

Recall: λ -Terms, M

are built up from a given, countable collection of

- ▶ variables x, y, z, \dots

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).

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 $M[N/x]$.

↑ result of substituting
 N for free x in M

Substitution $N[M/x]$

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

$N[M/x]$ = result of replacing all free occurrences of x in N with M , avoiding "capture" of free variables in M by λ -binders in N

Substitution $N[M/x]$

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

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 $N[M/x]$

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

Can always satisfy this  up to α -equivalence

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

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

In fact $N \mapsto N[M/x]$ induces a totally defined function from the set of α -equivalence classes of λ -terms to itself.

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

||

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

no possible capture

$$\begin{aligned} & \lambda x. (\lambda z. z) y x \left[\lambda z. y / y \right] \\ = & \lambda x. (\lambda z. z) (\lambda z. y) x \end{aligned}$$

$$\begin{aligned} & \lambda x. (\lambda u. u) x y \left[\lambda y. x / y \right] \\ = & \end{aligned}$$

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

$$\lambda x. (\lambda u. u) x y \left[\lambda y. x / y \right] \text{ possible capture}$$
$$=$$

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

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

possible capture...

$$=_{\alpha} \lambda z. (\lambda u. u) z y \left[\lambda y. x / y \right]$$

... α -convert to avoid

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

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

$$\lambda x. (\lambda u. u) x y \left[\lambda y. x / y \right] \quad \text{possible capture...}$$

$$\stackrel{\alpha}{=} \lambda z. (\lambda u. u) z y \left[\lambda y. x / y \right] \quad \text{...}\alpha\text{-convert to avoid}$$

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

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 $(\lambda x.M)N$

to the corresponding

β -reduct $M[N/x]$

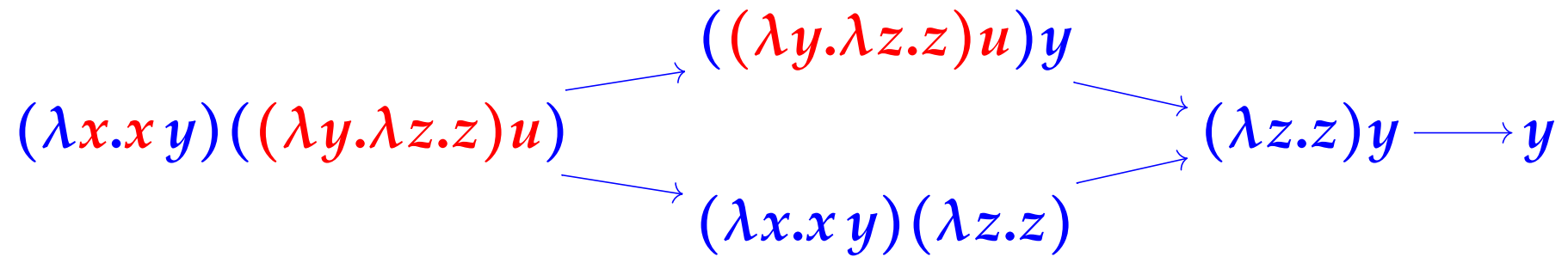
β -Reduction

One-step β -reduction, $M \rightarrow M'$:

$$\frac{}{(\lambda x.M)N \rightarrow M[N/x]} \qquad \frac{M \rightarrow M'}{\lambda x.M \rightarrow \lambda x.M'}$$
$$\frac{M \rightarrow M'}{MN \rightarrow M'N} \qquad \frac{M \rightarrow M'}{NM \rightarrow NM'}$$
$$\frac{N =_{\alpha} M \quad M \rightarrow M' \quad M' =_{\alpha} N'}{N \rightarrow N'}$$

