

Akim Demaille akim@lrde.epita.fr

EPITA — École Pour l'Informatique et les Techniques Avancées

March 22, 2009



About these lecture notes

Many of these slides are largely inspired from Andrew D. Ker's lecture notes [2, 3]. Some slides are even straightforward copies from them.

4□ > 4₫ > 4½ > 4½ > ½ 900

A. Demaille

imply Typed λ -Calculus

7 (0

Types λ^{\rightarrow} : Type Assignments

Simply Typed λ -Calculus

- Types

Types

- Types
 - Untyped λ -calculus
 - Paradoxes
 - Church vs. Curry
- 2 λ^{\rightarrow} : Type Assignments

4 D L 4 D L 4 E L 4 E L 5 O O

Simula Tanad \ Co

Simply Typed λ -Calculus

3 / 45

Types first appeared with

- Curry (1934) for Combinatory Logic
- Church (1940)

Simply Typed λ -Calculus

Types

Types first appeared with

- Curry (1934) for Combinatory Logic
- Church (1940)

Types are syntactic objects assigned to terms:

 $\lambda^{\longrightarrow}$: Type Assignments

M:AM has type A

For instance:

 $I:A\rightarrow A$

Simply Typed λ -Calculus

Types

Untyped λ -calculus



- Untyped λ -calculus
- Paradoxes
- Church vs. Curry
- $(2) \lambda^{\rightarrow}$: Type Assignments

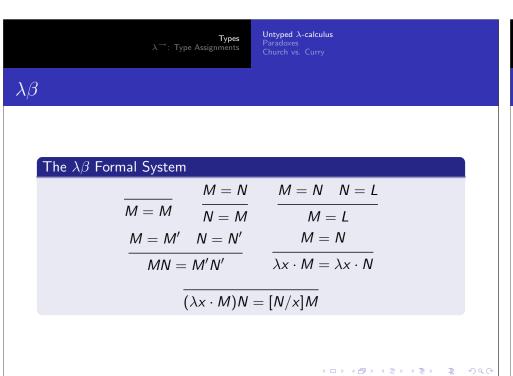
Untyped λ -calculus

 λ -terms

 Λ , set of λ -terms

$$\frac{1}{x \in \Lambda} x \in \mathcal{V} \qquad \frac{M \in \Lambda \quad N \in \Lambda}{(MN) \in \Lambda}$$

$$\frac{M \in \Lambda}{(\lambda x \cdot M) \in \Lambda} x \in \mathcal{V}$$



Simply Typed λ -Calculus

Simply Typed λ -Calculus

Untyped λ -calculus

Untyped λ -calculus

Properties of $\lambda\beta$

 β -reduction is Church-Rosser.

Any term has (at most) a unique NF.

 β -reduction is Church-Rosser.

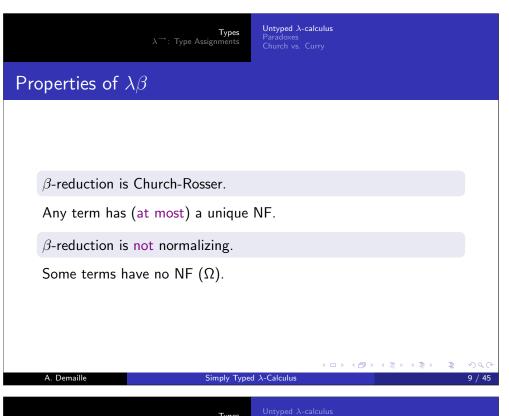
Any term has (at most) a unique NF.

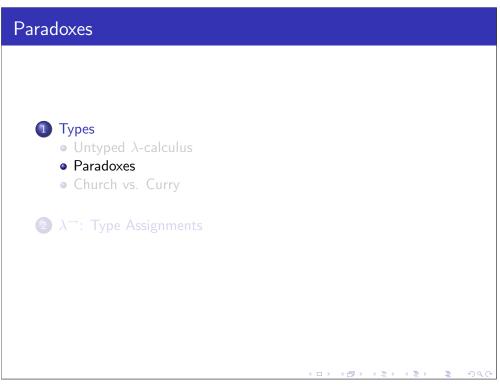
 β -reduction is **not** normalizing.

