

Programming Language Concepts: Lecture 18

Madhavan Mukund

Chennai Mathematical Institute

`madhavan@cmi.ac.in`

`http://www.cmi.ac.in/~madhavan/courses/pl2009`

PLC 2009, Lecture 18, 30 March 2009

One step reduction

- ▶ Can have other reduction rules like β

One step reduction

- ▶ Can have other reduction rules like β
- ▶ Observe that $\lambda x.(Mx)$ and M are equivalent with respect to β -reduction

One step reduction

- ▶ Can have other reduction rules like β
- ▶ Observe that $\lambda x.(Mx)$ and M are equivalent with respect to β -reduction
- ▶ New reduction rule η

$$\lambda x.(Mx) \rightarrow_{\eta} M$$

One step reduction

- ▶ Can have other reduction rules like β
- ▶ Observe that $\lambda x.(Mx)$ and M are equivalent with respect to β -reduction
- ▶ New reduction rule η

$$\lambda x.(Mx) \rightarrow_{\eta} M$$

- ▶ Given basic rules $\beta, \eta, \dots$, we are allowed to use them “in any context”

One step reduction

- ▶ Can have other reduction rules like β
- ▶ Observe that $\lambda x.(Mx)$ and M are equivalent with respect to β -reduction
- ▶ New reduction rule η

$$\lambda x.(Mx) \rightarrow_{\eta} M$$

- ▶ Given basic rules $\beta, \eta, \dots$, we are allowed to use them “in any context”
- ▶ Define a one step reduction relation $\rightarrow$ inductively

$$\frac{M \rightarrow_x M'}{M \rightarrow M} \quad \frac{M \rightarrow M'}{\lambda x.M \rightarrow \lambda x.M'} \quad \frac{M \rightarrow M'}{MN \rightarrow M'N} \quad \frac{N \rightarrow N'}{MN \rightarrow MN'}$$

$x \in \{\beta, \eta, \dots\}$

Normal forms

- ▶ Computation — a maximal sequence of reduction steps

Normal forms

- ▶ Computation — a maximal sequence of reduction steps
- ▶ “Values” are expressions that cannot be further reduced:
normal forms

Normal forms

- ▶ Computation — a maximal sequence of reduction steps
- ▶ “Values” are expressions that cannot be further reduced:
normal forms
- ▶ Allow reduction in any context $\Rightarrow$ multiple expressions may qualify for reduction in one step

Normal forms

- ▶ Computation — a maximal sequence of reduction steps
- ▶ “Values” are expressions that cannot be further reduced:
normal forms
- ▶ Allow reduction in any context $\Rightarrow$ multiple expressions may qualify for reduction in one step

Natural questions

Normal forms

- ▶ Computation — a maximal sequence of reduction steps
- ▶ “Values” are expressions that cannot be further reduced:
normal forms
- ▶ Allow reduction in any context $\Rightarrow$ multiple expressions may qualify for reduction in one step

Natural questions

- ▶ Does every term reduce to a normal form?

Normal forms

- ▶ Computation — a maximal sequence of reduction steps
- ▶ “Values” are expressions that cannot be further reduced:
normal forms
- ▶ Allow reduction in any context $\Rightarrow$ multiple expressions may qualify for reduction in one step

Natural questions

- ▶ Does every term reduce to a normal form?
- ▶ Can a term reduce to more than one normal form, depending on order reduction strategy?

Normal forms

- ▶ Computation — a maximal sequence of reduction steps
- ▶ “Values” are expressions that cannot be further reduced:
normal forms
- ▶ Allow reduction in any context $\Rightarrow$ multiple expressions may qualify for reduction in one step

Natural questions

- ▶ Does every term reduce to a normal form?
- ▶ Can a term reduce to more than one normal form, depending on order reduction strategy?
- ▶ If a term has a normal form, can we always find it?

Normal forms ...

Does every term reduce to a normal form?

- ▶ Consider $(\lambda x.xx)(\lambda x.xx)$

Normal forms ...

Does every term reduce to a normal form?

- ▶ Consider $(\lambda x.xx)(\lambda x.xx)$
- ▶ $(\lambda x.xx)(\lambda x.xx) \rightarrow_{\beta} (\lambda x.xx)(\lambda x.xx)$
 - ▶ Reduction never terminates

Normal forms . . .

Does every term reduce to a normal form?

