,.\,(\lambda y\,.\,f(f(f\,y)))\,((\lambda y\,.\,f(f(f\,y)))\,x) \\
&\longrightarrow^{*}_{\beta} \lambda f\,.\,\lambda x\,.\,(\lambda y\,.\,f(f(f\,y)))\,(f(f(f\,x))) \\
&\longrightarrow^{*}_{\beta} \lambda f\,.\,\lambda x\,.\,f(f(f\,(f(f(f\,x)))\,)) \\
&= 6
\end{aligned}
$$

Church Numerals

$$
\begin{aligned}
0 &= \lambda f.\lambda x.x & \text{succ} &= \lambda n.\lambda f.\lambda x.f\,(n\,f\,x) \\
1 &= \lambda f.\lambda x.f\,x & \text{plus} &= \lambda m.\lambda n.\lambda f.\lambda x.m\,f\,(n\,f\,x) \\
2 &= \lambda f.\lambda x.f\,(f\,x) & \text{mult} &= \lambda m.\lambda n.\lambda f.m\,(n\,f) \\
3 &= \lambda f.\lambda x.f\,(f\,(f\,x)) & \text{pred} &= \lambda n.\lambda f.\lambda x.n\,(\lambda g.\lambda h.h\,(g\,f))\,(\lambda u.x)\,(\lambda u.u)
\end{aligned}
$$

$$
\begin{aligned}
\text{pred}\,0 &= (\lambda n.\lambda f.\lambda x.n\,(\lambda g.\lambda h.h\,(g\,f))\,(\lambda u.x)\,(\lambda u.u))\,0 \\
&\longrightarrow_\beta \lambda f.\lambda x.0\,(\lambda g.\lambda h.h\,(g\,f))\,(\lambda u.x)\,(\lambda u.u) \\
&\longrightarrow_\beta^* \lambda f.\lambda x.(\lambda u.x)\,(\lambda u.u) \\
&\longrightarrow_\beta \lambda f.\lambda x.x \\
&= 0
\end{aligned}
$$

What other choice do we have?

Church Numerals

$$0 = \lambda f.\lambda x.x \qquad \text{succ} = \lambda n.\lambda f.\lambda x.f\,(n\,f\,x)$$

$$1 = \lambda f.\lambda x.f\,x \qquad \text{plus} = \lambda m.\lambda n.\lambda f.\lambda x.m f\,(n\,f\,x)$$

$$2 = \lambda f.\lambda x.f\,(f\,x) \qquad \text{mult} = \lambda m.\lambda n.\lambda f.m\,(n\,f)$$

$$3 = \lambda f.\lambda x.f\,(f\,(f\,x)) \qquad \text{pred} = \lambda n.\lambda f.\lambda x.n\,(\lambda g.\lambda h.h\,(g\,f))\,(\lambda u.x)\,(\lambda u.u)$$

$$\begin{aligned}
\text{pred}\,1 &= (\lambda n.\lambda f.\lambda x.n\,(\lambda g.\lambda h.h\,(g\,f))\,(\lambda u.x)\,(\lambda u.u))\,1\\
&\longrightarrow_\beta \lambda f.\lambda x.1\,(\lambda g.\lambda h.h\,(g\,f))\,(\lambda u.x)\,(\lambda u.u)\\
&\longrightarrow_\beta^* \lambda f.\lambda x.((\lambda g.\lambda h.h\,(g\,f))\,(\lambda u.x))\,(\lambda u.u)\\
&\longrightarrow_\beta \lambda f.\lambda x.(\lambda h.h\,((\lambda u.x)\,f))\,(\lambda u.u)\\
&\longrightarrow_\beta \lambda f.\lambda x.(\lambda h.h\,x)\,(\lambda u.u)\\
&\longrightarrow_\beta \lambda f.\lambda x.((\lambda u.u)\,x)\\
&\longrightarrow_\beta \lambda f.\lambda x.x\\
&= 0
\end{aligned}$$

Church Numerals

$$
\begin{aligned}
0 &= \lambda f\,.\,\lambda x\,.\,x & \text{succ} &= \lambda n\,.\,\lambda f\,.\,\lambda x\,.\,f\,(n\,f\,x)\\
1 &= \lambda f\,.\,\lambda x\,.\,f\,x & \text{plus} &= \lambda m\,.\,\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,m f\,(n\,f\,x)\\
2 &= \lambda f\,.\,\lambda x\,.\,f\,(f\,x) & \text{mult} &= \lambda m\,.\,\lambda n\,.\,\lambda f\,.\,m\,(n\,f)\\
3 &= \lambda f\,.\,\lambda x\,.\,f\,(f\,(f\,x)) & \text{pred} &= \lambda n\,.\,\lambda f\,.\,\lambda x\,.\,n\,(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda u\,.\,x)\,(\lambda u\,.\,u)
\end{aligned}
$$

$$
\begin{aligned}
\text{pred}\,2 &= \underline{(\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,n\,(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda u\,.\,x)\,(\lambda u\,.\,u))\,2}\\
&\longrightarrow_\beta \lambda f\,.\,\lambda x\,.\,\underline{2}\,(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda u\,.\,x)\,(\lambda u\,.\,u)\\
&= \lambda f\,.\,\lambda x\,.\,(\underline{(\lambda f\,.\,\lambda x\,.\,f\,(f\,x))\,(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda u\,.\,x)})\,(\lambda u\,.\,u)\\
&\longrightarrow^*_\beta \lambda f\,.\,\lambda x\,.\,((\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\underline{(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda u\,.\,x)}))\,(\lambda u\,.\,u)\\
&\longrightarrow_\beta \lambda f\,.\,\lambda x\,.\,(\underline{(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda h\,.\,h\,((\lambda u\,.\,x)\,f))})\,(\lambda u\,.\,u)\\
&\longrightarrow_\beta \lambda f\,.\,\lambda x\,.\,(\lambda h\,.\,h\,((\lambda h\,.\,h\,(\underline{(\lambda u\,.\,x)\,f}))\,f))\,(\lambda u\,.\,u)\\
&\longrightarrow_\beta \lambda f\,.\,\lambda x\,.\,(\lambda h\,.\,h\,(\underline{(\lambda h\,.\,h\,x)\,f}))\,(\lambda u\,.\,u)\\
&\longrightarrow_\beta \lambda f\,.\,\lambda x\,.\,\underline{(\lambda h\,.\,h\,(f\,x)))\,(\lambda u\,.\,u)}\\
&\longrightarrow_\beta \lambda f\,.\,\lambda x\,.\,f\,x\\
&= 1
\end{aligned}
$$

Church Numerals

$$0 = \lambda f . \lambda x . x$$
$$1 = \lambda f . \lambda x . f\, x$$
$$2 = \lambda f . \lambda x . f\,(f\, x)$$
$$3 = \lambda f . \lambda x . f\,(f\,(f\, x))$$

$$\mathrm{succ} = \lambda n . \lambda f . \lambda x . f\,(n\, f\, x)$$
$$\mathrm{plus} = \lambda m . \lambda n . \lambda f . \lambda x . m\, f\,(n\, f\, x)$$
$$\mathrm{mult} = \lambda m . \lambda n . \lambda f . m\,(n\, f)$$
$$\mathrm{pred} = \lambda n . \lambda f . \lambda x . n\,(\lambda g . \lambda h . h\,(g\, f))\,(\lambda u . x)\,(\lambda u . u)$$
$$\mathrm{minus} = \lambda m . \lambda n . (n\, \mathrm{pred})\, m$$

$$\begin{aligned}
\mathrm{minus}\, 3\, 2 &= (\lambda m . \lambda n . (n\, \mathrm{pred})\, m)\, 3\, 2 \\
&\longrightarrow_\beta^* (2\, \mathrm{pred})\, 3 \\
&\longrightarrow_\beta^* (\lambda x . \mathrm{pred}\,(\mathrm{pred}\, x))\, 3 \\
&\longrightarrow_\beta^* \mathrm{pred}\,(\mathrm{pred}\, 3) \\
&\longrightarrow_\beta^* \mathrm{pred}\, 2 \\
&\longrightarrow_\beta^* 1
\end{aligned}$$

Church Numerals

$$
\begin{aligned}
0 &= \lambda f\,.\,\lambda x\,.\,x \\
1 &= \lambda f\,.\,\lambda x\,.\,f\;x \\
2 &= \lambda f\,.\,\lambda x\,.\,f\,(f\;x) \\
3 &= \lambda f\,.\,\lambda x\,.\,f\,(f\,(f\;x))
\end{aligned}
$$

$$
\begin{aligned}
\text{succ} &= \lambda n\,.\,\lambda f\,.\,\lambda x\,.\,f\,(\,n\,f\;x\,) \\
\text{plus} &= \lambda m\,.\,\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,m\,f\,(\,n\,f\;x\,) \\
\text{mult} &= \lambda m\,.\,\lambda n\,.\,\lambda f\,.\,m\,(\,n\,f\,) \\
\text{pred} &= \lambda n\,.\,\lambda f\,.\,\lambda x\,.\,n\,(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda u\,.\,x)\,(\lambda u\,.\,u) \\
\text{minus} &= \lambda m\,.\,\lambda n\,.\,(n\,\text{pred})\;m \\
\text{isZero} &= \lambda n\,.\,n\,(\lambda x.\,\text{false})\;\text{true}
\end{aligned}
$$

$$
\begin{aligned}
\text{isZero}\;0 &= (\lambda n\,.\,n\,(\lambda x.\,\text{false})\;\text{true})\;0 \\
&\longrightarrow_\beta^* 0\,(\lambda x.\,\text{false})\;\text{true} \\
&\longrightarrow_\beta^* \text{true}
\end{aligned}
$$

Church Numerals

$$
\begin{aligned}
0 &= \lambda f\,.\,\lambda x\,.\,x \\
1 &= \lambda f\,.\,\lambda x\,.\,f\;x \\
2 &= \lambda f\,.\,\lambda x\,.\,f\,(f\;x) \\
3 &= \lambda f\,.\,\lambda x\,.\,f\,(f\,(f\;x))
\end{aligned}
$$

$$
\begin{aligned}
\text{succ} &= \lambda n\,.\,\lambda f\,.\,\lambda x\,.\,f\,(\,n\,f\;x\,) \\
\text{plus} &= \lambda m\,.\,\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,m\,f\,(\,n\,f\;x\,) \\
\text{mult} &= \lambda m\,.\,\lambda n\,.\,\lambda f\,.\,m\,(\,n\,f\,) \\
\text{pred} &= \lambda n\,.\,\lambda f\,.\,\lambda x\,.\,n\,(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda u\,.\,x)\,(\lambda u\,.\,u) \\
\text{minus} &= \lambda m\,.\,\lambda n\,.\,(n\,\text{pred})\;m \\
\text{isZero} &= \lambda n\,.\,n\,(\lambda x.\,\text{false})\,\text{true}
\end{aligned}
$$

$$
\begin{aligned}
\text{isZero}\,1 &= (\lambda n\,.\,n\,(\lambda x.\,\text{false})\,\text{true})\,1 \\
&\longrightarrow_{\beta}^{*} 1\,(\lambda x.\,\text{false})\,\text{true} \\
&\longrightarrow_{\beta}^{*} (\lambda x.\,\text{false})\,\text{true} \\
&\longrightarrow_{\beta}^{*} \text{false}
\end{aligned}
$$

Church Numerals

$$
\begin{aligned}
0 &= \lambda f\,.\,\lambda x\,.\,x \\
1 &= \lambda f\,.\,\lambda x\,.\,f\;x \\
2 &= \lambda f\,.\,\lambda x\,.\,f\,(f\;x) \\
3 &= \lambda f\,.\,\lambda x\,.\,f\,(f\,(f\;x))
\end{aligned}
$$

$$
\begin{aligned}
\text{succ} &= \lambda n\,.\,\lambda f\,.\,\lambda x\,.\,f\,(\,n\,f\;x\,) \\
\text{plus} &= \lambda m\,.\,\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,m\,f\,(\,n\,f\;x\,) \\
\text{mult} &= \lambda m\,.\,\lambda n\,.\,\lambda f\,.\,m\,(\,n\,f\,) \\
\text{pred} &= \lambda n\,.\,\lambda f\,.\,\lambda x\,.\,n\,(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda u\,.\,x)\,(\lambda u\,.\,u) \\
\text{minus} &= \lambda m\,.\,\lambda n\,.\,(n\,\text{pred})\;m \\
\text{isZero} &= \lambda n\,.\,n\,(\lambda x.\,\text{false})\;\text{true}
\end{aligned}
$$

$$
\begin{aligned}
\text{isZero}\;2 &= (\lambda n\,.\,n\,(\lambda x.\,\text{false})\;\text{true})\;2 \\
&\longrightarrow^{*}_{\beta}\; 2\,(\lambda x.\,\text{false})\;\text{true} \\
&\longrightarrow^{*}_{\beta}\; (\lambda x.\,\text{false})\,((\lambda x.\,\text{false})\;\text{true}) \\
&\longrightarrow^{*}_{\beta}\; \text{false}
\end{aligned}
$$

$$\text{pred}\, n \;=\; (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,n\,(\lambda \mathrm{g}\,.\,\lambda h\,.\,h\,(\mathrm{g}\,f))\,(\lambda u\,.\,x)\,(\lambda u\,.\,u))\;n$$