 β -reduction is Church-Rosser.

Properties of $\lambda \beta$

Untyped λ -calculus







What is the computational meaning of $\lambda x \cdot xx$?

- Stop considering anything can be applied to anything

Paradoxes

Self application

What is the computational meaning of $\lambda x \cdot xx$?

• Stop considering anything can be applied to anything

Types

• A function and its argument have different behaviors

Church vs. Curry

- Types
 - Untyped λ -calculus
 - Paradoxes
 - Church vs. Curry
- $(2) \lambda^{\rightarrow}$: Type Assignments



Simple Types

Church vs. Curry

A set of type variables

 α, β, \dots

 $\lambda^{\longrightarrow}$: Type Assignments

Simply Typed λ -Calculus

Types

Church vs. Curry

Simple Types

- A set of type variables α, β, \dots
- A symbol → for functions $\alpha \to \alpha$, $\alpha \to (\beta \to \gamma)$, $(\alpha \to \beta) \to \gamma$, ...

Types

Church vs. Curry

Simple Types

- A set of type variables α, β, \dots
- $\bullet \ \, \text{A symbol} \, \to \, \text{for functions}$ $\alpha \to \alpha$, $\alpha \to (\beta \to \gamma)$, $(\alpha \to \beta) \to \gamma$, ...
- Possibly constants for "primitive" types ι for integers, etc.

Simple Types

By convention \rightarrow is right-associative:

$$\alpha \to \beta \to \gamma = \alpha \to (\beta \to \gamma)$$

This matches the right-associativity of λ

$$\lambda x \cdot \lambda y \cdot M = \lambda x \cdot (\lambda y \cdot M)$$

Types

4□ → 4□ → 4 = → 4 = → 9 < 0</p>

Simply Typed λ -Calculus

 λ^{\rightarrow} : Type Assignments

14 / 45

Simple Types

By convention \rightarrow is right-associative:

$$\alpha \to \beta \to \gamma = \alpha \to (\beta \to \gamma)$$

This matches the right-associativity of λ :

$$\lambda x \cdot \lambda y \cdot M = \lambda x \cdot (\lambda y \cdot M)$$

A Demaille

Simply Typed λ -Calculus

_ / \

Untyped λ -calculus Paradoxes Church vs. Curry

Alonzo Style, or Haskell Way?

Church: Typed λ -calculus

$$\frac{x:\alpha \quad x:\alpha}{\lambda x^{\alpha}\cdot x:\alpha\to\alpha}$$

Curry:

 λ -calculus with Types

$$\frac{\mathbf{x} : \alpha \quad \mathbf{x} : \alpha}{\lambda \mathbf{x} \cdot \mathbf{x} : \alpha \to \alpha}$$

λ^{\rightarrow} : Type Assignments





- Types
- Type Deductions
- Subject Reduction Theorem
- Reducibility
- Typability

Types

- Types
- (2) λ^{\rightarrow} : Type Assignments
 - Types
 - Type Deductions
 - Subject Reduction Theorem
 - Reducibility
 - Typability



 λ^{\rightarrow} : Type Assignments

Type Contexts

Statement

A statement $M : \sigma$ is a pair with $M \in \Lambda, \sigma \in \mathcal{T}$. M is the subject, σ the predicate.

 λ^{\rightarrow} : Type Assignments

Type Deductions Subject Reduction Theorem

Simple Types

TV a set of type variables α, β, \dots

Simple Types

The set \mathcal{T} of types σ, τ, \ldots :

$$\frac{1}{\alpha \in \mathcal{T}} \alpha \in \mathcal{TV} \qquad \frac{\sigma \in \mathcal{T} \quad \tau \in \mathcal{T}}{(\sigma \to \tau) \in \mathcal{T}}$$

Simply Typed λ -Calculus

 λ : Type Assignments

Type Deductions

Type Contexts

Statement

A statement $M : \sigma$ is a pair with $M \in \Lambda, \sigma \in \mathcal{T}$. M is the subject, σ the predicate.