- ▶ Consider $(\lambda x.xx)(\lambda x.xx)$
- ▶ $(\lambda x.xx)(\lambda x.xx) \rightarrow_{\beta} (\lambda x.xx)(\lambda x.xx)$
 - ▶ Reduction never terminates
- ▶ Call this term Ω

Normal forms . . .

Does every term reduce to a normal form?

- ▶ Consider $(\lambda x.xx)(\lambda x.xx)$
- ▶ $(\lambda x.xx)(\lambda x.xx) \rightarrow_{\beta} (\lambda x.xx)(\lambda x.xx)$
 - ▶ Reduction never terminates
- ▶ Call this term Ω

Normal forms ...

Can a term reduce to more than one normal form, depending on order reduction strategy?

- ▶ Consider $\langle \text{False} \rangle \Omega = (\lambda yz.z)((\lambda x.xx)(\lambda x.xx))$

Normal forms ...

Can a term reduce to more than one normal form, depending on order reduction strategy?

- ▶ Consider $\langle \text{False} \rangle \Omega = (\lambda yz.z)((\lambda x.xx)(\lambda x.xx))$
- ▶ Outermost reduction:
 $(\lambda yz.z)((\lambda x.xx)(\lambda x.xx)) \rightarrow \lambda z.z$

Normal forms . . .

Can a term reduce to more than one normal form, depending on order reduction strategy?

- ▶ Consider $\langle \text{False} \rangle \Omega = (\lambda yz.z)((\lambda x.xx)(\lambda x.xx))$
- ▶ Outermost reduction:
 $(\lambda yz.z)((\lambda x.xx)(\lambda x.xx)) \rightarrow \lambda z.z$
- ▶ Innermost reduction:
 $(\lambda yz.z)((\lambda x.xx)(\lambda x.xx)) \rightarrow (\lambda yz.z)((\lambda x.xx)(\lambda x.xx)) \rightarrow \dots$

Normal forms ...

Can a term reduce to more than one normal form, depending on order reduction strategy?

- ▶ Consider $\langle \text{False} \rangle \Omega = (\lambda yz.z)((\lambda x.xx)(\lambda x.xx))$
- ▶ Outermost reduction:
 $(\lambda yz.z)((\lambda x.xx)(\lambda x.xx)) \rightarrow \lambda z.z$
- ▶ Innermost reduction:
 $(\lambda yz.z)((\lambda x.xx)(\lambda x.xx)) \rightarrow (\lambda yz.z)((\lambda x.xx)(\lambda x.xx)) \rightarrow \dots$
- ▶ Choice of reduction strategies may determine whether a normal form is reached ...

Normal forms ...

Can a term reduce to more than one normal form, depending on order reduction strategy?

- ▶ Consider $\langle \text{False} \rangle \Omega = (\lambda yz.z)((\lambda x.xx)(\lambda x.xx))$
- ▶ Outermost reduction:
 $(\lambda yz.z)((\lambda x.xx)(\lambda x.xx)) \rightarrow \lambda z.z$
- ▶ Innermost reduction:
 $(\lambda yz.z)((\lambda x.xx)(\lambda x.xx)) \rightarrow (\lambda yz.z)((\lambda x.xx)(\lambda x.xx)) \rightarrow \dots$
- ▶ Choice of reduction strategies may determine whether a normal form is reached ...
- ▶ ... but the question is, can more than one normal form be reached?

Normal forms . . .

If a term has a normal form, can we always find it?

Normal forms . . .

If a term has a normal form, can we always find it?

- ▶ We have seen how to encode recursive functions in λ -calculus

Normal forms . . .

If a term has a normal form, can we always find it?

- ▶ We have seen how to encode recursive functions in λ -calculus
- ▶ Given a recursive function f and an argument n , we cannot determine, in general, if computation of $f(n)$ terminates

Normal forms . . .

If a term has a normal form, can we always find it?

- ▶ We have seen how to encode recursive functions in λ -calculus
- ▶ Given a recursive function f and an argument n , we cannot determine, in general, if computation of $f(n)$ terminates
- ▶ Computing $f(n)$ is equivalent to asking if $\langle f \rangle \langle n \rangle$ achieves a normal form

Normal forms ...

Can a term reduce to more than one normal form, depending on order reduction strategy?

- Define an equivalence relation $\leftrightarrow$ on λ -terms

$$M \leftrightarrow N \text{ iff } \exists P. P \rightarrow^* M, P \rightarrow^* N$$

$M \leftrightarrow N$ if both M and N can be obtained by reduction from a common “ancestor” P

Normal forms ...

