λ-Terms, M

are built up from a given, countable collection of

- variables $x, y, z, \ldots$

by two operations for forming λ-terms:

- λ-abstraction: $(\lambda x.M)$
  (where x is a variable and M is a λ-term)
- application: $(M\,M')$
  (where M and M' are λ-terms).

Some random examples of λ-terms:

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

α-Equivalence $M =_\alpha M'$

is the binary relation inductively generated by the rules:

$$\frac{}{x =_\alpha x} \qquad \frac{z \# (M\,N) \qquad M\{z/x\} =_\alpha N\{z/y\}}{\lambda x.M =_\alpha \lambda y.N}$$

$$\frac{M =_\alpha M' \qquad N =_\alpha N'}{M\,N =_\alpha M'\,N'}$$

where $M\{z/x\}$ is M with all occurrences of x replaced by z.

Substitution $N[M/x]$

$$
\begin{array}{rcll}
x[M/x] & = & M & \\
y[M/x] & = & y & \text{if } y \neq x \\
(\lambda y.N)[M/x] & = & \lambda y.N[M/x] & \text{if } y \# (M\,x) \\
(N_1\,N_2)[M/x] & = & N_1[M/x]\,N_2[M/x] &
\end{array}
$$

$N[M/x]$ = result of replacing all <u>free</u> occurrences of x in N with M, avoiding <u>capture</u> of free variables in M by binders in N

Substitution $N[M/x]$

$$
\begin{array}{rcll}
x[M/x] & = & M & \\
y[M/x] & = & y & \text{if } y \neq x \\
(\lambda y.N)[M/x] & = & \lambda y.N[M/x] & \text{if } y \# (M\,x) \\
(N_1\,N_2)[M/x] & = & N_1[M/x]\,N_2[M/x] &
\end{array}
$$

Side-condition $y \# (M\,x)$ (y does not occur in M and $y \neq x$) makes substitution "capture-avoiding".

E.g. if $x \neq y$

$$(\lambda y.x)[y/x] \neq \lambda y.y$$

Substitution $N[M/x]$

$$\begin{array}{rcll} x[M/x] & = & M & \\ y[M/x] & = & y & \text{if } y \neq x \\ (\lambda y.N)[M/x] & = & \lambda y.N[M/x] & \text{if } y \,\#\, (M\, x) \\ (N_1\, N_2)[M/x] & = & N_1[M/x]\, N_2[M/x] & \end{array}$$

Side-condition $y \# (M\, x)$ (y does not occur in M and $y \neq x$) makes substitution "capture-avoiding".

E.g. if $x \neq y \neq z \neq x$

$$(\lambda y.x)[y/x] =_\alpha (\lambda z.x)[y/x] = \lambda z.y$$

$N \mapsto N[M/x]$ induces a total operation on α-equivalence classes.

$$\lambda x.(\lambda y.y)\,y\,x\left[\lambda z.y/y\right]$$

$=$

$$\lambda x.(\lambda y.y)\,y\,x\,[\,\lambda z.y/y\,]$$ no possible capture

$=$

$$\lambda x. (\lambda y. y)\, y\, x \, [\lambda z. y / y]$$

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

$$\lambda x. (\lambda y. y)\, x\, y \, [\lambda y. x / y]$$

$$=$$

$$\lambda x.(\lambda y.y)\,y\,x\,[\lambda x.y/y]$$

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

$$\lambda x.(\lambda y.y)\,x\,y\,[\lambda y.x/y]$$ possible capture

$$=$$

$$\lambda x.(\lambda y.y)\,y\,x\,[\lambda x.y/y]$$

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

$$\lambda x.(\lambda y.y)\,x\,y\,[\lambda y.x/y]$$ possible capture...

$$=_\alpha \lambda z.(\lambda y.y)\,z\,y\,[\lambda y.x/y]$$ ...α-convert to avoid

$$\lambda x.(\lambda y.y)\,y\,x\,[\lambda x.y/y]$$

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

$$\lambda x.(\lambda y.y)\,x\,y\,[\lambda y.x/y]$$ possible capture...

$$=_\alpha \lambda z.(\lambda y.y)\,z\,y\,[\lambda y.x/y]$$ ...α-convert to avoid

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

$$=_\alpha \lambda z.(\lambda y.y)\,z\,(\lambda y'.x)$$

β-Reduction

Recall that $\lambda x.M$ is intended to represent the function f such that $f(x) = M$ for all x. We can regard $\lambda x.M$ as a function on λ-terms via substitution: map each N to $M[N/x]$.

