universität innsbruck

WS 2023 lecture 13

Computability Theory

Aart Middeldorp

Outline

1. **Summary of Previous Lecture**
2. **β–Reduction**
3. **Church–Rosser Theorem**
4. **λ–Definability**
5. **η–Reduction**
6. **Normalization Theorem**
7. **Test Practice**
8. **Summary**

Definition

Hilbert system (for implication fragment) consists of two axioms and modus ponens:

$$\frac{}{\varphi \to \psi \to \varphi} \qquad \frac{}{(\varphi \to \psi \to \chi) \to (\varphi \to \psi) \to \varphi \to \chi} \qquad \frac{\varphi \qquad \varphi \to \psi}{\psi}$$

Deduction Theorem

$$\Gamma \cup \{\varphi\} \vdash_{\mathsf{h}} \psi \quad \iff \quad \Gamma \vdash_{\mathsf{h}} \varphi \to \psi$$

Theorem

Hilbert system is sound and complete with respect to Kripke models for implication fragment:

$$\Gamma \vdash_{\mathsf{h}} \varphi \quad \iff \quad \Gamma \Vdash \varphi$$

Theorem (Curry–Howard)

1. if $\Gamma \vdash t : \tau$ then $\text{types}(\Gamma) \vdash_h \tau$
2. if $\Gamma \vdash_h \varphi$ then $\Delta \vdash t : \varphi$ for some t and Δ with $\text{types}(\Delta) = \Gamma$

Definition

Hilbert system for intuitionistic propositional logic consists of modus ponens and axioms

① $\varphi \to \psi \to \varphi$

② $(\varphi \to \psi \to \chi) \to (\varphi \to \psi) \to \varphi \to \chi$

③ $\varphi \vee \psi \to \varphi$

④ $\varphi \vee \psi \to \psi$

⑤ $\varphi \to \psi \to \varphi \wedge \psi$

⑥ $\varphi \to \varphi \vee \psi$

⑦ $\psi \to \varphi \vee \psi$

⑧ $(\varphi \to \chi) \to (\psi \to \chi) \to \varphi \vee \psi \to \chi$

⑨ $\bot \to \varphi$

Remarks

- $\neg\varphi$ is shortcut for $\varphi \to \bot$
- adding axiom $\varphi \vee \neg\varphi$ (law of excluded middle) gives (classical) propositional logic

Theorem

Hilbert system is sound and complete with respect to Kripke models:

$$\Gamma \vdash_{\mathsf{h}} \varphi \iff \Gamma \Vdash \varphi$$

Theorem (Finite Model Property)

$\vdash_{\mathsf{h}} \varphi \iff \mathcal{C} \Vdash \varphi$ for all finite Kripke models $\mathcal{C}$

Theorem (Glivenko 1929)

$\vdash \varphi \iff \vdash_{\mathsf{h}} \neg\neg\varphi$

Theorem

problem

instance: formula φ

question: $\vdash_{\mathsf{h}} \varphi$?

is decidable and PSPACE–complete

Definition (Gödel's Negative Translation)

- $p^{\mathsf{n}} = \neg\neg p$ for propositional atoms p
- $(\varphi \wedge \psi)^{\mathsf{n}} = \varphi^{\mathsf{n}} \wedge \psi^{\mathsf{n}}$
- $(\varphi \vee \psi)^{\mathsf{n}} = \neg(\neg\varphi^{\mathsf{n}} \wedge \neg\psi^{\mathsf{n}})$
- $(\varphi \to \psi)^{\mathsf{n}} = \varphi^{\mathsf{n}} \to \psi^{\mathsf{n}}$
- $\bot^{\mathsf{n}} = \bot$

Theorem

$\vdash \varphi \iff \vdash_{\mathsf{h}} \varphi^{\mathsf{n}}$

Part I: Recursive Function Theory

Ackermann function, bounded minimization, bounded recursion, course−of−values recursion, diagonalization, diophantine sets, elementary functions, fixed point theorem, Fibonacci numbers, Gödel numbering, Gödel's β function, Grzegorczyk hierarchy, loop programs, minimization, normal form theorem, partial recursive functions, primitive recursion, recursive enumerability, recursive inseparability, s−m−n theorem, total recursive functions, undecidability, while programs, ...

Part II: Combinatory Logic and Lambda Calculus

α−equivalence, abstraction, arithmetization, β−reduction, CL−representability, combinators, combinatorial completeness, Church numerals, Church−Rosser theorem, Curry−Howard isomorphism, de Bruijn notation, η−reduction, fixed point theorem, intuitionistic propositional logic, λ−definability, normalization theorem, termination, typing, undecidability, Z property, ...

Part I: Recursive Function Theory

Ackermann function, bounded minimization, bounded recursion, course–of–values recursion, diagonalization, diophantine sets, elementary functions, fixed point theorem, Fibonacci numbers, Gödel numbering, Gödel's β function, Grzegorczyk hierarchy, loop programs, minimization, normal form theorem, partial recursive functions, primitive recursion, recursive enumerability, recursive inseparability, s–m–n theorem, total recursive functions, undecidability, while programs, ...

Part II: Combinatory Logic and Lambda Calculus

α–equivalence, abstraction, arithmetization, β–reduction, CL–representability, combinators, combinatorial completeness, Church numerals, Church–Rosser theorem, Curry–Howard isomorphism, de Bruijn notation, η–reduction, fixed point theorem, intuitionistic propositional logic, λ–definability, normalization theorem, termination, typing, undecidability, Z property, ...

Literature (Combinatory Logic and Lambda Calculus)

- Henk Barendregt
  The Lambda Calculus, Its Syntax and Semantics
  North Holland, 1984
- Henk Barendregt, Wil Dekkers and Richard Statman
  Lambda Calculus with Types
  Cambridge University Press, 2013
- Herman Geuvers and Rob Nederpelt
  Type Theory and Formal Proof
  Cambridge University Press, 2014
- Chris Hankin
  An Introduction to Lambda Calculi for Computer Scientists
  King's College Publications, 2000
- J. Roger Hindley and Jonathan P. Seldin
  Lambda−Calculus and Combinators, an Introduction
  Cambridge University Press, 2008

Outline

1. Summary of Previous Lecture
2. **β–Reduction**
3. Church–Rosser Theorem
4. λ–Definability
5. η–Reduction
6. Normalization Theorem
7. Test Practice
8. Summary

Definition

set of lambda terms (Λ) is built from

- infinite set of variables $\mathcal{V} = \{x, y, z, \ldots\}$ $\qquad x \in \mathcal{V} \implies x \in \Lambda$

Definition

set of lambda terms (Λ) is built from

- infinite set of variables $\mathcal{V} = \{x, y, z, \ldots\}$ $\qquad x \in \mathcal{V} \implies x \in \Lambda$
- application $\qquad M, N \in \Lambda \implies (MN) \in \Lambda$

Definition

set of lambda terms (Λ) is built from

- infinite set of variables $\mathcal{V} = \{x, y, z, \ldots\}$ $\qquad x \in \mathcal{V} \implies x \in \Lambda$
- application $\qquad M, N \in \Lambda \implies (MN) \in \Lambda$
- abstraction $\qquad x \in \mathcal{V},\ M \in \Lambda \implies (\lambda x.M) \in \Lambda$

Definition

set of lambda terms (Λ) is built from

- infinite set of variables $\mathcal{V} = \{x, y, z, \ldots\}$ $\quad x \in \mathcal{V} \implies x \in \Lambda$
- application $\quad M, N \in \Lambda \implies (MN) \in \Lambda$
- abstraction $\quad x \in \mathcal{V},\ M \in \Lambda \implies (\lambda x.M) \in \Lambda$

Examples

$(\lambda x.x) \qquad ((\lambda x.(xx))(\lambda y.(yy))) \qquad (\lambda f.(\lambda x.(f(fx))))$

Definition

set of lambda terms (Λ) is built from

- infinite set of variables $\mathcal{V} = \{x, y, z, \ldots\}$ $\quad x \in \mathcal{V} \implies x \in \Lambda$
- application $\quad M, N \in \Lambda \implies (MN) \in \Lambda$
- abstraction $\quad x \in \mathcal{V},\ M \in \Lambda \implies (\lambda x.M) \in \Lambda$

Examples

$(\lambda x.x) \qquad ((\lambda x.(xx))(\lambda y.(yy))) \qquad (\lambda f.(\lambda x.(f(fx))))$

Backus–Naur Form

$$M, N \ ::= \ x \mid (MN) \mid (\lambda x.M)$$

Conventions

- outermost parentheses are omitted

Conventions

- outermost parentheses are omitted
- application is left-associative: MNP stands for $(MN)P$

Conventions

- outermost parentheses are omitted
- application is left-associative: MNP stands for $(MN)P$
- body of lambda abstraction extends as far right as possible:
  $\lambda x.MN$ abbreviates $\lambda x.(MN)$ and not $(\lambda x.M)N$

Conventions

- outermost parentheses are omitted
- application is left-associative: MNP stands for $(MN)P$
- body of lambda abstraction extends as far right as possible:
  $\lambda x.MN$ abbreviates $\lambda x.(MN)$ and not $(\lambda x.M)N$
- $\lambda xyz.M$ abbreviates $\lambda x.\lambda y.\lambda z.M$

Conventions

- outermost parentheses are omitted
- application is left-associative: MNP stands for $(MN)P$
- body of lambda abstraction extends as far right as possible:
  $\lambda x.MN$ abbreviates $\lambda x.(MN)$ and not $(\lambda x.M)N$
- $\lambda xyz.M$ abbreviates $\lambda x.\lambda y.\lambda z.M$

Terminology

$$\lambda x.M$$

- λx is binder

Conventions

- outermost parentheses are omitted
- application is left-associative: MNP stands for $(MN)P$
- body of lambda abstraction extends as far right as possible:
  $\lambda x.MN$ abbreviates $\lambda x.(MN)$ and not $(\lambda x.M)N$
- $\lambda xyz.M$ abbreviates $\lambda x.\lambda y.\lambda z.M$

Terminology

$$\lambda x.M$$

- λx is binder
- M is scope of binder λx

Conventions

- outermost parentheses are omitted
- application is left-associative: MNP stands for $(MN)P$
- body of lambda abstraction extends as far right as possible:
  $\lambda x.MN$ abbreviates $\lambda x.(MN)$ and not $(\lambda x.M)N$
- $\lambda xyz.M$ abbreviates $\lambda x.\lambda y.\lambda z.M$

Terminology

$$\lambda x.M$$

- λx is binder
- M is scope of binder λx
- occurrence of x in $\lambda x.M$ is bound

Notation

$M \equiv N$ if M and N are identical

Notation

$M \equiv N$ if M and N are identical

Definition

- set $\mathsf{FV}(M)$ of free variables of lambda term M is inductively defined:

$$\mathsf{FV}(x) = \{x\}$$
$$\mathsf{FV}(MN) = \mathsf{FV}(M) \cup \mathsf{FV}(N)$$
$$\mathsf{FV}(\lambda x.M) = \mathsf{FV}(M) \setminus \{x\}$$

Notation

$M \equiv N$ if M and N are identical

Definition

- set $\mathsf{FV}(M)$ of free variables of lambda term M is inductively defined:

$$\begin{aligned} \mathsf{FV}(x) &= \{x\} \\ \mathsf{FV}(MN) &= \mathsf{FV}(M) \cup \mathsf{FV}(N) \\ \mathsf{FV}(\lambda x.M) &= \mathsf{FV}(M) \setminus \{x\} \end{aligned}$$

- lambda term M is closed (or combinator) if $\mathsf{FV}(M) = \varnothing$

Notation

$M \equiv N$ if M and N are identical

Definition

- set $\mathrm{FV}(M)$ of free variables of lambda term M is inductively defined:

$$\mathrm{FV}(x) = \{x\}$$

$$\mathrm{FV}(MN) = \mathrm{FV}(M) \cup \mathrm{FV}(N)$$

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

- lambda term M is closed (or combinator) if $\mathrm{FV}(M) = \varnothing$

Example

$$M \equiv (\lambda x.xy)(\lambda y.yz)$$

Notation

$M \equiv N$ if M and N are identical

Definition

- set $\mathsf{FV}(M)$ of free variables of lambda term M is inductively defined:

$$\begin{aligned} \mathsf{FV}(x) &= \{x\} \\ \mathsf{FV}(MN) &= \mathsf{FV}(M) \cup \mathsf{FV}(N) \\ \mathsf{FV}(\lambda x.M) &= \mathsf{FV}(M) \setminus \{x\} \end{aligned}$$

- lambda term M is closed (or combinator) if $\mathsf{FV}(M) = \varnothing$

Example

$M \equiv (\lambda x.xy)(\lambda y.yz)$ $\mathsf{FV}(M) = \{y, z\}$

Definition (Renaming)

$$x\{y/x\} \equiv y$$

Definition (Renaming)

$$x\{y/x\} \equiv y$$

$$z\{y/x\} \equiv z \qquad \text{if } x \neq z$$

Definition (Renaming)

$$
\begin{aligned}
x\{y/x\} &\equiv y \\
z\{y/x\} &\equiv z && \text{if } x \neq z \\
(MN)\{y/x\} &\equiv (M\{y/x\})(N\{y/x\})
\end{aligned}
$$