Can a term reduce to more than one normal form, depending on order reduction strategy?

- Define an equivalence relation $\leftrightarrow$ on λ -terms

$$M \leftrightarrow N \text{ iff } \exists P. P \rightarrow^* M, P \rightarrow^* N$$

$M \leftrightarrow N$ if both M and N can be obtained by reduction from a common “ancestor” P

- $\leftrightarrow$ is the **symmetric transitive closure** of $\rightarrow^*$

$$\frac{M \rightarrow^* N}{M \leftrightarrow N} \quad \frac{M \leftrightarrow N}{N \leftrightarrow M} \quad \frac{M \leftrightarrow N, N \leftrightarrow P}{M \leftrightarrow P}$$

Normal forms . . .

Can a term reduce to more than one normal form, depending on order reduction strategy?

- ▶ Define an equivalence relation $\leftrightarrow$ on λ -terms

$$M \leftrightarrow N \text{ iff } \exists P. P \rightarrow^* M, P \rightarrow^* N$$

$M \leftrightarrow N$ if both M and N can be obtained by reduction from a common “ancestor” P

- ▶ $\leftrightarrow$ is the **symmetric transitive closure** of $\rightarrow^*$

$$\frac{M \rightarrow^* N}{M \leftrightarrow N} \quad \frac{M \leftrightarrow N}{N \leftrightarrow M} \quad \frac{M \leftrightarrow N, N \leftrightarrow P}{M \leftrightarrow P}$$

- ▶ In general, for any reflexive, transitive relation R , can define the symmetric, transitive closure $\overset{R}{\leftrightarrow}$

Church-Rosser Theorem

Diamond property or Church-Rosser property

- ▶ Let R be any reflexive, transitive relation (such as $\rightarrow^*$)
- ▶ R has the **diamond property** if, whenever $X R Y$ and $X R Z$ there is W such that $Y R W$ and $Z R W$

Church-Rosser Theorem

Diamond property or Church-Rosser property

- ▶ Let R be any reflexive, transitive relation (such as $\rightarrow^*$)
- ▶ R has the **diamond property** if, whenever $X R Y$ and $X R Z$ there is W such that $Y R W$ and $Z R W$

Theorem [Church-Rosser]

Let R be Church-Rosser. Then $M \stackrel{R}{\leftrightarrow} N$ implies there exists Z , $M R Z$ and $N R Z$

Church-Rosser Theorem

Diamond property or Church-Rosser property

- ▶ Let R be any reflexive, transitive relation (such as $\rightarrow^*$)
- ▶ R has the **diamond property** if, whenever $X R Y$ and $X R Z$ there is W such that $Y R W$ and $Z R W$

Theorem [Church-Rosser]

Let R be Church-Rosser. Then $M \stackrel{R}{\leftrightarrow} N$ implies there exists Z , $M R Z$ and $N R Z$

Proof By induction on the definition of $\stackrel{R}{\leftrightarrow}$

Church-Rosser Theorem

Corollary [Church-Rosser]

*Let R be a reduction relation that is Church-Rosser.
Then a term can have at most one normal form with
respect to R*

Church-Rosser Theorem

Corollary [Church-Rosser]

*Let R be a reduction relation that is Church-Rosser.
Then a term can have at most one normal form with
respect to R*

Proof By picture

Church-Rosser Theorem

Is $\rightarrow^*$ Church-Rosser?

Church-Rosser Theorem

Is $\rightarrow^*$ Church-Rosser?

Consider $(\lambda x.xx)((\lambda x.x)(\lambda x.x))$

Church-Rosser Theorem

Is $\rightarrow^*$ Church-Rosser?

Consider $(\lambda x.xx)((\lambda x.x)(\lambda x.x))$

Two possible reductions

- ▶ $(\lambda x.xx)((\lambda x.x)(\lambda x.x)) \rightarrow$
 $((\lambda x.x)(\lambda x.x))((\lambda x.x)(\lambda x.x))$ (Outermost)
- ▶ $(\lambda x.xx)((\lambda x.x)(\lambda x.x)) \rightarrow ((\lambda x.xx)(\lambda x.x))$ (Innermost)

Church-Rosser Theorem

Is $\rightarrow^*$ Church-Rosser?