So the natural notion of computation for λ-terms is given by stepping from a

β-redex $\quad (\lambda x.M)N$

to the corresponding

β-reduct $\quad M[N/x]$

β-Reduction

One-step β-reduction, $M \to M'$:

$$\frac{}{(\lambda x.M)N \to M[N/x]} \qquad \frac{M \to M'}{\lambda x.M \to \lambda x.M'}$$

$$\frac{M \to M'}{M\,N \to M'\,N} \qquad \frac{M \to M'}{N\,M \to N\,M'}$$

$$\frac{N =_\alpha M \qquad M \to M' \qquad M' =_\alpha N'}{N \to N'}$$

β-Reduction

E.g.

$$
\begin{tikzcd}
 & ((\lambda y.\lambda z.z)u)y \arrow[dr] & & \\
(\lambda x.x\,y)((\lambda y.\lambda z.z)u) \arrow[ur] \arrow[dr] & & (\lambda z.z)y \arrow[r] & y \\
 & (\lambda x.x\,y)(\lambda z.z) \arrow[ur] & &
\end{tikzcd}
$$

β-Reduction

E.g.

$$
\begin{tikzcd}
 & ((\lambda y.\lambda z.z)\,u)\,y \arrow[dr] & & \\
(\lambda x.x\,y)((\lambda y.\lambda z.z)\,u) \arrow[ur] \arrow[dr] & & (\lambda z.z)\,y \arrow[r] & y \\
 & (\lambda x.x\,y)(\lambda z.z) \arrow[ur] & &
\end{tikzcd}
$$

β-Reduction

E.g.

$$
\begin{tikzcd}[column sep=small]
 & ((\lambda y.\lambda z.z)u)y \arrow[dr] & & \\
(\lambda x.x\,y)((\lambda y.\lambda z.z)u) \arrow[ur] \arrow[dr] & & (\lambda z.z)y \arrow[r] & y \\
 & (\lambda x.x\,y)(\lambda z.z) \arrow[ur] & &
\end{tikzcd}
$$

β-Reduction

E.g.

$$
\begin{tikzcd}[column sep=small]
 & ((\lambda y.\lambda z.z)u)y \arrow[dr] & & \\
(\lambda x.x\,y)((\lambda y.\lambda z.z)u) \arrow[ur] \arrow[dr] & & (\lambda z.z)y \arrow[r] & y \\
 & (\lambda x.x\,y)(\lambda z.z) \arrow[ur] & &
\end{tikzcd}
$$

β-Reduction

E.g.

$$
\begin{tikzcd}
& ((\lambda y.\lambda z.z)u)y \arrow[dr] & & \\
(\lambda x.x\,y)((\lambda y.\lambda z.z)u) \arrow[ur] \arrow[dr] & & (\lambda z.z)y \arrow[r] & y \\
& (\lambda x.x\,y)(\lambda z.z) \arrow[ur] & &
\end{tikzcd}
$$

β-Reduction

E.g.

$$
\begin{tikzcd}[column sep=small]
 & ((\lambda y.\lambda z.z)u)y \arrow[dr] & & \\
(\lambda x.x\,y)((\lambda y.\lambda z.z)u) \arrow[ur] \arrow[dr] & & (\lambda z.z)y \arrow[r] & y \\
 & (\lambda x.x\,y)(\lambda z.z) \arrow[ur] & &
\end{tikzcd}
$$

E.g. of "up to α-equivalence" aspect of reduction:

$$(\lambda x.\lambda y.x)y =_\alpha (\lambda x.\lambda z.x)y \to \lambda z.y$$

Many-step β-reduction, $M \twoheadrightarrow M'$:

$$\frac{M =_\alpha M'}{M \twoheadrightarrow M'} \qquad \frac{M \to M'}{M \twoheadrightarrow M'} \qquad \frac{M \twoheadrightarrow M' \quad M' \to M''}{M \twoheadrightarrow M''}$$

(no steps) (1 step) (1 more step)

E.g.

$$(\lambda x.x\, y)((\lambda y\, z.z)u) \twoheadrightarrow y$$

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

β-Conversion $M =_\beta N$

Informally: $M =_\beta N$ holds if N can be obtained from M by performing zero or more steps of α-equivalence, β-reduction, or β-*expansion* ($=$ inverse of a reduction).