Definition (Renaming)

$$
\begin{aligned}
x\{y/x\} &\equiv y \\
z\{y/x\} &\equiv z && \text{if } x \neq z \\
(MN)\{y/x\} &\equiv (M\{y/x\})(N\{y/x\}) \\
(\lambda x.M)\{y/x\} &\equiv \lambda y.(M\{y/x\})
\end{aligned}
$$

Definition (Renaming)

$$
\begin{aligned}
x\{y/x\} &\equiv y \\
z\{y/x\} &\equiv z && \text{if } x \neq z \\
(MN)\{y/x\} &\equiv (M\{y/x\})(N\{y/x\}) \\
(\lambda x.M)\{y/x\} &\equiv \lambda y.(M\{y/x\}) \\
(\lambda z.M)\{y/x\} &\equiv \lambda z.(M\{y/x\}) && \text{if } x \neq z
\end{aligned}
$$

Definition (Renaming)

$$
\begin{aligned}
x\{y/x\} &\equiv y \\
z\{y/x\} &\equiv z && \text{if } x \neq z \\
(MN)\{y/x\} &\equiv (M\{y/x\})(N\{y/x\}) \\
(\lambda x.M)\{y/x\} &\equiv \lambda y.(M\{y/x\}) \\
(\lambda z.M)\{y/x\} &\equiv \lambda z.(M\{y/x\}) && \text{if } x \neq z
\end{aligned}
$$

Definition

α–equivalence is smallest congruence relation $\equiv_\alpha$ on lambda terms such that

$$\lambda x.M \equiv_\alpha \lambda y.(M\{y/x\})$$

for all terms M and variables y that do not occur in M

Definition (Renaming)

$$
\begin{aligned}
x\{y/x\} &\equiv y \\
z\{y/x\} &\equiv z && \text{if } x \neq z \\
(MN)\{y/x\} &\equiv (M\{y/x\})(N\{y/x\}) \\
(\lambda x.M)\{y/x\} &\equiv \lambda y.(M\{y/x\}) \\
(\lambda z.M)\{y/x\} &\equiv \lambda z.(M\{y/x\}) && \text{if } x \neq z
\end{aligned}
$$

Definition

α–equivalence is smallest congruence relation $\equiv_\alpha$ on lambda terms such that

$$\lambda x.M \equiv_\alpha \lambda y.(M\{y/x\})$$

for all terms M and variables y that do not occur in M

α–equivalence

$$\frac{y \notin M}{\lambda x.M \equiv_\alpha \lambda y.(M\{y/x\})} \quad (\alpha)$$

α–equivalence

(reflexivity) $$\frac{}{M \equiv_\alpha M}$$

(symmetry) $$\frac{M \equiv_\alpha N}{N \equiv_\alpha M}$$

(transitivity) $$\frac{M \equiv_\alpha N \quad N \equiv_\alpha P}{M \equiv_\alpha P}$$

$$\frac{y \notin M}{\lambda x.M \equiv_\alpha \lambda y.(M\{y/x\})} \ (\alpha)$$

α–equivalence

$$\text{(reflexivity)} \quad \frac{}{M \equiv_\alpha M} \qquad \frac{M \equiv_\alpha M' \quad N \equiv_\alpha N'}{MN \equiv_\alpha M'N'} \quad \text{(congruence)}$$

$$\text{(symmetry)} \quad \frac{M \equiv_\alpha N}{N \equiv_\alpha M} \qquad \frac{M \equiv_\alpha M'}{\lambda x.M \equiv_\alpha \lambda x.M'} \quad (\xi)$$

$$\text{(transitivity)} \quad \frac{M \equiv_\alpha N \quad N \equiv_\alpha P}{M \equiv_\alpha P} \qquad \frac{y \notin M}{\lambda x.M \equiv_\alpha \lambda y.(M\{y/x\})} \quad (\alpha)$$

α–equivalence

(reflexivity) $$\frac{}{M \equiv_\alpha M}$$

$$\frac{M \equiv_\alpha M' \quad N \equiv_\alpha N'}{MN \equiv_\alpha M'N'}$$ (congruence)

(symmetry) $$\frac{M \equiv_\alpha N}{N \equiv_\alpha M}$$

$$\frac{M \equiv_\alpha M'}{\lambda x.M \equiv_\alpha \lambda x.M'}$$ (ξ)

(transitivity) $$\frac{M \equiv_\alpha N \quad N \equiv_\alpha P}{M \equiv_\alpha P}$$

$$\frac{y \notin M}{\lambda x.M \equiv_\alpha \lambda y.(M\{y/x\})}$$ (α)

Examples

$$\lambda x.y \equiv_\alpha \lambda z.y$$

α–equivalence

(reflexivity) $$\frac{}{M \equiv_\alpha M}$$

(congruence) $$\frac{M \equiv_\alpha M' \quad N \equiv_\alpha N'}{MN \equiv_\alpha M'N'}$$

(symmetry) $$\frac{M \equiv_\alpha N}{N \equiv_\alpha M}$$

(ξ) $$\frac{M \equiv_\alpha M'}{\lambda x.M \equiv_\alpha \lambda x.M'}$$

(transitivity) $$\frac{M \equiv_\alpha N \quad N \equiv_\alpha P}{M \equiv_\alpha P}$$

(α) $$\frac{y \notin M}{\lambda x.M \equiv_\alpha \lambda y.(M\{y/x\})}$$

Examples

$$\lambda x.y \equiv_\alpha \lambda z.y \qquad \lambda x.y \not\equiv_\alpha \lambda y.y$$

α–equivalence

(reflexivity) $$\frac{}{M \equiv_\alpha M}$$

$$\frac{M \equiv_\alpha M' \quad N \equiv_\alpha N'}{MN \equiv_\alpha M'N'}$$ (congruence)

(symmetry) $$\frac{M \equiv_\alpha N}{N \equiv_\alpha M}$$

$$\frac{M \equiv_\alpha M'}{\lambda x.M \equiv_\alpha \lambda x.M'}$$ (ξ)

(transitivity) $$\frac{M \equiv_\alpha N \quad N \equiv_\alpha P}{M \equiv_\alpha P}$$

$$\frac{y \notin M}{\lambda x.M \equiv_\alpha \lambda y.(M\{y/x\})}$$ (α)

Examples

$\lambda x.y \equiv_\alpha \lambda z.y \qquad \lambda x.y \not\equiv_\alpha \lambda y.y \qquad (\lambda x.y)z \not\equiv_\alpha (\lambda x.w)z$

α–equivalence

(reflexivity) $$\frac{}{M \equiv_\alpha M}$$

(congruence) $$\frac{M \equiv_\alpha M' \quad N \equiv_\alpha N'}{MN \equiv_\alpha M'N'}$$

(symmetry) $$\frac{M \equiv_\alpha N}{N \equiv_\alpha M}$$

(ξ) $$\frac{M \equiv_\alpha M'}{\lambda x.M \equiv_\alpha \lambda x.M'}$$

(transitivity) $$\frac{M \equiv_\alpha N \quad N \equiv_\alpha P}{M \equiv_\alpha P}$$

(α) $$\frac{y \notin M}{\lambda x.M \equiv_\alpha \lambda y.(M\{y/x\})}$$

Examples

$$\lambda x.y \equiv_\alpha \lambda z.y \qquad \lambda x.y \not\equiv_\alpha \lambda y.y \qquad (\lambda x.y)z \not\equiv_\alpha (\lambda x.w)z \qquad (\lambda x.y)(\lambda z.z) \equiv_\alpha (\lambda z.y)(\lambda z.z)$$

Barendregt's Variable Convention

free variables are different from bound variables in lambda terms occurring in certain context

Barendregt's Variable Convention

free variables are different from bound variables in lambda terms occurring in certain context

Definition

$M[N/x]$ denotes result of substituting N for free occurrences of x in M:

$$x[N/x] \equiv N$$

Barendregt's Variable Convention

free variables are different from bound variables in lambda terms occurring in certain context

Definition

$M[N/x]$ denotes result of substituting N for free occurrences of x in M:

$$x[N/x] \equiv N$$

$$y[N/x] \equiv y \qquad \text{if } x \neq y$$

Barendregt's Variable Convention

free variables are different from bound variables in lambda terms occurring in certain context

Definition

$M[N/x]$ denotes result of substituting N for free occurrences of x in M:

$$
\begin{aligned}
x[N/x] &\equiv N \\
y[N/x] &\equiv y && \text{if } x \neq y \\
(MP)[N/x] &\equiv (M[N/x])(P[N/x])
\end{aligned}
$$

Barendregt's Variable Convention

free variables are different from bound variables in lambda terms occurring in certain context

Definition

$M[N/x]$ denotes result of substituting N for free occurrences of x in M:

$$
\begin{aligned}
x[N/x] &\equiv N \\
y[N/x] &\equiv y && \text{if } x \neq y \\
(MP)[N/x] &\equiv (M[N/x])(P[N/x]) \\
(\lambda x.M)[N/x] &\equiv \lambda x.M
\end{aligned}
$$

Barendregt's Variable Convention

free variables are different from bound variables in lambda terms occurring in certain context

Definition

$M[N/x]$ denotes result of substituting N for free occurrences of x in M:

$$
\begin{aligned}
x[N/x] &\equiv N \\
y[N/x] &\equiv y && \text{if } x \neq y \\
(MP)[N/x] &\equiv (M[N/x])(P[N/x]) \\
(\lambda x.M)[N/x] &\equiv \lambda x.M \\
(\lambda y.M)[N/x] &\equiv \lambda y.(M[N/x]) && \text{if } x \neq y \text{ and } y \notin \mathrm{FV}(N)
\end{aligned}
$$

Barendregt's Variable Convention

free variables are different from bound variables in lambda terms occurring in certain context

Definition

$M[N/x]$ denotes result of substituting N for free occurrences of x in M:

$$
\begin{aligned}
x[N/x] &\equiv N && \\
y[N/x] &\equiv y && \text{if } x \neq y \\
(MP)[N/x] &\equiv (M[N/x])(P[N/x]) && \\
(\lambda x.M)[N/x] &\equiv \lambda x.M && \\
(\lambda y.M)[N/x] &\equiv \lambda y.(M[N/x]) && \text{if } x \neq y \text{ and } y \notin \mathrm{FV}(N) \\
(\lambda y.M)[N/x] &\equiv \lambda z.(M\{z/y\}[N/x]) && \text{if } x \neq y,\ y \in \mathrm{FV}(N) \text{ and } z \text{ is fresh}
\end{aligned}
$$

Barendregt's Variable Convention

free variables are different from bound variables in lambda terms occurring in certain context

Definition

$M[N/x]$ denotes result of substituting N for free occurrences of x in M:

$$
\begin{aligned}
x[N/x] &\equiv N \\
y[N/x] &\equiv y && \text{if } x \neq y \\
(MP)[N/x] &\equiv (M[N/x])(P[N/x]) \\
(\lambda x.M)[N/x] &\equiv \lambda x.M \\
(\lambda y.M)[N/x] &\equiv \lambda y.(M[N/x]) && \text{if } x \neq y \text{ and } y \notin \mathrm{FV}(N) \\
(\lambda y.M)[N/x] &\equiv \lambda z.(M\{z/y\}[N/x]) && \text{if } x \neq y,\ y \in \mathrm{FV}(N) \text{ and } z \text{ is fresh}
\end{aligned}
$$

Convention

lambda terms are identified up to α–equivalence

Definition

one-step β–reduction is smallest relation $\to_\beta$ on lambda terms satisfying

$$(\beta) \quad \frac{}{(\lambda x.M)N \to_\beta M[N/x]}$$

Definition

one-step β–reduction is smallest relation $\rightarrow_\beta$ on lambda terms satisfying

$$(\beta) \quad \frac{}{(\lambda x.M)N \rightarrow_\beta M[N/x]} \qquad\qquad \frac{M \rightarrow_\beta M'}{MN \rightarrow_\beta M'N} \quad \text{(congruence)}$$

$$(\xi) \quad \frac{M \rightarrow_\beta M'}{\lambda x.M \rightarrow_\beta \lambda x.M'} \qquad\qquad \frac{N \rightarrow_\beta N'}{MN \rightarrow_\beta MN'} \quad \text{(congruence)}$$

Definition

one-step β–reduction is smallest relation $\rightarrow_\beta$ on lambda terms satisfying

$$(\beta) \quad \frac{}{(\lambda x.M)N \rightarrow_\beta M[N/x]} \qquad \frac{M \rightarrow_\beta M'}{MN \rightarrow_\beta M'N} \quad \text{(congruence)}$$

$$(\xi) \quad \frac{M \rightarrow_\beta M'}{\lambda x.M \rightarrow_\beta \lambda x.M'} \qquad \frac{N \rightarrow_\beta N'}{MN \rightarrow_\beta MN'} \quad \text{(congruence)}$$

Example

$$(\lambda x.y)((\lambda z.zz)(\lambda w.w)) \rightarrow_\beta (\lambda x.y)((\lambda w.w)(\lambda w.w))$$

Definition

one-step β–reduction is smallest relation $\rightarrow_\beta$ on lambda terms satisfying

$$(\beta) \quad \frac{}{(\lambda x.M)N \rightarrow_\beta M[N/x]} \qquad \frac{M \rightarrow_\beta M'}{MN \rightarrow_\beta M'N} \quad \text{(congruence)}$$

$$(\xi) \quad \frac{M \rightarrow_\beta M'}{\lambda x.M \rightarrow_\beta \lambda x.M'} \qquad \frac{N \rightarrow_\beta N'}{MN \rightarrow_\beta MN'} \quad \text{(congruence)}$$