Type Context, Basis

A type context Γ is a finite set of statements over distinct variables $\{x_1:\sigma_1,\ldots\}.$

Type Contexts

Statement

A statement $M : \sigma$ is a pair with $M \in \Lambda, \sigma \in \mathcal{T}$. M is the subject, σ the predicate.

Type Context, Basis

A type context Γ is a finite set of statements over distinct variables $\{x_1 : \sigma_1, \ldots\}$.

Assignment

The variable x is assigned the type σ in Γ iff $x : \sigma \in \Gamma$.

4□ > 4回 > 4 = > 4 = > = 990

A. Demaille

Simply Typed λ -Calculus

19 / 45

Type Context Restrictions

 $\Gamma - x$ is the Γ with all assignment $x : \sigma$ removed. $\Gamma \upharpoonright M$ is $\Gamma - FV(M)$.

4 D > 4 D > 4 E > 4 E > E 9 Q G

A. Demaille

Type Contexts

Simply Typed λ -Calculus

20 /

Type Deductions





- Types
- Type Deductions
- Subject Reduction Theorem
- Reducibility
- Typability

 λ $\stackrel{\longrightarrow}{-}$: Type Assignments

Types
Type Deductions
Subject Reduction Theorer
Reducibility
Typability

A Natural Presentation

Type derivations are trees built from the following nodes.

 $M: \sigma \rightarrow \tau \quad N: \sigma$

Types
Type Deductions
Subject Reduction Theorem
Reducibility
Typability

Types λ^{\rightarrow} : Type Assignments

Types
Type Deductions
Subject Reduction Theorem
Reducibility
Typability

A Natural Presentation

Type derivations are trees built from the following nodes.

$$\frac{\textit{M}:\sigma\rightarrow\tau\quad\textit{N}:\sigma}{\textit{MN}:\tau}$$

 λ^{\rightarrow} : Type Assignments

4□ → 4□ → 4 E → 4 E → 9 Q @

(= \ = \ \) \ (\(\psi \)

A. Demaille

Simply Typed λ -Calculus

22 / 4

Types
Type Deductions
Subject Reduction Theorem
Reducibility
Typobility

A Natural Presentation

Type derivations are trees built from the following nodes.

$$\frac{M: \sigma \to \tau \quad N: \sigma}{MN: \tau} \qquad \frac{[x: \sigma]}{\vdots} \\
\frac{M: \tau}{\lambda x \cdot M: \sigma \to \tau}$$

Simply Typed λ -Calculus

A Natural Presentation

Type derivations are trees built from the following nodes.

$$\frac{M:\sigma\to\tau\quad N:\sigma}{MN:\tau}$$

 λ^{\rightarrow} : Type Assignments

Types
Type Deductions
Subject Reduction Theorer
Reducibility
Typability

Type Statement

Type Statement

A statement $M : \sigma$ is derivable from the type context Γ ,

 $\Gamma \vdash M : \sigma$

if there is a derivation of $M:\sigma$ which all non-canceled assumptions are in Γ .

Type Deductions Subject Reduction Theorem Reducibility

Prove

Type Statements

$$\vdash \lambda f x \cdot f(f x) : (\sigma \rightarrow \sigma) \rightarrow \sigma \rightarrow \sigma$$

Simply Typed λ -Calculus

 λ^{\rightarrow} : Type Assignments

Type Deductions Subject Reduction Theorem Reducibility

Type Statements

Prove

$$\vdash \lambda \mathit{fx} \cdot \mathit{f(fx)} : (\sigma \to \sigma) \to \sigma \to \sigma$$

$$\frac{[f:\sigma\to\sigma]^{(2)} \quad [x:\sigma]^{(1)}}{fx:\sigma} \\
\frac{f(fx):\sigma}{\lambda x\cdot f(fx):\sigma\to\sigma} \\
\frac{\lambda fx\cdot f(fx):\sigma\to\sigma}{\lambda fx\cdot f(fx):(\sigma\to\sigma)\to\sigma\to\sigma}$$
(2)

