Lambda Calculus for Language Modeling

Day One: Lambda Calculus

Greg Kobele and Sylvain Salvati

23. June 2014
NASSLLI 6

Course Outline

Monday	Intro to λ-calculus
Tuesday	Using λ-calculus for syntax (I)
Wednesday	Using λ-calculus for syntax (II)
Thursday	Models of the λ-calculus
Friday	Using λ-calculus for semantics

Broad Overview

- Why the λ-calculus?
- Why model language with it?

Main points today:

1. Church Rosser Theorem
2. Strong Normalization for simple types
3. Inhabitation and η-long forms

λ-Terms

- Intuition: λ-terms represent functions

  $$f(x) = x^2 + 2x + 1 \qquad \rightsquigarrow \qquad \lambda x.x^2 + 2x + 1$$

  - can *apply* functions to arguments:

    $$f(4)$$

  - can *create* new functions from old ones:

    $$\begin{array}{ll} \mathbf{plus}(x, y) = x + y & \mathbf{square}(x) = x^2 \\ \mathbf{double}(x) = 2x & \mathbf{one} = 1 \end{array}$$

    $$f(x) = \mathbf{plus}(\mathbf{square}(x), \mathbf{plus}(\mathbf{double}(x), \mathbf{one}))$$

Concrete Syntax

A λ-term is either

1. a variable
2. the application of one term to another

$$(M\ N)$$

3. the abstraction over a variable in another term

$$(\lambda x.M)$$

Examples

$$V \rightarrow x \mid V'$$
$$\Lambda \rightarrow V \mid (\Lambda\ \Lambda) \mid (\lambda V.\Lambda)$$

1. x
2. $(x\ y)$
3. $(\lambda z.(x\ y))$

Notations

1. $M\ N\ O := ((M\ N)\ O)$

2. $\lambda x, y, z.M := (\lambda x.(\lambda y.(\lambda z.M)))$

3. $$\begin{aligned} M^0\ N &:= N \\ M^{n+1}\ N &:= M\ (M^n\ N) \end{aligned}$$

α-equivalence (I)

$$f(x) = x^2 + 2x + 1 \qquad g(x) = (x+1)^2$$

$$f'(y) = y^2 + 2y + 1 \qquad g'(y) = (y+1)^2$$

- All compute the same function (qua graph)
- Syntactic difference between f and g is meaningful (different algorithm)
- Syntactic difference between f and f' is not

α-equivalence (II)

We would like to say:

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

but

$$x \not\equiv_\alpha y$$
$$(\lambda x, y.(y\ x)) \not\equiv_\alpha (\lambda y, y.(y\ y))$$

What is important is:

1. which variables are *bound* by which binders
2. which *free* variables are identical to which other free variables

Free and Bound Variables

An occurrence of a variable x in M

- $(\lambda x.(y\ (\lambda z.(x\ (z\ (\lambda w.(x\ w)))))))$

$$(\lambda x.\underbrace{M}_{scope})$$

Free and Bound Occurrences

An occurrence of z is *Free* in M iff

- it does not occur in the scope of any λz

An occurrence of z is *Bound* in M iff

- it occurs in the scope of some λz

Free and Bound Variables

An occurrence of a variable x in M

- $(\lambda x.(y\ (\lambda z.(x\ (z\ (\lambda w.(x\ w)))))))$

$$(\lambda x. \underbrace{M}_{scope})$$

Free and Bound Occurrences

An occurrence of z is *Free* in M iff

- it does not occur in the scope of any λz

An occurrence of z is *Bound* in M iff

- it occurs in the scope of some λz

Free and Bound Variables

M	$BV(M)$	$FV(M)$
x	$\emptyset$	$\{x\}$
$(M\ N)$	$BV(M) \cup BV(N)$	$FV(M) \cup FV(N)$
$(\lambda x.N)$	$BV(N) \cup \{x\}$	$FV(N) - \{x\}$

The variable convention

In a term M

- all bound variables are distinct from all free ones
- all binders bind different variables

renaming bound variables

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

The variable convention

In a term M

- all bound variables are distinct from all free ones
- all binders bind different variables

renaming bound variables

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

The variable convention

In a term M

- all bound variables are distinct from all free ones
- all binders bind different variables

renaming bound variables

$(\lambda u.(y\ (\lambda y.(u\ (y\ (\lambda y.(u\ y)))))))$

The variable convention

In a term M

- all bound variables are distinct from all free ones
- all binders bind different variables

renaming bound variables

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

The variable convention

In a term M

- all bound variables are distinct from all free ones
- all binders bind different variables

renaming bound variables

$(\lambda u.(y\ (\lambda v.(u\ (v\ (\lambda w.(u\ w)))))))$

An embarassment of riches

Our representation is too rich

using variables makes *too many* distinctions
we want to *equate* different representations

1. work with *equivalence classes* of terms
2. do this semantically

De Bruijn notation

$\lambda x.\lambda y.x\ y\ (\lambda z.z\ y) \rightsquigarrow \lambda.\lambda.1\ 0\ (\lambda.0\ 1)$

Substitution

$M[x := N] \approx$ Substitute N for x in M

$$
\begin{aligned}
x[x := N] &= N \\
y[x := N] &= y \\
(P\ Q)[x := N] &= (P[x := N]\ Q[x := N]) \\
(\lambda y.P)[x := N] &= (\lambda y.P[x := N])
\end{aligned}
$$

by our variable convention, all bound variables in M, x, and N are distinct, and different from all free variables

Substitution (II)

over concrete terms

$$\begin{aligned}
x[x := N] &= N \\
y[x := N] &= y \\
(P\ Q)[x := N] &= (P[x := N]\ Q[x := N]) \\
(\lambda x.P)[x := N] &= (\lambda x.P) \\
(\lambda y.P)[x := N] &= (\lambda y.P[x := N])
\end{aligned}$$