Consider $(\lambda x.xx)((\lambda x.x)(\lambda x.x))$

Two possible reductions

- ▶ $(\lambda x.xx)((\lambda x.x)(\lambda x.x)) \rightarrow$
 $((\lambda x.x)(\lambda x.x))((\lambda x.x)(\lambda x.x))$ (Outermost)
- ▶ $(\lambda x.xx)((\lambda x.x)(\lambda x.x)) \rightarrow ((\lambda x.xx)(\lambda x.x))$ (Innermost)

From second option, in one step we get

$$(\lambda x.xx)(\lambda x.x) \rightarrow ((\lambda x.x)(\lambda x.x))$$

Church-Rosser Theorem

Is $\rightarrow^*$ Church-Rosser?

Consider $(\lambda x.xx)((\lambda x.x)(\lambda x.x))$

Two possible reductions

- ▶ $(\lambda x.xx)((\lambda x.x)(\lambda x.x)) \rightarrow ((\lambda x.x)(\lambda x.x))((\lambda x.x)(\lambda x.x))$ (Outermost)
- ▶ $(\lambda x.xx)((\lambda x.x)(\lambda x.x)) \rightarrow ((\lambda x.xx)(\lambda x.x))$ (Innermost)

From second option, in one step we get

$$(\lambda x.xx)(\lambda x.x) \rightarrow ((\lambda x.x)(\lambda x.x))$$

Can reach this term from the first option as well, but it requires **two** steps!

Church-Rosser Theorem

Solution: Define a new notion of one step reduction $\rightarrow$ such that

- ▶ This new reduction is Church-Rosser.
- ▶ Its reflexive, transitive closure is equal to $\rightarrow^*$.

Church-Rosser Theorem

Solution: Define a new notion of one step reduction $\twoheadrightarrow$ such that

- ▶ This new reduction is Church-Rosser.
- ▶ Its reflexive, transitive closure is equal to $\rightarrow^*$.

Define $\twoheadrightarrow$ as follows.

$$\begin{array}{c} M \twoheadrightarrow M \\[10pt] \frac{M \twoheadrightarrow M', N \twoheadrightarrow N'}{MN \twoheadrightarrow M'N'} \\[10pt] \frac{M \twoheadrightarrow M'}{\lambda x.M \twoheadrightarrow \lambda x.M'} \\[10pt] \frac{M \twoheadrightarrow M', N \twoheadrightarrow N'}{(\lambda x.M)N \twoheadrightarrow M'\{x \leftarrow N'\}} \end{array}$$

- ▶ $\twoheadrightarrow$ combines nonoverlapping $\rightarrow$ reductions into one parallel step

Recursive definitions

Suppose $F = \lambda x_1 x_2 \dots x_n E$, where E contains an occurrence of F

- ▶ Choose a new variable f
- ▶ Convert E to E^* replacing every F in E by ff
 - ▶ If E is of the form $\dots F \dots F \dots$ then E^* is $\dots (ff) \dots (ff) \dots$.

Now write

$$\begin{aligned} G &= \lambda f x_1 x_2 \dots x_n. E^* \\ &= \lambda f x_1 x_2 \dots x_n. \dots (ff) \dots (ff) \dots \end{aligned}$$

Then

$$GG = \lambda x_1 x_2 \dots x_n. \dots (GG) \dots (GG) \dots$$

- ▶ GG satisfies the equation defining F
- ▶ Write $F = GG$, where $G = \lambda f x_1 x_2 \dots x_n. E^*$.

Fixed point combinator

- ▶ Consider recursive definition $F = \lambda x.x(Fx)$

Fixed point combinator

- ▶ Consider recursive definition $F = \lambda x.x(Fx)$
- ▶ Can use the GG trick to get a λ -expression of F

$$\begin{aligned} F &= GG, \text{ where } G = \lambda fx.x(ffx) \\ &= \lambda fx.x(\lambda fx.x(ffx)\lambda fx.x(ffx)x) \end{aligned}$$

Fixed point combinator

- ▶ Consider recursive definition $F = \lambda x.x(Fx)$
- ▶ Can use the GG trick to get a λ -expression of F

$$\begin{aligned} F &= GG, \text{ where } G = \lambda fx.x(ffx) \\ &= \lambda fx.x(\lambda fx.x(ffx)\lambda fx.x(ffx)x) \end{aligned}$$

- ▶ Note that $FX = X(FX)$ for any term X

Fixed point combinator

- ▶ Consider recursive definition $F = \lambda x.x(Fx)$
- ▶ Can use the GG trick to get a λ -expression of F

$$\begin{aligned} F &= GG, \text{ where } G = \lambda fx.x(ffx) \\ &= \lambda fx.x(\lambda fx.x(ffx)\lambda fx.x(ffx)x) \end{aligned}$$

- ▶ Note that $FX = X(FX)$ for any term X
- ▶ Fixed point : Given Z , M such that $ZM = M$.