$$
\begin{aligned}
\mathrm{pred}\, n &= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,n\,(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda u\,.\,x)\,(\lambda u\,.\,u))\,n \\
&= \lambda f\,.\,\lambda x\,.\,\Big(n\,\underbrace{(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))}_{p}\,\underbrace{(\lambda u\,.\,x)}_{y}\Big)\,(\lambda u\,.\,u)
\end{aligned}
$$

$$0 = \lambda p \,.\, \lambda y \,.\, y \qquad\qquad 0\, p\, y = y$$

$$\begin{aligned}\mathrm{pred}\, n &= (\lambda n \,.\, \lambda f \,.\, \lambda x \,.\, n\, (\lambda g \,.\, \lambda h \,.\, h\, (g\, f))\, (\lambda u \,.\, x)\, (\lambda u \,.\, u))\, n \\ &= \lambda f \,.\, \lambda x \,.\, \Big(n\, \underbrace{(\lambda g \,.\, \lambda h \,.\, h\, (g\, f))}_{p}\, \underbrace{(\lambda u \,.\, x)}_{y}\Big)\, (\lambda u \,.\, u)\end{aligned}$$

$$0 = \lambda p\,.\,\lambda y\,.\,y$$

$$1 = \lambda p\,.\,\lambda y\,.\,p\,y$$

$$0\,p\,y = y$$

$$\begin{aligned} 1\,p\,y &= p\,y \\ &= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,y \\ &= \lambda h\,.\,h\,(y\,f) \end{aligned}$$

$$\begin{aligned} \mathrm{pred}\,n &= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,n\,(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda u\,.\,x)\,(\lambda u\,.\,u))\,n \\ &= \lambda f\,.\,\lambda x\,.\,\Big(n\,\underbrace{(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))}_{p}\,\underbrace{(\lambda u\,.\,x)}_{y}\Big)\,(\lambda u\,.\,u) \end{aligned}$$

$$
\begin{aligned}
0 &= \lambda p\,.\,\lambda y\,.\,y \\
1 &= \lambda p\,.\,\lambda y\,.\,p\,y \\
2 &= \lambda p\,.\,\lambda y\,.\,p\,(p\,y)
\end{aligned}
$$

$$
\begin{aligned}
\text{pred}\,n &= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,n\,(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda u\,.\,x)\,(\lambda u\,.\,u))\,n \\
&= \lambda f\,.\,\lambda x\,.\,\Big(n\,\underbrace{(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))}_{p}\,\underbrace{(\lambda u\,.\,x)}_{y}\Big)\,(\lambda u\,.\,u)
\end{aligned}
$$

$$
\begin{aligned}
0\,p\,y &= y \\
1\,p\,y &= p\,y \\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,y \\
&= \lambda h\,.\,h\,(y\,f) \\
2\,p\,y &= p\,(1\,p\,y) \\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda h\,.\,h\,(y\,f)) \\
&= \lambda h\,.\,h\,((\lambda h\,.\,h\,(y\,f))\,f) \\
&= \lambda h\,.\,h\,(f\,(y\,f))
\end{aligned}
$$

$$
\begin{aligned}
0 &= \lambda p\,.\,\lambda y\,.\,y \\
1 &= \lambda p\,.\,\lambda y\,.\,p\,y \\
2 &= \lambda p\,.\,\lambda y\,.\,p\,(p\,y) \\
3 &= \lambda p\,.\,\lambda y\,.\,p\,(p\,(p\,y)) \\
\mathrm{pred}\,n &= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,n\,(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda u\,.\,x)\,(\lambda u\,.\,u))\,n \\
&= \lambda f\,.\,\lambda x\,.\,\Big(n\,\underbrace{(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))}_{p}\,\underbrace{(\lambda u\,.\,x)}_{y}\Big)\,(\lambda u\,.\,u)
\end{aligned}
$$

$$
\begin{aligned}
0\,p\,y &= y \\
1\,p\,y &= p\,y \\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,y \\
&= \lambda h\,.\,h\,(y\,f) \\
2\,p\,y &= p\,(1\,p\,y) \\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda h\,.\,h\,(y\,f)) \\
&= \lambda h\,.\,h\,((\lambda h\,.\,h\,(y\,f))\,f) \\
&= \lambda h\,.\,h\,(f\,(y\,f)) \\
3\,p\,y &= p\,(2\,p\,y) \\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda h\,.\,h\,(f\,(y\,f))) \\
&= \lambda h\,.\,h\,((\lambda h\,.\,h\,(f\,(y\,f)))\,f)) \\
&= \lambda h\,.\,h\,(f\,(f\,(y\,f)))
\end{aligned}
$$

$$
\begin{aligned}
0 &= \lambda p\,.\,\lambda y\,.\,y\\
1 &= \lambda p\,.\,\lambda y\,.\,p\,y\\
2 &= \lambda p\,.\,\lambda y\,.\,p\,(p\,y)\\
3 &= \lambda p\,.\,\lambda y\,.\,p\,(p\,(p\,y))\\
\mathrm{pred}\,n &= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,n\,(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda u\,.\,x)\,(\lambda u\,.\,u))\,n\\
&= \lambda f\,.\,\lambda x\,.\,\Big(n\,\underbrace{(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))}_{p}\,\underbrace{(\lambda u\,.\,x)}_{y}\Big)\,(\lambda u\,.\,u)
\end{aligned}
$$

$$
\begin{aligned}
0\,p\,y &= y\\
1\,p\,y &= p\,y\\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,y\\
&= \lambda h\,.\,h\,(y\,f)\\
2\,p\,y &= p\,(1\,p\,y)\\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda h\,.\,h\,(y\,f))\\
&= \lambda h\,.\,h\,((\lambda h\,.\,h\,(y\,f))\,f)\\
&= \lambda h\,.\,h\,(f\,(y\,f))\\
3\,p\,y &= p\,(2\,p\,y)\\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda h\,.\,h\,(f\,(y\,f)))\\
&= \lambda h\,.\,h\,((\lambda h\,.\,h\,(f\,(y\,f)))\,f)\\
&= \lambda h\,.\,h\,(f\,(f\,(y\,f)))\\
4\,p\,y &= \lambda h\,.\,h\,(f\,(f\,(f\,(y\,f))))
\end{aligned}
$$

$$
\begin{aligned}
0 &= \lambda p\,.\,\lambda y\,.\,y \\
1 &= \lambda p\,.\,\lambda y\,.\,p\,y \\
2 &= \lambda p\,.\,\lambda y\,.\,p\,(p\,y) \\
3 &= \lambda p\,.\,\lambda y\,.\,p\,(p\,(p\,y)) \\
\mathrm{pred}\,n &= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,n\,(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda u\,.\,x)\,(\lambda u\,.\,u))\,n \\
&= \lambda f\,.\,\lambda x\,.\,\Big(n\,\underbrace{(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))}_{p}\,\underbrace{(\lambda u\,.\,x)}_{y}\Big)\,(\lambda u\,.\,u) \\
&= \lambda f\,.\,\lambda x\,.\,\Big(\lambda h\,.\,h\,\underbrace{(f\,(f\,\cdots\,(f\,(y\,f))))}_{n\times f}\Big)\,(\lambda u\,.\,u)
\end{aligned}
$$

$$
\begin{aligned}
0\,p\,y &= y \\
1\,p\,y &= p\,y \\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,y \\
&= \lambda h\,.\,h\,(y\,f) \\
2\,p\,y &= p\,(1\,p\,y) \\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda h\,.\,h\,(y\,f)) \\
&= \lambda h\,.\,h\,((\lambda h\,.\,h\,(y\,f))\,f) \\
&= \lambda h\,.\,h\,(f\,(y\,f)) \\
3\,p\,y &= p\,(2\,p\,y) \\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda h\,.\,h\,(f\,(y\,f))) \\
&= \lambda h\,.\,h\,((\lambda h\,.\,h\,(f\,(y\,f)))\,f) \\
&= \lambda h\,.\,h\,(f\,(f\,(y\,f))) \\
4\,p\,y &= \lambda h\,.\,h\,(f\,(f\,(f\,(y\,f))))
\end{aligned}
$$

$$
\begin{aligned}
0 &= \lambda p\,.\,\lambda y\,.\,y \\
1 &= \lambda p\,.\,\lambda y\,.\,p\,y \\
2 &= \lambda p\,.\,\lambda y\,.\,p\,(p\,y) \\
3 &= \lambda p\,.\,\lambda y\,.\,p\,(p\,(p\,y)) \\
\mathrm{pred}\,n &= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,n\,(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda u\,.\,x)\,(\lambda u\,.\,u))\,n \\
&= \lambda f\,.\,\lambda x\,.\,\Big(n\,\underbrace{(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))}_{p}\,\underbrace{(\lambda u\,.\,x)}_{y}\Big)\,(\lambda u\,.\,u) \\
&= \lambda f\,.\,\lambda x\,.\,\Big(\lambda h\,.\,h\,\underbrace{(f\,(f\,\cdots\,(f\,(y\,f))))}_{n\times f}\Big)\,(\lambda u\,.\,u) \\
&= \lambda f\,.\,\lambda x\,.\,\Big(\lambda h\,.\,h\,\underbrace{(f\,(f\,\cdots\,(f\,x)))}_{n-1\times f}\Big)\,(\lambda u\,.\,u)
\end{aligned}
$$

$$
\begin{aligned}
0\,p\,y &= y \\
1\,p\,y &= p\,y \\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,y \\
&= \lambda h\,.\,h\,(y\,f) \\
2\,p\,y &= p\,(1\,p\,y) \\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda h\,.\,h\,(y\,f)) \\
&= \lambda h\,.\,h\,((\lambda h\,.\,h\,(y\,f))\,f) \\
&= \lambda h\,.\,h\,(f\,(y\,f)) \\
3\,p\,y &= p\,(2\,p\,y) \\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda h\,.\,h\,(f\,(y\,f))) \\
&= \lambda h\,.\,h\,((\lambda h\,.\,h\,(f\,(y\,f)))\,f) \\
&= \lambda h\,.\,h\,(f\,(f\,(y\,f))) \\
4\,p\,y &= \lambda h\,.\,h\,(f\,(f\,(f\,(y\,f))))
\end{aligned}
$$

$$
\begin{aligned}
0 &= \lambda p\,.\,\lambda y\,.\,y \\
1 &= \lambda p\,.\,\lambda y\,.\,p\,y \\
2 &= \lambda p\,.\,\lambda y\,.\,p\,(p\,y) \\
3 &= \lambda p\,.\,\lambda y\,.\,p\,(p\,(p\,y)) \\
\text{pred}\,n &= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,n\,(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda u\,.\,x)\,(\lambda u\,.\,u))\,n \\
&= \lambda f\,.\,\lambda x\,.\,\Big(n\,\underbrace{(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))}_{p}\,\underbrace{(\lambda u\,.\,x)}_{y}\Big)\,(\lambda u\,.\,u) \\
&= \lambda f\,.\,\lambda x\,.\,\Big(\lambda h\,.\,h\,\underbrace{(f\,(f\,\cdots\,(f\,(y\,f))))}_{n\times f}\Big)\,(\lambda u\,.\,u) \\
&= \lambda f\,.\,\lambda x\,.\,\Big(\lambda h\,.\,h\,\underbrace{(f\,(f\,\cdots\,(f\,x)))}_{n-1\times f}\Big)\,(\lambda u\,.\,u) \\
&= \lambda f\,.\,\lambda x\,.\,\underbrace{f\,(f\,\cdots\,(f\,x))}_{n-1\times f}
\end{aligned}
$$

$$
\begin{aligned}
0\,p\,y &= y \\
1\,p\,y &= p\,y \\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,y \\
&= \lambda h\,.\,h\,(y\,f) \\
2\,p\,y &= p\,(1\,p\,y) \\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda h\,.\,h\,(y\,f)) \\
&= \lambda h\,.\,h\,((\lambda h\,.\,h\,(y\,f))\,f) \\
&= \lambda h\,.\,h\,(f\,(y\,f)) \\
3\,p\,y &= p\,(2\,p\,y) \\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda h\,.\,h\,(f\,(y\,f))) \\
&= \lambda h\,.\,h\,((\lambda h\,.\,h\,(f\,(y\,f)))\,f)) \\
&= \lambda h\,.\,h\,(f\,(f\,(y\,f))) \\
4\,p\,y &= \lambda h\,.\,h\,(f\,(f\,(f\,(y\,f))))
\end{aligned}
$$

$$
\begin{aligned}
0 &= \lambda p\,.\,\lambda y\,.\,y \\
1 &= \lambda p\,.\,\lambda y\,.\,p\,y \\
2 &= \lambda p\,.\,\lambda y\,.\,p\,(p\,y) \\
3 &= \lambda p\,.\,\lambda y\,.\,p\,(p\,(p\,y)) \\
\text{pred}\,n &= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,n\,(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda u\,.\,x)\,(\lambda u\,.\,u))\,n \\
&= \lambda f\,.\,\lambda x\,.\left(n\,\underbrace{(\lambda g\,.\,\lambda h\,.\,h\,(g\,f))}_{p}\,\underbrace{(\lambda u\,.\,x)}_{y}\right)(\lambda u\,.\,u) \\
&= \lambda f\,.\,\lambda x\,.\left(\lambda h\,.\,h\,\underbrace{(f\,(f\,\cdots\,(f\,(y\,f))))}_{n\times f}\right)(\lambda u\,.\,u) \\
&= \lambda f\,.\,\lambda x\,.\left(\lambda h\,.\,h\,\underbrace{(f\,(f\,\cdots\,(f\,x)))}_{n-1\times f}\right)(\lambda u\,.\,u) \\
&= \lambda f\,.\,\lambda x\,.\,\underbrace{f\,(f\,\cdots\,(f\,x))}_{n-1\times f} \\
&= n - 1
\end{aligned}
$$

$$
\begin{aligned}
0\,p\,y &= y \\
1\,p\,y &= p\,y \\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,y \\
&= \lambda h\,.\,h\,(y\,f) \\
2\,p\,y &= p\,(1\,p\,y) \\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda h\,.\,h\,(y\,f)) \\
&= \lambda h\,.\,h\,((\lambda h\,.\,h\,(y\,f))\,f) \\
&= \lambda h\,.\,h\,(f\,(y\,f)) \\
3\,p\,y &= p\,(2\,p\,y) \\
&= (\lambda g\,.\,\lambda h\,.\,h\,(g\,f))\,(\lambda h\,.\,h\,(f\,(y\,f))) \\
&= \lambda h\,.\,h\,((\lambda h\,.\,h\,(f\,(y\,f)))\,f)) \\
&= \lambda h\,.\,h\,(f\,(f\,(y\,f))) \\
4\,p\,y &= \lambda h\,.\,h\,(f\,(f\,(f\,(y\,f))))
\end{aligned}
$$

Recursion

$$\mathrm{fac} = (\lambda n\,.\,(\mathrm{isZero}\ n)\ 1\ (\mathrm{mult}\ n\ (\mathrm{fac}\ (\mathrm{pred}\ n))))$$

This isn't a definition; it's a fixed-point equation defined in terms of itself.