Substitution (II)

over concrete terms

$$
\begin{aligned}
x[x := N] &= N \\
y[x := N] &= y \\
(P\ Q)[x := N] &= (P[x := N]\ Q[x := N]) \\
(\lambda x.P)[x := N] &= (\lambda x.P) \\
(\lambda y.P)[x := N] &= (\lambda y.P[x := N])
\end{aligned}
$$

Variable Capture

$$(\lambda y.\, \underbrace{P}_{scope})[x := N] = (\lambda y.\, \underbrace{P[x := N]}_{scope})$$

Substitution (III)

over concrete terms

$$x[x := N] = N$$

$$y[x := N] = y$$

$$(P\ Q)[x := N] = (P[x := N]\ Q[x := N])$$

$$(\lambda x.P)[x := N] = (\lambda x.P)$$

$$(\lambda y.P)[x := N] = (\lambda z.P[y := z][x := N])$$

z must not be free in P or in N!

α-equivalence

$$(\lambda x.M) \equiv_\alpha (\lambda y.M[x := y]) \qquad\qquad (\text{if } y \notin FV(M))$$

If $M \equiv_\alpha N$, we will treat M and N as the same term

The variable convention guides our choice of which α-equivalent term to use

Classes of λ-terms

Combinators

no free variables

λI

each binder binds at least one variable

no deleting

affine (BCK)

each binder binds at most one variable

no copying

linear (BCI)

each binder binds exactly one variable

Interpreting λ-terms

Operational

- 'External'
- Meaning emerges from use
- Today

Denotational

- 'Internal'
- Use emerges from meaning
- Thursday

What makes sense?

$$\frac{(M\ N)}{\begin{array}{c}(x\ N)\\[1em]((P\ Q)\ N)\\[1em]((\lambda x.M)\ N)\end{array}}$$

$$\frac{(\lambda x.(M\ N))}{\begin{array}{c}(\lambda x.(M\ x))\\[1em](\lambda x.(M\ (P\ Q)))\\[1em](\lambda x.(M\ (\lambda y.N)))\end{array}}$$

What makes sense?

$$\frac{(M\ N)}{}$$

$$(x\ N)$$

$$((P\ Q)\ N)$$

$$((\lambda x.M)\ N)$$

$$\frac{(\lambda x.(M\ N))}{}$$

$$(\lambda x.(M\ x))$$

$$(\lambda x.(M\ (P\ Q)))$$

$$(\lambda x.(M\ (\lambda y.N)))$$

Applying functions to arguments

β-**red**ucible **ex**pression

$$\underbrace{(\underbrace{(\lambda x.M)}_{\textit{abstraction}} \;\; N)}_{\textit{application}}$$

$$((\lambda x.M)\ N) \rightsquigarrow M[x := N] \qquad (\beta)$$

Abstracting over application

η-**red**ucible **ex**pression

$$\underbrace{(\lambda x.\ \underbrace{(M\ x)}_{application}\ )}_{abstraction}$$

- provided $x \notin FV(M)$

$$(\lambda x.(M\ x)) \rightsquigarrow M \qquad (\eta)$$

Compatible closure

- The rules (β,η) tell us how to apply a function we've created to an argument.
- We also need to know *where* they may apply

$$\frac{M \rightsquigarrow N}{M \Rightarrow N}$$

$$\frac{M \Rightarrow M'}{(M\ N) \Rightarrow (M'\ N)} \qquad \frac{N \Rightarrow N'}{(M\ N) \Rightarrow (M\ N')}$$

$$\frac{M \Rightarrow M'}{(\lambda x.M) \Rightarrow (\lambda x.M')}$$

Reduction

β-reduction

is the compatible closure of the rule β

$$M \Rightarrow_{\beta} N$$

$\beta\eta$-reduction

is the compatible closure of the rules β and η

$$M \Rightarrow_{\beta\eta} N$$

Expansion

is the opposite of reduction:
if $M \Rightarrow N$, then M is an expansion of N
we write $N \Leftarrow M$

Multiple steps

$$\frac{}{M \Rightarrow^0 M}$$

$$\frac{M \Rightarrow^n N \qquad N \Rightarrow O}{M \Rightarrow^{n+1} O}$$

$$\frac{M \Rightarrow^n N}{M \Rightarrow^* N}$$

Normal Forms

Algorithm

A λ-term is a description of a sequence of instructions
(wait for an argument)
(when you get it, put it here)

Computation

reduction is carrying out the instructions of the algorithm

Value

the result of a computation is what you are left with once there is nothing more to do

M is a normal form iff it cannot be further reduced

Example

Computable functions

we can define λ-terms representing numbers and functions so that, for any computable $f \in \mathbb{N}^k \to \mathbb{N}$, and all $n_1, \ldots, n_k \in \mathbb{N}$,

$$((\ulcorner f \urcorner \ulcorner n_1 \urcorner) \ \ldots \ \ulcorner n_k \urcorner) \Rightarrow_\beta^* \ulcorner f(n_1, \cdots, n_k) \urcorner$$

Church encodings

- $\ulcorner n \urcorner := \lambda s, z.(s^n \ z)$
- $\ulcorner \text{plus} \urcorner := \lambda m, n, s, z.m \ s \ (n \ s \ z)$

$$\ulcorner \text{plus} \urcorner \ulcorner 3 \urcorner \ulcorner 2 \urcorner \Rightarrow_\beta^* \ulcorner 5 \urcorner$$

Example

Tests

We can define λ-terms representing boolean values, and a conditional statement, so that for all M, N:

$$\begin{aligned}\textbf{if-then-else true } M\ N &\Rightarrow_\beta^* M \\ \textbf{if-then-else false } M\ N &\Rightarrow_\beta^* N\end{aligned}$$