Example

$$(\lambda x.y)((\lambda z.zz)(\lambda w.w)) \rightarrow_\beta (\lambda x.y)((\lambda w.w)(\lambda w.w))$$

β-redex

Definition

one-step β–reduction is smallest relation $\rightarrow_\beta$ on lambda terms satisfying

$$(\beta) \quad \frac{}{(\lambda x.M)N \rightarrow_\beta M[N/x]} \qquad\qquad \frac{M \rightarrow_\beta M'}{MN \rightarrow_\beta M'N} \quad \text{(congruence)}$$

$$(\xi) \quad \frac{M \rightarrow_\beta M'}{\lambda x.M \rightarrow_\beta \lambda x.M'} \qquad\qquad \frac{N \rightarrow_\beta N'}{MN \rightarrow_\beta MN'} \quad \text{(congruence)}$$

Example

$$(\lambda x.y)((\lambda z.zz)(\lambda w.w)) \rightarrow_\beta (\lambda x.y)((\lambda w.w)(\lambda w.w))$$ reduct

Definition

one-step β–reduction is smallest relation $\to_\beta$ on lambda terms satisfying

$$(\beta) \quad \frac{}{(\lambda x.M)N \to_\beta M[N/x]} \qquad \frac{M \to_\beta M'}{MN \to_\beta M'N} \quad \text{(congruence)}$$

$$(\xi) \quad \frac{M \to_\beta M'}{\lambda x.M \to_\beta \lambda x.M'} \qquad \frac{N \to_\beta N'}{MN \to_\beta MN'} \quad \text{(congruence)}$$

Example

$$(\lambda x.y)((\lambda z.zz)(\lambda w.w)) \to_\beta (\lambda x.y)((\lambda w.w)(\lambda w.w)) \qquad \beta\text{-redex}$$

$$\to_\beta (\lambda x.y)(\lambda w.w)$$

Definition

one-step β–reduction is smallest relation $\to_\beta$ on lambda terms satisfying

$$(\beta) \quad \frac{}{(\lambda x.M)N \to_\beta M[N/x]} \qquad\qquad \frac{M \to_\beta M'}{MN \to_\beta M'N} \quad \text{(congruence)}$$

$$(\xi) \quad \frac{M \to_\beta M'}{\lambda x.M \to_\beta \lambda x.M'} \qquad\qquad \frac{N \to_\beta N'}{MN \to_\beta MN'} \quad \text{(congruence)}$$

Example

$$\begin{aligned}(\lambda x.y)((\lambda z.zz)(\lambda w.w)) &\to_\beta (\lambda x.y)((\lambda w.w)(\lambda w.w)) \\ &\to_\beta (\lambda x.y)(\lambda w.w) \to_\beta y\end{aligned}$$

Definition

one-step β–reduction is smallest relation $\rightarrow_\beta$ on lambda terms satisfying

$$(\beta) \quad \frac{}{(\lambda x.M)N \rightarrow_\beta M[N/x]} \qquad \frac{M \rightarrow_\beta M'}{MN \rightarrow_\beta M'N} \quad \text{(congruence)}$$

$$(\xi) \quad \frac{M \rightarrow_\beta M'}{\lambda x.M \rightarrow_\beta \lambda x.M'} \qquad \frac{N \rightarrow_\beta N'}{MN \rightarrow_\beta MN'} \quad \text{(congruence)}$$

Example

$$\begin{aligned}(\lambda x.y)((\lambda z.zz)(\lambda w.w)) &\rightarrow_\beta (\lambda x.y)((\lambda w.w)(\lambda w.w)) \\ &\rightarrow_\beta (\lambda x.y)(\lambda w.w) \rightarrow_\beta y\end{aligned}$$

Definitions

- β–**normal form** is lambda term without β–redexes

Definition

one-step β–reduction is smallest relation $\rightarrow_\beta$ on lambda terms satisfying

$$(\beta) \quad \frac{}{(\lambda x.M)N \rightarrow_\beta M[N/x]} \qquad \frac{M \rightarrow_\beta M'}{MN \rightarrow_\beta M'N} \quad \text{(congruence)}$$

$$(\xi) \quad \frac{M \rightarrow_\beta M'}{\lambda x.M \rightarrow_\beta \lambda x.M'} \qquad \frac{N \rightarrow_\beta N'}{MN \rightarrow_\beta MN'} \quad \text{(congruence)}$$

Example

$$\begin{aligned}(\lambda x.y)((\lambda z.zz)(\lambda w.w)) &\rightarrow_\beta (\lambda x.y)((\lambda w.w)(\lambda w.w)) \\ &\rightarrow_\beta (\lambda x.y)(\lambda w.w) \rightarrow_\beta y\end{aligned}$$

Definitions

- β–normal form is lambda term without β–redexes
- β–conversion ($=_\beta$) is transitive symmetric reflexive closure of $\rightarrow_\beta$

Definition

lambda term N is fixed point of lambda term F if $FN =_\beta N$

Definition

lambda term N is fixed point of lambda term F if $FN =_\beta N$

Definition (Turing's Fixed Point Combinator)

$\Theta \equiv AA$ with $A = \lambda xy.y(xxy)$

Definition

lambda term N is fixed point of lambda term F if $FN =_\beta N$

Definition (Turing's Fixed Point Combinator)

$\Theta \equiv AA$ with $A = \lambda xy.y(xxy)$

Theorem

every lambda term has fixed point

Definition

lambda term N is fixed point of lambda term F if $FN =_\beta N$

Definition (Turing's Fixed Point Combinator)

$\Theta \equiv AA$ with $A = \lambda xy.y(xxy)$

Theorem

every lambda term has fixed point

Proof

$N \equiv \Theta F$ is fixed point of F:

$$N \equiv (\lambda xy.y(xxy))AF$$

Definition

lambda term N is fixed point of lambda term F if $FN =_\beta N$

Definition (Turing's Fixed Point Combinator)

$\Theta \equiv AA$ with $A = \lambda xy.y(xxy)$

Theorem

every lambda term has fixed point

Proof

$N \equiv \Theta F$ is fixed point of F:

$$N \equiv (\lambda xy.y(xxy))AF \to_\beta (\lambda y.y(AAy))F$$

Definition

lambda term N is fixed point of lambda term F if $FN =_\beta N$

Definition (Turing's Fixed Point Combinator)

$\Theta \equiv AA$ with $A = \lambda xy.y(xxy)$

Theorem

every lambda term has fixed point

Proof

$N \equiv \Theta F$ is fixed point of F:

$$N \equiv (\lambda xy.y(xxy))AF \to_\beta (\lambda y.y(AAy))F \to_\beta F(AAF)$$

Definition

lambda term N is fixed point of lambda term F if $FN =_\beta N$

Definition (Turing's Fixed Point Combinator)

$\Theta \equiv AA$ with $A = \lambda xy.y(xxy)$

Theorem

every lambda term has fixed point

Proof

$N \equiv \Theta F$ is fixed point of F:

$$N \equiv (\lambda xy.y(xxy))AF \to_\beta (\lambda y.y(AAy))F \to_\beta F(AAF) \equiv F(\Theta F)$$

Definition

lambda term N is fixed point of lambda term F if $FN =_\beta N$

Definition (Turing's Fixed Point Combinator)

$\Theta \equiv AA$ with $A = \lambda xy.y(xxy)$

Theorem

every lambda term has fixed point

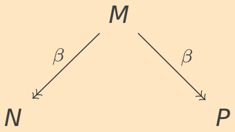
Proof

$N \equiv \Theta F$ is fixed point of F:

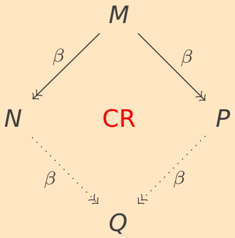
$$N \equiv (\lambda xy.y(xxy))AF \to_\beta (\lambda y.y(AAy))F \to_\beta F(AAF) \equiv F(\Theta F) \equiv FN$$

Outline

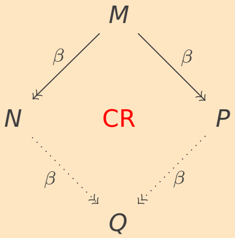
Church–Rosser Theorem



Church–Rosser Theorem



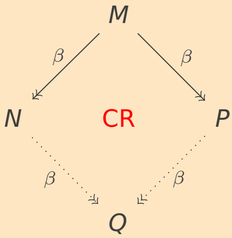
Church–Rosser Theorem



Corollary

- $M =_\beta N \quad \Longrightarrow \quad \exists\, Q$ such that $M \to_\beta^* Q$ and $N \to_\beta^* Q$

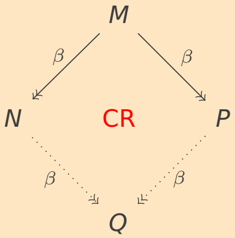
Church–Rosser Theorem



Corollary

- $M =_\beta N \quad \Longrightarrow \quad \exists\, Q$ such that $M \to_\beta^* Q$ and $N \to_\beta^* Q$
- $M =_\beta N$ and N is β–normal form $\quad \Longrightarrow \quad M \to_\beta^* N$

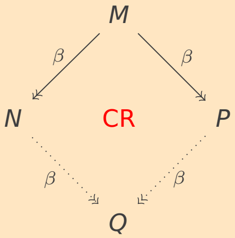
Church–Rosser Theorem



Corollary

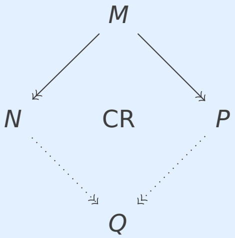
- $M =_\beta N \implies \exists\, Q$ such that $M \to_\beta^* Q$ and $N \to_\beta^* Q$
- $M =_\beta N$ and N is β–normal form $\implies M \to_\beta^* N$
- $M =_\beta N$ and M, N are β–normal forms $\implies M \equiv_\alpha N$

Church–Rosser Theorem

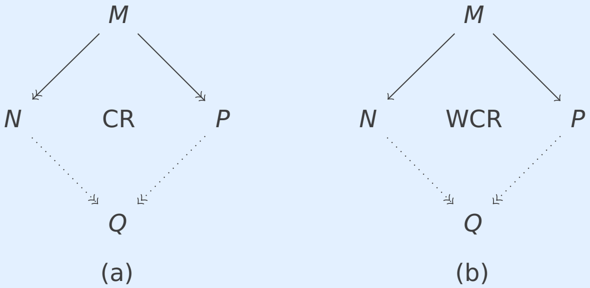


Corollary

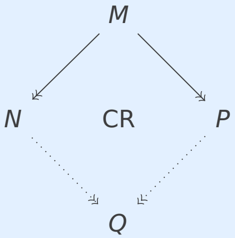
- $M =_\beta N \quad \Longrightarrow \quad \exists\, Q$ such that $M \to_\beta^* Q$ and $N \to_\beta^* Q$
- $M =_\beta N$ and N is β–normal form $\quad \Longrightarrow \quad M \to_\beta^* N$
- $M =_\beta N$ and M, N are β–normal forms $\quad \Longrightarrow \quad M \equiv_\alpha N$
- $M =_\beta N \quad \Longrightarrow \quad$ both or neither of M, N have β–normal form



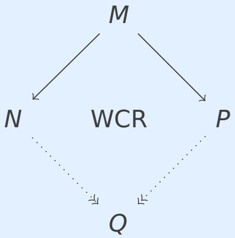
(a)



- β–reduction satisfies (b)

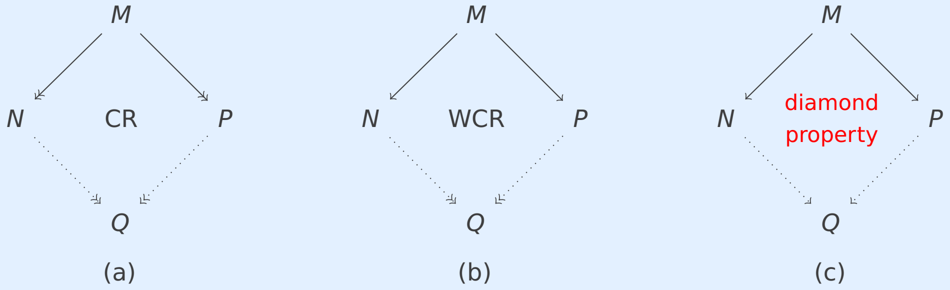


(a)

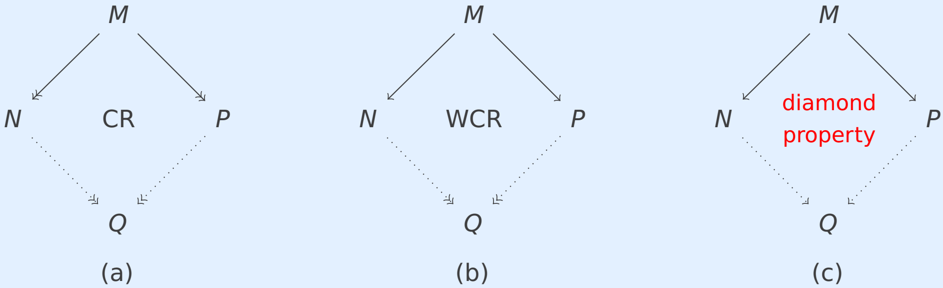


(b)