Sometimes these are OK:

$$x = \mathrm{cons}(1, x)$$

has the solution

$$x = \mathrm{cons}(1, \mathrm{cons}(1, \mathrm{cons}(1, \cdots))).$$

Other times, they might not have a solution:

$$x = x + 1$$

Recursion

$$
\begin{aligned}
\mathrm{fac} &= (\lambda n \,.\, (\mathrm{isZero}\; n)\; 1\; (\mathrm{mult}\; n\; (\mathrm{fac}\; (\mathrm{pred}\; n)))) \\
&= (\lambda f \,.\, \lambda n \,.\, (\mathrm{isZero}\; n)\; 1\; (\mathrm{mult}\; n\; (f\; (\mathrm{pred}\; n))))\; \mathrm{fac} \\
&= H\; \mathrm{fac}
\end{aligned}
$$

Use β-abstraction to pull fac out of the equation.

H is a well-defined lambda expression that performs one step of the factorial computation

Recursion

$$\begin{aligned} \text{fac} &= (\lambda n\,.\,(\text{isZero}\ n)\ 1\ (\text{mult}\ n\ (\text{fac}\ (\text{pred}\ n)))) \\ &= (\lambda f\,.\,\lambda n\,.\,(\text{isZero}\ n)\ 1\ (\text{mult}\ n\ (f\ (\text{pred}\ n))))\ \text{fac} \\ &= H\ \text{fac} \end{aligned}$$

$$\begin{aligned} \text{fac} &= H\ \text{fac} \\ Y\ H &= H\ (Y\ H) \end{aligned}$$

Let's make a leap of faith and assume there's some function Y that computes the fixed point of fac from H:

$$\text{fac} = Y\ H$$

Recursion

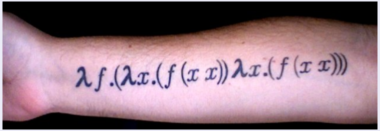
$$\begin{aligned}\text{fac} &= (\lambda n\,.\,(\text{isZero}\ n)\ 1\ (\text{mult}\ n\ (\text{fac}\ (\text{pred}\ n))))\\ &= (\lambda f\,.\,\lambda n\,.\,(\text{isZero}\ n)\ 1\ (\text{mult}\ n\ (f\ (\text{pred}\ n))))\ \text{fac}\\ &= H\ \text{fac}\end{aligned}$$

$$\begin{aligned}\text{fac} &= H\ \text{fac}\\ Y\ H &= H\ (Y\ H)\end{aligned}$$

$$\begin{aligned}\text{fac}\ 1 &= Y\ H\ 1\\ &= H\ (Y\ H)\ 1\\ &= (\lambda f\,.\,\lambda n\,.\,(\text{isZero}\ n)\ 1\ (\text{mult}\ n\ (f\ (\text{pred}\ n))))\ (Y\ H)\ 1\\ &\longrightarrow_\beta^* (\lambda n\,.\,(\text{isZero}\ n)\ 1\ (\text{mult}\ n\ (Y\ H\ (\text{pred}\ n))))\ 1\\ &\longrightarrow_\beta^* (\text{isZero}\ 1)\ 1\ (\text{mult}\ 1\ (Y\ H\ (\text{pred}\ 1)))\\ &\longrightarrow_\beta^* \text{mult}\ 1\ (Y\ H\ 0)\\ &= \text{mult}\ 1\ (H\ (Y\ H)\ 0)\\ &= \text{mult}\ 1\ ((\lambda f\,.\,\lambda n\,.\,(\text{isZero}\ n)\ 1\ (\text{mult}\ n\ (f\ (\text{pred}\ n))))\ (Y\ H)\ 0)\\ &\longrightarrow_\beta^* \text{mult}\ 1\ ((\lambda n\,.\,(\text{isZero}\ n)\ 1\ (\text{mult}\ n\ (Y\ H\ (\text{pred}\ n))))\ 0)\\ &\longrightarrow_\beta^* \text{mult}\ 1\ ((\text{isZero}\ 0)\ 1\ (\text{mult}\ 0\ (Y\ H\ (\text{pred}\ 0))))\\ &\longrightarrow_\beta^* \text{mult}\ 1\ 1\\ &\longrightarrow_\beta^* 1\end{aligned}$$

The Y Combinator

$Y =$



$$= \lambda f\,.\,(\lambda x\,.\,f\,(x\,x))\,(\lambda x\,.\,f\,(x\,x))$$

$$\begin{aligned}
Y\,H &= (\lambda f\,.\,(\lambda x\,.\,f\,(x\,x))\,(\lambda x\,.\,f\,(x\,x)))\,H \\
&\longrightarrow_\beta (\lambda x\,.\,H\,(x\,x))\,(\lambda x\,.\,H\,(x\,x)) \\
&\longrightarrow_\beta H\,((\lambda x\,.\,H\,(x\,x))\,(\lambda x\,.\,H\,(x\,x))) \\
&= H\,((\lambda f\,.\,(\lambda x\,.\,f\,(x\,x))\,(\lambda x\,.\,f\,(x\,x)))\,H) \\
&= H\,(Y\,H)
\end{aligned}$$

$$Y\,f = f\,(f\,(f\,(f\,(f\,(\cdots)))))$$

Reduction Rules

$$\frac{M[x := N] \;=\; M'}{(\lambda x\,.\,M)\,N \longrightarrow M'}\,\text{beta} \qquad \frac{M \longrightarrow M'}{M\,N \longrightarrow M'\,N}\,\text{func} \qquad \frac{N \longrightarrow N'}{M\,N \longrightarrow M\,N'}\,\text{arg} \qquad \frac{M \longrightarrow M'}{\lambda x\,.\,M \longrightarrow \lambda x\,.\,M'}\,\text{body}$$

Reduction Order Matters

$$(\lambda x\,.\,\lambda y\,.\,y)\,((\lambda z\,.\,z\,z)\,(\lambda z\,.\,z\,z)) \longrightarrow_\beta \lambda y\,.\,y \qquad \text{beta}$$

$$(\lambda x\,.\,\lambda y\,.\,y)\,((\lambda z\,.\,z\,z)\,(\lambda z\,.\,z\,z)) \longrightarrow_\beta (\lambda x\,.\,\lambda y\,.\,y)\,((\lambda z\,.\,z\,z)\,(\lambda z\,.\,z\,z)) \qquad \text{arg-beta}$$

Reduction Rules are Nondeterministic

$$\frac{M[x := N] \;=\; M'}{(\lambda x\,.\,M)\,N \longrightarrow M'}\,\text{beta} \qquad \frac{M \longrightarrow M'}{M\,N \longrightarrow M'\,N}\,\text{func} \qquad \frac{N \longrightarrow N'}{M\,N \longrightarrow M\,N'}\,\text{arg} \qquad \frac{M \longrightarrow M'}{\lambda x\,.\,M \longrightarrow \lambda x\,.\,M'}\,\text{body}$$

Reduction Rules are Nondeterministic

$$\frac{M[x := N] \;=\; M'}{(\lambda x\,.\,M)\,N \longrightarrow M'}\,\text{beta} \quad \frac{M \longrightarrow M'}{M\,N \longrightarrow M'\,N}\,\text{func} \quad \frac{N \longrightarrow N'}{M\,N \longrightarrow M\,N'}\,\text{arg} \quad \frac{M \longrightarrow M'}{\lambda x\,.\,M \longrightarrow \lambda x\,.\,M'}\,\text{body}$$

x	$\lambda x\,.\,e$	$e\,e$

There are three expression forms

Reduction Rules are Nondeterministic

$$\frac{M[x := N] \;=\; M'}{(\lambda x\,.\,M)\,N \longrightarrow M'}\,\text{beta} \qquad \frac{M \longrightarrow M'}{M\,N \longrightarrow M'\,N}\,\text{func} \qquad \frac{N \longrightarrow N'}{M\,N \longrightarrow M\,N'}\,\text{arg} \qquad \frac{M \longrightarrow M'}{\lambda x\,.\,M \longrightarrow \lambda x\,.\,M'}\,\text{body}$$

$e\;e$	x	$\lambda x\,.\,e$	$e\;e$

x is irreducible

only body applies to $\lambda.e$

$e\;e$ can lead to choice

Reduction Rules are Nondeterministic

$$\frac{M[x := N] \;=\; M'}{(\lambda x\,.\,M)\,N \longrightarrow M'}\;\text{beta} \qquad \frac{M \longrightarrow M'}{M\,N \longrightarrow M'\,N}\;\text{func} \qquad \frac{N \longrightarrow N'}{M\,N \longrightarrow M\,N'}\;\text{arg} \qquad \frac{M \longrightarrow M'}{\lambda x\,.\,M \longrightarrow \lambda x\,.\,M'}\;\text{body}$$

$e\,e$	x	$\lambda x\,.\,e$	$e\,e$
x			
$\lambda x\,.\,e$			
$e\,e$			

There are nine possibilities

Reduction Rules are Nondeterministic

$$\frac{M[x := N] \;=\; M'}{(\lambda x\,.\,M)\,N \longrightarrow M'}\,\text{beta} \quad \frac{M \longrightarrow M'}{M\,N \longrightarrow M'\,N}\,\text{func} \quad \frac{N \longrightarrow N'}{M\,N \longrightarrow M\,N'}\,\text{arg} \quad \frac{M \longrightarrow M'}{\lambda x\,.\,M \longrightarrow \lambda x\,.\,M'}\,\text{body}$$

$e\ e$	x	$\lambda x\,.\,e$	$e\ e$
x	$x\ x$	$x\,(\lambda x\,.\,e)$	$x\,(e\ e)$
$\lambda x\,.\,e$	$(\lambda x\,.\,e)\,x$	$(\lambda x\,.\,e)\,(\lambda x\,.\,e)$	$(\lambda x\,.\,e)\,(e\ e)$
$e\ e$	$(e\ e)\,x$	$(e\ e)\,(\lambda x\,.\,e)$	$(e\ e)\,(e\ e)$

Reduction Rules are Nondeterministic

$$\frac{M[x := N] \;=\; M'}{(\lambda x\,.\,M)\,N \longrightarrow M'}\,\text{beta} \qquad \frac{M \longrightarrow M'}{M\,N \longrightarrow M'\,N}\,\text{func} \qquad \frac{N \longrightarrow N'}{M\,N \longrightarrow M\,N'}\,\text{arg} \qquad \frac{M \longrightarrow M'}{\lambda x\,.\,M \longrightarrow \lambda x\,.\,M'}\,\text{body}$$

$e\,e$	x	$\lambda x\,.\,e$	$e\,e$
x	$x\,x$ (NF)	$x\,(\lambda x\,.\,e)$	$x\,(e\,e)$
$\lambda x\,.\,e$	$(\lambda x\,.\,e)\,x$	$(\lambda x\,.\,e)\,(\lambda x\,.\,e)$	$(\lambda x\,.\,e)\,(e\,e)$
$e\,e$	$(e\,e)\,x$	$(e\,e)\,(\lambda x\,.\,e)$	$(e\,e)\,(e\,e)$

One is in normal form already

Reduction Rules are Nondeterministic

$$\frac{M[x := N] \;=\; M'}{(\lambda x\,.\,M)\,N \longrightarrow M'}\,\text{beta} \qquad \frac{M \longrightarrow M'}{M\,N \longrightarrow M'\,N}\,\text{func} \qquad \frac{N \longrightarrow N'}{M\,N \longrightarrow M\,N'}\,\text{arg} \qquad \frac{M \longrightarrow M'}{\lambda x\,.\,M \longrightarrow \lambda x\,.\,M'}\,\text{body}$$