Encodings

- **true** $:= \lambda x, y.x$
- **false** $:= \lambda x, y.y$
- **if-then-else** $:= \lambda b, x, y.b\ x\ y$
- **not** $:= \lambda b.b$ **false true**
- **and** $:= \lambda b, c.b\ c$ **false**
- **is-zero?** $:= \lambda n.n\ (\lambda z.\textbf{false})$ **true**

Example

Pairs

We can define λ-terms representing pairs, and projections, so that for all M, N:

$$\begin{aligned} \textbf{fst}\ (\textbf{pair}\ M\ \ N)\ &\Rightarrow^*_\beta\ M \\ \textbf{snd}\ (\textbf{pair}\ M\ \ N)\ &\Rightarrow^*_\beta\ N \end{aligned}$$

Encodings

- **pair** $:= \lambda x, y, f.f\ \ x\ \ y$
- **fst** $:= \lambda p.p\ (\lambda u, v.u)$
- **snd** $:= \lambda p.p\ (\lambda u, v.v)$

$$\textbf{pair}\ M\ \ N \Rightarrow^*_\beta \lambda f.f\ \ M\ \ N$$

Decrement

$$\mathbf{shift}\ (\mathbf{pair}\ \ulcorner m \urcorner\ \ulcorner n \urcorner) \Rightarrow_\beta^* \mathbf{pair}\ \ulcorner n \urcorner\ \ulcorner n+1 \urcorner$$

$$\begin{aligned} \mathbf{dec}\ \ulcorner 0 \urcorner &\Rightarrow_\beta^* \ulcorner 0 \urcorner \\ \mathbf{dec}\ \ulcorner m+1 \urcorner &\Rightarrow_\beta^* \ulcorner m \urcorner \end{aligned}$$

Encodings

- $\mathbf{shift} := \lambda p.p\ (\lambda x, y, f.f\ \ y\ (\mathbf{suc}\ y))$
- $\mathbf{dec} := \lambda n.\mathbf{fst}\ (n\ \mathbf{shift}\ (\mathbf{pair}\ \ulcorner 0 \urcorner\ \ulcorner 0 \urcorner))$

The shape of values

(head) normal form

$$\lambda x_1, \ldots, x_n.(y\ M_1 \cdots\ M_k) \qquad \text{(where } M_1, \ldots, M_k \text{ are hnfs)}$$

unsolvable terms

let $\omega := \lambda x.x\ x$
$\Omega := \omega\ \omega$ has no normal form.

$$\begin{aligned} \Omega &= \omega\ \omega \\ &= (\lambda x.x\ x)\ \omega \\ &\Rightarrow_\beta \omega\ \omega \\ &= \Omega \end{aligned}$$

Confluence

A relation R is **confluent** iff

if aRb and aRc then
there is some d such that
bRd and cRd

Theorem (Church-Rosser Theorem):

$\Rightarrow_{\beta}^{*}$ and $\Rightarrow_{\beta\eta}^{*}$ are confluent

Corollary:

If a term has a normal form it is unique

Finding normal forms

Reduction strategies

leftmost/call-by-name/outside-in:
: if M has multiple redices, reduce the one whose λ occurs furthest to the left

applicative/call-by-value/inside-out:
: reduce $((\lambda x.M)\ N)$ only if N is a normal form

Theorem (Standardization):

if M has a normal form, it can be reached by a *leftmost* reduction strategy

Finding normal forms (II)

which terms have normal forms?

- all non-duplicating terms (BCI,BCK)
- some duplicating terms

Types

A type is a syntactic object which describes the behaviour of a term

we will have:
All well-typed terms have normal forms

Simple types

A type is either

1. a type variable
2. an implication between two types

$$(\alpha \to \beta)$$

Intuition

a is a set, $(a \to b)$ a set of functions between a and b

Notations

1. $\alpha \to \beta \to \gamma := (\alpha \to (\beta \to \gamma))$

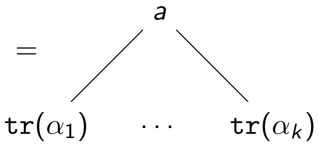
2. $$\begin{aligned} \alpha^0 \to \beta &:= \beta \\ \alpha^{n+1} \to \beta &:= \alpha \to \alpha^n \to \beta \end{aligned}$$

Types as trees

All types have the following form:

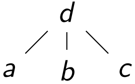
$\alpha_1 \to \cdots \to \alpha_k \to a$

$$\mathtt{tr}(\alpha_1 \to \cdots \to \alpha_k \to a) \quad =$$



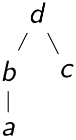
Examples

$(a \rightarrow (b \rightarrow (c \rightarrow d)))$

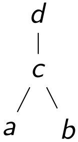


$(((a \rightarrow b) \rightarrow c) \rightarrow d)$

$((a \rightarrow b) \rightarrow (c \rightarrow d))$



$((a \rightarrow (b \rightarrow c)) \rightarrow d)$



Complexity measures

Order

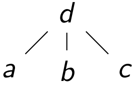
$$\begin{aligned} \mathtt{ord}(a) &= 1 \\ \mathtt{ord}(\alpha \to \beta) &= \mathtt{max}(\{\mathtt{ord}(\alpha) + 1, \mathtt{ord}(\beta)\}) \end{aligned}$$

The order of a type

is length of the longest path from the root to a leaf

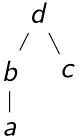
Examples

$(a \rightarrow (b \rightarrow (c \rightarrow d)))$

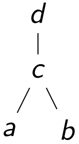


$(((a \rightarrow b) \rightarrow c) \rightarrow d)$

$((a \rightarrow b) \rightarrow (c \rightarrow d))$



$((a \rightarrow (b \rightarrow c)) \rightarrow d)$



Typing

$$\underbrace{M}_{\textit{subject}} : \underbrace{\alpha}_{\textit{predicate}}$$

Type environments

- finite set of type declarations $(x : \alpha)$
- *consistent* iff no variable is declared with two types