E.g. $u\,((\lambda x\,y.\,v\,x)y) =_\beta (\lambda x.\,u\,x)(\lambda x.\,v\,y)$

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

and so we have

$$
\begin{array}{lcll}
u\,((\lambda x\,y.\,v\,x)y) & =_\alpha & u\,((\lambda x\,y'.\,v\,x)y) & \\
 & \to & u(\lambda y'.\,v\,y) & \text{reduction} \\
 & =_\alpha & u(\lambda x.\,v\,y) & \\
 & \leftarrow & (\lambda x.\,u\,x)(\lambda x.\,v\,y) & \text{expansion}
\end{array}
$$

β-Conversion $M =_\beta N$

is the binary relation inductively generated by the rules:

$$\frac{M =_\alpha M'}{M =_\beta M'} \qquad \frac{M \to M'}{M =_\beta M'} \qquad \frac{M =_\beta M'}{M' =_\beta M}$$

$$\frac{M =_\beta M' \qquad M' =_\beta M''}{M =_\beta M''} \qquad \frac{M =_\beta M'}{\lambda x.M =_\beta \lambda x.M'}$$

$$\frac{M =_\beta M' \qquad N =_\beta N'}{M\,N =_\beta M'\,N'}$$

Church-Rosser Theorem

Theorem. $\twoheadrightarrow$ is confluent, that is, if $M_1 \twoheadleftarrow M \twoheadrightarrow M_2$, then there exists M' such that $M_1 \twoheadrightarrow M' \twoheadleftarrow M_2$.

[Proof omitted.]

Church-Rosser Theorem

Theorem. $\twoheadrightarrow$ is confluent, that is, if $M_1 \twoheadleftarrow M \twoheadrightarrow M_2$, then there exists M' such that $M_1 \twoheadrightarrow M' \twoheadleftarrow M_2$.

Corollary. Two show that two terms are β-convertible, it suffices to show that they both reduce to the same term. More precisely: $M_1 =_\beta M_2$ iff $\exists M \, (M_1 \twoheadrightarrow M \twoheadleftarrow M_2)$.

Church-Rosser Theorem

Theorem. $\twoheadrightarrow$ is confluent, that is, if $M_1 \twoheadleftarrow M \twoheadrightarrow M_2$, then there exists M' such that $M_1 \twoheadrightarrow M' \twoheadleftarrow M_2$.

Corollary. $M_1 =_\beta M_2$ iff $\exists M \, (M_1 \twoheadrightarrow M \twoheadleftarrow M_2)$.

Proof. $=_\beta$ satisfies the rules generating $\twoheadrightarrow$; so $M \twoheadrightarrow M'$ implies $M =_\beta M'$. Thus if $M_1 \twoheadrightarrow M \twoheadleftarrow M_2$, then $M_1 =_\beta M =_\beta M_2$ and so $M_1 =_\beta M_2$.

Conversely, the relation $\{(M_1, M_2) \mid \exists M \, (M_1 \twoheadrightarrow M \twoheadleftarrow M_2)\}$ satisfies the rules generating $=_\beta$: the only difficult case is closure of the relation under transitivity and for this we use the Church-Rosser theorem: $M_1 \longrightarrow\!\!\!\!\!\rightarrow M \twoheadleftarrow\!\!\!\!\!\longleftarrow M_2 \longrightarrow\!\!\!\!\!\rightarrow M' \twoheadleftarrow\!\!\!\!\!\longleftarrow M_3$

Church-Rosser Theorem

Theorem. $\twoheadrightarrow$ is confluent, that is, if $M_1 \twoheadleftarrow M \twoheadrightarrow M_2$, then there exists M' such that $M_1 \twoheadrightarrow M' \twoheadleftarrow M_2$.

Corollary. $M_1 =_\beta M_2$ iff $\exists M\,(M_1 \twoheadrightarrow M \twoheadleftarrow M_2)$.

Proof. $=_\beta$ satisfies the rules generating $\twoheadrightarrow$; so $M \twoheadrightarrow M'$ implies $M =_\beta M'$. Thus if $M_1 \twoheadrightarrow M \twoheadleftarrow M_2$, then $M_1 =_\beta M =_\beta M_2$ and so $M_1 =_\beta M_2$.