 $\lambda^{\longrightarrow}$: Type Assignments

Type Deductions Subject Reduction Theorem Reducibility

Type Statements

Prove

$$\vdash \lambda f x \cdot f(f x) : (\sigma \rightarrow \sigma) \rightarrow \sigma \rightarrow \sigma$$

$$\lambda f x \cdot f(f x) : (\sigma \to \sigma) \to \sigma \to \sigma$$

Simply Typed λ -Calculus

 λ^{\rightarrow} : Type Assignments

Type Deductions Subject Reduction Theorem Reducibility

Type Statements

Prove

$$\vdash \lambda xy \cdot x : \sigma \rightarrow \tau \rightarrow \sigma$$

Type Statements

Prove

$$\vdash \lambda xy \cdot x : \sigma \rightarrow \tau \rightarrow \sigma$$

$$\lambda xy \cdot x : \sigma \to \tau \to \sigma$$

4□ > 4回 > 4 = > 4 = > ■ 900

Simply Typed λ -Calculus

Type Statements

Prove

$$\vdash \lambda xy \cdot x : \sigma \rightarrow \tau \rightarrow \sigma$$

$$\frac{[x:\sigma]^{(1)}}{\lambda y \cdot x : \tau \to \sigma}$$
$$\frac{\lambda x y \cdot x : \sigma \to \tau \to \sigma}{\lambda x y \cdot x : \sigma \to \tau \to \sigma}$$
(1)

Simply Typed λ -Calculus

 λ^{\rightarrow} : Type Assignments

Type Deductions

Alternative Presentation of Type Derivations

Type Derivations

Type derivations are trees built from the following nodes.

$$\{x:\sigma\}\mapsto x:\sigma$$

$$\frac{\Gamma \mapsto M : \sigma \to \tau \quad \Delta \mapsto N : \sigma}{\Gamma \sqcup \Delta \mapsto MN : \tau} \quad \Gamma, \Delta \text{ consistent}$$

$$\frac{\Gamma \mapsto M : \tau}{\Gamma \setminus \{x : \sigma\} \mapsto \lambda x \cdot M : \sigma \to \tau} \quad \Gamma, \{x : \sigma\} \text{ consistent}$$

$$\frac{\Gamma \mapsto M : \tau}{\Gamma \land \vdash M \cdot \tau}$$

Simply Typed λ -Calculus

 $\lambda^{\longrightarrow}$: Type Assignments

Type Deductions

Alternative Presentation of Type Derivations

Type Derivations

Type derivations are trees built from the following nodes.

$$\{x:\sigma\}\mapsto x:\sigma$$

$$\frac{\Gamma \mapsto M : \sigma \to \tau \quad \Delta \mapsto N : \sigma}{\Gamma \cup \Delta \mapsto MN : \tau} \quad \Gamma, \Delta \text{ consistent}$$

$$\frac{\Gamma \mapsto M : \tau}{\Gamma \setminus \{x : \sigma\} \mapsto \lambda x \cdot M : \sigma \to \tau} \quad \Gamma, \{x : \sigma\} \text{ consistent}$$