Notation

- $x : \alpha := \{x : \alpha\}$
- $\Gamma, \Delta := \Gamma \cup \Delta$, just in case $\Gamma \cup \Delta$ is consistent

Typing judgments

$$\Gamma \vdash M : \alpha \qquad\qquad (M \text{ has type } \alpha \text{ in environment } \Gamma)$$

Typing rules

$$\frac{}{\Gamma, x : \alpha \vdash x : \alpha}\ \mathsf{Ax}$$

$$\frac{\Gamma, x : \alpha \vdash M : \beta}{\Gamma \vdash \lambda x.M : \alpha \to \beta}\ {\to}\mathsf{I}$$

$$\frac{\Gamma \vdash M : \alpha \to \beta \qquad \Gamma \vdash N : \alpha}{\Gamma \vdash M\ N : \beta}\ {\to}\mathsf{E}$$

Minimal logic

$$\dfrac{}{\Gamma, \alpha \vdash \alpha}\ \mathsf{Ax}$$

$$\dfrac{\Gamma, \alpha \vdash \beta}{\Gamma \vdash \alpha \to \beta}\ {\to}\mathsf{I}$$

$$\dfrac{\Gamma \vdash \alpha \to \beta \qquad \Gamma \vdash \alpha}{\Gamma \vdash \beta}\ {\to}\mathsf{E}$$

Normalization

Theorem (Weak normalization):

if $\Gamma \vdash M : \alpha$, then M has a normal form

Theorem (Strong normalization):

if $\Gamma \vdash M : \alpha$, then there is no infinite reduction sequence starting at M

Examples

$\mathbb{I} := \lambda x.x$

$$\vdash \lambda x.x : \alpha$$

Examples

$\mathbb{I} := \lambda x.x$

$$\vdash \lambda x.x : \alpha$$

$$\frac{\Gamma, x : \alpha \vdash M : \beta}{\Gamma \vdash \lambda x.M : \alpha \to \beta} \to\mathsf{I}$$

Examples

$\mathbb{I} := \lambda x.x$

$$\frac{x : \beta \vdash x : \gamma}{\vdash \lambda x.x : \beta \to \gamma} \to\mathsf{I}$$

Examples

$\mathbb{I} := \lambda x.x$

$$\dfrac{x : \beta \vdash x : \gamma}{\vdash \lambda x.x : \beta \to \gamma} \to\mathsf{I}$$

$$\dfrac{}{\Gamma, x : \alpha \vdash x : \alpha}\ \mathsf{Ax}$$

Examples

$\mathbb{I} := \lambda x.x$

$$\dfrac{\dfrac{}{x : \beta \vdash x : \beta}\ \mathsf{Ax}}{\vdash \lambda x.x : \beta \to \beta}\ {\to}\mathsf{I}$$

Examples

$\mathbb{I} := \lambda x.x$

$$\dfrac{\dfrac{}{\beta \vdash \beta}\ \mathsf{Ax}}{\vdash \beta \to \beta}\ {\to}\mathsf{I}$$

Examples

$\mathbb{K} := \lambda x, y.x$

$$\vdash \lambda x, y.x : \alpha$$

Examples

$\mathbb{K} := \lambda x, y.x$

$$\vdash \lambda x, y.x : \alpha$$

$$\frac{\Gamma, x : \alpha \vdash M : \beta}{\Gamma \vdash \lambda x.M : \alpha \to \beta} \to\mathsf{I}$$

Examples

$\mathbb{K} := \lambda x, y.x$

$$\frac{x : \beta \vdash \lambda y.x : \gamma}{\vdash \lambda x, y.x : \beta \to \gamma} \to\mathsf{I}$$

$$\frac{\Gamma, x : \alpha \vdash M : \beta}{\Gamma \vdash \lambda x.M : \alpha \to \beta} \to\mathsf{I}$$

Examples

$\mathbb{K} := \lambda x, y.x$

$$\dfrac{\dfrac{x : \beta, y : \delta \vdash x : \eta}{x : \beta \vdash \lambda y.x : \delta \to \eta}\ {\to}\mathsf{I}}{\vdash \lambda x, y.x : \beta \to \delta \to \eta}\ {\to}\mathsf{I}$$

Examples

$\mathbb{K} := \lambda x, y.x$

$$\dfrac{\dfrac{x : \beta, y : \delta \vdash x : \eta}{x : \beta \vdash \lambda y.x : \delta \to \eta} \to\mathsf{I}}{\vdash \lambda x, y.x : \beta \to \delta \to \eta} \to\mathsf{I}$$

$$\dfrac{}{\Gamma, x : \alpha \vdash x : \alpha} \mathsf{Ax}$$

Examples

$\mathbb{K} := \lambda x, y.x$

$$
\dfrac{\dfrac{\dfrac{}{x : \beta, y : \delta \vdash x : \beta}\,\text{Ax}}{x : \beta \vdash \lambda y.x : \delta \to \beta}\,\to\text{I}}{\vdash \lambda x, y.x : \beta \to \delta \to \beta}\,\to\text{I}
$$

Examples

$\mathbb{K} := \lambda x, y.x$

$$\dfrac{\dfrac{\dfrac{}{\beta, \delta \vdash \beta}\ \mathsf{Ax}}{\beta \vdash \delta \rightarrow \beta}\ {\rightarrow}\mathsf{I}}{\vdash \beta \rightarrow \delta \rightarrow \beta}\ {\rightarrow}\mathsf{I}$$

Examples

$\mathbb{W} := \lambda x, y.x\ y\ y$

$$\vdash \lambda x, y.x\ y\ y : \alpha$$

Examples

$$\mathbb{W} := \lambda x, y.x\ y\ y$$

$$\vdash \lambda x, y.x\ y\ y : \alpha$$

$$\frac{\Gamma, x : \alpha \vdash M : \beta}{\Gamma \vdash \lambda x.M : \alpha \to \beta} \to\mathsf{I}$$