$e\,e$	x	$\lambda x\,.\,e$	$e\,e$
x	$x\,x$ (NF)	$x\,(\lambda x\,.\,e)$ arg-body	$x\,(e\,e)$ arg
$\lambda x\,.\,e$	$(\lambda x\,.\,e)\,x$	$(\lambda x\,.\,e)\,(\lambda x\,.\,e)$	$(\lambda x\,.\,e)\,(e\,e)$
$e\,e$	$(e\,e)\,x$ func	$(e\,e)\,(\lambda x\,.\,e)$	$(e\,e)\,(e\,e)$

For some, there is only one way to proceed

Reduction Rules are Nondeterministic

$$\frac{M[x := N] \;=\; M'}{(\lambda x\,.\,M)\,N \longrightarrow M'}\,\text{beta} \qquad \frac{M \longrightarrow M'}{M\,N \longrightarrow M'\,N}\,\text{func} \qquad \frac{N \longrightarrow N'}{M\,N \longrightarrow M\,N'}\,\text{arg} \qquad \frac{M \longrightarrow M'}{\lambda x\,.\,M \longrightarrow \lambda x\,.\,M'}\,\text{body}$$

$e\,e$	x	$\lambda x\,.\,e$	$e\,e$
x	$x\,x$ (NF)	$x\,(\lambda x\,.\,e)$ arg-body	$x\,(e\,e)$ arg
$\lambda x\,.\,e$	$(\lambda x\,.\,e)\,x$ beta | func-body	$(\lambda x\,.\,e)\,(\lambda x\,.\,e)$ beta | func-body | arg-body	$(\lambda x\,.\,e)\,(e\,e)$ beta | func-body | arg
$e\,e$	$(e\,e)\,x$ func	$(e\,e)\,(\lambda x\,.\,e)$ func | arg-body	$(e\,e)\,(e\,e)$ func | arg

For others, there are choices

Call-By-Name: beta; func

$$\frac{M[x := N] \;=\; M'}{(\lambda x\,.\,M)\,N \longrightarrow M'}\,\text{beta} \qquad \frac{M_1\,M_2 \longrightarrow M'}{M_1\,M_2\,N \longrightarrow M'\,N}\,\text{func}$$

Does not reduce arguments (the arg rule) or under abstractions (the body rule)

Results in Weak Head Normal Form (WHNF) $E ::= \lambda x\,.\,e \mid x\,e\,\cdots\,e$

$e\,e$	x	$\lambda x\,.\,e$	$e\,e$
x	$x\,x$ (NF)	$x\,(\lambda x\,.\,e)$ (WHNF)	$x\,(e\,e)$ (WHNF)
$\lambda x\,.\,e$	$(\lambda x\,.\,e)\,x$ beta	$(\lambda x\,.\,e)\,(\lambda x\,.\,e)$ beta	$(\lambda x\,.\,e)\,(e\,e)$ beta
$e\,e$	$(e\,e)\,x$ func	$(e\,e)\,(\lambda x\,.\,e)$ func	$(e\,e)\,(e\,e)$ func

Call-By-Name: beta; func

$$\frac{M[x := N] \; = \; M'}{(\lambda x \,.\, M)\, N \longrightarrow M'}\,\text{beta} \qquad \frac{M_1\, M_2 \longrightarrow M'}{M_1\, M_2\, N \longrightarrow M'\, N}\,\text{func}$$

Does not reduce arguments (the arg rule) or under abstractions (the body rule)
Results in Weak Head Normal Form (WHNF) $E ::= \lambda x\,.\,e \,|\, x\, e\, \cdots\, e$

$$\begin{aligned}
(\lambda x\,.\,\lambda y\,.\,y\,x)\,(\text{plus}\;5\;2)\,(\lambda x\,.\,\text{succ}\;x) &= (\lambda y\,.\,y\,(\text{plus}\;5\;2))\,(\lambda x\,.\,\text{succ}\;x) \\
&= (\lambda x\,.\,\text{succ}\;x)\,(\text{plus}\;5\;2) \\
&= \text{succ}\,(\text{plus}\;5\;2) \\
&= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,f\,(\,n\,f\;x\,))\,(\text{plus}\;5\;2) \\
&= \lambda f\,.\,\lambda x\,.\,f\,(\,(\text{plus}\;5\;2)\,f\;x\,)
\end{aligned}$$

Call-By-Value: func; arg; beta   substitute values: $e ::= v \mid e\ e$ $v ::= x \mid \lambda x\,.\,e$

$$\frac{M_1\ M_2 \longrightarrow M'}{M_1\ M_2\ N \longrightarrow M'\ N}\ \text{func} \qquad \frac{N \longrightarrow N'}{v\ N \longrightarrow v\ N'}\ \text{arg} \qquad \frac{M[x := v]\ =\ M'}{(\lambda x\,.\,M)\ v \longrightarrow M'}\ \text{beta}$$

Left to right, reduce arguments to a value then substitute, no reduction under abstraction
Results in Head Normal Form (HNF) $E ::= \lambda x\,.\,E \mid x\ e\ \cdots\ e$

$e\ e$	x	$\lambda x\,.\,e$	$e\ e$
x	$x\ x$ (NF)	$x\ (\lambda x\,.\,e)$ (HNF)	$x\ (e\ e)$ arg
$\lambda x\,.\,e$	$(\lambda x\,.\,e)\ x$ beta	$(\lambda x\,.\,e)\ (\lambda x\,.\,e)$ beta	$(\lambda x\,.\,e)\ (e\ e)$ arg
$e\ e$	$(e\ e)\ x$ func	$(e\ e)\ (\lambda x\,.\,e)$ func	$(e\ e)\ (e\ e)$ func

Call-By-Value: func; arg; beta   substitute values: $e ::= v \mid e\ e \qquad v ::= x \mid \lambda x\,.\,e$

$$\frac{M_1\ M_2 \longrightarrow M'}{M_1\ M_2\ N \longrightarrow M'\ N}\text{func} \qquad \frac{N \longrightarrow N'}{v\ N \longrightarrow v\ N'}\text{arg} \qquad \frac{M[x := v]\ =\ M'}{(\lambda x\,.\,M)\ v \longrightarrow M'}\text{beta}$$

Left to right, reduce arguments to a value then substitute, no reduction under abstraction
Results in Head Normal Form (HNF) $E ::= \ \lambda x\,.\,E \mid x\ e\ \cdots\ e$

$$(\lambda x\,.\,\lambda y\,.\,y\ x)\ (\text{plus}\ 5\ 2)\ (\lambda x\,.\,\text{succ}\ x)\ =$$

Call-By-Value: func; arg; beta   substitute values: $e ::= v \mid e\ e \qquad v ::= x \mid \lambda x\,.\,e$

$$\frac{M_1\ M_2 \longrightarrow M'}{M_1\ M_2\ N \longrightarrow M'\ N}\,\text{func} \qquad \frac{N \longrightarrow N'}{v\ N \longrightarrow v\ N'}\,\text{arg} \qquad \frac{M[x := v]\ =\ M'}{(\lambda x\,.\,M)\ v \longrightarrow M'}\,\text{beta}$$

Left to right, reduce arguments to a value then substitute, no reduction under abstraction

Results in Head Normal Form (HNF) $E ::= \lambda x\,.\,E \mid x\ e\ \cdots\ e$

$$\begin{aligned} \text{plus}\ 5\ 2 &= (\lambda m\,.\,\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,m f\,(n f\ x))\ 5\ 2 \\ &= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,5 f\,(n f\ x))\ 2 \\ &= \lambda f\,.\,\lambda x\,.\,5 f\,(2 f\ x) \end{aligned}$$

$$(\lambda x\,.\,\lambda y\,.\,y\ x)\ (\text{plus}\ 5\ 2)\ (\lambda x\,.\,\text{succ}\ x) =$$

Call-By-Value: func; arg; beta   substitute values: $e ::= v \mid e\ e \quad v ::= x \mid \lambda x\,.\,e$

$$\frac{M_1\ M_2 \longrightarrow M'}{M_1\ M_2\ N \longrightarrow M'\ N}\ \text{func} \qquad \frac{N \longrightarrow N'}{v\ N \longrightarrow v\ N'}\ \text{arg} \qquad \frac{M[x := v]\ =\ M'}{(\lambda x\,.\,M)\ v \longrightarrow M'}\ \text{beta}$$

Left to right, reduce arguments to a value then substitute, no reduction under abstraction
Results in Head Normal Form (HNF) $E ::= \lambda x\,.\,E \mid x\ e \cdots e$

$$\begin{aligned}
\text{plus}\ 5\ 2 &= (\lambda m\,.\,\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,m f\,(n f\ x))\ 5\ 2 \\
&= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,5 f\,(n f\ x))\ 2 \\
&= \lambda f\,.\,\lambda x\,.\,5 f\,(2 f\ x) \\
(\lambda x\,.\,\lambda y\,.\,y\,x)\ (\text{plus}\ 5\ 2)\ (\lambda x\,.\,\text{succ}\ x) &= (\lambda x\,.\,\lambda y\,.\,y\,x)\ (\lambda f\,.\,\lambda x\,.\,5 f\,(2 f\ x))\ (\lambda x\,.\,\text{succ}\ x)
\end{aligned}$$

Call-By-Value: func; arg; beta   substitute values: $e ::= v \mid e\ e \qquad v ::= x \mid \lambda x\,.\,e$

$$\frac{M_1\ M_2 \longrightarrow M'}{M_1\ M_2\ N \longrightarrow M'\ N}\ \text{func} \qquad \frac{N \longrightarrow N'}{v\ N \longrightarrow v\ N'}\ \text{arg} \qquad \frac{M[x := v]\ =\ M'}{(\lambda x\,.\,M)\ v \longrightarrow M'}\ \text{beta}$$

Left to right, reduce arguments to a value then substitute, no reduction under abstraction

Results in Head Normal Form (HNF) $E ::= \lambda x\,.\,E \mid x\ e\ \cdots\ e$

$$\begin{aligned}
\text{plus}\ 5\ 2 &= (\lambda m\,.\,\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,m f\,(n f\ x))\ 5\ 2 \\
&= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,5 f\,(n f\ x))\ 2 \\
&= \lambda f\,.\,\lambda x\,.\,5 f\,(2 f\ x) \\
(\lambda x\,.\,\lambda y\,.\,y\ x)\ (\text{plus}\ 5\ 2)\ (\lambda x\,.\,\text{succ}\ x) &= (\lambda x\,.\,\lambda y\,.\,y\ x)\ (\lambda f\,.\,\lambda x\,.\,5 f\,(2 f\ x))\ (\lambda x\,.\,\text{succ}\ x) \\
&= (\lambda y\,.\,y (\lambda f\,.\,\lambda x\,.\,5 f\,(2 f\ x)))\ (\lambda x\,.\,\text{succ}\ x)
\end{aligned}$$

Call-By-Value: func; arg; beta   substitute values: $e ::= v \mid e\ e \qquad v ::= x \mid \lambda x\,.\,e$

$$\frac{M_1\ M_2 \longrightarrow M'}{M_1\ M_2\ N \longrightarrow M'\ N}\ \text{func} \qquad \frac{N \longrightarrow N'}{v\ N \longrightarrow v\ N'}\ \text{arg} \qquad \frac{M[x := v]\ =\ M'}{(\lambda x\,.\,M)\ v \longrightarrow M'}\ \text{beta}$$

Left to right, reduce arguments to a value then substitute, no reduction under abstraction
Results in Head Normal Form (HNF) $E ::= \lambda x\,.\,E \mid x\ e \cdots e$

$$\begin{aligned}
\text{plus}\ 5\ 2 &= (\lambda m\,.\,\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,m f\,(n f\ x))\ 5\ 2 \\
&= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,5 f\,(n f\ x))\ 2 \\
&= \lambda f\,.\,\lambda x\,.\,5 f\,(2 f\ x) \\
(\lambda x\,.\,\lambda y\,.\,y\ x)\ (\text{plus}\ 5\ 2)\ (\lambda x\,.\,\text{succ}\ x) &= (\lambda x\,.\,\lambda y\,.\,y\ x)\ (\lambda f\,.\,\lambda x\,.\,5 f\,(2 f\ x))\ (\lambda x\,.\,\text{succ}\ x) \\
&= (\lambda y\,.\,y(\lambda f\,.\,\lambda x\,.\,5 f\,(2 f\ x)))\ (\lambda x\,.\,\text{succ}\ x) \\
&= (\lambda x\,.\,\text{succ}\ x)\ (\lambda f\,.\,\lambda x\,.\,5 f\,(2 f\ x))
\end{aligned}$$

Call-By-Value: func; arg; beta   substitute values: $e ::= v \mid e\ e \quad v ::= x \mid \lambda x\,.\,e$

$$\frac{M_1\ M_2 \longrightarrow M'}{M_1\ M_2\ N \longrightarrow M'\ N}\,\text{func} \qquad \frac{N \longrightarrow N'}{v\ N \longrightarrow v\ N'}\,\text{arg} \qquad \frac{M[x := v]\ =\ M'}{(\lambda x\,.\,M)\ v \longrightarrow M'}\,\text{beta}$$

Left to right, reduce arguments to a value then substitute, no reduction under abstraction