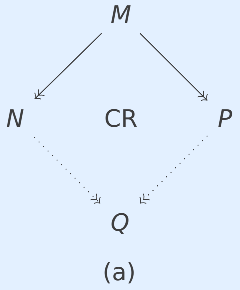
- β–reduction satisfies (b)
- (b) $\not\Longrightarrow$ (a)



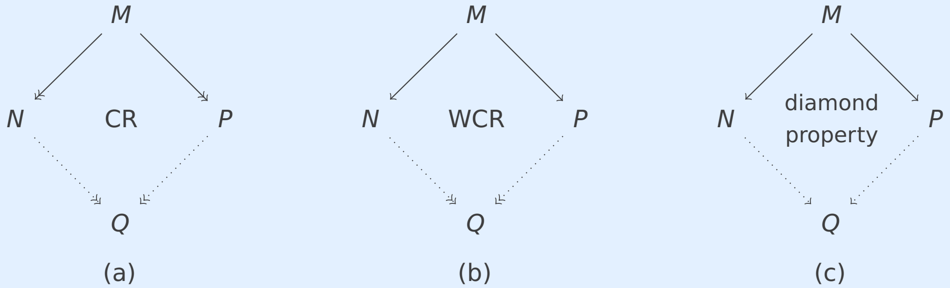
- β–reduction satisfies (b)
- (b) $\;\not\Longrightarrow\;$ (a)
- (c) $\;\Longrightarrow\;$ (a)



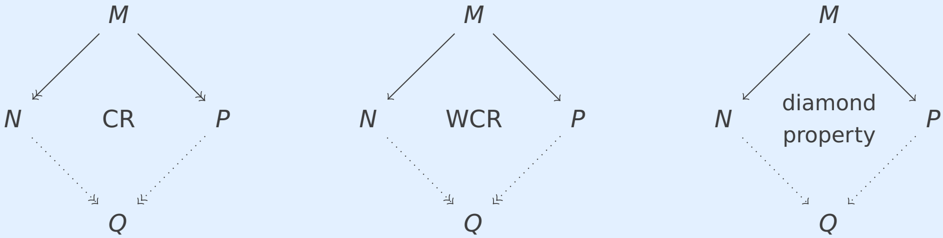
- β–reduction satisfies (b)
- (b) $\not\Longrightarrow$ (a)
- (c) $\Longrightarrow$ (a)
- β–reduction does not satisfy (c)



- β–reduction satisfies (b)
- (b) $\quad\not\Longrightarrow\quad$ (a)
- (c) $\quad\Longrightarrow\quad$ (a)
- β–reduction does not satisfy (c): $\qquad (\lambda x.xx)((\lambda y.y)z)$



- β–reduction satisfies (b)
- (b) $\not\Longrightarrow$ (a)
- (c) $\Longrightarrow$ (a)
- β–reduction does not satisfy (c): $(\lambda x.xx)z \;{}_{\beta}\!\leftarrow (\lambda x.xx)((\lambda y.y)z)$



(a) (b) (c)

- β–reduction satisfies (b)
- (b) $\not\Longrightarrow$ (a)
- (c) $\Longrightarrow$ (a)
- β–reduction does not satisfy (c): $(\lambda x.xx)z \; {}_{\beta}\!\leftarrow (\lambda x.xx)((\lambda y.y)z) \rightarrow_{\beta} (\lambda y.y)z((\lambda y.y)z)$

Definition (Parallel Reduction)

$$\frac{}{M \;\Longrightarrow\!\!\!\!\!\!\!\!\;\;\|\;_\beta\; M}$$

Definition (Parallel Reduction)

$$\frac{}{M \;\twoheadrightarrow\!\!\!\!\!\!\parallel\;_\beta\; M} \qquad \frac{}{(\lambda x.M)N \;\twoheadrightarrow\!\!\!\!\!\!\parallel\;_\beta\; M[N/x]}$$

Definition (Parallel Reduction)

$$\frac{}{M \;\twoheadrightarrow\!\!\!\!\!\!\!\!\;\,+\!\!\!\!\;\; _\beta\; M}$$

$$\frac{}{M \mathrel{\Vert\!\!\!\rightarrow}_\beta M} \qquad \frac{}{(\lambda x.M)N \mathrel{\Vert\!\!\!\rightarrow}_\beta M[N/x]} \qquad \frac{M \mathrel{\Vert\!\!\!\rightarrow}_\beta M'}{\lambda x.M \mathrel{\Vert\!\!\!\rightarrow}_\beta \lambda x.M'}$$

Definition (Parallel Reduction)

$$\frac{}{M \mathrel{\rightarrow\!\!\!\!\!\!\!\parallel\;}_\beta M} \qquad \frac{}{(\lambda x.M)N \mathrel{\rightarrow\!\!\!\!\!\!\!\parallel\;}_\beta M[N/x]} \qquad \frac{M \mathrel{\rightarrow\!\!\!\!\!\!\!\parallel\;}_\beta M'}{\lambda x.M \mathrel{\rightarrow\!\!\!\!\!\!\!\parallel\;}_\beta \lambda x.M'} \qquad \frac{M \mathrel{\rightarrow\!\!\!\!\!\!\!\parallel\;}_\beta M' \qquad N \mathrel{\rightarrow\!\!\!\!\!\!\!\parallel\;}_\beta N'}{MN \mathrel{\rightarrow\!\!\!\!\!\!\!\parallel\;}_\beta M'N'}$$

Definition (Parallel Reduction)

$$\frac{}{M \twoheadrightarrow\!\!\!\!\!\!\!\mid\mid_\beta M} \qquad \frac{}{(\lambda x.M)N \twoheadrightarrow\!\!\!\!\!\!\!\mid\mid_\beta M[N/x]} \qquad \frac{M \twoheadrightarrow\!\!\!\!\!\!\!\mid\mid_\beta M'}{\lambda x.M \twoheadrightarrow\!\!\!\!\!\!\!\mid\mid_\beta \lambda x.M'} \qquad \frac{M \twoheadrightarrow\!\!\!\!\!\!\!\mid\mid_\beta M' \quad N \twoheadrightarrow\!\!\!\!\!\!\!\mid\mid_\beta N'}{MN \twoheadrightarrow\!\!\!\!\!\!\!\mid\mid_\beta M'N'}$$

Problem

$\twoheadrightarrow\!\!\!\!\!\!\!\mid\mid$ lacks diamond property

Definition (Parallel Reduction)

$$\frac{}{M \twoheadrightarrow_\beta M} \qquad \frac{}{(\lambda x.M)N \twoheadrightarrow_\beta M[N/x]} \qquad \frac{M \twoheadrightarrow_\beta M'}{\lambda x.M \twoheadrightarrow_\beta \lambda x.M'} \qquad \frac{M \twoheadrightarrow_\beta M' \quad N \twoheadrightarrow_\beta N'}{MN \twoheadrightarrow_\beta M'N'}$$

Problem

$\twoheadrightarrow$ lacks diamond property: $(\lambda x.x)I \ {}_\beta\!\twoheadleftarrow (\lambda x.(\lambda y.x)I)(II) \twoheadrightarrow_\beta (\lambda y.II)I$ with $I = \lambda x.x$

Definition (Parallel Reduction)

$$\frac{}{M \mathrel{+\!\!\!\!\!\!\!\rightarrow}_\beta M} \qquad \frac{}{(\lambda x.M)N \mathrel{+\!\!\!\!\!\!\!\rightarrow}_\beta M[N/x]} \qquad \frac{M \mathrel{+\!\!\!\!\!\!\!\rightarrow}_\beta M'}{\lambda x.M \mathrel{+\!\!\!\!\!\!\!\rightarrow}_\beta \lambda x.M'} \qquad \frac{M \mathrel{+\!\!\!\!\!\!\!\rightarrow}_\beta M' \quad N \mathrel{+\!\!\!\!\!\!\!\rightarrow}_\beta N'}{MN \mathrel{+\!\!\!\!\!\!\!\rightarrow}_\beta M'N'}$$

Problem

$\mathrel{+\!\!\!\!\!\!\!\rightarrow}$ lacks diamond property: $(\lambda x.x)I \;{}_\beta\!\!\mathrel{\leftarrow\!\!\!\!\!\!\!+} (\lambda x.(\lambda y.x)I)(II) \mathrel{+\!\!\!\!\!\!\!\rightarrow}_\beta (\lambda y.II)I$ with $I = \lambda x.x$

Definition (Parallel Reduction Revisited)

$$\frac{}{M \mathrel{\multimap\!\!\!\rightarrow}_\beta M} \qquad \frac{M \mathrel{\multimap\!\!\!\rightarrow}_\beta M' \quad N \mathrel{\multimap\!\!\!\rightarrow}_\beta N'}{(\lambda x.M)N \mathrel{\multimap\!\!\!\rightarrow}_\beta M'[N'/x]} \qquad \frac{M \mathrel{\multimap\!\!\!\rightarrow}_\beta M'}{\lambda x.M \mathrel{\multimap\!\!\!\rightarrow}_\beta \lambda x.M'} \qquad \frac{M \mathrel{\multimap\!\!\!\rightarrow}_\beta M' \quad N \mathrel{\multimap\!\!\!\rightarrow}_\beta N'}{MN \mathrel{\multimap\!\!\!\rightarrow}_\beta M'N'}$$

Definition (Parallel Reduction)

$$\frac{}{M \mathrel{+\!\!\!\!\!\!\!\longrightarrow}_\beta M} \qquad \frac{}{(\lambda x.M)N \mathrel{+\!\!\!\!\!\!\!\longrightarrow}_\beta M[N/x]} \qquad \frac{M \mathrel{+\!\!\!\!\!\!\!\longrightarrow}_\beta M'}{\lambda x.M \mathrel{+\!\!\!\!\!\!\!\longrightarrow}_\beta \lambda x.M'} \qquad \frac{M \mathrel{+\!\!\!\!\!\!\!\longrightarrow}_\beta M' \qquad N \mathrel{+\!\!\!\!\!\!\!\longrightarrow}_\beta N'}{MN \mathrel{+\!\!\!\!\!\!\!\longrightarrow}_\beta M'N'}$$

Problem

$\mathrel{+\!\!\!\!\!\!\!\longrightarrow}$ lacks diamond property: $(\lambda x.x)I \;{}_\beta\!\mathrel{\longleftarrow\!\!\!\!\!\!\!+} (\lambda x.(\lambda y.x)I)(II) \mathrel{+\!\!\!\!\!\!\!\longrightarrow}_\beta (\lambda y.II)I$ with $I = \lambda x.x$

Definition (Parallel Reduction Revisited)

$$\frac{}{M \mathrel{\circ\!\!\!\!\!\!\longrightarrow}_\beta M} \qquad \frac{M \mathrel{\circ\!\!\!\!\!\!\longrightarrow}_\beta M' \qquad N \mathrel{\circ\!\!\!\!\!\!\longrightarrow}_\beta N'}{(\lambda x.M)N \mathrel{\circ\!\!\!\!\!\!\longrightarrow}_\beta M'[N'/x]} \qquad \frac{M \mathrel{\circ\!\!\!\!\!\!\longrightarrow}_\beta M'}{\lambda x.M \mathrel{\circ\!\!\!\!\!\!\longrightarrow}_\beta \lambda x.M'} \qquad \frac{M \mathrel{\circ\!\!\!\!\!\!\longrightarrow}_\beta M' \qquad N \mathrel{\circ\!\!\!\!\!\!\longrightarrow}_\beta N'}{MN \mathrel{\circ\!\!\!\!\!\!\longrightarrow}_\beta M'N'}$$

Lemma

$\to_\beta \;\subseteq\; \mathrel{\circ\!\!\!\!\!\!\longrightarrow}_\beta \;\subseteq\; \to_\beta^*$

Lemma (Substitution)

if $M \multimap\!\!\rightarrow_\beta M'$ and $U \multimap\!\!\rightarrow_\beta U'$ then $M[U/y] \multimap\!\!\rightarrow_\beta M'[U'/y]$

Lemma (Substitution)

if $M \multimap\!\!\to_\beta M'$ and $U \multimap\!\!\to_\beta U'$ then $M[U/y] \multimap\!\!\to_\beta M'[U'/y]$

Definition

M^* is maximal parallel one-step reduct of M:

① $x^* = x$

Lemma (Substitution)

if $M \twoheadrightarrow\!\!\!\!\circ_\beta M'$ and $U \twoheadrightarrow\!\!\!\!\circ_\beta U'$ then $M[U/y] \twoheadrightarrow\!\!\!\!\circ_\beta M'[U'/y]$

Definition

M^* is maximal parallel one-step reduct of M:

① $x^* = x$

② $(PN)^* = P^*N^*$ if PN is no β–redex

Lemma (Substitution)

if $M \twoheadrightarrow_\beta M'$ and $U \twoheadrightarrow_\beta U'$ then $M[U/y] \twoheadrightarrow_\beta M'[U'/y]$

Definition

M^* is maximal parallel one-step reduct of M:

① $x^* = x$

② $(PN)^* = P^*N^*$ if PN is no β–redex

③ $((\lambda x.Q)N)^* = Q^*[N^*/x]$

Lemma (Substitution)

if $M \twoheadrightarrow_\beta M'$ and $U \twoheadrightarrow_\beta U'$ then $M[U/y] \twoheadrightarrow_\beta M'[U'/y]$

Definition

M^* is maximal parallel one-step reduct of M:

① $x^* = x$

② $(PN)^* = P^*N^*$ if PN is no β–redex

③ $((\lambda x.Q)N)^* = Q^*[N^*/x]$

④ $(\lambda x.N)^* = \lambda x.N^*$

Lemma (Substitution)

if $M \multimap\!\!\!\rightarrow_\beta M'$ and $U \multimap\!\!\!\rightarrow_\beta U'$ then $M[U/y] \multimap\!\!\!\rightarrow_\beta M'[U'/y]$

Definition

M^* is maximal parallel one-step reduct of M:

① $x^* = x$

② $(PN)^* = P^*N^*$ if PN is no β–redex

③ $((\lambda x.Q)N)^* = Q^*[N^*/x]$

④ $(\lambda x.N)^* = \lambda x.N^*$

Lemma

if $M \multimap\!\!\!\rightarrow_\beta N$ then $N \multimap\!\!\!\rightarrow_\beta M^*$

Lemma (Diamond Property of Parallel Reduction)

$\forall\, M, N, P \in \Lambda$ such that $M \twoheadrightarrow\!\!\!\!\!\circ\;_\beta N$ and $M \twoheadrightarrow\!\!\!\!\!\circ\;_\beta P$

$\exists\, Q \in \Lambda$ such that $N \twoheadrightarrow\!\!\!\!\!\circ\;_\beta Q$ and $P \twoheadrightarrow\!\!\!\!\!\circ\;_\beta Q$

Lemma (Diamond Property of Parallel Reduction)

$\forall\, M, N, P \in \Lambda$ such that $M \twoheadrightarrow\!\!\!\!\!\circ\;_\beta N$ and $M \twoheadrightarrow\!\!\!\!\!\circ\;_\beta P$

$\exists\, Q \in \Lambda$ such that $N \twoheadrightarrow\!\!\!\!\!\circ\;_\beta Q$ and $P \twoheadrightarrow\!\!\!\!\!\circ\;_\beta Q$

Proof

take $Q \equiv M^*$

Lemma (Diamond Property of Parallel Reduction)

$\forall\, M, N, P \in \Lambda$ such that $M \twoheadrightarrow_\beta N$ and $M \twoheadrightarrow_\beta P$

$\exists\, Q \in \Lambda$ such that $N \twoheadrightarrow_\beta Q$ and $P \twoheadrightarrow_\beta Q$

Proof

take $Q \equiv M^*$

Corollary

β–reduction has Church–Rosser property

Outline

Definitions

- $\mathsf{T} \equiv \lambda xy.x$

Definitions

- $\mathsf{T} \equiv \lambda xy.x \qquad \mathsf{F} \equiv \lambda xy.y$

Definitions

- $\mathsf{T} \equiv \lambda xy.x \qquad \mathsf{F} \equiv \lambda xy.y \qquad \mathsf{and} \equiv \lambda ab.ab\,\mathsf{F}$

Definitions

- $\mathsf{T} \equiv \lambda xy.x \qquad \mathsf{F} \equiv \lambda xy.y \qquad \mathsf{and} \equiv \lambda ab.ab\,\mathsf{F}$

Lemmata

- $\mathsf{and}\,\mathsf{T}\,\mathsf{T} \to_\beta^* \mathsf{T} \qquad \mathsf{and}\,\mathsf{T}\,\mathsf{F} \to_\beta^* \mathsf{F} \qquad \mathsf{and}\,\mathsf{F}\,\mathsf{T} \to_\beta^* \mathsf{F} \qquad \mathsf{and}\,\mathsf{F}\,\mathsf{F} \to_\beta^* \mathsf{F}$

Definitions

- $\mathsf{T} \equiv \lambda xy.x \qquad \mathsf{F} \equiv \lambda xy.y \qquad \mathsf{and} \equiv \lambda ab.ab\,\mathsf{F}$
- $\mathsf{ite} \equiv \lambda x.x$

Lemmata

- $\mathsf{and}\,\mathsf{T}\,\mathsf{T} \to_\beta^* \mathsf{T} \qquad \mathsf{and}\,\mathsf{T}\,\mathsf{F} \to_\beta^* \mathsf{F} \qquad \mathsf{and}\,\mathsf{F}\,\mathsf{T} \to_\beta^* \mathsf{F} \qquad \mathsf{and}\,\mathsf{F}\,\mathsf{F} \to_\beta^* \mathsf{F}$

Definitions

- $\mathsf{T} \equiv \lambda xy.x \qquad \mathsf{F} \equiv \lambda xy.y \qquad \mathsf{and} \equiv \lambda ab.ab\,\mathsf{F}$
- $\mathsf{ite} \equiv \lambda x.x$

Lemmata

- $\mathsf{and}\,\mathsf{T}\,\mathsf{T} \to_\beta^* \mathsf{T} \qquad \mathsf{and}\,\mathsf{T}\,\mathsf{F} \to_\beta^* \mathsf{F} \qquad \mathsf{and}\,\mathsf{F}\,\mathsf{T} \to_\beta^* \mathsf{F} \qquad \mathsf{and}\,\mathsf{F}\,\mathsf{F} \to_\beta^* \mathsf{F}$
- $\mathsf{ite}\,\mathsf{T}\,M\,N \to_\beta^* M \qquad \mathsf{ite}\,\mathsf{F}\,M\,N \to_\beta^* N$

Definitions

- $\mathsf{T} \equiv \lambda xy.x \qquad \mathsf{F} \equiv \lambda xy.y \qquad \mathsf{and} \equiv \lambda ab.ab\,\mathsf{F}$
- $\mathsf{ite} \equiv \lambda x.x$

Lemmata

- $\mathsf{and}\,\mathsf{T}\,\mathsf{T} \to_\beta^* \mathsf{T} \qquad \mathsf{and}\,\mathsf{T}\,\mathsf{F} \to_\beta^* \mathsf{F} \qquad \mathsf{and}\,\mathsf{F}\,\mathsf{T} \to_\beta^* \mathsf{F} \qquad \mathsf{and}\,\mathsf{F}\,\mathsf{F} \to_\beta^* \mathsf{F}$
- $\mathsf{ite}\,\mathsf{T}\,M\,N \to_\beta^* M \qquad \mathsf{ite}\,\mathsf{F}\,M\,N \to_\beta^* N$

Definitions (Church Numerals)

- for every natural number n

$$\underline{n} \equiv \lambda fx.f^n x \quad \text{where} \quad F^n M \equiv \begin{cases} M & \text{if } n = 0 \\ F(F^{n-1}M) & \text{if } n > 0 \end{cases}$$

Definitions

- $\mathsf{T} \equiv \lambda xy.x \qquad \mathsf{F} \equiv \lambda xy.y \qquad \mathsf{and} \equiv \lambda ab.ab\,\mathsf{F}$
- $\mathsf{ite} \equiv \lambda x.x$

Lemmata

- $\mathsf{and}\,\mathsf{T}\,\mathsf{T} \to_\beta^* \mathsf{T} \qquad \mathsf{and}\,\mathsf{T}\,\mathsf{F} \to_\beta^* \mathsf{F} \qquad \mathsf{and}\,\mathsf{F}\,\mathsf{T} \to_\beta^* \mathsf{F} \qquad \mathsf{and}\,\mathsf{F}\,\mathsf{F} \to_\beta^* \mathsf{F}$
- $\mathsf{ite}\,\mathsf{T}\,M\,N \to_\beta^* M \qquad \mathsf{ite}\,\mathsf{F}\,M\,N \to_\beta^* N$

Definitions (Church Numerals)

- for every natural number n

$$\underline{n} \equiv \lambda fx.f^n x \quad \text{where} \quad F^n M \equiv \begin{cases} M & \text{if } n = 0 \\ F(F^{n-1}M) & \text{if } n > 0 \end{cases}$$

- $\mathsf{succ} \equiv \lambda nfx.f(nfx)$

Lemma

$\mathsf{succ}\ \underline{n} \rightarrow_\beta^* \underline{n+1}$

Lemma

$\mathsf{succ}\ \underline{n} \to_\beta^* \underline{n+1}$

Proof

$$\mathsf{succ}\ \underline{n} \equiv (\lambda n f x. f(n f x))(\lambda f x. f^n x)$$

Lemma

$\text{succ}\ \underline{n} \rightarrow_\beta^* \underline{n+1}$

Proof

$$\text{succ}\ \underline{n} \equiv (\lambda n f x. f(n f x))(\lambda f x. f^n x) \rightarrow_\beta \lambda f x. f((\lambda f x. f^n x) f x)$$

Lemma

$\text{succ}\ \underline{n} \to_\beta^* \underline{n+1}$

Proof

$$\text{succ}\ \underline{n} \equiv (\lambda n f x. f(n f x))(\lambda f x. f^n x) \to_\beta \lambda f x. f((\lambda f x. f^n x) f x) \to_\beta \lambda f x. f((\lambda x. f^n x) x)$$

Lemma

$\text{succ}\ \underline{n} \to_\beta^* \underline{n+1}$

Proof

$$\text{succ}\ \underline{n} \equiv (\lambda n f x. f\,(n\,f\,x))(\lambda f x. f^n x) \to_\beta \lambda f x. f\,((\lambda f x. f^n x)\,f\,x) \to_\beta \lambda f x. f\,((\lambda x. f^n x)\,x)$$
$$\to_\beta \lambda f x. f\,(f^n x)$$

Lemma

$\text{succ}\ \underline{n} \rightarrow_\beta^* \underline{n+1}$

Proof

$$\begin{aligned}\text{succ}\ \underline{n} \equiv (\lambda n f x. f(n f x))(\lambda f x. f^n x) &\rightarrow_\beta \lambda f x. f((\lambda f x. f^n x) f x) \rightarrow_\beta \lambda f x. f((\lambda x. f^n x) x) \\ &\rightarrow_\beta \lambda f x. f(f^n x) \equiv \lambda f x. f^{n+1} x \equiv \underline{n+1}\end{aligned}$$

Lemma

$\text{succ}\ \underline{n} \to_\beta^* \underline{n+1}$

Proof

$$\text{succ}\ \underline{n} \equiv (\lambda n f x. f(n f x))(\lambda f x. f^n x) \to_\beta \lambda f x. f((\lambda f x. f^n x) f x) \to_\beta \lambda f x. f((\lambda x. f^n x) x)$$
$$\to_\beta \lambda f x. f(f^n x) \equiv \lambda f x. f^{n+1} x \equiv \underline{n+1}$$

Definitions

- $\text{zero?} \equiv \lambda n. n(\lambda x. \text{F})\,\text{T}$

Lemma

$\text{succ}\,\underline{n} \to_\beta^* \underline{n+1}$

Proof

$$\text{succ}\,\underline{n} \equiv (\lambda n f x.f(n f x))(\lambda f x.f^n x) \to_\beta \lambda f x.f((\lambda f x.f^n x) f x) \to_\beta \lambda f x.f((\lambda x.f^n x) x)$$
$$\to_\beta \lambda f x.f(f^n x) \equiv \lambda f x.f^{n+1} x \equiv \underline{n+1}$$

Definitions

- $\text{zero?} \equiv \lambda n.n(\lambda x.\text{F})\,\text{T}$

Lemmata

- $\text{zero?}\,\underline{n} \to_\beta^* \begin{cases} \text{T} & \text{if } n = 0 \\ \text{F} & \text{if } n > 0 \end{cases}$

Lemma

$\text{succ}\ \underline{n} \rightarrow_\beta^* \underline{n+1}$

Proof

$$\begin{aligned} \text{succ}\ \underline{n} \equiv (\lambda nfx.f(n\,f\,x))(\lambda fx.f^n x) &\rightarrow_\beta \lambda fx.f((\lambda fx.f^n x)\,f\,x) \rightarrow_\beta \lambda fx.f((\lambda x.f^n x)\,x) \\ &\rightarrow_\beta \lambda fx.f(f^n x) \equiv \lambda fx.f^{n+1} x \equiv \underline{n+1} \end{aligned}$$

Definitions

- $\text{zero?} \equiv \lambda n.n(\lambda x.\text{F})\,\text{T}$ $\qquad$ $\text{add} \equiv \lambda nmfx.nf(mfx)$

Lemmata

- $\text{zero?}\ \underline{n} \rightarrow_\beta^* \begin{cases} \text{T} & \text{if } n = 0 \\ \text{F} & \text{if } n > 0 \end{cases}$

Lemma

$\mathsf{succ}\ \underline{n} \to_\beta^* \underline{n+1}$

Proof

$$\mathsf{succ}\ \underline{n} \equiv (\lambda nfx.f(nfx))(\lambda fx.f^n x) \to_\beta \lambda fx.f((\lambda fx.f^n x)fx) \to_\beta \lambda fx.f((\lambda x.f^n x)x)$$
$$\to_\beta \lambda fx.f(f^n x) \equiv \lambda fx.f^{n+1}x \equiv \underline{n+1}$$

Definitions

- $\mathsf{zero?} \equiv \lambda n.n(\lambda x.\mathsf{F})\mathsf{T}$ $\quad$ $\mathsf{add} \equiv \lambda nmfx.nf(mfx)$ $\quad$ $\mathsf{mul} \equiv \lambda nmf.n(mf)$

Lemmata

- $\mathsf{zero?}\ \underline{n} \to_\beta^* \begin{cases} \mathsf{T} & \text{if } n = 0 \\ \mathsf{F} & \text{if } n > 0 \end{cases}$