Examples

$\mathbb{W} := \lambda x, y.x\ y\ y$

$$\frac{x : \beta \vdash \lambda y.x\ y\ y : \gamma}{\vdash \lambda x, y.x\ y\ y : \beta \to \gamma} \to\mathsf{I}$$

$$\frac{\Gamma, x : \alpha \vdash M : \beta}{\Gamma \vdash \lambda x.M : \alpha \to \beta} \to\mathsf{I}$$

Examples

$\mathbb{W} := \lambda x, y.x\ y\ y$

$$\cfrac{\cfrac{x : \alpha, y : \beta \vdash x\ y\ y : \gamma}{x : \alpha \vdash \lambda y.x\ y\ y : \beta \to \gamma}\ {\to}\mathrm{I}}{\vdash \lambda x, y.x\ y\ y : \alpha \to \beta \to \gamma}\ {\to}\mathrm{I}$$

Examples

$\mathbb{W} := \lambda x, y.x\ y\ y$

$$\dfrac{\dfrac{x : \alpha, y : \beta \vdash x\ y\ y : \gamma}{x : \alpha \vdash \lambda y.x\ y\ y : \beta \to \gamma}\,{\to}\mathrm{I}}{\vdash \lambda x, y.x\ y\ y : \alpha \to \beta \to \gamma}\,{\to}\mathrm{I}$$

$$\dfrac{\Gamma \vdash M : \alpha \to \beta \qquad \Gamma \vdash N : \alpha}{\Gamma \vdash M\ N : \beta}\,{\to}\mathrm{E}$$

Examples

$\mathbb{W} := \lambda x, y.x\ y\ y$

$$\dfrac{\dfrac{\dfrac{x : \alpha, y : \beta \vdash x\ y : \eta \to \gamma \qquad y : \beta \vdash y : \eta}{x : \alpha, y : \beta \vdash x\ y\ y : \gamma}\ \to\mathsf{E}}{x : \alpha \vdash \lambda y.x\ y\ y : \beta \to \gamma}\ \to\mathsf{I}}{\vdash \lambda x, y.x\ y\ y : \alpha \to \beta \to \gamma}\ \to\mathsf{I}$$

$$\dfrac{\Gamma \vdash M : \alpha \to \beta \qquad \Gamma \vdash N : \alpha}{\Gamma \vdash M\ N : \beta}\ \to\mathsf{E}$$

Examples

$\mathbb{W} := \lambda x, y.x\ y\ y$

$$\dfrac{\dfrac{\dfrac{\dfrac{x : \alpha \vdash x : \delta \to \eta \to \gamma \qquad y : \beta \vdash y : \delta}{x : \alpha, y : \beta \vdash x\ y : \eta \to \gamma}\ {\to}\mathsf{E} \qquad y : \beta \vdash y : \eta}{x : \alpha, y : \beta \vdash x\ y\ y : \gamma}\ {\to}\mathsf{E}}{x : \alpha \vdash \lambda y.x\ y\ y : \beta \to \gamma}\ {\to}\mathsf{I}}{\vdash \lambda x, y.x\ y\ y : \alpha \to \beta \to \gamma}\ {\to}\mathsf{I}$$

Examples

$\mathbb{W} := \lambda x, y.x\ y\ y$

$$
\cfrac{\cfrac{\cfrac{\cfrac{x : \alpha \vdash x : \delta \to \eta \to \gamma \qquad y : \beta \vdash y : \delta}{x : \alpha, y : \beta \vdash x\ y : \eta \to \gamma}\ {\to}\mathsf{E} \qquad y : \beta \vdash y : \eta}{x : \alpha, y : \beta \vdash x\ y\ y : \gamma}\ {\to}\mathsf{E}}{x : \alpha \vdash \lambda y.x\ y\ y : \beta \to \gamma}\ {\to}\mathsf{I}}{\vdash \lambda x, y.x\ y\ y : \alpha \to \beta \to \gamma}\ {\to}\mathsf{I}
$$

$$
\frac{}{\Gamma, x : \alpha \vdash x : \alpha}\ \mathsf{Ax}
$$

Examples

$\mathbb{W} := \lambda x, y.x\ y\ y$

$$\dfrac{\dfrac{\dfrac{\dfrac{x : \alpha \vdash x : \delta \to \beta \to \gamma \qquad y : \beta \vdash y : \delta}{x : \alpha, y : \beta \vdash x\ y : \beta \to \gamma}\ {\to}\mathsf{E} \qquad \dfrac{}{y : \beta \vdash y : \beta}\ \mathsf{Ax}}{x : \alpha, y : \beta \vdash x\ y\ y : \gamma}\ {\to}\mathsf{E}}{x : \alpha \vdash \lambda y.x\ y\ y : \beta \to \gamma}\ {\to}\mathsf{I}}{\vdash \lambda x, y.x\ y\ y : \alpha \to \beta \to \gamma}\ {\to}\mathsf{I}$$

$$\dfrac{}{\Gamma, x : \alpha \vdash x : \alpha}\ \mathsf{Ax}$$

Examples

$\mathbb{W} := \lambda x, y.x\ y\ y$

$$\dfrac{\dfrac{x : \alpha \vdash x : \beta \to \beta \to \gamma \qquad \dfrac{}{y : \beta \vdash y : \beta}\,\text{Ax}}{x : \alpha, y : \beta \vdash x\ y : \beta \to \gamma}\,{\to}\text{E} \qquad \dfrac{}{y : \beta \vdash y : \beta}\,\text{Ax}}{\dfrac{x : \alpha, y : \beta \vdash x\ y\ y : \gamma}{\dfrac{x : \alpha \vdash \lambda y.x\ y\ y : \beta \to \gamma}{\vdash \lambda x, y.x\ y\ y : \alpha \to \beta \to \gamma}\,{\to}\text{I}}\,{\to}\text{I}}\,{\to}\text{E}$$