Results in Head Normal Form (HNF) $E ::= \lambda x\,.\,E \mid x\ e\ \cdots\ e$

$$\begin{aligned}
\text{plus}\ 5\ 2 &= (\lambda m\,.\,\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,m f\,(n f\ x))\ 5\ 2 \\
&= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,5 f\,(n f\ x))\ 2 \\
&= \lambda f\,.\,\lambda x\,.\,5 f\,(2 f\ x) \\
(\lambda x\,.\,\lambda y\,.\,y\ x)\ (\text{plus}\ 5\ 2)\ (\lambda x\,.\,\text{succ}\ x) &= (\lambda x\,.\,\lambda y\,.\,y\ x)\ (\lambda f\,.\,\lambda x\,.\,5 f\,(2 f\ x))\ (\lambda x\,.\,\text{succ}\ x) \\
&= (\lambda y\,.\,y(\lambda f\,.\,\lambda x\,.\,5 f\,(2 f\ x)))\ (\lambda x\,.\,\text{succ}\ x) \\
&= (\lambda x\,.\,\text{succ}\ x)\ (\lambda f\,.\,\lambda x\,.\,5 f\,(2 f\ x)) \\
&= \text{succ}\ (\lambda f\,.\,\lambda x\,.\,5 f\,(2 f\ x))
\end{aligned}$$

Call-By-Value: func; arg; beta   substitute values: $e ::= v \mid e\ e \qquad v ::= x \mid \lambda x\,.\,e$

$$\frac{M_1\ M_2 \longrightarrow M'}{M_1\ M_2\ N \longrightarrow M'\ N}\text{func} \qquad \frac{N \longrightarrow N'}{v\ N \longrightarrow v\ N'}\text{arg} \qquad \frac{M[x := v]\ =\ M'}{(\lambda x\,.\,M)\ v \longrightarrow M'}\text{beta}$$

Left to right, reduce arguments to a value then substitute, no reduction under abstraction
Results in Head Normal Form (HNF) $E ::= \lambda x\,.\,E \mid x\ e \cdots e$

$$\begin{aligned}
\text{plus}\ 5\ 2 &= (\lambda m\,.\,\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,m f\,(n f\,x))\ 5\ 2\\
&= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,5 f\,(n f\,x))\ 2\\
&= \lambda f\,.\,\lambda x\,.\,5 f\,(2 f\,x)\\
(\lambda x\,.\,\lambda y\,.\,y\,x)\ (\text{plus}\ 5\ 2)\ (\lambda x\,.\,\text{succ}\ x) &= (\lambda x\,.\,\lambda y\,.\,y\,x)\ (\lambda f\,.\,\lambda x\,.\,5 f\,(2 f\,x))\ (\lambda x\,.\,\text{succ}\ x)\\
&= (\lambda y\,.\,y(\lambda f\,.\,\lambda x\,.\,5 f\,(2 f\,x)))\ (\lambda x\,.\,\text{succ}\ x)\\
&= (\lambda x\,.\,\text{succ}\ x)\ (\lambda f\,.\,\lambda x\,.\,5 f\,(2 f\,x))\\
&= \text{succ}\ (\lambda f\,.\,\lambda x\,.\,5 f\,(2 f\,x))\\
&= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,f\,(n f\,x))\ (\lambda f\,.\,\lambda x\,.\,5 f\,(2 f\,x))\\
&= \lambda f\,.\,\lambda x\,.\,f\,((\lambda f\,.\,\lambda x\,.\,5 f\,(2 f\,x))\,f\,x)
\end{aligned}$$

Normal Order Reduction: beta; func; arg

$$\frac{M[x := N] \;=\; M'}{(\lambda x\,.\,M)\,N \longrightarrow M'}\,\text{beta} \qquad \frac{M \longrightarrow M'}{\lambda x\,.\,M \longrightarrow \lambda x\,.\,M'}\,\text{body}$$

$$\frac{M_1\,M_2 \longrightarrow M'}{M_1\,M_2\,N \longrightarrow M'\,N}\,\text{func} \qquad \frac{M \not\longrightarrow M' \qquad N \longrightarrow N'}{M\,N \longrightarrow M\,N'}\,\text{arg}$$

Leftmost, outermost (not enclosed in another) redex first

$e\ e$	x	$\lambda x\,.\,e$	$e\ e$
x	$x\,x$ (NF)	$x\,(\lambda x\,.\,e)$ arg-body	$x\,(e\,e)$ arg
$\lambda x\,.\,e$	$(\lambda x\,.\,e)\,x$ beta	$(\lambda x\,.\,e)\,(\lambda x\,.\,e)$ beta	$(\lambda x\,.\,e)\,(e\,e)$ beta
$e\ e$	$(e\,e)\,x$ func	$(e\,e)\,(\lambda x\,.\,e)$ func or arg-body	$(e\,e)\,(e\,e)$ func or arg

Normal Order Reduction: beta; func; arg

$$\frac{M[x := N] \;=\; M'}{(\lambda x \,.\, M)\, N \longrightarrow M'}\,\text{beta} \qquad \frac{M \longrightarrow M'}{\lambda x \,.\, M \longrightarrow \lambda x \,.\, M'}\,\text{body}$$

$$\frac{M_1\, M_2 \longrightarrow M'}{M_1\, M_2\, N \longrightarrow M'\, N}\,\text{func} \qquad \frac{M \not\longrightarrow M' \qquad N \longrightarrow N'}{M\, N \longrightarrow M\, N'}\,\text{arg}$$

Leftmost, outermost (not enclosed in another) redex first

$$\begin{aligned}
(\lambda x \,.\, \lambda y \,.\, y\, x)\,(\text{plus}\ 5\ 2)\,(\lambda x \,.\, \text{succ}\ x) &= (\lambda y \,.\, y\,(\text{plus}\ 5\ 2))\,(\lambda x \,.\, \text{succ}\ x) \\
&= (\lambda x \,.\, \text{succ}\ x)\,(\text{plus}\ 5\ 2) \\
&= \text{succ}\,(\text{plus}\ 5\ 2) \\
&= (\lambda n \,.\, \lambda f \,.\, \lambda x \,.\, f\,(\,n\, f\, x\,))\,(\text{plus}\ 5\ 2) \\
&= \lambda f \,.\, \lambda x \,.\, f\,(\,(\text{plus}\ 5\ 2)\, f\, x\,)
\end{aligned}$$

Normal Order Reduction: beta; func; arg

$$\frac{M[x := N] \;=\; M'}{(\lambda x \,.\, M)\, N \longrightarrow M'}\,\text{beta} \qquad \frac{M \longrightarrow M'}{\lambda x \,.\, M \longrightarrow \lambda x \,.\, M'}\,\text{body}$$

$$\frac{M_1\, M_2 \longrightarrow M'}{M_1\, M_2\, N \longrightarrow M'\, N}\,\text{func} \qquad \frac{M \not\longrightarrow M' \qquad N \longrightarrow N'}{M\, N \longrightarrow M\, N'}\,\text{arg}$$

Leftmost, outermost (not enclosed in another) redex first

$$\begin{aligned}
(\lambda x \,.\, \lambda y \,.\, y\, x)\, (\text{plus}\, 5\, 2)\, (\lambda x \,.\, \text{succ}\, x) &= (\lambda y \,.\, y\, (\text{plus}\, 5\, 2))\, (\lambda x \,.\, \text{succ}\, x) \\
&= (\lambda x \,.\, \text{succ}\, x)\, (\text{plus}\, 5\, 2) \\
&= \text{succ}\, (\text{plus}\, 5\, 2) \\
&= (\lambda n \,.\, \lambda f \,.\, \lambda x \,.\, f\, (\, n\, f\, x\,))\, (\text{plus}\, 5\, 2) \\
&= \lambda f \,.\, \lambda x \,.\, f\, (\, (\text{plus}\, 5\, 2)\, f\, x\,) \\
&= \lambda f \,.\, \lambda x \,.\, f\, (\, ((\lambda m \,.\, \lambda n \,.\, \lambda f \,.\, \lambda x \,.\, m\, f\, (\, n\, f\, x\,))\, 5\, 2)\, f\, x\,)
\end{aligned}$$

Normal Order Reduction: beta; func; arg

$$\frac{M[x := N] \;=\; M'}{(\lambda x\,.\,M)\,N \longrightarrow M'}\,\text{beta} \qquad \frac{M \longrightarrow M'}{\lambda x\,.\,M \longrightarrow \lambda x\,.\,M'}\,\text{body}$$

$$\frac{M_1\,M_2 \longrightarrow M'}{M_1\,M_2\,N \longrightarrow M'\,N}\,\text{func} \qquad \frac{M \not\longrightarrow M' \qquad N \longrightarrow N'}{M\,N \longrightarrow M\,N'}\,\text{arg}$$

Leftmost, outermost (not enclosed in another) redex first

$$\begin{aligned}
(\lambda x\,.\,\lambda y\,.\,y\,x)\,(\text{plus}\;5\;2)\,(\lambda x\,.\,\text{succ}\;x) &= (\lambda y\,.\,y\,(\text{plus}\;5\;2))\,(\lambda x\,.\,\text{succ}\;x)\\
&= (\lambda x\,.\,\text{succ}\;x)\,(\text{plus}\;5\;2)\\
&= \text{succ}\,(\text{plus}\;5\;2)\\
&= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,f\,(\,n\,f\,x\,))\,(\text{plus}\;5\;2)\\
&= \lambda f\,.\,\lambda x\,.\,f\,(\,(\text{plus}\;5\;2)\,f\,x\,)\\
&= \lambda f\,.\,\lambda x\,.\,f\,(\,((\lambda m\,.\,\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,m\,f\,(\,n\,f\,x\,))\;5\;2)\,f\,x\,)\\
&= \lambda f\,.\,\lambda x\,.\,f\,(\,((\lambda f\,.\,\lambda x\,.\,5\,f\,(\,2\,f\,x\,)))\,f\,x\,)
\end{aligned}$$

Normal Order Reduction: beta; func; arg

$$\frac{M[x := N] \;=\; M'}{(\lambda x \,.\, M)\, N \longrightarrow M'}\,\text{beta} \qquad \frac{M \longrightarrow M'}{\lambda x \,.\, M \longrightarrow \lambda x \,.\, M'}\,\text{body}$$

$$\frac{M_1\, M_2 \longrightarrow M'}{M_1\, M_2\, N \longrightarrow M'\, N}\,\text{func} \qquad \frac{M \not\longrightarrow M' \qquad N \longrightarrow N'}{M\, N \longrightarrow M\, N'}\,\text{arg}$$

Leftmost, outermost (not enclosed in another) redex first

$$\begin{aligned}
(\lambda x \,.\, \lambda y \,.\, y\, x)\, (\text{plus}\; 5\; 2)\, (\lambda x \,.\, \text{succ}\; x) &= (\lambda y \,.\, y\, (\text{plus}\; 5\; 2))\, (\lambda x \,.\, \text{succ}\; x) \\
&= (\lambda x \,.\, \text{succ}\; x)\, (\text{plus}\; 5\; 2) \\
&= \text{succ}\, (\text{plus}\; 5\; 2) \\
&= (\lambda n \,.\, \lambda f \,.\, \lambda x \,.\, f\, (\, n\, f\, x\,))\, (\text{plus}\; 5\; 2) \\
&= \lambda f \,.\, \lambda x \,.\, f\, (\, (\text{plus}\; 5\; 2)\, f\, x\,) \\
&= \lambda f \,.\, \lambda x \,.\, f\, (\, ((\lambda m \,.\, \lambda n \,.\, \lambda f \,.\, \lambda x \,.\, m\, f\, (\, n\, f\, x\,))\; 5\; 2)\, f\, x\,) \\
&= \lambda f \,.\, \lambda x \,.\, f\, (\, ((\lambda f \,.\, \lambda x \,.\, 5\, f\, (\, 2\, f\, x\,)))\, f\, x\,) \\
&= \lambda f \,.\, \lambda x \,.\, f\, (5\, f\, (\, 2\, f\, x\,))
\end{aligned}$$

Normal Order Reduction: beta; func; arg

$$\frac{M[x := N] \;=\; M'}{(\lambda x\,.\,M)\,N \longrightarrow M'}\,\text{beta} \qquad \frac{M \longrightarrow M'}{\lambda x\,.\,M \longrightarrow \lambda x\,.\,M'}\,\text{body}$$

$$\frac{M_1\,M_2 \longrightarrow M'}{M_1\,M_2\,N \longrightarrow M'\,N}\,\text{func} \qquad \frac{M \not\longrightarrow M' \qquad N \longrightarrow N'}{M\,N \longrightarrow M\,N'}\,\text{arg}$$

Leftmost, outermost (not enclosed in another) redex first

$$\begin{aligned}
(\lambda x\,.\,\lambda y\,.\,y\,x)\,(\text{plus}\;5\;2)\,(\lambda x\,.\,\text{succ}\;x) &= (\lambda y\,.\,y\,(\text{plus}\;5\;2))\,(\lambda x\,.\,\text{succ}\;x)\\
&= (\lambda x\,.\,\text{succ}\;x)\,(\text{plus}\;5\;2)\\
&= \text{succ}\,(\text{plus}\;5\;2)\\
&= (\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,f\,(\,n\,f\,x\,))\,(\text{plus}\;5\;2)\\
&= \lambda f\,.\,\lambda x\,.\,f\,(\,(\text{plus}\;5\;2)\,f\,x\,)\\
&= \lambda f\,.\,\lambda x\,.\,f\,(\,((\lambda m\,.\,\lambda n\,.\,\lambda f\,.\,\lambda x\,.\,m\,f\,(\,n\,f\,x\,))\,5\;2)\,f\,x\,)\\
&= \lambda f\,.\,\lambda x\,.\,f\,(\,((\lambda f\,.\,\lambda x\,.\,5\,f\,(\,2\,f\,x\,)))\,f\,x\,)\\
&= \lambda f\,.\,\lambda x\,.\,f\,(5\,f\,(\,2\,f\,x\,))\\
&= \lambda f\,.\,\lambda x\,.\,f\,(f\,(f\,(f\,(f\,(f\,(2\,f\,x))))))
\end{aligned}$$