Lemma

$\text{succ}\ \underline{n} \to_\beta^* \underline{n+1}$

Proof

$$\text{succ}\ \underline{n} \equiv (\lambda nfx.f(nfx))(\lambda fx.f^n x) \to_\beta \lambda fx.f((\lambda fx.f^n x)fx) \to_\beta \lambda fx.f((\lambda x.f^n x)x)$$
$$\to_\beta \lambda fx.f(f^n x) \equiv \lambda fx.f^{n+1}x \equiv \underline{n+1}$$

Definitions

- $\text{zero?} \equiv \lambda n.n(\lambda x.\text{F})\text{T}$ $\qquad$ $\text{add} \equiv \lambda nmfx.nf(mfx)$ $\qquad$ $\text{mul} \equiv \lambda nmf.n(mf)$

Lemmata

- $\text{zero?}\ \underline{n} \to_\beta^* \begin{cases} \text{T} & \text{if } n = 0 \\ \text{F} & \text{if } n > 0 \end{cases}$ $\qquad$ $\text{add}\ \underline{n}\ \underline{m} \to_\beta^* \underline{n+m}$ $\qquad$ $\text{mul}\ \underline{n}\ \underline{m} \to_\beta^* \underline{n \cdot m}$

Definition

$\mathsf{pred} \equiv \lambda n.n(\lambda uv.v(u\,\mathsf{succ}))(\lambda z.\underline{0})(\lambda z.z)$

Definition

$\mathsf{pred} \equiv \lambda n.n(\lambda uv.v(u\,\mathsf{succ}))(\lambda z.\underline{0})(\lambda z.z)$

Lemma

$$\mathsf{pred}\ \underline{n} \rightarrow_\beta^* \begin{cases} \underline{0} & \text{if } n = 0 \\ \underline{n-1} & \text{if } n > 0 \end{cases}$$

Definition

$\mathsf{pred} \equiv \lambda n.n(\lambda uv.v(u\,\mathsf{succ}))(\lambda z.\underline{0})(\lambda z.z)$

Lemma

$\mathsf{pred}\ \underline{n} \to_\beta^* \begin{cases} \underline{0} & \text{if } n = 0 \\ \underline{n-1} & \text{if } n > 0 \end{cases}$

Proof

- $\mathsf{pred}\ \underline{0} \to_\beta \underline{0}(\lambda uv.v(u\,\mathsf{succ}))(\lambda z.\underline{0})(\lambda z.z)$

Definition

$\mathsf{pred} \equiv \lambda n.n(\lambda uv.v(u\,\mathsf{succ}))(\lambda z.\underline{0})(\lambda z.z)$

Lemma

$$\mathsf{pred}\ \underline{n} \rightarrow_\beta^* \begin{cases} \underline{0} & \text{if } n = 0 \\ \underline{n-1} & \text{if } n > 0 \end{cases}$$

Proof

- $\mathsf{pred}\ \underline{0} \rightarrow_\beta \underline{0}(\lambda uv.v(u\,\mathsf{succ}))(\lambda z.\underline{0})(\lambda z.z) \rightarrow_\beta^* (\lambda z.\underline{0})(\lambda z.z)$

Definition

$\text{pred} \equiv \lambda n.n(\lambda uv.v(u\,\text{succ}))(\lambda z.\underline{0})(\lambda z.z)$

Lemma

$$\text{pred}\ \underline{n} \to_\beta^* \begin{cases} \underline{0} & \text{if } n = 0 \\ \underline{n-1} & \text{if } n > 0 \end{cases}$$

Proof

- $\text{pred}\ \underline{0} \to_\beta \underline{0}(\lambda uv.v(u\,\text{succ}))(\lambda z.\underline{0})(\lambda z.z) \to_\beta^* (\lambda z.\underline{0})(\lambda z.z) \to_\beta \underline{0}$

Definition

$\text{pred} \equiv \lambda n.n(\lambda uv.v(u\,\text{succ}))(\lambda z.\underline{0})(\lambda z.z)$

Lemma

$$\text{pred}\ \underline{n} \to_\beta^* \begin{cases} \underline{0} & \text{if } n = 0 \\ \underline{n-1} & \text{if } n > 0 \end{cases}$$

Proof

- $\text{pred}\ \underline{0} \to_\beta \underline{0}(\lambda uv.v(u\,\text{succ}))(\lambda z.\underline{0})(\lambda z.z) \to_\beta^* (\lambda z.\underline{0})(\lambda z.z) \to_\beta \underline{0}$
- $\text{pred}\ \underline{n+1} \to_\beta^* \underline{n}$ (homework exercise)

Example

representing factorial function in lambda calculus

$$\mathsf{fac}\, n = \mathsf{ite}\,(\mathsf{zero?}\, n)\, \underline{1}\, (\mathsf{mul}\, n\, (\mathsf{fac}\, (\mathsf{pred}\, n)))$$

Example

representing factorial function in lambda calculus

$$\mathsf{fac}\; n = \mathsf{ite}\,(\mathsf{zero?}\; n)\; \underline{1}\; (\mathsf{mul}\; n\; (\mathsf{fac}\,(\mathsf{pred}\; n)))$$

- $\mathsf{fac} = \lambda n.\, \mathsf{ite}\,(\mathsf{zero?}\; n)\; \underline{1}\; (\mathsf{mul}\; n\; (\mathsf{fac}\,(\mathsf{pred}\; n)))$

Example

representing factorial function in lambda calculus

$$\mathsf{fac}\,n = \mathsf{ite}\,(\mathsf{zero?}\,n)\,\underline{1}\,(\mathsf{mul}\,n\,(\mathsf{fac}\,(\mathsf{pred}\,n)))$$

- $\mathsf{fac} = \lambda n.\,\mathsf{ite}\,(\mathsf{zero?}\,n)\,\underline{1}\,(\mathsf{mul}\,n\,(\mathsf{fac}\,(\mathsf{pred}\,n)))$
- $\mathsf{fac} = (\lambda f\,n.\,\mathsf{ite}\,(\mathsf{zero?}\,n)\,\underline{1}\,(\mathsf{mul}\,n\,(f\,(\mathsf{pred}\,n))))\,\mathsf{fac}$

Example

representing factorial function in lambda calculus

$$\mathsf{fac}\; n = \mathsf{ite}\,(\mathsf{zero?}\; n)\; \underline{1}\; (\mathsf{mul}\; n\; (\mathsf{fac}\,(\mathsf{pred}\; n)))$$

- $\mathsf{fac} = \lambda n.\, \mathsf{ite}\,(\mathsf{zero?}\; n)\; \underline{1}\; (\mathsf{mul}\; n\; (\mathsf{fac}\,(\mathsf{pred}\; n)))$
- $\mathsf{fac} = (\lambda f\, n.\, \mathsf{ite}\,(\mathsf{zero?}\; n)\; \underline{1}\; (\mathsf{mul}\; n\; (f\,(\mathsf{pred}\; n))))\; \mathsf{fac}$
- $\mathsf{fac} = \Theta\, F$ with $F \equiv (\lambda f\, n.\, \mathsf{ite}\,(\mathsf{zero?}\; n)\; \underline{1}\; (\mathsf{mul}\; n\; (f\,(\mathsf{pred}\; n))))$

Example

representing factorial function in lambda calculus

$$\mathsf{fac}\, n = \mathsf{ite}\,(\mathsf{zero?}\, n)\, \underline{1}\, (\mathsf{mul}\, n\, (\mathsf{fac}\, (\mathsf{pred}\, n)))$$

- $\mathsf{fac} = \lambda n.\, \mathsf{ite}\,(\mathsf{zero?}\, n)\, \underline{1}\, (\mathsf{mul}\, n\, (\mathsf{fac}\, (\mathsf{pred}\, n)))$
- $\mathsf{fac} = (\lambda f\, n.\, \mathsf{ite}\,(\mathsf{zero?}\, n)\, \underline{1}\, (\mathsf{mul}\, n\, (f\, (\mathsf{pred}\, n))))\, \mathsf{fac}$
- $\mathsf{fac} = \Theta F$ with $F \equiv (\lambda f\, n.\, \mathsf{ite}\,(\mathsf{zero?}\, n)\, \underline{1}\, (\mathsf{mul}\, n\, (f\, (\mathsf{pred}\, n))))$

$$\mathsf{fac}\, \underline{2} \to_\beta^* F\, \mathsf{fac}\, \underline{2}$$

Example

representing factorial function in lambda calculus

$$\mathsf{fac}\,n = \mathsf{ite}\,(\mathsf{zero?}\,n)\,\underline{1}\,(\mathsf{mul}\,n\,(\mathsf{fac}\,(\mathsf{pred}\,n)))$$

- $\mathsf{fac} = \lambda n.\,\mathsf{ite}\,(\mathsf{zero?}\,n)\,\underline{1}\,(\mathsf{mul}\,n\,(\mathsf{fac}\,(\mathsf{pred}\,n)))$
- $\mathsf{fac} = (\lambda f\,n.\,\mathsf{ite}\,(\mathsf{zero?}\,n)\,\underline{1}\,(\mathsf{mul}\,n\,(f\,(\mathsf{pred}\,n))))\,\mathsf{fac}$
- $\mathsf{fac} = \Theta\,F$ with $F \equiv (\lambda f\,n.\,\mathsf{ite}\,(\mathsf{zero?}\,n)\,\underline{1}\,(\mathsf{mul}\,n\,(f\,(\mathsf{pred}\,n))))$

$$\begin{aligned}\mathsf{fac}\,\underline{2} &\to_\beta^* F\,\mathsf{fac}\,\underline{2}\\ &\to_\beta^* \mathsf{ite}\,(\mathsf{zero?}\,\underline{2})\,\underline{1}\,(\mathsf{mul}\,\underline{2}\,(\mathsf{fac}\,(\mathsf{pred}\,\underline{2})))\end{aligned}$$

Example

representing factorial function in lambda calculus

$$\mathsf{fac}\ n = \mathsf{ite}\,(\mathsf{zero?}\ n)\ \underline{1}\ (\mathsf{mul}\ n\ (\mathsf{fac}\,(\mathsf{pred}\ n)))$$

- $\mathsf{fac} = \lambda n.\,\mathsf{ite}\,(\mathsf{zero?}\ n)\ \underline{1}\ (\mathsf{mul}\ n\ (\mathsf{fac}\,(\mathsf{pred}\ n)))$
- $\mathsf{fac} = (\lambda f\,n.\,\mathsf{ite}\,(\mathsf{zero?}\ n)\ \underline{1}\ (\mathsf{mul}\ n\ (f\,(\mathsf{pred}\ n))))\ \mathsf{fac}$
- $\mathsf{fac} = \Theta F$ with $F \equiv (\lambda f\,n.\,\mathsf{ite}\,(\mathsf{zero?}\ n)\ \underline{1}\ (\mathsf{mul}\ n\ (f\,(\mathsf{pred}\ n))))$

$$\begin{aligned}\mathsf{fac}\ \underline{2} &\to_\beta^* F\ \mathsf{fac}\ \underline{2}\\ &\to_\beta^* \mathsf{ite}\,(\mathsf{zero?}\ \underline{2})\ \underline{1}\ (\mathsf{mul}\ \underline{2}\ (\mathsf{fac}\,(\mathsf{pred}\ \underline{2})))\\ &\to_\beta^* \mathsf{ite}\ \mathsf{F}\ \underline{1}\ (\mathsf{mul}\ \underline{2}\ (\mathsf{fac}\,(\mathsf{pred}\ \underline{2})))\end{aligned}$$

Example

representing factorial function in lambda calculus

$$\mathsf{fac}\; n \;=\; \mathsf{ite}\,(\mathsf{zero?}\; n)\; \underline{1}\; (\mathsf{mul}\; n\; (\mathsf{fac}\,(\mathsf{pred}\; n)))$$

- $\mathsf{fac} \;=\; \lambda n.\, \mathsf{ite}\,(\mathsf{zero?}\; n)\; \underline{1}\; (\mathsf{mul}\; n\; (\mathsf{fac}\,(\mathsf{pred}\; n)))$
- $\mathsf{fac} \;=\; (\lambda f\, n.\, \mathsf{ite}\,(\mathsf{zero?}\; n)\; \underline{1}\; (\mathsf{mul}\; n\; (f\,(\mathsf{pred}\; n))))\; \mathsf{fac}$
- $\mathsf{fac} \;=\; \Theta\, F$ with $F \equiv (\lambda f\, n.\, \mathsf{ite}\,(\mathsf{zero?}\; n)\; \underline{1}\; (\mathsf{mul}\; n\; (f\,(\mathsf{pred}\; n))))$

$$\begin{aligned}
\mathsf{fac}\; \underline{2} &\rightarrow_\beta^* F\; \mathsf{fac}\; \underline{2} \\
&\rightarrow_\beta^* \mathsf{ite}\,(\mathsf{zero?}\; \underline{2})\; \underline{1}\; (\mathsf{mul}\; \underline{2}\; (\mathsf{fac}\,(\mathsf{pred}\; \underline{2}))) \\
&\rightarrow_\beta^* \mathsf{ite\; F}\; \underline{1}\; (\mathsf{mul}\; \underline{2}\; (\mathsf{fac}\,(\mathsf{pred}\; \underline{2}))) \\
&\rightarrow_\beta^* \mathsf{mul}\; \underline{2}\; (\mathsf{fac}\,(\mathsf{pred}\; \underline{2}))
\end{aligned}$$

Example

representing factorial function in lambda calculus

$$\text{fac}\ n = \text{ite}\,(\text{zero?}\ n)\ \underline{1}\ (\text{mul}\ n\ (\text{fac}\,(\text{pred}\ n)))$$