$$\dfrac{}{\Gamma, x : \alpha \vdash x : \alpha}\,\text{Ax}$$

Examples

$$\mathbb{W} := \lambda x, y.x\ y\ y$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x : \beta \to \beta \to \gamma \vdash x : \beta \to \beta \to \gamma}\,\text{Ax} \qquad \dfrac{}{y : \beta \vdash y : \beta}\,\text{Ax}}{x : \beta \to \beta \to \gamma, y : \beta \vdash x\ y : \beta \to \gamma}\,{\to}\text{E} \qquad \dfrac{}{y : \beta \vdash y :}}{x : \beta \to \beta \to \gamma, y : \beta \vdash x\ y\ y : \gamma}}{x : \beta \to \beta \to \gamma \vdash \lambda y.x\ y\ y : \beta \to \gamma}\,{\to}\text{I}}{\vdash \lambda x, y.x\ y\ y : (\beta \to \beta \to \gamma) \to \beta \to \gamma}\,{\to}\text{I}$$

Examples

$\mathbb{W} := \lambda x, y.x\ y\ y$

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{}{\beta \to \beta \to \gamma \vdash \beta \to \beta \to \gamma}\,\mathsf{Ax}
      \qquad
      \dfrac{}{\beta \vdash \beta}\,\mathsf{Ax}
    }{\beta \to \beta \to \gamma, \beta \vdash \beta \to \gamma}\,{\to}\mathsf{E}
    \qquad
    \dfrac{}{\beta \vdash \beta}\,\mathsf{Ax}
  }{\dfrac{\beta \to \beta \to \gamma, \beta \vdash \gamma}{\beta \to \beta \to \gamma \vdash \beta \to \gamma}\,{\to}\mathsf{I}}\,{\to}\mathsf{E}
}{\vdash (\beta \to \beta \to \gamma) \to \beta \to \gamma}\,{\to}\mathsf{I}
$$

Examples

$\omega := \lambda x.x\ x$

$$\vdash \lambda x.x\ x : \alpha$$

Examples

$\omega := \lambda x.x\ x$

$$\frac{x : \beta \vdash x\ x : \gamma}{\vdash \lambda x.x\ x : \beta \to \gamma} \to\mathsf{I}$$

Examples

$\omega := \lambda x.x\ x$

$$\dfrac{\dfrac{x : \beta \vdash x : \alpha \to \gamma \qquad x : \beta \vdash x : \alpha}{x : \beta \vdash x\ x : \gamma}\ {\to}\mathsf{E}}{\vdash \lambda x.x\ x : \beta \to \gamma}\ {\to}\mathsf{I}$$

Examples

$\omega := \lambda x.x\ x$

$$\dfrac{\dfrac{x : \beta \vdash x : \beta \to \gamma \qquad \dfrac{}{x : \beta \vdash x : \beta}\ \mathsf{Ax}}{x : \beta \vdash x\ x : \gamma}\ {\to}\mathsf{E}}{\vdash \lambda x.x\ x : \beta \to \gamma}\ {\to}\mathsf{I}$$

Examples

$\omega := \lambda x.x\ x$

$$\dfrac{x : \beta \vdash x : \beta \to \gamma \qquad \dfrac{}{x : \beta \vdash x : \beta}\,\mathsf{Ax}}{\dfrac{x : \beta \vdash x\ x : \gamma}{\vdash \lambda x.x\ x : \beta \to \gamma}\,{\to}\mathsf{I}}\,{\to}\mathsf{E}$$

Church Typing

Idea:

A typed λ-term encodes the shape of its typing proof.
We can make it encode the entire proof!

$$\dfrac{\dfrac{}{x : \beta \vdash x : \beta}\,\mathsf{Ax}}{\vdash \lambda x.x : \beta \to \beta}\,{\to}\mathsf{I}$$

Church Typing

Idea:
A typed λ-term encodes the shape of its typing proof.
We can make it encode the entire proof!

$$\dfrac{\dfrac{}{x^{\beta}}\,\mathsf{Ax}}{\vdash \lambda x.x : \beta \to \beta}\,\to\mathsf{I}$$

Church Typing

Idea:

A typed λ-term encodes the shape of its typing proof.

We can make it encode the entire proof!

$$\dfrac{\dfrac{}{x^\beta}\,\mathsf{Ax}}{(\lambda x^\beta . x^\beta)^{\beta \to \beta}}\,{\to}\mathsf{I}$$

Church Typing

Idea:
A typed λ-term encodes the shape of its typing proof.
We can make it encode the entire proof!

$$\cfrac{\cfrac{\cfrac{}{x : \beta, y : \delta \vdash x : \beta}\ \mathsf{Ax}}{x : \beta \vdash \lambda y.x : \delta \to \beta}\ {\to}\mathsf{I}}{\vdash \lambda x, y.x : \beta \to \delta \to \beta}\ {\to}\mathsf{I}$$

Church Typing

Idea:
A typed λ-term encodes the shape of its typing proof.
We can make it encode the entire proof!

$$\dfrac{\dfrac{\dfrac{}{x^{\beta}}\,\text{Ax}}{x : \beta \vdash \lambda y.x : \delta \to \beta}\,{\to}\text{I}}{\vdash \lambda x, y.x : \beta \to \delta \to \beta}\,{\to}\text{I}$$

Church Typing

Idea:

A typed λ-term encodes the shape of its typing proof.
We can make it encode the entire proof!