Normal Order Reduction: beta; func; arg

$$\frac{M[x := N] \; = \; M'}{(\lambda x \,.\, M)\, N \longrightarrow M'}\,\text{beta} \qquad \frac{M \longrightarrow M'}{\lambda x \,.\, M \longrightarrow \lambda x \,.\, M'}\,\text{body}$$

$$\frac{M_1\, M_2 \longrightarrow M'}{M_1\, M_2\, N \longrightarrow M'\, N}\,\text{func} \qquad \frac{M \not\longrightarrow M' \qquad N \longrightarrow N'}{M\, N \longrightarrow M\, N'}\,\text{arg}$$

Leftmost, outermost (not enclosed in another) redex first

$$\begin{aligned}
(\lambda x \,.\, \lambda y \,.\, y\, x)\,(\text{plus } 5\; 2)\,(\lambda x \,.\, \text{succ } x) &= (\lambda y \,.\, y\,(\text{plus } 5\; 2))\,(\lambda x \,.\, \text{succ } x) \\
&= (\lambda x \,.\, \text{succ } x)\,(\text{plus } 5\; 2) \\
&= \text{succ}\,(\text{plus } 5\; 2) \\
&= (\lambda n \,.\, \lambda f \,.\, \lambda x \,.\, f\,(\, n\, f\, x\,))\,(\text{plus } 5\; 2) \\
&= \lambda f \,.\, \lambda x \,.\, f\,(\,(\text{plus } 5\; 2)\, f\, x\,) \\
&= \lambda f \,.\, \lambda x \,.\, f\,(\,((\lambda m \,.\, \lambda n \,.\, \lambda f \,.\, \lambda x \,.\, m\, f\,(\, n\, f\, x\,))\, 5\; 2)\, f\, x\,) \\
&= \lambda f \,.\, \lambda x \,.\, f\,(\,((\lambda f \,.\, \lambda x \,.\, 5\, f\,(\, 2\, f\, x\,)))\, f\, x\,) \\
&= \lambda f \,.\, \lambda x \,.\, f\,(5\, f\,(\, 2\, f\, x\,)) \\
&= \lambda f \,.\, \lambda x \,.\, f\,(f\,(f\,(f\,(f\,(f\,(2\, f\, x)))))) \\
&= \lambda f \,.\, \lambda x \,.\, f\,(f\,(f\,(f\,(f\,(f\,(f\,(f\, x)))))))
\end{aligned}$$

α-, β-, and η-Conversions

$$\frac{M[x := N] \;=\; M'}{(\lambda x \,.\, M)\, N \leftrightarrow M'}\,\text{beta} \qquad \frac{y \notin \text{FV}(M)}{\lambda x \,.\, M \leftrightarrow \lambda y \,.\, M[x := y]}\,\text{alpha} \qquad \frac{x \notin \text{FV}(M)}{(\lambda x \,.\, M\, x) \leftrightarrow M}\,\text{eta}$$

Church-Rosser Theorem I

If $e_1 \leftrightarrow e_2$ then there exists an e such that $e_1 \longrightarrow e$ and $e_2 \longrightarrow e$

Corollary: An expression may only have a single normal form.

Church-Rosser Theorem II

If $e_1 \longrightarrow e_2$ and e_2 is in normal form, then normal order reduction can transform e_1 into e_2

α-conversion renders names irrelevant; make them canonical?

$$\lambda x. \lambda y. \lambda z. x \quad y \quad z \quad z \quad y$$

α-conversion renders names irrelevant; make them canonical? Idea: number by depth

$$\begin{array}{llllllll} \lambda x. & \lambda y. & \lambda z. & x & y & z & z & y \\ \lambda 2 & \lambda 1 & \lambda 0 & 2 & 1 & 0 & 0 & 1 \end{array}$$

α-conversion renders names irrelevant; make them canonical? Idea: number by depth

$$
\begin{array}{llllllll}
\lambda x. & \lambda y. & \lambda z. & x & y & z & z & y \\
\lambda 2 & \lambda 1 & \lambda 0 & 2 & 1 & 0 & 0 & 1 \\
\lambda & \lambda & \lambda & 2 & 1 & 0 & 0 & 1
\end{array}
$$

α-conversion renders names irrelevant; make them canonical? Idea: number by depth

$$
\begin{array}{llllllll}
\lambda x. & \lambda y. & \lambda z. & x & y & z & z & y \\
\lambda 2 & \lambda 1 & \lambda 0 & 2 & 1 & 0 & 0 & 1 \\
\lambda & \lambda & \lambda & 2 & 1 & 0 & 0 & 1
\end{array}
$$

Why number innermost 0? Nothing changes when instantiating bound variables

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

α-conversion renders names irrelevant; make them canonical? Idea: number by depth

$$\begin{array}{llllllll} \lambda x. & \lambda y. & \lambda z. & x & y & z & z & y \\ \lambda 2 & \lambda 1 & \lambda 0 & 2 & 1 & 0 & 0 & 1 \\ \lambda & \lambda & \lambda & 2 & 1 & 0 & 0 & 1 \end{array}$$

Why number innermost 0? Nothing changes when instantiating bound variables

$$(\lambda w.\lambda p.\ \ \lambda q.\ \ w \quad q \quad p)\ \ (\lambda x.\ \ \lambda y.\ \ \lambda z.\ \ x \quad y \quad z \quad z \quad y)$$

α-conversion renders names irrelevant; make them canonical? Idea: number by depth

$$
\begin{array}{llllllll}
\lambda x. & \lambda y. & \lambda z. & x & y & z & z & y \\
\lambda 2 & \lambda 1 & \lambda 0 & 2 & 1 & 0 & 0 & 1 \\
\lambda & \lambda & \lambda & 2 & 1 & 0 & 0 & 1
\end{array}
$$

Why number innermost 0? Nothing changes when instantiating bound variables

$$
\begin{array}{llllllllllllll}
(\lambda w. & \lambda p. & \lambda q. & w & q & p) & (\lambda x. & \lambda y. & \lambda z. & x & y & z & z & y) \\
(\lambda 2 & \lambda 1 & \lambda 0 & 2 & 0 & 1) & (\lambda 2 & \lambda 1 & \lambda 0 & 2 & 1 & 0 & 0 & 1)
\end{array}
$$

α-conversion renders names irrelevant; make them canonical? Idea: number by depth

$$\begin{array}{llllllll} \lambda x. & \lambda y. & \lambda z. & x & y & z & z & y \\ \lambda 2 & \lambda 1 & \lambda 0 & 2 & 1 & 0 & 0 & 1 \\ \lambda & \lambda & \lambda & 2 & 1 & 0 & 0 & 1 \end{array}$$

Why number innermost 0? Nothing changes when instantiating bound variables

$$\begin{array}{llllllllllllll} (\lambda w. & \lambda p. & \lambda q. & w & q & p) & (\lambda x. & \lambda y. & \lambda z. & x & y & z & z & y) \\ (\lambda 2 & \lambda 1 & \lambda 0 & 2 & 0 & 1) & (\lambda 2 & \lambda 1 & \lambda 0 & 2 & 1 & 0 & 0 & 1) \end{array}$$

$$\rightarrow \lambda p.\ \lambda q.\ (\lambda x.\ \lambda y.\ \lambda z\ \ x\ \ y\ \ z\ \ z\ \ y)\ \ q\ \ p$$

α-conversion renders names irrelevant; make them canonical? Idea: number by depth

$$\begin{array}{llllllll}
\lambda x. & \lambda y. & \lambda z. & x & y & z & z & y \\
\lambda 2 & \lambda 1 & \lambda 0 & 2 & 1 & 0 & 0 & 1 \\
\lambda & \lambda & \lambda & 2 & 1 & 0 & 0 & 1
\end{array}$$

Why number innermost 0? Nothing changes when instantiating bound variables

$$\begin{array}{llllllllllllll}
(\lambda w. & \lambda p. & \lambda q. & w & q & p) & (\lambda x. & \lambda y. & \lambda z. & x & y & z & z & y) \\
(\lambda 2 & \lambda 1 & \lambda 0 & 2 & 0 & 1) & (\lambda 2 & \lambda 1 & \lambda 0 & 2 & 1 & 0 & 0 & 1)
\end{array}$$

$$\begin{array}{lllllllllllll}
\longrightarrow & \lambda p. & \lambda q. & (\lambda x. & \lambda y. & \lambda z & x & y & z & z & y) & q & p \\
\longrightarrow & \lambda 1 & \lambda 0 & (\lambda 2 & \lambda 1 & \lambda 0 & 2 & 1 & 0 & 0 & 1) & 0 & 1
\end{array}$$

de Bruijn indices

A *de Bruijn index* for a variable is the count of the number of λ binder scopes between its use and its corresponding λ.

$$\begin{array}{lll} \lambda x. & x & \\ \lambda & 0 & \end{array}$$

$$\begin{array}{lll} \lambda x. & \lambda y. & x \\ \lambda & \lambda & 1 \end{array}$$

de Bruijn indices

A *de Bruijn index* for a variable is the count of the number of λ binder scopes between its use and its corresponding λ.

$\lambda x.\ x$

$\lambda \quad 0$

$\lambda x.\ \lambda y.\ x$

$\lambda \quad \lambda \quad 1$

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

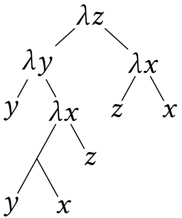
de Bruijn indices

A *de Bruijn index* for a variable is the count of the number of λ binder scopes between its use and its corresponding λ.

$$\begin{array}{ll} \lambda x. & x \\ \lambda & 0 \end{array}$$

$$\begin{array}{lll} \lambda x. & \lambda y. & x \\ \lambda & \lambda & 1 \end{array}$$

$$\begin{array}{lllllllll} \lambda z. & (\lambda y. & y & (\lambda x. & y & x & z)) & (\lambda x. & z & x) \\ \lambda & (\lambda & 0 & (\lambda & 1 & 0 & 2)) & (\lambda & 1 & 0) \end{array}$$



Assigning de Bruijn Indices

$$\frac{\Gamma(x) = n}{\Gamma \vdash x \xrightarrow{\text{dB}} n}\ \text{var} \qquad \frac{\Gamma \vdash e_1 \xrightarrow{\text{dB}} e_1' \quad \Gamma \vdash e_2 \xrightarrow{\text{dB}} e_2'}{\Gamma \vdash e_1\, e_2 \xrightarrow{\text{dB}} e_1'\, e_2'}\ \text{app} \qquad \frac{\Gamma, x \vdash e \xrightarrow{\text{dB}} e'}{\Gamma \vdash \lambda x.e \xrightarrow{\text{dB}} \lambda e'}\ \text{lam}$$

Judgments: $\Gamma \vdash e \xrightarrow{\text{dB}} e'$ Expression e rewrites to e' in environment Γ

Variables: x Variables $e,\ e_1,\ e_2$ Expressions n Natural numbers

Environments: Γ Sequence of variable names

Assigning de Bruijn Indices

$$\frac{\Gamma(x) = n}{\Gamma \vdash x \xrightarrow{\text{dB}} n}\ \text{var} \qquad \frac{\Gamma \vdash e_1 \xrightarrow{\text{dB}} e_1' \quad \Gamma \vdash e_2 \xrightarrow{\text{dB}} e_2'}{\Gamma \vdash e_1\, e_2 \xrightarrow{\text{dB}} e_1'\, e_2'}\ \text{app} \qquad \frac{\Gamma, x \vdash e \xrightarrow{\text{dB}} e'}{\Gamma \vdash \lambda x.e \xrightarrow{\text{dB}} \lambda e'}\ \text{lam}$$

Judgments: $\Gamma \vdash e \xrightarrow{\text{dB}} e'$ Expression e rewrites to e' in environment Γ

Variables: x Variables $e,\ e_1,\ e_2$ Expressions n Natural numbers

Environments: Γ Sequence of variable names

The comma operator extends environments on the right.

If Γ is the empty environment, then $\Gamma, x = x$

If $\Gamma = a\, b\, c$ then $\Gamma, d = a\, b\, c\, d$

Assigning de Bruijn Indices

$$\frac{\Gamma(x) = n}{\Gamma \vdash x \xrightarrow{\text{dB}} n}\ \text{var} \qquad \frac{\Gamma \vdash e_1 \xrightarrow{\text{dB}} e_1' \quad \Gamma \vdash e_2 \xrightarrow{\text{dB}} e_2'}{\Gamma \vdash e_1\, e_2 \xrightarrow{\text{dB}} e_1' e_2'}\ \text{app} \qquad \frac{\Gamma, x \vdash e \xrightarrow{\text{dB}} e'}{\Gamma \vdash \lambda x.e \xrightarrow{\text{dB}} \lambda e'}\ \text{lam}$$

Judgments:	$\Gamma \vdash e \xrightarrow{\text{dB}} e'$	Expression e rewrites to e' in environment Γ				
Variables:	x	Variables	$e,\ e_1,\ e_2$	Expressions	n	Natural numbers
Environments:	Γ	Sequence of variable names				

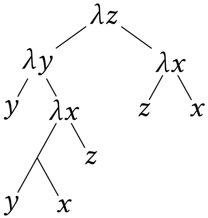
The comma operator extends environments on the right.
If Γ is the empty environment, then $\Gamma, x = x$
If $\Gamma = a\ b\ c$ then $\Gamma, d = a\ b\ c\ d$

$\Gamma(x)$ is the count of how many variables are to the right of the rightmost appearance of x.
If $\Gamma = d\ a\ b\ c\ d$, then $\Gamma(d) = 0$, $\Gamma(c) = 1$, $\Gamma(b) = 2$, and $\Gamma(a) = 3$.