$$\frac{\Gamma \mapsto M : \tau}{\Gamma, \Delta \vdash M : \tau}$$

Types
Type Deductions
Subject Reduction Theorem
Reducibility

Types λ^{\rightarrow} : Type Assignments

Types
Type Deductions
Subject Reduction Theorem
Reductibility
Type bility

Type Statements

Prove

$$\vdash \lambda xy \cdot x : \sigma \rightarrow \tau \rightarrow \sigma$$

Prove

Type Statements

$$\vdash \lambda xy \cdot x : \sigma \rightarrow \tau \rightarrow \sigma$$

$$\mapsto \lambda xy \cdot x : \sigma \to \tau \to \sigma$$

◆ロト ◆昼ト ◆夏ト ◆夏ト 夏 りへで

A. Demaille

Simply Typed λ -Calculus

27 / 45

. Demaille

Simply Typed λ -Calculus

 λ^{\rightarrow} : Type Assignments

27 / 4

 $\lambda^{\longrightarrow}$: Type Assignments

Types
Type Deductions
Subject Reduction Theorem
Reducibility
Typobility

Type Statements

Types
Type Deductions
Subject Reduction Theorem
Reducibility
Typebility

Type Statements

Prove

$$\vdash \lambda xy \cdot x : \sigma \rightarrow \tau \rightarrow \sigma$$

$$\frac{\{x:\sigma\}\mapsto x:\sigma}{\{x:\sigma\}\mapsto \lambda y\cdot x:\tau\to\sigma}$$
$$\mapsto \lambda xy\cdot x:\sigma\to\tau\to\sigma$$

Prove

$$\vdash \lambda xy \cdot x : \sigma \to \tau \to \sigma$$

$$\frac{[x:\sigma] \mapsto x:\sigma}{\{x:\sigma\} \mapsto \lambda y \cdot x:\tau \to \sigma} \qquad \frac{[x:\sigma]^{(1)}}{\lambda y \cdot x:\tau \to \sigma} \\
\mapsto \lambda x y \cdot x:\sigma \to \tau \to \sigma \qquad \frac{\lambda x y \cdot x:\tau \to \sigma}{\lambda x y \cdot x:\sigma \to \tau \to \sigma} \qquad (1)$$

A. Demaille

Type Deductions Subject Reduction Theorem Reducibility

Type $\omega = \lambda x \cdot xx$.

Type Statements

Simply Typed λ -Calculus

 $\lambda^{\longrightarrow}$: Type Assignments

Type Deductions Subject Reduction Theorem
Reducibility

Type Statements

Type $\omega = \lambda x \cdot xx$.

$$\frac{\{x:\sigma_1\}\mapsto xx:\sigma_2}{\Longrightarrow \lambda x\cdot xx:\sigma} \sigma = \sigma_1 \to \sigma_2$$

 λ^{\rightarrow} : Type Assignments

Type Deductions Subject Reduction Theorem Reducibility

Type Statements

Type $\omega = \lambda x \cdot xx$.

 $\mapsto \lambda x \cdot xx : \sigma$

Simply Typed λ -Calculus

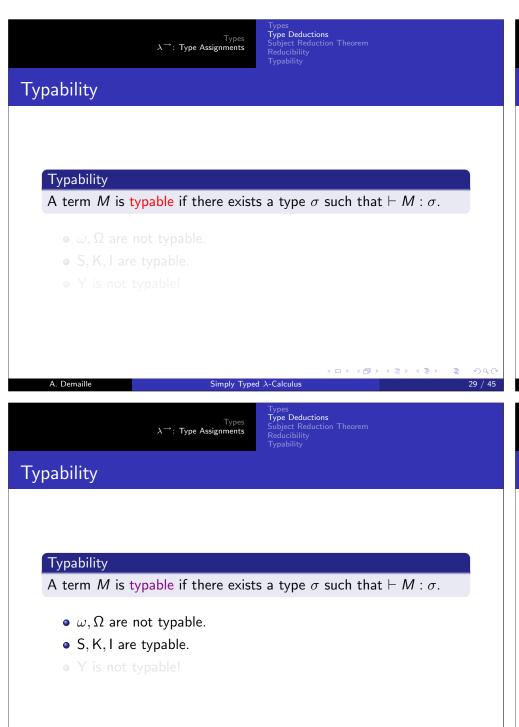
Type Deductions Subject Reduction Theorem
Reducibility

Type Statements

Type $\omega = \lambda x \cdot xx$.