$$\dfrac{\dfrac{\dfrac{}{x^{\beta}}\ \mathsf{Ax}}{(\lambda y^{\delta}.x^{\beta})^{\delta\to\beta}}\ \to\mathsf{I}}{\vdash \lambda x, y.x : \beta \to \delta \to \beta}\ \to\mathsf{I}$$

Church Typing

Idea:
A typed λ-term encodes the shape of its typing proof.
We can make it encode the entire proof!

$$\dfrac{\dfrac{\dfrac{}{x^\beta}\,\mathsf{Ax}}{(\lambda y^\delta.x^\beta)^{\delta\to\beta}}\,\to\mathsf{I}}{(\lambda x^\beta.(\lambda y^\delta.x^\beta)^{\delta\to\beta})^{\beta\to\delta\to\beta}}\,\to\mathsf{I}$$

Church Types

$$\frac{}{x^\alpha}\ \mathsf{Ax}$$

$$\frac{M^\beta}{(\lambda x^\alpha . M^\beta)^{\alpha\to\beta}}\ {\to}\mathsf{I}$$

$$\frac{M^{\alpha\to\beta} \qquad N^\alpha}{(M^{\alpha\to\beta}\ N^\alpha)^\beta}\ {\to}\mathsf{E}$$

Principal Types

If a term has a type, how many does it have?

1. ∞
2. one

One type to rule them all...

ignoring type variables, to each λ-term corresponds at most one typing proof.

the most general type we can assign a λ-term is its principal type.

Decision Problems

Typability

given Γ, M, is there some α such that $\Gamma \vdash M : \alpha$?

Inhabitation

given Γ, α, is there some M such that $\Gamma \vdash M : \alpha$?

Given Γ, α

- We construct M_α such that $\Gamma \vdash M_\alpha : \alpha$ (or return that there is no such term).
- M_α will be in hnf, and so will be of the form
  $\lambda \underbrace{x_1, \ldots, x_k}_{\textit{prefix}} . \underbrace{y}_{\textit{head}} \underbrace{M_1 \ \ldots \ M_j}_{\textit{args}}$

  prefix $\alpha = \alpha_1 \to \ldots \to \alpha_n \to a$; we take the prefix to be $x_1, \ldots, x_n$

  head **choose** some $y : \beta$ in $\Gamma \cup \{x_1 : \alpha_1, \ldots, x_n : \alpha_n\}$, such that $\beta = \beta_1 \to \ldots \to \beta_i \to a$

  args construct $M_1, \ldots M_i$ such that $\Gamma, x_1 : \alpha_1, \ldots, x_n : \alpha_n \vdash M_h : \beta_h$, for $1 \leq h \leq i$

Example

Given $\Gamma = \emptyset$, $\alpha = (a \to a \to b) \to a \to b$

prefix x_1, x_2

head **choose** some $y : \beta$ in $\{x_1 : a \to a \to b, x_2 : a\}$ with β ending in b.
Only choice: $x_1 : a \to a \to b$

arg construct M_1, M_2 such that $\Delta \vdash M_1 : a$ and $\Delta \vdash M_2 : a$, where $\Delta = \{x_1 : a \to a \to b, x_2 : b\}$:

> prefix (none)
>
> head **choose** some $z : \eta$ in $\{x_1 : a \to a \to b, x_2 : a\}$ with β ending in a.
> Only choice: $x_2 : a$
>
> arg (none)

so $M_1 = M_2 = x_2$

so $M = \lambda x_1, x_2 . x_1\ x_2\ x_2$

η-long normal forms

The terms we obtain via the previous procedure have a special property:
their principal types are exactly the type we wanted

Syntactic characterization:

every variable in M occurs with the maximum number of arguments permitted by its type

Proof characterization:

every judgment $\Gamma \vdash M : \alpha \to \beta$ is either

- the conclusion of $\to$I, or
- the major premise of $\to$E

Example

$\mathbb{I} := \lambda x.x$

$a \to a$:

$\lambda x.x$ is η-long

$(a \to b) \to a \to b$:

$\lambda x, y.x\ y$ is η-long

$((a \to b) \to c) \to (a \to b) \to c$:

$\lambda x, y.x\ (\lambda z.y\ z)$ is η-long

all are in β-normal form (they cannot be further reduced).

$$\lambda x, y.x\ (\lambda z.y\ z) \Rightarrow_\eta \lambda x, y.x\ y \Rightarrow_\eta \lambda x.x$$

Example

$\mathbb{I} := \lambda x.x$

$a \to a$:

$\lambda x.x$ is η-long

$(a \to b) \to a \to b$:

$\lambda x, y.x\ y$ is η-long

$((a \to b) \to c) \to (a \to b) \to c$:

$\lambda x, y.x\ (\lambda z.y\ z)$ is η-long

all are in β-normal form (they cannot be further reduced).

$$\lambda x, y.x\ (\lambda z.y\ z) \Rightarrow_\eta \lambda x, y.x\ y \Rightarrow_\eta \lambda x.x$$

Example

$\mathbb{I} := \lambda x.x$

$a \to a$:

$\lambda x.x$ is η-long

$(a \to b) \to a \to b$:

$\lambda x, y.x\ y$ is η-long

$((a \to b) \to c) \to (a \to b) \to c$:

$\lambda x, y.x\ (\lambda z.y\ z)$ is η-long

all are in β-normal form (they cannot be further reduced).

$$\lambda x, y.x\ (\lambda z.y\ z) \Rightarrow_\eta \lambda x, y.x\ y \Rightarrow_\eta \lambda x.x$$

Substitution and Typing

Theorem (Substitution):

if $\Gamma, x : \alpha \vdash M : \beta$ and $\Delta \vdash N : \alpha$, then $\Gamma, \Delta \vdash M[x := N] : \beta$.