Assigning de Bruijn Indices

$$\frac{\Gamma(x) = n}{\Gamma \vdash x \xrightarrow{\text{dB}} n}\ \text{var} \qquad \frac{\Gamma \vdash e_1 \xrightarrow{\text{dB}} e_1' \quad \Gamma \vdash e_2 \xrightarrow{\text{dB}} e_2'}{\Gamma \vdash e_1\, e_2 \xrightarrow{\text{dB}} e_1'\, e_2'}\ \text{app} \qquad \frac{\Gamma, x \vdash e \xrightarrow{\text{dB}} e'}{\Gamma \vdash \lambda x.e \xrightarrow{\text{dB}} \lambda e'}\ \text{lam}$$

$$\lambda z.(\lambda y.\ y \quad (\lambda x.\ y \quad x \quad z))\,(\lambda x.\ z \quad x)$$

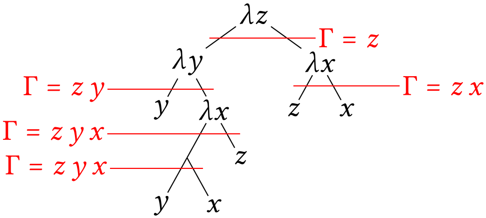


Assigning de Bruijn Indices

$$\frac{\Gamma(x) = n}{\Gamma \vdash x \xrightarrow{\text{dB}} n}\ \text{var} \qquad \frac{\Gamma \vdash e_1 \xrightarrow{\text{dB}} e_1' \quad \Gamma \vdash e_2 \xrightarrow{\text{dB}} e_2'}{\Gamma \vdash e_1\, e_2 \xrightarrow{\text{dB}} e_1'\, e_2'}\ \text{app} \qquad \frac{\Gamma, x \vdash e \xrightarrow{\text{dB}} e'}{\Gamma \vdash \lambda x.e \xrightarrow{\text{dB}} \lambda e'}\ \text{lam}$$

$\lambda z.(\lambda y.\ y \quad (\lambda x.\ y \quad x \quad z))\,(\lambda x.\ z \quad x)$

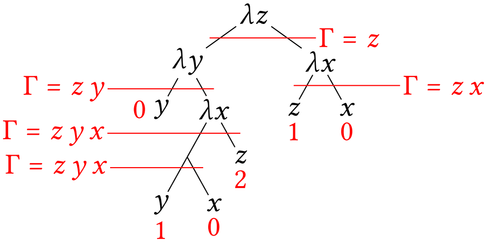
$\lambda \quad (\lambda \qquad (\lambda \qquad\qquad))\,(\lambda \qquad)$



Assigning de Bruijn Indices

$$\frac{\Gamma(x) = n}{\Gamma \vdash x \xrightarrow{\text{dB}} n}\ \text{var} \qquad \frac{\Gamma \vdash e_1 \xrightarrow{\text{dB}} e_1' \quad \Gamma \vdash e_2 \xrightarrow{\text{dB}} e_2'}{\Gamma \vdash e_1\, e_2 \xrightarrow{\text{dB}} e_1'\, e_2'}\ \text{app} \qquad \frac{\Gamma, x \vdash e \xrightarrow{\text{dB}} e'}{\Gamma \vdash \lambda x.e \xrightarrow{\text{dB}} \lambda e'}\ \text{lam}$$

$$\lambda z.(\lambda y.\ y\ \ (\lambda x.\ y\ \ x\ \ z))\,(\lambda x.\ z\ \ x)$$
$$\lambda\ \ (\lambda\ \ 0\ \ (\lambda\ \ 1\ \ 0\ \ 2))\,(\lambda\ \ 1\ \ 0)$$



$$\lambda x.(\lambda y.\ \lambda z.\quad y \qquad z \quad x)\ (\lambda w.x)$$
$$\lambda \quad (\lambda \quad \lambda \qquad\qquad\qquad)\ (\lambda \qquad)$$

$$\begin{array}{llllllll}\lambda x.(\lambda y.\ \lambda z. & y & z & x) & (\lambda w. & x) \\ \lambda\ \ (\lambda\ \ \lambda & 1 & 0 & 2) & (\lambda & 1)\end{array}$$

$$
\begin{array}{lllllllll}
 & \lambda x.(\lambda y. & \lambda z. & y & z & x) & (\lambda w. x) \\
 & \lambda \ (\lambda & \lambda & 1 & 0 & 2) & (\lambda \quad 1) \\
\to & \lambda x.(& \lambda z.(\lambda w. x) & & z & x) &
\end{array}
$$

$$
\begin{array}{lllllllll}
& \lambda x.(\lambda y.\ \lambda z. & y & z & x) & (\lambda w.\ x) \\
& \lambda\ \ (\lambda\ \ \ \lambda & 1 & 0 & 2) & (\lambda\ \ \ 1) \\
\rightarrow & \lambda x.(\ \ \ \lambda z.(\lambda w.\ x) & & z & x) & \\
\rightarrow & \lambda\ \ (\ \ \ \lambda\ \ (\lambda\ \ \ ) & & &) &
\end{array}
$$

$$
\begin{array}{llllllll}
 & \lambda x.(\lambda y. & \lambda z. & y & z & x) & (\lambda w.x) \\
 & \lambda\ \ (\lambda & \lambda & 1 & 0 & 2) & (\lambda\ \ 1) \\
\longrightarrow & \lambda x.(& \lambda z.(\lambda w.x) & & z & x) & \\
\longrightarrow & \lambda\ \ (& \lambda\ \ (\lambda\ \ 2) & & 0 & 1) &
\end{array}
$$

$$\begin{array}{llllllll} & \lambda x.(\lambda y.\ \lambda z. & y & z & x) & (\lambda w. x) \\ & \lambda\ \ (\lambda\ \ \lambda & 1 & 0 & 2) & (\lambda\ \ 1) \\ \rightarrow & \lambda x.(\ \ \lambda z.(\lambda w. x) & & z & x) & \\ \rightarrow & \lambda\ \ (\ \ \lambda\ \ (\lambda\ \ 2) & & 0 & 1) & \end{array}$$

Leave a bound variable alone

Adjust a free variable down when a lambda is removed

Adjust a free variable up when a term enters a deeper context

$$\begin{array}{llllllll} & \lambda x.(\lambda y. & \lambda z. & y & z & x) & (\lambda w. x) \\ & \lambda \; (\lambda & \lambda & 1 & 0 & 2) & (\lambda \; 1) \\ \longrightarrow & \lambda x.(& \lambda z.(\lambda w. x) & z & x) & & \\ \longrightarrow & \lambda \; (& \lambda \; (\lambda \; 2) & 0 & 1) & & \end{array}$$

Leave a bound variable alone

Adjust a free variable down when a lambda is removed

Adjust a free variable up when a term enters a deeper context

$\uparrow_b^d$ "Leave b bound variables alone; add d to the free variables"

$$\uparrow_b^d x = \begin{cases} x & \text{if } x < b \\ x + d & \text{otherwise} \end{cases}$$

A bound variable: do not touch

A free variable: adjust

$$\uparrow_b^d \lambda M = \lambda \uparrow_{b+1}^d M$$

Entering a lambda: remember additional bound variable

$$\uparrow_b^d M_1 \, M_2 = \uparrow_b^d M_1 \uparrow_b^d M_2$$

Recurse on application

Beta Reduction with de Bruijn Indices

$$\begin{aligned}
\uparrow^d_b x &= \begin{cases} x & \text{if } x < b \\ x + d & \text{otherwise} \end{cases} \\
\uparrow^d_b \lambda M &= \lambda \uparrow^d_{b+1} M \\
\uparrow^d_b M_1\, M_2 &= \uparrow^d_b M_1 \uparrow^d_b M_2
\end{aligned}$$

Beta Reduction with de Bruijn Indices

$$\uparrow^d_b x = \begin{cases} x & \text{if } x < b \\ x + d & \text{otherwise} \end{cases}$$

$$\uparrow^d_b \lambda M = \lambda \uparrow^d_{b+1} M$$

$$\uparrow^d_b M_1\, M_2 = \uparrow^d_b M_1 \uparrow^d_b M_2$$

$$y[x := N] = \begin{cases} N & \text{if } y = x \\ y & \text{otherwise} \end{cases}$$

Base case: substitute N for x and leave everything else alone

Beta Reduction with de Bruijn Indices

$$\uparrow^d_b x = \begin{cases} x & \text{if } x < b \\ x + d & \text{otherwise} \end{cases}$$

$$\uparrow^d_b \lambda M = \lambda \uparrow^d_{b+1} M$$

$$\uparrow^d_b M_1\, M_2 = \uparrow^d_b M_1 \uparrow^d_b M_2$$

$$y[x := N] = \begin{cases} N & \text{if } y = x \\ y & \text{otherwise} \end{cases}$$

$$(\lambda M)[x := N] = \lambda M[(x+1) := \uparrow^1_0 N]$$

Base case: substitute N for x and leave everything else alone

Entering a λ: "x" now named "$x + 1$" and all free variables in N should be one higher

Beta Reduction with de Bruijn Indices

$$\uparrow^d_b x = \begin{cases} x & \text{if } x < b \\ x + d & \text{otherwise} \end{cases}$$

$$\uparrow^d_b \lambda\, M = \lambda \uparrow^d_{b+1} M$$

$$\uparrow^d_b M_1\, M_2 = \uparrow^d_b M_1 \uparrow^d_b M_2$$

$$y[x := N] = \begin{cases} N & \text{if } y = x \\ y & \text{otherwise} \end{cases}$$

$$(\lambda\, M)[x := N] = \lambda\, M[(x+1) := \uparrow^1_0 N]$$

$$(M_1\, M_2)[x := N] = M_1[x := N]\, M_2[x := N]$$

Base case: substitute N for x and leave everything else alone

Entering a λ: "x" now named "$x + 1$" and all free variables in N should be one higher

Just recurse on application

Beta Reduction with de Bruijn Indices

$$\uparrow^d_b x = \begin{cases} x & \text{if } x < b \\ x + d & \text{otherwise} \end{cases} \qquad y[x := N] = \begin{cases} N & \text{if } y = x \\ y & \text{otherwise} \end{cases}$$

$$\uparrow^d_b \lambda\, M = \lambda\, \uparrow^d_{b+1} M \qquad (\lambda\, M)[x := N] = \lambda\, M[(x+1) := \uparrow^1_0 N]$$

$$\uparrow^d_b M_1\, M_2 = \uparrow^d_b M_1\, \uparrow^d_b M_2 \qquad (M_1\, M_2)[x := N] = M_1[x := N]\, M_2[x := N]$$

$$(\lambda M)N \longrightarrow_\beta\, \uparrow^{-1}_0 (M[0 := \uparrow^1_0 N])$$

Base case: substitute N for x and leave everything else alone

Entering a λ: "x" now named "$x + 1$" and all free variables in N should be one higher

Just recurse on application

Beta reduction:
Adjust N's free variables for the scope of M; replace the bound variable "0" with it in M.
Finally, correct the free variables in the result since the λ was removed.