 λ^{\rightarrow} : Type Assignments

 $\frac{\{x:\sigma_1\}\mapsto x:\tau\to\sigma_2}{\{x:\sigma_1\}\mapsto xx:\tau}$ $\frac{\{x:\sigma_1\}\mapsto xx:\sigma_2}{\mapsto \lambda x\cdot xx:\sigma} \sigma = \sigma_1\to\sigma_2$



Type Deductions Subject Reduction Theorem Reducibility λ^{\rightarrow} : Type Assignments **Typability Typability** A term M is typable if there exists a type σ such that $\vdash M : \sigma$. • ω, Ω are not typable. Simply Typed λ -Calculus

 $\lambda^{\longrightarrow}$: Type Assignments

Type Deductions

Typability

Typability

A term M is typable if there exists a type σ such that $\vdash M : \sigma$.

- ω, Ω are not typable.
- S, K, I are typable.
- Y is not typable!

Subject Construction Lemma I

Consider the derivation π for $M:\sigma$.

• If M = x, then $\Gamma = \{x : \sigma\}$ and

$$\pi = \overline{\{x : \sigma\} \mapsto x : \sigma}$$

$$\pi = \frac{\Gamma \upharpoonright N \mapsto N : \tau \to \sigma \quad \Gamma \upharpoonright L \mapsto L : \tau}{\Gamma \mapsto NL : \sigma}$$

Type Deductions

Type Deductions

Subject Construction Lemma I

Consider the derivation π for $M:\sigma$.

• If $M = \lambda x \cdot N$, then $\sigma = \sigma_1 \rightarrow \sigma_2$ and

• If
$$x \in FV(N)$$

$$\pi = \frac{\Gamma \cup \{x : \sigma_1\} \mapsto N : \sigma_2}{\Gamma \mapsto \lambda x \cdot N : \sigma_1 \to \sigma_2}$$

$$\pi = \frac{\Gamma \mapsto N : \sigma_2}{\Gamma \mapsto \lambda x \cdot N : \sigma_1 \to \sigma_2}$$

Subject Construction Lemma I

Consider the derivation π for $M:\sigma$.

• If M = x, then $\Gamma = \{x : \sigma\}$ and

$$\pi = \overline{\{x : \sigma\} \mapsto x : \sigma}$$

• If M = NL, then

$$\pi = \frac{\Gamma \upharpoonright N \mapsto N : \tau \to \sigma \quad \Gamma \upharpoonright L \mapsto L : \tau}{\Gamma \mapsto NL : \sigma}$$

 λ : Type Assignments

Subject Construction Lemma I

Consider the derivation π for $M:\sigma$.

• If $M = \lambda x \cdot N$, then $\sigma = \sigma_1 \rightarrow \sigma_2$ and

• If $x \in FV(N)$

$$\pi = \frac{\Gamma \cup \{x : \sigma_1\} \mapsto N : \sigma_2}{\Gamma \mapsto \lambda x \cdot N : \sigma_1 \to \sigma_2}$$

$$\tau = \frac{\Gamma \mapsto N : \sigma_2}{\Gamma \mapsto \lambda x \cdot N : \sigma_1 \to \sigma_2}$$

Subject Construction Lemma I

Consider the derivation π for $M: \sigma$.

- If $M = \lambda x \cdot N$, then $\sigma = \sigma_1 \rightarrow \sigma_2$ and
 - If $x \in FV(N)$

$$\pi = \frac{\Gamma \cup \{x : \sigma_1\} \mapsto N : \sigma_2}{\Gamma \mapsto \lambda x \cdot N : \sigma_1 \to \sigma_2}$$

• If $x \notin FV(N)$

$$\pi = \frac{\Gamma \mapsto \mathsf{N} : \sigma_2}{\Gamma \mapsto \lambda \mathsf{x} \cdot \mathsf{N} : \sigma_1 \to \sigma_2}$$

Simply Typed λ -Calculus

Derivations are not Unique

They are for β -normal forms.