Subjects

Theorem (Subject reduction):

if $\Gamma \vdash M : \alpha$, and $M \Rightarrow^* N$, then $\Gamma \vdash N : \alpha$

Theorem (Subject expansion):

if $\Gamma \vdash M : \alpha$, and $M \ {}^*\!\!\Leftarrow N$ via linear β-reductions, then $\Gamma \vdash N : \alpha$

Classes of typed λ-terms

λI (non-deleting)

$$\dfrac{\dfrac{\dfrac{}{x:\beta, \color{red}{y:\delta} \vdash x:\beta}\,\text{Ax}}{x:\beta \vdash \lambda y.x : \delta \to \eta}\,{\to}\text{I}}{\vdash \lambda x, y.x : \beta \to \delta \to \beta}\,{\to}\text{I}$$

Revised Axiom Rule:

$$\dfrac{}{x:\alpha \vdash x:\alpha}\,\text{Ax}$$

Classes of typed λ-terms

λI (non-deleting)

$$\dfrac{\dfrac{\dfrac{}{x : \beta, y : \delta \vdash x : \beta}\,\text{Ax}}{x : \beta \vdash \lambda y.x : \delta \to \eta}\,{\to}\text{I}}{\vdash \lambda x, y.x : \beta \to \delta \to \beta}\,{\to}\text{I}$$

Revised Axiom Rule:

$$\dfrac{}{x : \alpha \vdash x : \alpha}\,\text{Ax}$$

Classes of typed λ-terms

BCK (non-duplicating)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x : \beta \to \beta \to \gamma \vdash x : \beta \to \beta \to \gamma}\,\text{Ax} \qquad \dfrac{}{y : \beta \vdash y : \beta}\,\text{Ax}}{x : \beta \to \beta \to \gamma, y : \beta \vdash x\ y : \beta \to \gamma}\,\to\text{E} \qquad \dfrac{}{y : \beta \vdash y : \beta}\,\text{Ax}}{x : \beta \to \beta \to \gamma, y : \beta \vdash x\ y\ y : \gamma}\,\to\text{E}}{x : \beta \to \beta \to \gamma \vdash \lambda y.x\ y\ y : \beta \to \gamma}\,\to\text{I}}{\vdash \lambda x, y.x\ y\ y : (\beta \to \beta \to \gamma) \to \beta \to \gamma}\,\to\text{I}$$

Revised $\to$E Rule:

$$\dfrac{\Gamma \vdash M : \alpha \to \beta \qquad \Delta \vdash N : \alpha \qquad \Gamma \cap \Delta = \emptyset}{\Gamma, \Delta \vdash M\ N : \beta}\,\to\text{E}$$

Classes of typed λ-terms

BCK (non-duplicating)

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x : \beta \to \beta \to \gamma \vdash x : \beta \to \beta \to \gamma}\text{Ax} \qquad \dfrac{}{y : \beta \vdash y : \beta}\text{Ax}}{x : \beta \to \beta \to \gamma, {\color{red} y : \beta} \vdash x\ y : \beta \to \gamma}\to\text{E} \qquad \dfrac{}{{\color{red} y : \beta} \vdash y : \beta}\text{Ax}}{x : \beta \to \beta \to \gamma, {\color{red} y : \beta} \vdash x\ y\ y : \gamma}\to\text{E}}{x : \beta \to \beta \to \gamma \vdash \lambda y.x\ y\ y : \beta \to \gamma}\to\text{I}}{\vdash \lambda x, y.x\ y\ y : (\beta \to \beta \to \gamma) \to \beta \to \gamma}\to\text{I}
$$

Revised $\to$E Rule:

$$
\dfrac{\Gamma \vdash M : \alpha \to \beta \qquad \Delta \vdash N : \alpha \qquad {\color{red} \Gamma \cap \Delta = \emptyset}}{\Gamma, \Delta \vdash M\ N : \beta}\to\text{E}
$$

Classes of typed λ-terms

BCK (non-duplicating)

$$
\dfrac{
\dfrac{
\dfrac{}{x : \beta \to \beta \to \gamma \vdash x : \beta \to \beta \to \gamma}\,\text{Ax}
\qquad
\dfrac{}{y : \beta \vdash y : \beta}\,\text{Ax}
}{x : \beta \to \beta \to \gamma, \color{red}{y : \beta} \vdash x\ y : \beta \to \gamma}\,{\to}\text{E}
\qquad
\dfrac{}{\color{red}{y : \beta} \vdash y : \beta}\,\text{Ax}
}{
\dfrac{
\dfrac{x : \beta \to \beta \to \gamma, \color{red}{y : \beta} \vdash x\ y\ y : \gamma}{x : \beta \to \beta \to \gamma \vdash \lambda y.x\ y\ y : \beta \to \gamma}\,{\to}\text{I}
}{\vdash \lambda x, y.x\ y\ y : (\beta \to \beta \to \gamma) \to \beta \to \gamma}\,{\to}\text{I}
}\,{\to}\text{E}
$$

Revised $\to$E Rule:

$$
\dfrac{\Gamma \vdash M : \alpha \to \beta \qquad \Delta \vdash N : \alpha}{\Gamma \color{red}{;} \Delta \vdash M\ N : \beta}\,{\to}\text{E}
$$

Classes of λ-terms

Linear

$$\frac{}{\emptyset, x : \alpha \vdash x : \alpha}\ \text{Ax}$$

$$\frac{\Gamma; x : \alpha \vdash M : \beta}{\Gamma \vdash \lambda x.M : \alpha \to \beta}\ {\to}\text{I}$$

$$\frac{\Gamma \vdash M : \alpha \to \beta \qquad \Delta \vdash N : \alpha}{\Gamma; \Delta \vdash M\ N : \beta}\ {\to}\text{E}$$

Affine terms have types

If M is affine, then $M : \alpha$ for some α.

Theorem (Coherence):

If M, N are affine and $M : \alpha$, then $N : \alpha$ implies that $M \equiv_{\beta\eta} N$