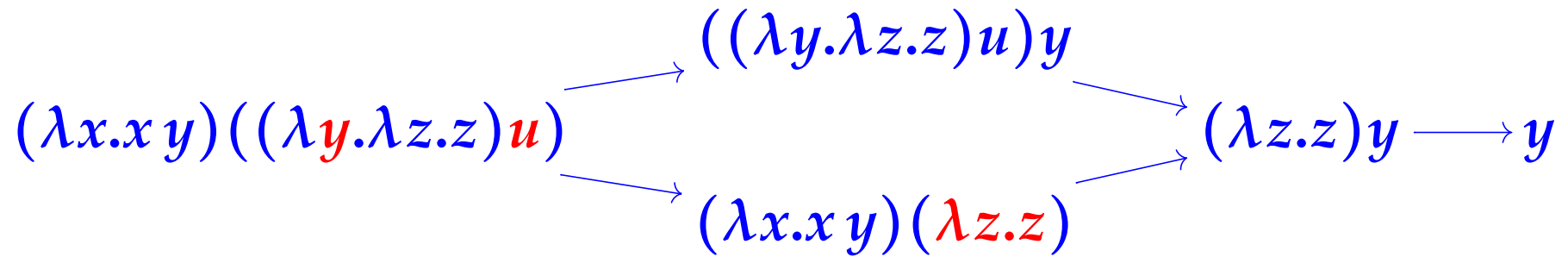
β -Reduction

E.g.



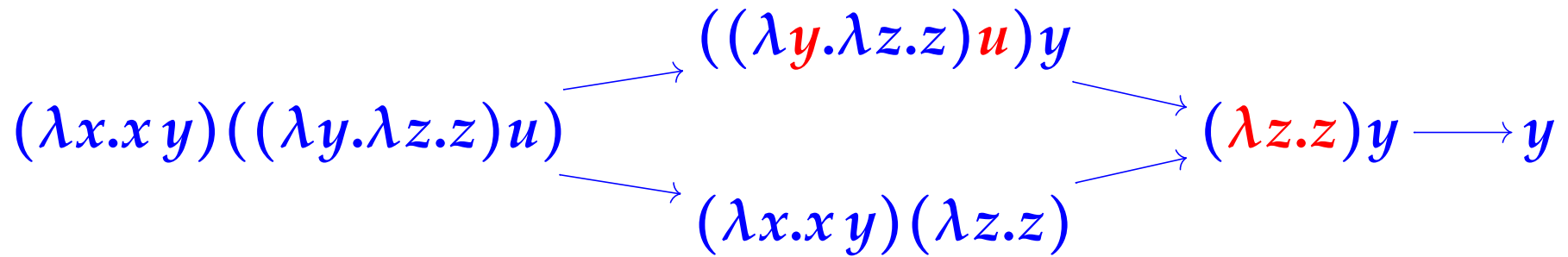
β -Reduction

E.g.



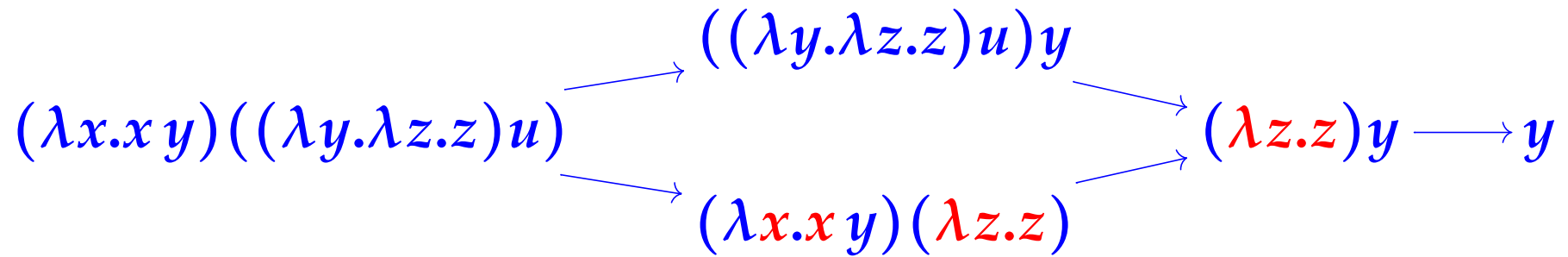
β -Reduction

E.g.



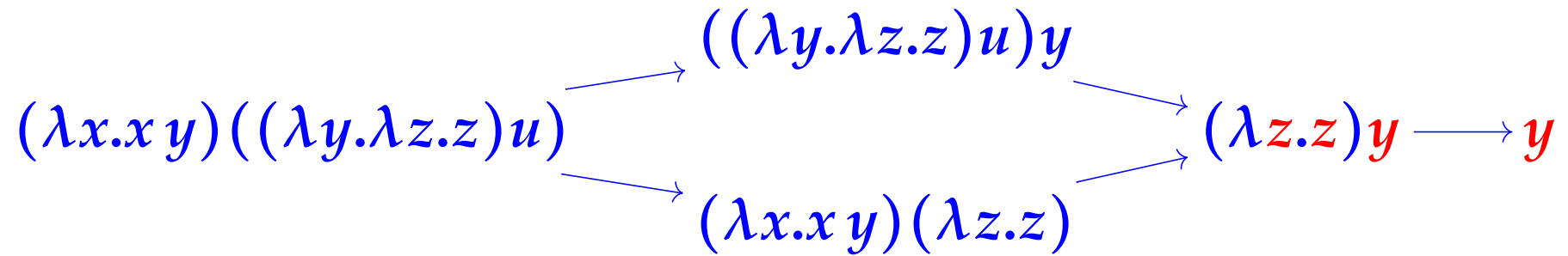
β -Reduction

E.g.