Types
Type Deductions
Subject Reduction Theorem

Simply Typed λ -Calculus

Subject Reduction Theorem





- Types
- Type Deductions
- Subject Reduction Theorem
- Reducibility
- Typability

Conversions and Types

α -Invariance

If $\Gamma \mapsto M : \sigma$ and $M \equiv_{\alpha} N$ then $\Gamma \mapsto N : \sigma$.

 $\lambda^{\longrightarrow}$: Type Assignments

Types
Type Deductions
Subject Reduction Theorem
Reducibility
Typability

Conversions and Types

lpha-Invariance

If $\Gamma \mapsto M : \sigma$ and $M \equiv_{\alpha} N$ then $\Gamma \mapsto N : \sigma$.

Substitution

lf

- $\Gamma \cup \{x : \tau\} \vdash M : \sigma$,
- \bullet $\Delta \vdash N : \tau$,
- \bullet Γ , Δ consistent,
- $\bullet x \notin FV(N)$

then

 $\Gamma \cup \Delta \vdash [N/x]M : \sigma$

< ロ ト 4 回 ト 4 重 ト 4 重 ト 9 Q (~)

A. Demaille

Simply Typed λ -Calculus

Types
Type Deductions
Subject Reduction Theorem
Reducibility

 $\lambda^{\longrightarrow}$: Type Assignments

Subject Reduction Theorem

Subject Reduction Theorem

If $\Gamma \vdash M : \sigma$ and $M \rightarrow_{\beta} N$ then $\Gamma \vdash N : \sigma$.

What about the converse?

30013-0-3-0-30113131

n + 4 a + 4 = + 4 = + 5 0 0 0

 $\lambda^{\longrightarrow}$: Type Assignments

Types
Type Deductions
Subject Reduction Theorem
Reducibility
Typability

Conversions and Types

α -Invariance

If $\Gamma \mapsto M : \sigma$ and $M \equiv_{\alpha} N$ then $\Gamma \mapsto N : \sigma$.

Substitution

Ιf

- $\Gamma \cup \{x : \tau\} \vdash M : \sigma$,
- \bullet $\Delta \vdash N : \tau$,
- Γ, Δ consistent,
- $x \notin FV(N)$

then

 $\Gamma \cup \Delta \vdash [N/x]M : \sigma$

A. Demaille

Simply Typed λ -Calculus

Types
Type Deductions
Subject Reduction Theorem

Reducibility

Subject Reduction Theorem

Subject Reduction Theorem

If $\Gamma \vdash M : \sigma$ and $M \rightarrow_{\beta} N$ then $\Gamma \vdash N : \sigma$.

 $\lambda^{\longrightarrow}$: Type Assignments

What about the converse?

Subject Expansion

If $\Gamma \vdash N : \sigma$ and $M \rightarrow_{\beta} N$ then $\Gamma \vdash M : \sigma$.

Types $\lambda^{
ightharpoonup}$: Type Assignments

Types
Type Deductions
Subject Reduction Theorem
Reducibility
Typability

 $\lambda \overset{\mathsf{Types}}{\rightarrow} \colon \mathsf{Type} \; \mathsf{Assignments}$

Subject Reduction Theorem

Subject Reduction Theorem

What about the converse?

 Ω is not typable, but $KI\Omega \rightarrow I$.

Subject Expansion

If $\Gamma \vdash M : \sigma$ and $M \rightarrow_{\beta} N$ then $\Gamma \vdash N : \sigma$.

If $\Gamma \vdash N : \sigma$ and $M \rightarrow_{\beta} N$ then $\Gamma \vdash M : \sigma$.

Types
Type Deductions
Subject Reduction Theorem
Reducibility
Typability

Subject Reduction Theorem

Subject Reduction Theorem

If $\Gamma \vdash M : \sigma$ and $M \rightarrow_{\beta} N$ then $\Gamma \vdash N : \sigma$.

What about the converse?

Subject Expansion

If $\Gamma \vdash N : \sigma$ and $M \rightarrow_{\beta} N$ then $\Gamma \vdash M : \sigma$.