- $\text{fac} = \lambda n.\ \text{ite}\,(\text{zero?}\ n)\ \underline{1}\ (\text{mul}\ n\ (\text{fac}\,(\text{pred}\ n)))$
- $\text{fac} = (\lambda f\, n.\ \text{ite}\,(\text{zero?}\ n)\ \underline{1}\ (\text{mul}\ n\ (f\,(\text{pred}\ n))))\ \text{fac}$
- $\text{fac} = \Theta F$ with $F \equiv (\lambda f\, n.\ \text{ite}\,(\text{zero?}\ n)\ \underline{1}\ (\text{mul}\ n\ (f\,(\text{pred}\ n))))$

$$\begin{aligned}
\text{fac}\ \underline{2} &\to_\beta^* F\ \text{fac}\ \underline{2} \\
&\to_\beta^* \text{ite}\,(\text{zero?}\ \underline{2})\ \underline{1}\ (\text{mul}\ \underline{2}\ (\text{fac}\,(\text{pred}\ \underline{2}))) \\
&\to_\beta^* \text{ite F}\ \underline{1}\ (\text{mul}\ \underline{2}\ (\text{fac}\,(\text{pred}\ \underline{2}))) \\
&\to_\beta^* \text{mul}\ \underline{2}\ (\text{fac}\,(\text{pred}\ \underline{2})) \\
&\to_\beta^* \text{mul}\ \underline{2}\ (\text{fac}\ \underline{1})
\end{aligned}$$

Example

representing factorial function in lambda calculus

$$\mathsf{fac}\ n = \mathsf{ite}\,(\mathsf{zero?}\,n)\ \underline{1}\ (\mathsf{mul}\ n\ (\mathsf{fac}\,(\mathsf{pred}\ n)))$$

- $\mathsf{fac} = \lambda n.\,\mathsf{ite}\,(\mathsf{zero?}\,n)\ \underline{1}\ (\mathsf{mul}\ n\ (\mathsf{fac}\,(\mathsf{pred}\ n)))$
- $\mathsf{fac} = (\lambda f\,n.\,\mathsf{ite}\,(\mathsf{zero?}\,n)\ \underline{1}\ (\mathsf{mul}\ n\ (f\,(\mathsf{pred}\ n))))\ \mathsf{fac}$
- $\mathsf{fac} = \Theta F$ with $F \equiv (\lambda f\,n.\,\mathsf{ite}\,(\mathsf{zero?}\,n)\ \underline{1}\ (\mathsf{mul}\ n\ (f\,(\mathsf{pred}\ n))))$

$$\begin{aligned}
\mathsf{fac}\ \underline{2} &\to_\beta^* F\ \mathsf{fac}\ \underline{2} \\
&\to_\beta^* \mathsf{ite}\,(\mathsf{zero?}\,\underline{2})\ \underline{1}\ (\mathsf{mul}\ \underline{2}\ (\mathsf{fac}\,(\mathsf{pred}\ \underline{2}))) \\
&\to_\beta^* \mathsf{ite}\ \mathsf{F}\ \underline{1}\ (\mathsf{mul}\ \underline{2}\ (\mathsf{fac}\,(\mathsf{pred}\ \underline{2}))) \\
&\to_\beta^* \mathsf{mul}\ \underline{2}\ (\mathsf{fac}\,(\mathsf{pred}\ \underline{2})) \\
&\to_\beta^* \mathsf{mul}\ \underline{2}\ (\mathsf{fac}\ \underline{1}) \\
&\to_\beta^* \cdots \\
&\to_\beta^* \mathsf{mul}\ \underline{2}\ (\mathsf{mul}\ \underline{1}\ (\mathsf{fac}\ \underline{0}))
\end{aligned}$$

Example

representing factorial function in lambda calculus

$$\mathsf{fac}\ n = \mathsf{ite}\,(\mathsf{zero?}\ n)\ \underline{1}\ (\mathsf{mul}\ n\ (\mathsf{fac}\,(\mathsf{pred}\ n)))$$

- $\mathsf{fac} = \lambda n.\,\mathsf{ite}\,(\mathsf{zero?}\ n)\ \underline{1}\ (\mathsf{mul}\ n\ (\mathsf{fac}\,(\mathsf{pred}\ n)))$
- $\mathsf{fac} = (\lambda f\,n.\,\mathsf{ite}\,(\mathsf{zero?}\ n)\ \underline{1}\ (\mathsf{mul}\ n\ (f\,(\mathsf{pred}\ n))))\ \mathsf{fac}$
- $\mathsf{fac} = \Theta\,F$ with $F \equiv (\lambda f\,n.\,\mathsf{ite}\,(\mathsf{zero?}\ n)\ \underline{1}\ (\mathsf{mul}\ n\ (f\,(\mathsf{pred}\ n))))$

$$\begin{aligned}
\mathsf{fac}\ \underline{2} &\to_\beta^* F\ \mathsf{fac}\ \underline{2} \\
&\to_\beta^* \mathsf{ite}\,(\mathsf{zero?}\ \underline{2})\ \underline{1}\ (\mathsf{mul}\ \underline{2}\ (\mathsf{fac}\,(\mathsf{pred}\ \underline{2}))) \\
&\to_\beta^* \mathsf{ite\ F}\ \underline{1}\ (\mathsf{mul}\ \underline{2}\ (\mathsf{fac}\,(\mathsf{pred}\ \underline{2}))) \\
&\to_\beta^* \mathsf{mul}\ \underline{2}\ (\mathsf{fac}\,(\mathsf{pred}\ \underline{2})) \\
&\to_\beta^* \mathsf{mul}\ \underline{2}\ (\mathsf{fac}\ \underline{1}) \\
&\to_\beta^* \cdots \\
&\to_\beta^* \mathsf{mul}\ \underline{2}\ (\mathsf{mul}\ \underline{1}\ (\mathsf{fac}\ \underline{0})) \to_\beta^* \mathsf{mul}\ \underline{2}\ (\mathsf{mul}\ \underline{1}\ \underline{1})
\end{aligned}$$

Example

representing factorial function in lambda calculus

$$\mathsf{fac}\; n \;=\; \mathsf{ite}\,(\mathsf{zero?}\; n)\; \underline{1}\; (\mathsf{mul}\; n\; (\mathsf{fac}\,(\mathsf{pred}\; n)))$$

- $\mathsf{fac} \;=\; \lambda n.\, \mathsf{ite}\,(\mathsf{zero?}\; n)\; \underline{1}\; (\mathsf{mul}\; n\; (\mathsf{fac}\,(\mathsf{pred}\; n)))$
- $\mathsf{fac} \;=\; (\lambda f\, n.\, \mathsf{ite}\,(\mathsf{zero?}\; n)\; \underline{1}\; (\mathsf{mul}\; n\; (f\,(\mathsf{pred}\; n))))\; \mathsf{fac}$
- $\mathsf{fac} \;=\; \Theta\, F$ with $F \equiv (\lambda f\, n.\, \mathsf{ite}\,(\mathsf{zero?}\; n)\; \underline{1}\; (\mathsf{mul}\; n\; (f\,(\mathsf{pred}\; n))))$

$$\begin{aligned}
\mathsf{fac}\; \underline{2} &\to_\beta^* F\; \mathsf{fac}\; \underline{2} \\
&\to_\beta^* \mathsf{ite}\,(\mathsf{zero?}\; \underline{2})\; \underline{1}\; (\mathsf{mul}\; \underline{2}\; (\mathsf{fac}\,(\mathsf{pred}\; \underline{2}))) \\
&\to_\beta^* \mathsf{ite}\; \mathsf{F}\; \underline{1}\; (\mathsf{mul}\; \underline{2}\; (\mathsf{fac}\,(\mathsf{pred}\; \underline{2}))) \\
&\to_\beta^* \mathsf{mul}\; \underline{2}\; (\mathsf{fac}\,(\mathsf{pred}\; \underline{2})) \\
&\to_\beta^* \mathsf{mul}\; \underline{2}\; (\mathsf{fac}\; \underline{1}) \\
&\to_\beta^* \cdots \\
&\to_\beta^* \mathsf{mul}\; \underline{2}\; (\mathsf{mul}\; \underline{1}\; (\mathsf{fac}\; \underline{0})) \to_\beta^* \mathsf{mul}\; \underline{2}\; (\mathsf{mul}\; \underline{1}\; \underline{1}) \to_\beta^* \underline{2}
\end{aligned}$$

Definition

partial function $f\colon \mathbb{N}^n \to \mathbb{N}$ is λ–definable if $\exists$ combinator F such that

$$f(x_1, \ldots, x_n) = y \qquad \Longrightarrow \qquad F\, \underline{x_1} \cdots \underline{x_n} \to_\beta^* \underline{y}$$

for all $x_1, \ldots, x_n, y \in \mathbb{N}$

Definition

partial function $f\colon \mathbb{N}^n \to \mathbb{N}$ is λ–definable if $\exists$ combinator F such that

$$f(x_1,\dots,x_n) = y \quad\Longrightarrow\quad F\,\underline{x_1}\,\cdots\,\underline{x_n} \to_\beta^* \underline{y}$$

$$f(x_1,\dots,x_n) \text{ is undefined} \quad\Longrightarrow\quad F\,\underline{x_1}\,\cdots\,\underline{x_n} \text{ is not normalizing}$$

for all $x_1,\dots,x_n, y \in \mathbb{N}$

Theorem

partial recursive functions are λ–definable

Definition

partial function $f\colon \mathbb{N}^n \to \mathbb{N}$ is λ–definable if $\exists$ combinator F such that

$$f(x_1,\ldots,x_n) = y \quad\Longrightarrow\quad F\,\underline{x_1}\cdots\underline{x_n} \to_\beta^* \underline{y}$$

$$f(x_1,\ldots,x_n) \text{ is undefined} \quad\Longrightarrow\quad F\,\underline{x_1}\cdots\underline{x_n} \text{ is not normalizing}$$

for all $x_1,\ldots,x_n,y \in \mathbb{N}$

Theorem

partial recursive functions are λ–definable

Proof

- zero function $\quad z(x) = 0$

Definition

partial function $f\colon \mathbb{N}^n \to \mathbb{N}$ is λ–definable if $\exists$ combinator F such that

$$f(x_1,\dots,x_n) = y \quad\Longrightarrow\quad F\,\underline{x_1}\cdots\underline{x_n} \to_\beta^* \underline{y}$$
$$f(x_1,\dots,x_n) \text{ is undefined} \quad\Longrightarrow\quad F\,\underline{x_1}\cdots\underline{x_n} \text{ is not normalizing}$$

for all $x_1,\dots,x_n,y \in \mathbb{N}$

Theorem

partial recursive functions are λ–definable

Proof

- zero function $\quad z(x) = 0 \quad$ zero $\equiv \lambda x.\underline{0}$

Definition

partial function $f \colon \mathbb{N}^n \to \mathbb{N}$ is λ–definable if $\exists$ combinator F such that

$$f(x_1, \dots, x_n) = y \quad\Longrightarrow\quad F\,\underline{x_1} \cdots \underline{x_n} \to_\beta^* \underline{y}$$

$$f(x_1, \dots, x_n) \text{ is undefined} \quad\Longrightarrow\quad F\,\underline{x_1} \cdots \underline{x_n} \text{ is not normalizing}$$

for all $x_1, \dots, x_n, y \in \mathbb{N}$

Theorem

partial recursive functions are λ–definable

Proof

- zero function $z(x) = 0$ zero $\equiv \lambda x.\underline{0}$
- successor function $s(x) = x + 1$ succ $\equiv \lambda n f x. f(n f x)$

Definition

partial function $f\colon \mathbb{N}^n \to \mathbb{N}$ is λ–definable if $\exists$ combinator F such that

$$f(x_1,\dots,x_n) = y \qquad \Longrightarrow \qquad F\,\underline{x_1}\cdots\underline{x_n} \to_\beta^* \underline{y}$$

$$f(x_1,\dots,x_n) \text{ is undefined} \qquad \Longrightarrow \qquad F\,\underline{x_1}\cdots\underline{x_n} \text{ is not normalizing}$$

for all $x_1,\dots,x_n,y \in \mathbb{N}$

Theorem

partial recursive functions are λ–definable

Proof

- zero function $z(x) = 0$ $\quad$ zero $\equiv \lambda x.\underline{0}$
- successor function $s(x) = x + 1$ $\quad$ succ $\equiv \lambda nfx.f(nfx)$
- projection functions $\pi_i^n(x_1,\dots,x_n) = x_i$

Definition

partial function $f\colon \mathbb{N}^n \to \mathbb{N}$ is λ–definable if $\exists$ combinator F such that

$$f(x_1,\dots,x_n) = y \quad\Longrightarrow\quad F\,\underline{x_1}\cdots\underline{x_n} \to_\beta^* \underline{y}$$

$$f(x_1,\dots,x_n) \text{ is undefined} \quad\Longrightarrow\quad F\,\underline{x_1}\cdots\underline{x_n} \text{ is not normalizing}$$

for all $x_1,\dots,x_n,y \in \mathbb{N}$

Theorem

partial recursive functions are λ–definable

Proof

- zero function $z(x) = 0$ zero $\equiv \lambda x.\underline{0}$
- successor function $s(x) = x + 1$ succ $\equiv \lambda nfx.f(nfx)$
- projection functions $\pi_i^n(x_1,\dots,x_n) = x_i$ $\pi_i^n \equiv \lambda x_1\cdots x_n.x_i$