β -Reduction

E.g.



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

$$\frac{M =_{\alpha} M'}{M \twoheadrightarrow M'}$$

(no steps)

$$\frac{M \twoheadrightarrow M' \quad M' \rightarrow M''}{M \twoheadrightarrow M''}$$

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

and so we have

$u((\lambda x y. v x)y) =_{\alpha} u((\lambda x y'. v x)y)$

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

and so we have

$$\begin{array}{lcl} u((\lambda x y. v x)y) & =_{\alpha} & u((\lambda x y'. v x)y) \\ & \rightarrow & u(\lambda y'. v y) \end{array} \quad \text{reduction}$$

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

and so we have

$$\begin{aligned} u((\lambda x y. v x)y) &=_{\alpha} u((\lambda x y'. v x)y) \\ &\rightarrow u(\lambda y'. v y) && \text{reduction} \\ &=_{\alpha} u(\lambda x. v y) \end{aligned}$$

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

and so we have

$$\begin{aligned} u((\lambda x y. v x)y) &=_{\alpha} u((\lambda x y'. v x)y) \\ &\rightarrow u(\lambda y'. v y) && \text{reduction} \\ &=_{\alpha} u(\lambda x. v y) \\ &\leftarrow (\lambda x. u x)(\lambda x. v y) && \text{expansion} \end{aligned}$$

β -Conversion $M =_{\beta} N$

is the binary relation inductively generated by the rules:

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

$$\frac{M \rightarrow M'}{M =_{\beta} M'}$$

$$\frac{M =_{\beta} M'}{M' =_{\beta} M}$$

$$\frac{M =_{\beta} M' \quad M' =_{\beta} M''}{M =_{\beta} M''}$$

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

$$\frac{M =_{\beta} M' \quad N =_{\beta} N'}{M N =_{\beta} M' N'}$$

Church-Rosser Theorem

Theorem. \rightarrow is **confluent**, that is, if $M_1 \leftarrow M \rightarrow M_2$, then there exists M' such that $M_1 \rightarrow M' \leftarrow M_2$.

[Proof omitted.]

Church-Rosser Theorem

Theorem. \rightarrow is **confluent**, that is, if $M_1 \leftarrow M \rightarrow M_2$, then there exists M' such that $M_1 \rightarrow M' \leftarrow M_2$.

Corollary. To 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 $\exists M (M_1 \rightarrow M \leftarrow M_2)$.

Church-Rosser Theorem

Theorem. \rightarrow is **confluent**, that is, if $M_1 \leftarrow M \rightarrow M_2$, then there exists M' such that $M_1 \rightarrow M' \leftarrow M_2$.

Corollary. $M_1 =_{\beta} M_2$ iff $\exists M (M_1 \rightarrow M \leftarrow M_2)$.

Proof. $=_{\beta}$ satisfies the rules generating \rightarrow ; so $M \rightarrow M'$ implies $M =_{\beta} M'$. Thus if $M_1 \rightarrow M \leftarrow M_2$, then $M_1 =_{\beta} M =_{\beta} M_2$ and so $M_1 =_{\beta} M_2$.

Conversely,

Church-Rosser Theorem

Theorem. $\rightarrow\!\!\rightarrow$ is **confluent**, that is, if $M_1 \leftarrow M \rightarrow\!\!\rightarrow M_2$, then there exists M' such that $M_1 \rightarrow\!\!\rightarrow M' \leftarrow M_2$.

Corollary. $M_1 =_\beta M_2$ iff $\exists M (M_1 \rightarrow\!\!\rightarrow M \leftarrow M_2)$.