Beta Reduction with de Bruijn Indices

$$\uparrow^d_b x = \begin{cases} x & \text{if } x < b \\ x + d & \text{otherwise} \end{cases} \qquad y[x := N] = \begin{cases} N & \text{if } y = x \\ y & \text{otherwise} \end{cases}$$

$$\uparrow^d_b \lambda M = \lambda \uparrow^d_{b+1} M \qquad (\lambda M)[x := N] = \lambda M[(x+1) := \uparrow^1_0 N]$$

$$\uparrow^d_b M_1 M_2 = \uparrow^d_b M_1 \uparrow^d_b M_2 \qquad (M_1 M_2)[x := N] = M_1[x := N] M_2[x := N]$$

$$(\lambda M)N \longrightarrow_\beta \uparrow^{-1}_0 (M[0 := \uparrow^1_0 N])$$

$\lambda y. \lambda x. (\lambda u. \lambda v. u\, x)\, y$

Beta Reduction with de Bruijn Indices

$$\uparrow^d_b x = \begin{cases} x & \text{if } x < b \\ x + d & \text{otherwise} \end{cases} \qquad y[x := N] = \begin{cases} N & \text{if } y = x \\ y & \text{otherwise} \end{cases}$$

$$\uparrow^d_b \lambda M = \lambda \uparrow^d_{b+1} M \qquad (\lambda M)[x := N] = \lambda M[(x+1) := \uparrow^1_0 N]$$

$$\uparrow^d_b M_1 M_2 = \uparrow^d_b M_1 \uparrow^d_b M_2 \qquad (M_1 M_2)[x := N] = M_1[x := N]\, M_2[x := N]$$

$$(\lambda M)N \longrightarrow_\beta \uparrow^{-1}_0 (M[0 := \uparrow^1_0 N])$$

$$\lambda y.\, \lambda x.\, (\lambda u.\, \lambda v.\, u\, x)\, y$$

$$= (\lambda\, \lambda\, 1\; 2)\; 1$$

Switch to de Bruijn indices

Beta Reduction with de Bruijn Indices

$$\uparrow^d_b x = \begin{cases} x & \text{if } x < b \\ x + d & \text{otherwise} \end{cases} \qquad y[x := N] = \begin{cases} N & \text{if } y = x \\ y & \text{otherwise} \end{cases}$$

$$\uparrow^d_b \lambda M = \lambda \uparrow^d_{b+1} M \qquad (\lambda M)[x := N] = \lambda M[(x+1) := \uparrow^1_0 N]$$

$$\uparrow^d_b M_1 M_2 = \uparrow^d_b M_1 \uparrow^d_b M_2 \qquad (M_1 M_2)[x := N] = M_1[x := N]\, M_2[x := N]$$

$$(\lambda M)N \longrightarrow_\beta \uparrow^{-1}_0 (M[0 := \uparrow^1_0 N])$$

$\lambda y.\, \lambda x.\, (\lambda u.\, \lambda v.\, u\, x)\, y$

$= (\lambda \lambda\, 1\, 2)\, 1$ — Switch to de Bruijn indices

$= \uparrow^{-1}_0 (\lambda\, 1\, 2)[0 := \uparrow^1_0 1]$ — Apply beta reduction rule

Beta Reduction with de Bruijn Indices

$$\uparrow^d_b x = \begin{cases} x & \text{if } x < b \\ x + d & \text{otherwise} \end{cases} \qquad y[x := N] = \begin{cases} N & \text{if } y = x \\ y & \text{otherwise} \end{cases}$$

$$\uparrow^d_b \lambda M = \lambda \uparrow^d_{b+1} M \qquad (\lambda M)[x := N] = \lambda M[(x+1) := \uparrow^1_0 N]$$

$$\uparrow^d_b M_1 M_2 = \uparrow^d_b M_1 \uparrow^d_b M_2 \qquad (M_1 M_2)[x := N] = M_1[x := N]\, M_2[x := N]$$

$$(\lambda M)N \longrightarrow_\beta \uparrow^{-1}_0 (M[0 := \uparrow^1_0 N])$$

$\lambda y.\, \lambda x.\, (\lambda u.\, \lambda v.\, u\, x)\, y$

$= (\lambda \lambda 1\ 2)\ 1$	Switch to de Bruijn indices
$= \uparrow^{-1}_0 (\lambda 1\ 2)[0 := \uparrow^1_0 1]$	Apply beta reduction rule
$= \uparrow^{-1}_0 (\lambda 1\ 2)[0 := 2]$	Rewrite free variable $\uparrow^1_0 1$

Beta Reduction with de Bruijn Indices

$$\uparrow^d_b x = \begin{cases} x & \text{if } x < b \\ x + d & \text{otherwise} \end{cases} \qquad y[x := N] = \begin{cases} N & \text{if } y = x \\ y & \text{otherwise} \end{cases}$$

$$\uparrow^d_b \lambda M = \lambda \uparrow^d_{b+1} M \qquad (\lambda M)[x := N] = \lambda M[(x+1) := \uparrow^1_0 N]$$

$$\uparrow^d_b M_1 M_2 = \uparrow^d_b M_1 \uparrow^d_b M_2 \qquad (M_1 M_2)[x := N] = M_1[x := N] M_2[x := N]$$

$$(\lambda M)N \longrightarrow_\beta \uparrow^{-1}_0 (M[0 := \uparrow^1_0 N])$$

$\lambda y. \lambda x. (\lambda u. \lambda v. u\, x)\, y$

$= (\lambda \lambda 1\ 2)\ 1$	Switch to de Bruijn indices
$= \uparrow^{-1}_0 (\lambda 1\ 2)[0 := \uparrow^1_0 1]$	Apply beta reduction rule
$= \uparrow^{-1}_0 (\lambda 1\ 2)[0 := 2]$	Rewrite free variable $\uparrow^1_0 1$
$= \uparrow^{-1}_0 \lambda (1\ 2)[1 := \uparrow^1_0 2]$	Enter λ: replace 1 with N rewritten for new scope

Beta Reduction with de Bruijn Indices

$$\uparrow^d_b x = \begin{cases} x & \text{if } x < b \\ x + d & \text{otherwise} \end{cases} \qquad y[x := N] = \begin{cases} N & \text{if } y = x \\ y & \text{otherwise} \end{cases}$$

$$\uparrow^d_b \lambda M = \lambda \uparrow^d_{b+1} M \qquad (\lambda M)[x := N] = \lambda M[(x+1) := \uparrow^1_0 N]$$

$$\uparrow^d_b M_1 M_2 = \uparrow^d_b M_1 \uparrow^d_b M_2 \qquad (M_1 M_2)[x := N] = M_1[x := N]\, M_2[x := N]$$

$$(\lambda M)N \longrightarrow_\beta \uparrow^{-1}_0 (M[0 := \uparrow^1_0 N])$$

$\lambda y.\, \lambda x.\, (\lambda u.\, \lambda v.\, u\, x)\, y$

$= (\lambda\, \lambda\, 1\; 2)\, 1$	Switch to de Bruijn indices
$= \uparrow^{-1}_0 (\lambda\, 1\; 2)[0 := \uparrow^1_0 1]$	Apply beta reduction rule
$= \uparrow^{-1}_0 (\lambda\, 1\; 2)[0 := 2]$	Rewrite free variable $\uparrow^1_0 1$
$= \uparrow^{-1}_0 \lambda\, (1\; 2)[1 := \uparrow^1_0 2]$	Enter λ: replace 1 with N rewritten for new scope
$= \uparrow^{-1}_0 \lambda\, (1\; 2)[1 := 3]$	Rewrite free variable $\uparrow^1_0 2$

Beta Reduction with de Bruijn Indices

$$\uparrow^d_b x = \begin{cases} x & \text{if } x < b \\ x + d & \text{otherwise} \end{cases} \qquad y[x := N] = \begin{cases} N & \text{if } y = x \\ y & \text{otherwise} \end{cases}$$

$$\uparrow^d_b \lambda M = \lambda \uparrow^d_{b+1} M \qquad (\lambda M)[x := N] = \lambda M[(x+1) := \uparrow^1_0 N]$$

$$\uparrow^d_b M_1 M_2 = \uparrow^d_b M_1 \uparrow^d_b M_2 \qquad (M_1 M_2)[x := N] = M_1[x := N]\ M_2[x := N]$$

$$(\lambda M)N \longrightarrow_\beta \uparrow^{-1}_0 (M[0 := \uparrow^1_0 N])$$

$\lambda y.\, \lambda x.\, (\lambda u.\, \lambda v.\, u\, x)\, y$

$= (\lambda\, \lambda\, 1\, 2)\, 1$	Switch to de Bruijn indices
$= \uparrow^{-1}_0 (\lambda\, 1\, 2)[0 := \uparrow^1_0 1]$	Apply beta reduction rule
$= \uparrow^{-1}_0 (\lambda\, 1\, 2)[0 := 2]$	Rewrite free variable $\uparrow^1_0 1$
$= \uparrow^{-1}_0 \lambda\, (1\, 2)[1 := \uparrow^1_0 2]$	Enter λ: replace 1 with N rewritten for new scope
$= \uparrow^{-1}_0 \lambda\, (1\, 2)[1 := 3]$	Rewrite free variable $\uparrow^1_0 2$
$= \uparrow^{-1}_0 \lambda\, 1[1 := 3]\ 2[1 := 3]$	Recurse into application

Beta Reduction with de Bruijn Indices

$$\uparrow_b^d x = \begin{cases} x & \text{if } x < b \\ x + d & \text{otherwise} \end{cases} \qquad y[x := N] = \begin{cases} N & \text{if } y = x \\ y & \text{otherwise} \end{cases}$$

$$\uparrow_b^d \lambda M = \lambda \uparrow_{b+1}^d M \qquad (\lambda M)[x := N] = \lambda M[(x+1) := \uparrow_0^1 N]$$

$$\uparrow_b^d M_1 M_2 = \uparrow_b^d M_1 \uparrow_b^d M_2 \qquad (M_1 M_2)[x := N] = M_1[x := N] M_2[x := N]$$

$$(\lambda M)N \longrightarrow_\beta \uparrow_0^{-1} (M[0 := \uparrow_0^1 N])$$

$\lambda y.\, \lambda x.\, (\lambda u.\, \lambda v.\, u\, x)\, y$

$= (\lambda\, \lambda\, 1\, 2)\, 1$	Switch to de Bruijn indices
$= \uparrow_0^{-1} (\lambda\, 1\, 2)[0 := \uparrow_0^1 1]$	Apply beta reduction rule
$= \uparrow_0^{-1} (\lambda\, 1\, 2)[0 := 2]$	Rewrite free variable $\uparrow_0^1 1$
$= \uparrow_0^{-1} \lambda\, (1\, 2)[1 := \uparrow_0^1 2]$	Enter λ: replace 1 with N rewritten for new scope
$= \uparrow_0^{-1} \lambda\, (1\, 2)[1 := 3]$	Rewrite free variable $\uparrow_0^1 2$
$= \uparrow_0^{-1} \lambda\, 1[1 := 3]\; 2[1 := 3]$	Recurse into application
$= \uparrow_0^{-1} \lambda\, 3\, 2$	Substitute 3 for 1; leave 2 unchanged

Beta Reduction with de Bruijn Indices

$$\uparrow^d_b x = \begin{cases} x & \text{if } x < b \\ x + d & \text{otherwise} \end{cases} \qquad y[x := N] = \begin{cases} N & \text{if } y = x \\ y & \text{otherwise} \end{cases}$$

$$\uparrow^d_b \lambda M = \lambda \uparrow^d_{b+1} M \qquad (\lambda M)[x := N] = \lambda M[(x+1) := \uparrow^1_0 N]$$

$$\uparrow^d_b M_1 M_2 = \uparrow^d_b M_1 \uparrow^d_b M_2 \qquad (M_1 M_2)[x := N] = M_1[x := N]\, M_2[x := N]$$

$$(\lambda M)N \longrightarrow_\beta \uparrow^{-1}_0 (M[0 := \uparrow^1_0 N])$$

$\lambda y.\, \lambda x.\, (\lambda u.\, \lambda v.\, u\, x)\, y$

$= (\lambda\, \lambda\, 1\, 2)\, 1$	Switch to de Bruijn indices
$= \uparrow^{-1}_0 (\lambda\, 1\, 2)[0 := \uparrow^1_0 1]$	Apply beta reduction rule
$= \uparrow^{-1}_0 (\lambda\, 1\, 2)[0 := 2]$	Rewrite free variable $\uparrow^1_0 1$
$= \uparrow^{-1}_0 \lambda\, (1\, 2)[1 := \uparrow^1_0 2]$	Enter λ: replace 1 with N rewritten for new scope
$= \uparrow^{-1}_0 \lambda\, (1\, 2)[1 := 3]$	Rewrite free variable $\uparrow^1_0 2$
$= \uparrow^{-1}_0 \lambda\, 1[1 := 3]\, 2[1 := 3]$	Recurse into application
$= \uparrow^{-1}_0 \lambda\, 3\, 2$	Substitute 3 for 1; leave 2 unchanged
$= \lambda\, 2\, 1$	Shift the free variables down by 1

Beta Reduction with de Bruijn Indices

$$\uparrow^d_b x = \begin{cases} x & \text{if } x < b \\ x + d & \text{otherwise} \end{cases} \qquad y[x := N] = \begin{cases} N & \text{if } y = x \\ y & \text{otherwise} \end{cases}$$

$$\uparrow^d_b \lambda M = \lambda \uparrow^d_{b+1} M \qquad (\lambda M)[x := N] = \lambda M[(x+1) := \uparrow^1_0 N]$$

$$\uparrow^d_b M_1 M_2 = \uparrow^d_b M_1 \uparrow^d_b M_2 \qquad (M_1 M_2)[x := N] = M_1[x := N]\, M_2[x := N]$$

$$(\lambda M)N \longrightarrow_\beta \uparrow^{-1}_0 (M[0 := \uparrow^1_0 N])$$

$\lambda y.\, \lambda x.\, (\lambda u.\, \lambda v.\, u\, x)\, y$		
	$= (\lambda\, \lambda\, 1\, 2)\, 1$	Switch to de Bruijn indices
	$= \uparrow^{-1}_0 (\lambda\, 1\, 2)[0 := \uparrow^1_0 1]$	Apply beta reduction rule
	$= \uparrow^{-1}_0 (\lambda\, 1\, 2)[0 := 2]$	Rewrite free variable $\uparrow^1_0 1$
	$= \uparrow^{-1}_0 \lambda\, (1\, 2)[1 := \uparrow^1_0 2]$	Enter λ: replace 1 with N rewritten for new scope
	$= \uparrow^{-1}_0 \lambda\, (1\, 2)[1 := 3]$	Rewrite free variable $\uparrow^1_0 2$
	$= \uparrow^{-1}_0 \lambda\, 1[1 := 3]\; 2[1 := 3]$	Recurse into application
	$= \uparrow^{-1}_0 \lambda\, 3\, 2$	Substitute 3 for 1; leave 2 unchanged
	$= \lambda\, 2\, 1$	Shift the free variables down by 1
$\lambda y.\, \lambda x.\; \lambda v.\, y\, x$		Switch to named variables