Fixed point combinator

- ▶ Consider recursive definition $F = \lambda x.x(Fx)$
- ▶ Can use the GG trick to get a λ -expression of F

$$\begin{aligned} F &= GG, \text{ where } G = \lambda fx.x(ffx) \\ &= \lambda fx.x(\lambda fx.x(ffx)\lambda fx.x(ffx)x) \end{aligned}$$

- ▶ Note that $FX = X(FX)$ for any term X
- ▶ Fixed point : Given Z , M such that $ZM = M$.
- ▶ FZ is a fixed point for Z

Fixed point combinator

- ▶ Consider recursive definition $F = \lambda x.x(Fx)$
- ▶ Can use the GG trick to get a λ -expression of F

$$\begin{aligned} F &= GG, \text{ where } G = \lambda fx.x(ffx) \\ &= \lambda fx.x(\lambda fx.x(ffx)\lambda fx.x(ffx)x) \end{aligned}$$

- ▶ Note that $FX = X(FX)$ for any term X
- ▶ Fixed point : Given Z , M such that $ZM = M$.
- ▶ FZ is a fixed point for Z
- ▶ Due to Turing — Θ

Terms without normal forms

Are all terms without normal forms equally “meaningless”?

Can we define an equivalence $\approx$ on λ -terms such that:

- ▶ $(\lambda x M)N \approx M\{x \leftarrow N\}$ —that is, $\approx$ the equivalence induced by the β reduction.

Terms without normal forms

Are all terms without normal forms equally “meaningless”?

Can we define an equivalence $\approx$ on λ -terms such that:

- ▶ $(\lambda x.M)N \approx M\{x \leftarrow N\}$ —that is, $\approx$ the equivalence induced by the β reduction.
- ▶ If M and N do not have normal forms, then $M \equiv N$.

Terms without normal forms

Are all terms without normal forms equally “meaningless”?

Can we define an equivalence $\approx$ on λ -terms such that:

- ▶ $(\lambda x.M)N \approx M\{x \leftarrow N\}$ —that is, $\approx$ the equivalence induced by the β reduction.
- ▶ If M and N do not have normal forms, then $M \equiv N$.
- ▶ Functions that are equated by $\approx$ yield equivalent results for the same arguments. That is, if $M \approx N$ then for all R , $MR \approx NR$.

Terms without normal forms

Consider the function F defined by

$$Fxb = \text{if } b \text{ then } x \text{ else } (Fxb)$$

Terms without normal forms

Consider the function F defined by

$$Fxb = \text{if } b \text{ then } x \text{ else } (Fxb)$$

If we unravel FF , we get

$$F = GG, \text{ where } G = \lambda fxb. (\text{if } b \text{ then } x \text{ else } (ffxb))$$

Terms without normal forms

Consider the function F defined by

$$Fxb = \text{if } b \text{ then } x \text{ else } (Fxb)$$

If we unravel FF , we get

$$F = GG, \text{ where } G = \lambda fxb. (\text{if } b \text{ then } x \text{ else } (ffxb))$$

Consider $FX\langle true \rangle$ and $FX\langle false \rangle$

Terms without normal forms

Consider the function F defined by

$$Fxb = \text{if } b \text{ then } x \text{ else } (Fxb)$$

If we unravel FF , we get

$$F = GG, \text{ where } G = \lambda fxb. (\text{if } b \text{ then } x \text{ else } (ffxb))$$

Consider $FX\langle true \rangle$ and $FX\langle false \rangle$

- $FX\langle true \rangle \rightarrow \text{if } \langle T \rangle \text{ then } X \text{ else } (FX\langle true \rangle) \rightarrow X.$

Terms without normal forms

Consider the function F defined by

$$Fxb = \text{if } b \text{ then } x \text{ else } (Fxb)$$

If we unravel FF , we get

$$F = GG, \text{ where } G = \lambda fxb. (\text{if } b \text{ then } x \text{ else } (ffxb))$$