4 □ ▶ 4 □ ▶ 4 □ ▶ 4 □ ▶ 9 ♀ ○

Simply Typed λ -Calculus

/ 45

 $\lambda^{\longrightarrow}$: Type Assignments

Simply Typed λ -Calculus

35 /

Simply Typed X-Calculus

35 / 45

Types
Type Deductions
Subject Reduction Theorem
Reducibility

Reducibility





- Types
- Type Deductions
- Subject Reduction Theorem
- Reducibility
- Typability

λ^{\rightarrow} is Strongly Normalizing

Types and Normalization

All typable terms are β -strongly normalizing.

Typability

- Types
- - Types
 - Type Deductions
 - Subject Reduction Theorem
 - Reducibility
 - Typability

 λ \rightarrow : Type Assignments

Decidability of Type Assignment

Type checking Given M, σ , does $\vdash M : \sigma$?

 $\vdash M : \sigma$?

Typability Given M, does there exist σ such that $\vdash M : \sigma$?

⊢ *M* :?

Inhabitation Given σ , does there exist M such that $\vdash M : \sigma$?

 \vdash ? : σ

 λ^{\rightarrow} : Type Assignments

Type Deductions Subject Reduction Theorem Typability

$\vdash M : \sigma \text{ suffices}$

Note that with $\Gamma = \{x_1 : \sigma_1, \dots, x_n : \sigma_n\}$

 $\Gamma \vdash M : \tau$

is equivalent to

 $\vdash \lambda x_1 \dots x_n \cdot M : \sigma_1 \to \dots \to \sigma_n \to \tau$

 $\lambda^{\longrightarrow}$: Type Assignments

Decidability of Type Assignment

Type Checking is Decidable

It is decidable whether a statement of λ^{\rightarrow} is provable.

Types
Type Deductions
Subject Reduction Theorer
Reducibility
Typability

 $\lambda^{\longrightarrow}$: Type Assignments

Typable terms do not have unique types.

Types
Type Deductions
Subject Reduction Theorem
Reducibility
Typability

Decidability of Type Assignment

Type Checking is Decidable

It is decidable whether a statement of λ^{\rightarrow} is provable.

Typability is Decidable

It is decidable whether a term of λ^{\rightarrow} has a type.

4□ > 4□ > 4□ > 4□ > 4□ > 4□

A Demaille

etc.

Simply Typed λ -Calculus

 $\lambda^{\longrightarrow}$: Type Assignments

is valid for any σ, τ , including $\sigma = \sigma' \to \sigma'', \tau = (\tau' \to \tau'') \to \tau'$

42 / 4

Types $\lambda^{
ightharpoonup}$: Type Assignments

Types
Type Deductions
Subject Reduction Theorem
Reducibility
Typability

Simply Typed λ -Calculus

Types are not Unique. . .

but some are more Unique than others ...

All the types of K are instances of $\sigma \to \tau \to \sigma$.

Types are not Unique...

lypes Type Deductions Subject Reduction Theorem Reducibility Typability

Bibliography Notes

- [2] Complete and readable lecture notes on λ -calculus. Uses conventions different from ours.
- [3] Additional information, including slides.
- [1] A classical introduction to λ -calculus.

 λ $\stackrel{}{
ightharpoonup}$: Types Assignments

Types
Type Deductions
Subject Reduction Theorem
Reducibility
Typability

Bibliography I

Henk Barendregt and Erik Barendsen.

Introduction to lambda calculus, March 2000.

http://www.cs.ru.nl/~erikb/onderwijs/T3/materiaal/ lambda.pdf.

Andrew D. Ker.

Lambda calculus and types, May 2005.

http://web.comlab.ox.ac.uk/oucl/work/andrew.ker/lambda-calculus-notes-full-v3.pdf.

Andrew D. Ker.

Lambda calculus notes, May 2005.

http://web.comlab.ox.ac.uk/oucl/work/andrew.ker/.



A. Demaille

Simply Typed λ -Calculus

5 / 45