Conversely, the relation $\{(M_1, M_2) \mid \exists M\,(M_1 \twoheadrightarrow M \twoheadleftarrow M_2)\}$ satisfies the rules generating $=_\beta$: the only difficult case is closure of the relation under transitivity and for this we use the Church-Rosser theorem:

$$\begin{tikzcd}
M_1 \arrow[r, two heads] & M \arrow[dr, two heads] & M_2 \arrow[l, two heads] \arrow[r, two heads] \arrow[d, phantom, "\text{C-R}"] & M' \arrow[dl, two heads] & M_3 \arrow[l, two heads] \\
& & M_2' & &
\end{tikzcd}$$

Church-Rosser Theorem

Theorem. $\twoheadrightarrow$ is confluent, that is, if $M_1 \twoheadleftarrow M \twoheadrightarrow M_2$, then there exists M' such that $M_1 \twoheadrightarrow M' \twoheadleftarrow M_2$.

Corollary. $M_1 =_\beta M_2$ iff $\exists M\,(M_1 \twoheadrightarrow M \twoheadleftarrow M_2)$.

Proof. $=_\beta$ satisfies the rules generating $\twoheadrightarrow$; so $M \twoheadrightarrow M'$ implies $M =_\beta M'$. Thus if $M_1 \twoheadrightarrow M \twoheadleftarrow M_2$, then $M_1 =_\beta M =_\beta M_2$ and so $M_1 =_\beta M_2$.

Conversely, the relation $\{(M_1, M_2) \mid \exists M\,(M_1 \twoheadrightarrow M \twoheadleftarrow M_2)\}$ satisfies the rules generating $=_\beta$: the only difficult case is closure of the relation under transitivity and for this we use the Church-Rosser theorem. Hence $M_1 =_\beta M_2$ implies $\exists M\,(M_1 \twoheadrightarrow M' \twoheadleftarrow M_2)$.

β-Normal Forms

Definition. A λ-term N is in β-normal form (nf) if it contains no β-redexes (no sub-terms of the form $(\lambda x.M)M'$). M has β-nf N if $M =_\beta N$ with N a β-nf.

β-Normal Forms

Definition. A λ-term N is in β-normal form (nf) if it contains no β-redexes (no sub-terms of the form $(\lambda x.M)M'$). M has β-nf N if $M =_\beta N$ with N a β-nf.

Note that if N is a β-nf and $N \twoheadrightarrow N'$, then it must be that $N =_\alpha N'$ (why?).

Hence if $N_1 =_\beta N_2$ with N_1 and N_2 both β-nfs, then $N_1 =_\alpha N_2$. (For if $N_1 =_\beta N_2$, then ~~$N_1 \twoheadleftarrow M \twoheadrightarrow N_2$ for some M, hence~~ by Church-Rosser, $N_1 \twoheadrightarrow M' \twoheadleftarrow N_2$ for some M', so $N_1 =_\alpha M' =_\alpha N_2$.)

So the β-nf of M is unique up to α-equivalence if it exists.

Non-termination

Some λ terms have no β-nf.

E.g. $\Omega \triangleq (\lambda x.x\,x)(\lambda x.x\,x)$ satisfies

- $\Omega \to (x\,x)[(\lambda x.x\,x)/x] = \Omega$,
- $\Omega \twoheadrightarrow M$ implies $\Omega =_\alpha M$.

So there is no β-nf N such that $\Omega =_\beta N$.

Non-termination

Some λ terms have no β-nf.

E.g. $\Omega \triangleq (\lambda x.x\,x)(\lambda x.x\,x)$ satisfies

- $\Omega \to (x\,x)[(\lambda x.x\,x)/x] = \Omega$,
- $\Omega \twoheadrightarrow M$ implies $\Omega =_\alpha M$.

So there is no β-nf N such that $\Omega =_\beta N$.

A term can possess both a β-nf and infinite chains of reduction from it.

E.g. $(\lambda x.y)\Omega \to y$, but also $(\lambda x.y)\Omega \to (\lambda x.y)\Omega \to \cdots$.

Non-termination

Normal-order reduction is a deterministic strategy for reducing λ-terms: reduce the "left-most, outer-most" redex first.

- left-most: reduce M before N in $M\,N$, and then
- outer-most: reduce $(\lambda x.M)N$ rather than either of M or N.

(cf. call-by-name evaluation).

Fact: normal-order reduction of M always reaches the β-nf of M if it possesses one.