Consider $FX\langle true \rangle$ and $FX\langle false \rangle$

- ▶ $FX\langle true \rangle \rightarrow \text{if } \langle T \rangle \text{ then } X \text{ else } (FX\langle true \rangle) \rightarrow X.$
- ▶ $FX\langle false \rangle \rightarrow \text{if } \langle F \rangle \text{ then } X \text{ else } (FX\langle false \rangle) \rightarrow FX\langle false \rangle.$

Terms without normal forms

$FZ \rightarrow (\lambda x b. (\text{if } b \text{ then } x \text{ else } (Fxb)))Z$
 $\rightarrow \lambda b. (\text{if } b \text{ then } Z \text{ else } (FZb))$
 $\rightarrow \lambda b. (\text{if } b \text{ then } Z \text{ else } G), \text{ where } G = \text{if } b \text{ then } Z \text{ else } (fZB)$
 $\rightarrow \lambda b. (\text{if } b \text{ then } Z \text{ else } (\text{if } b \text{ then } Z \text{ else } G))$
 $\rightarrow \dots$

Terms without normal forms

$FZ \rightarrow (\lambda x b. (\text{if } b \text{ then } x \text{ else } (Fxb)))Z$
 $\rightarrow \lambda b. (\text{if } b \text{ then } Z \text{ else } (FZb))$
 $\rightarrow \lambda b. (\text{if } b \text{ then } Z \text{ else } G), \text{ where } G = \text{if } b \text{ then } Z \text{ else } (fZB)$
 $\rightarrow \lambda b. (\text{if } b \text{ then } Z \text{ else } (\text{if } b \text{ then } Z \text{ else } G))$
 $\rightarrow \dots$

- FZ does not terminate for any $Z \Rightarrow FX \approx FY$ for all X, Y

Terms without normal forms

$FZ \rightarrow (\lambda x b. (\text{if } b \text{ then } x \text{ else } (Fxb)))Z$
 $\rightarrow \lambda b. (\text{if } b \text{ then } Z \text{ else } (FZb))$
 $\rightarrow \lambda b. (\text{if } b \text{ then } Z \text{ else } G), \text{ where } G = \text{if } b \text{ then } Z \text{ else } (fZB)$
 $\rightarrow \lambda b. (\text{if } b \text{ then } Z \text{ else } (\text{if } b \text{ then } Z \text{ else } G))$
 $\rightarrow \dots$

- ▶ FZ does not terminate for any $Z \Rightarrow FX \approx FY$ for all X, Y
- ▶ $FX \approx FY$ implies $FXM \approx FYM$ for all M

Terms without normal forms

$FZ \rightarrow (\lambda x b. (\text{if } b \text{ then } x \text{ else } (Fxb)))Z$
 $\rightarrow \lambda b. (\text{if } b \text{ then } Z \text{ else } (FZb))$
 $\rightarrow \lambda b. (\text{if } b \text{ then } Z \text{ else } G), \text{ where } G = \text{if } b \text{ then } Z \text{ else } (fZB)$
 $\rightarrow \lambda b. (\text{if } b \text{ then } Z \text{ else } (\text{if } b \text{ then } Z \text{ else } G))$
 $\rightarrow \dots$

- ▶ FZ does not terminate for any $Z \Rightarrow FX \approx FY$ for all X, Y
- ▶ $FX \approx FY$ implies $FXM \approx FYM$ for all M
- ▶ $FX\langle \text{true} \rangle \approx FY\langle \text{true} \rangle$

Terms without normal forms

$FZ \rightarrow (\lambda x b. (\text{if } b \text{ then } x \text{ else } (Fxb)))Z$
 $\rightarrow \lambda b. (\text{if } b \text{ then } Z \text{ else } (FZb))$
 $\rightarrow \lambda b. (\text{if } b \text{ then } Z \text{ else } G), \text{ where } G = \text{if } b \text{ then } Z \text{ else } (fZB)$
 $\rightarrow \lambda b. (\text{if } b \text{ then } Z \text{ else } (\text{if } b \text{ then } Z \text{ else } G))$
 $\rightarrow \dots$

- ▶ FZ does not terminate for any $Z \Rightarrow FX \approx FY$ for all X, Y
- ▶ $FX \approx FY$ implies $FXM \approx FYM$ for all M
- ▶ $FX\langle \text{true} \rangle \approx FY\langle \text{true} \rangle$
- ▶ $FZ\langle \text{true} \rangle \rightarrow Z$ for all Z , so $X \approx Y$ for all X and Y !