Proof (cont'd)

- composition $f(\vec{x}) = g(h_1(\vec{x}), \ldots, h_m(\vec{x}))$

Proof (cont'd)

- composition $f(\vec{x}) = g(h_1(\vec{x}), \ldots, h_m(\vec{x}))$

$$F \equiv \lambda \vec{x}.\, G\,(H_1\,\vec{x}) \cdots (H_m\,\vec{x})$$

Proof (cont'd)

- composition $$f(\vec{x}) = g(h_1(\vec{x}), \dots, h_m(\vec{x}))$$

$$F \equiv \lambda \vec{x}.\, G\,(H_1\,\vec{x}) \cdots (H_m\,\vec{x})$$

- primitive recusion $$f(0, \vec{y}) = g(\vec{y})$$ $$f(x+1, \vec{y}) = h(f(x, \vec{y}), x, \vec{y})$$

Proof (cont'd)

- composition $$f(\vec{x}) = g(h_1(\vec{x}), \ldots, h_m(\vec{x}))$$

$$F \equiv \lambda \vec{x}.\, G\,(H_1\,\vec{x}) \cdots (H_m\,\vec{x})$$

- primitive recusion
$$f(0, \vec{y}) = g(\vec{y})$$
$$f(x+1, \vec{y}) = h(f(x, \vec{y}), x, \vec{y})$$

$$F = \lambda x\,\vec{y}.\, \mathsf{ite}\,(\mathsf{zero?}\,x)\,(G\,\vec{y})\,(H\,(F\,(\mathsf{pred}\,x)\,\vec{y})\,(\mathsf{pred}\,x)\,\vec{y})$$

Proof (cont'd)

- composition $$f(\vec{x}) = g(h_1(\vec{x}), \ldots, h_m(\vec{x}))$$

$$F \equiv \lambda\vec{x}.\, G\,(H_1\,\vec{x}) \cdots (H_m\,\vec{x})$$

- primitive recusion
$$f(0, \vec{y}) = g(\vec{y})$$
$$f(x+1, \vec{y}) = h(f(x, \vec{y}), x, \vec{y})$$

$$F = \lambda x\,\vec{y}.\, \mathsf{ite}\,(\mathsf{zero?}\,x)\,(G\,\vec{y})\,(H\,(F\,(\mathsf{pred}\,x)\,\vec{y})\,(\mathsf{pred}\,x)\,\vec{y})$$
$$= \Theta\,(\lambda f\,x\,\vec{y}.\, \mathsf{ite}\,(\mathsf{zero?}\,x)\,(G\,\vec{y})\,(H\,(f\,(\mathsf{pred}\,x)\,\vec{y})\,(\mathsf{pred}\,x)\,\vec{y}))$$

Proof (cont'd)

- minimization $f(\vec{x}) = (\mu\, i)\,(g(i, x_1, \ldots, x_n) = 0)$

Proof (cont'd)

- minimization $f(\vec{x}) = (\mu\, i)\, (g(i, x_1, \dots, x_n) = 0)$

$$F \equiv H\, \underline{0}$$

with

$$H = \lambda i\, \vec{x}.\, \mathsf{ite}\, (\mathsf{zero?}\, (G\, i\, \vec{x}))\, i\, (H\, (\mathsf{succ}\, i)\, \vec{x})$$

Proof (cont'd)

- minimization $f(\vec{x}) = (\mu\, i)\, (g(i, x_1, \ldots, x_n) = 0)$

$$F \equiv H\, \underline{0}$$

with

$$H = \lambda i\, \vec{x}.\, \mathsf{ite}\, (\mathsf{zero?}\, (G\, i\, \vec{x}))\, i\, (H\, (\mathsf{succ}\, i)\, \vec{x})$$
$$= \Theta\, (\lambda h\, i\, \vec{x}.\, \mathsf{ite}\, (\mathsf{zero?}\, (G\, i\, \vec{x}))\, i\, (h\, (\mathsf{succ}\, i)\, \vec{x}))$$

Proof (cont'd)

- minimization $f(\vec{x}) = (\mu\, i)\,(g(i, x_1, \dots, x_n) = 0)$

$$F \equiv H\,\underline{0}$$

with

$$H = \lambda i\,\vec{x}.\,\mathsf{ite}\,(\mathsf{zero?}\,(G\,i\,\vec{x}))\;i\;(H\,(\mathsf{succ}\;i)\;\vec{x})$$
$$= \Theta\,(\lambda h\,i\,\vec{x}.\,\mathsf{ite}\,(\mathsf{zero?}\,(G\,i\,\vec{x}))\;i\;(h\,(\mathsf{succ}\;i)\;\vec{x}))$$

Theorem

λ–definable function are partial recursive

Proof (cont'd)

- minimization $\quad f(\vec{x}) = (\mu\, i)\,(g(i, x_1, \ldots, x_n) = 0)$

$$F \equiv H\,\underline{0}$$

with

$$\begin{aligned} H &= \lambda i\,\vec{x}.\,\mathsf{ite}\,(\mathsf{zero?}\,(G\,i\,\vec{x}))\;i\;(H\,(\mathsf{succ}\;i)\,\vec{x}) \\ &= \Theta\,(\lambda h\,i\,\vec{x}.\,\mathsf{ite}\,(\mathsf{zero?}\,(G\,i\,\vec{x}))\;i\;(h\,(\mathsf{succ}\;i)\,\vec{x})) \end{aligned}$$

Theorem

λ–definable function are partial recursive

Remark

however, cf. slide 28 of lecture 8 and slides 19–21 of lecture 9

Outline

Theorem

β–reduction has Church–Rosser property

Theorem

β–reduction has Church–Rosser property

Corollary

$x \neq_{\beta} \lambda y.xy$

Theorem

β–reduction has Church–Rosser property

Corollary

$x \neq_\beta \lambda y.xy$

Corollary

λ–calculus is consistent

Theorem

β–reduction has Church–Rosser property

Corollary

$x \neq_\beta \lambda y.xy$

Corollary

λ–calculus is consistent

Remark

x and $\lambda y.xy$ are extensionally equivalent: $xM =_\beta (\lambda y.xy)M$ for all $M \in \Lambda$

Definition

one-step η–reduction is smallest relation $\to_\eta$ on lambda terms satisfying

$$(\eta) \qquad \frac{x \notin \mathsf{FV}(M)}{\lambda x.Mx \to_\eta M}$$

Definition

one-step η–reduction is smallest relation $\to_\eta$ on lambda terms satisfying

$$(\eta)\quad \frac{x \notin \mathsf{FV}(M)}{\lambda x.Mx \to_\eta M} \qquad\qquad \frac{M \to_\eta M'}{MN \to_\eta M'N}\quad \text{(congruence)}$$

$$(\xi)\quad \frac{M \to_\eta M'}{\lambda x.M \to_\eta \lambda x.M'} \qquad\qquad \frac{N \to_\eta N'}{MN \to_\eta MN'}\quad \text{(congruence)}$$

Definition

one-step $\beta\eta$–reduction $\to_{\beta\eta}$ is union of $\to_\beta$ and $\to_\eta$

Definition

one-step η–reduction is smallest relation $\rightarrow_\eta$ on lambda terms satisfying

$$(\eta)\quad \frac{x \notin \mathrm{FV}(M)}{\lambda x.Mx \rightarrow_\eta M} \qquad\qquad \frac{M \rightarrow_\eta M'}{MN \rightarrow_\eta M'N}\quad \text{(congruence)}$$

$$(\xi)\quad \frac{M \rightarrow_\eta M'}{\lambda x.M \rightarrow_\eta \lambda x.M'} \qquad\qquad \frac{N \rightarrow_\eta N'}{MN \rightarrow_\eta MN'}\quad \text{(congruence)}$$

Definition

one-step $\beta\eta$–reduction $\rightarrow_{\beta\eta}$ is union of $\rightarrow_\beta$ and $\rightarrow_\eta$

Theorem

$\beta\eta$–reduction has Church-Rosser property

Outline

1. **Summary of Previous Lecture**
2. **β–Reduction**
3. **Church–Rosser Theorem**
4. **λ–Definability**
5. **η–Reduction**
6. **Normalization Theorem**
7. **Test Practice**
8. **Summary**

Example

- $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$

Example

- $\Omega \equiv (\lambda x.xx)(\lambda x.xx) \to_\beta \Omega$

Example

- $\Omega \equiv (\lambda x.xx)(\lambda x.xx) \rightarrow_\beta \Omega$
- $(\lambda x.y)\,\Omega$ has β–normal form

Example

- $\Omega \equiv (\lambda x.xx)(\lambda x.xx) \rightarrow_\beta \Omega$
- $(\lambda x.y)\,\Omega$ has β–normal form: $(\lambda x.y)\,\Omega \rightarrow_\beta y$

Example

- $\Omega \equiv (\lambda x.xx)(\lambda x.xx) \to_\beta \Omega$
- $(\lambda x.y)\,\Omega$ has β–normal form: $(\lambda x.y)\,\Omega \to_\beta y$
- $(\lambda x.y)\,\Omega$ admits infinite reduction

Example

- $\Omega \equiv (\lambda x.xx)(\lambda x.xx) \rightarrow_\beta \Omega$
- $(\lambda x.y)\,\Omega$ has β–normal form: $(\lambda x.y)\,\Omega \rightarrow_\beta y$
- $(\lambda x.y)\,\Omega$ admits infinite reduction: $(\lambda x.y)\,\Omega \rightarrow_\beta (\lambda x.y)\,\Omega \rightarrow_\beta \cdots$

Example

- $\Omega \equiv (\lambda x.xx)(\lambda x.xx) \to_\beta \Omega$
- $(\lambda x.y)\,\Omega$ has β–normal form: $(\lambda x.y)\,\Omega \to_\beta y$
- $(\lambda x.y)\,\Omega$ admits infinite reduction: $(\lambda x.y)\,\Omega \to_\beta (\lambda x.y)\,\Omega \to_\beta \cdots$

Question

how to compute β–normal forms (or $\beta\eta$–normal forms) ?

Example

- $\Omega \equiv (\lambda x.xx)(\lambda x.xx) \to_\beta \Omega$
- $(\lambda x.y)\,\Omega$ has β–normal form: $(\lambda x.y)\,\Omega \to_\beta y$
- $(\lambda x.y)\,\Omega$ admits infinite reduction: $(\lambda x.y)\,\Omega \to_\beta (\lambda x.y)\,\Omega \to_\beta \cdots$

Question

how to compute β–normal forms (or $\beta\eta$–normal forms) ?

Answer

always select leftmost redex

Example

- $\Omega \equiv (\lambda x.xx)(\lambda x.xx) \rightarrow_\beta \Omega$
- $(\lambda x.y)\,\Omega$ has β–normal form: $(\lambda x.y)\,\Omega \rightarrow_\beta y$
- $(\lambda x.y)\,\Omega$ admits infinite reduction: $(\lambda x.y)\,\Omega \rightarrow_\beta (\lambda x.y)\,\Omega \rightarrow_\beta \cdots$

Question

how to compute β–normal forms (or $\beta\eta$–normal forms) ?

Answer

always select leftmost redex

Normalization Theorem

leftmost reduction strategy is normalizing

Outline

1. **Summary of Previous Lecture**
2. **β–Reduction**
3. **Church–Rosser Theorem**
4. **λ–Definability**
5. **η–Reduction**
6. **Normalization Theorem**
7. **Test Practice**
8. **Summary**

Test on January 29

- 15:15 – 18:00 in HS 10
- online registration required before 10 am on January 23
- closed book

Test on January 29

- 15:15 – 18:00 in HS 10
- online registration required before 10 am on January 23
- closed book
- score $=$ $\min\left(\max\left(\frac{2}{3}(E+P)+\frac{1}{3}T+B, T+B\right), 100\right)$

Test on January 29

- 15:15 – 18:00 in HS 10
- online registration required before 10 am on January 23
- closed book
- score $= \min\left(\max\left(\frac{2}{3}(E + P) + \frac{1}{3}T + B, T + B\right), 100\right)$

Earlier Exams / Tests

- SS 2022 (test)
- WS 2017 – 2
- WS 2017 – 1
- WS 2014 – 2
- WS 2014 – 1
- SS 2012
- SS 2008 – 2
- SS 2008 – 1
- SS 2007
- SS 2006 – 2
- SS 2006 – 1
- WS 2004

Test on January 29

- 15:15 – 18:00 in HS 10
- online registration required before 10 am on January 23
- closed book
- score $= \min\left(\max\left(\frac{2}{3}(E+P)+\frac{1}{3}T+B, T+B\right), 100\right)$

Earlier Exams / Tests

- SS 2022 (test)
- WS 2017 – 2
- WS 2017 – 1
- WS 2014 – 2
- WS 2014 – 1
- SS 2012
- SS 2008 – 2
- SS 2008 – 1
- SS 2007
- SS 2006 – 2
- SS 2006 – 1
- WS 2004

test practice on January 22

Outline

Important Concepts

- α–conversion
- β–conversion
- β–reduction
- Church–Rosser theorem
- η–reduction
- lambda calculus
- λ–definability
- normalization theorem

Important Concepts

- α–conversion
- β–conversion
- β–reduction
- Church–Rosser theorem
- η–reduction
- lambda calculus
- λ–definability
- normalization theorem

homework for January 22