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

Conversely, the relation $\{(M_1, M_2) \mid \exists M (M_1 \rightarrow\!\!\rightarrow M \leftarrow M_2)\}$ 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: $M_1 \rightarrow\!\!\rightarrow M \leftarrow M_2 \rightarrow\!\!\rightarrow M' \leftarrow M_3$

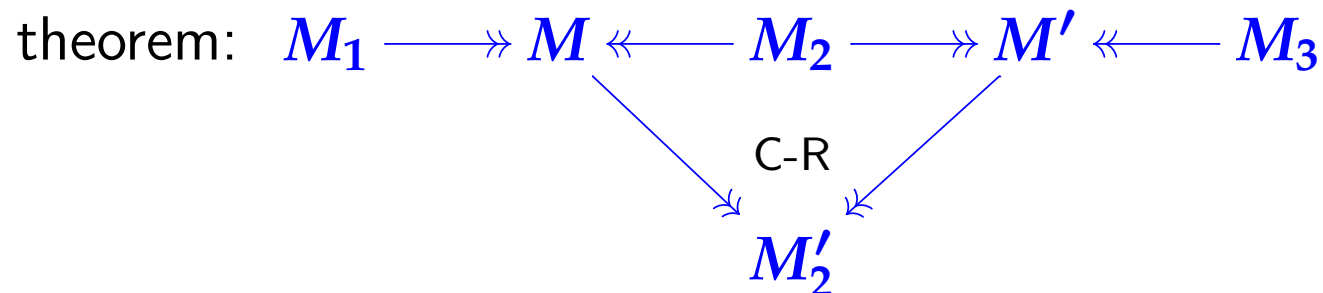
Church-Rosser Theorem

Theorem. $\rightarrow\!\!\rightarrow$ is **confluent**, that is, if $M_1 \leftarrow M \rightarrow\!\!\rightarrow M_2$, then there exists M' such that $M_1 \rightarrow\!\!\rightarrow M' \leftarrow M_2$.

Corollary. $M_1 =_\beta M_2$ iff $\exists M (M_1 \rightarrow\!\!\rightarrow M \leftarrow M_2)$.

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

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



Church-Rosser Theorem

Theorem. \rightarrow is **confluent**, that is, if $M_1 \leftarrow M \rightarrow M_2$, then there exists M' such that $M_1 \rightarrow M' \leftarrow M_2$.

Corollary. $M_1 =_{\beta} M_2$ iff $\exists M (M_1 \rightarrow M \leftarrow M_2)$.

Proof. $=_{\beta}$ satisfies the rules generating \rightarrow ; so $M \rightarrow M'$ implies $M =_{\beta} M'$. Thus if $M_1 \rightarrow M \leftarrow M_2$, then $M_1 =_{\beta} M =_{\beta} M_2$ and so $M_1 =_{\beta} M_2$.

Conversely, the relation $\{(M_1, M_2) \mid \exists M (M_1 \rightarrow M \leftarrow M_2)\}$ 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 $\exists M (M_1 \rightarrow M' \leftarrow M_2)$.

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 by Church-Rosser $N_1 \twoheadrightarrow M' \leftarrow N_2$ 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.

(and if M does have β -nf N , then
 $M \twoheadrightarrow N$)

Non-termination

Some λ terms have no β -nf.

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

- ▶ $\Omega \rightarrow (x x)[(\lambda x.x x)/x] = \Omega$,
- ▶ $\Omega \rightarrow M$ 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

- ▶ $\Omega \rightarrow (x x)[(\lambda x.x x)/x] = \Omega$,
- ▶ $\Omega \rightarrow M$ 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. $(\lambda x.y)\Omega \rightarrow y$, but also $(\lambda x.y)\Omega \rightarrow (\lambda x.y)\Omega \rightarrow \dots$.

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.

$$\frac{M_1 =_{\alpha} M_1' \quad M_1' \rightarrow_{\text{NOR}} M_2' \quad M_2' =_{\alpha} M_2}{M_1 \rightarrow_{\text{NOR}} M_2}$$

$$\frac{M \rightarrow_{\text{NOR}} M'}{\lambda x. M \rightarrow_{\text{NOR}} \lambda x. M'}$$

$$\frac{M_1 \rightarrow_{\text{NOR}} M_1'}{M_1 M_2 \rightarrow_{\text{NOR}} M_1' M_2}$$

$$\frac{}{(\lambda x. N) M \rightarrow_{\text{NOR}} N[M/x]}$$

$$\frac{M \rightarrow_{\text{NOR}} M'}{U M \rightarrow_{\text{NOR}} U M'}$$

where

$$\begin{cases} U ::= x \mid U N \\ N ::= \lambda x. N \mid U \end{cases}$$

β -normal forms

"neutral" forms