# Simply Typed $\lambda$-Calculus

Akim Demaille akim@lrde.epita.fr

EPITA — École Pour l'Informatique et les Techniques Avancées

June 14, 2016

## About these lecture notes

Many of these slides are largely inspired from Andrew D. Ker's lecture notes [Ker, 2005a, Ker, 2005b]. Some slides are even straightforward copies.



## Simply Typed $\lambda$-Calculus





## Types

## Types

Alonzo Church (1903–1995)

Haskell Curry (1900–1982)



## Types

Types first appeared with

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

Types are syntactic objects assigned to terms:

$$M : A \qquad M \text{ has type } A$$

For instance:

$$I : A \to A$$



## Untyped $\lambda$-calculus





## $\lambda$-terms

**$\Lambda$, set of $\lambda$-terms**

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

## $\lambda\beta$

### The $\lambda\beta$ Formal System

$$\frac{}{M = M} \qquad \frac{M = N}{N = M} \qquad \frac{M = N \quad N = L}{M = L}$$

$$\frac{M = M' \quad N = N'}{MN = M'N'} \qquad \frac{M = N}{\lambda x \cdot M = \lambda x \cdot N}$$

$$\frac{}{(\lambda x \cdot M)N = [N/x]M}$$



## Properties of $\lambda\beta$

**$\beta$-reduction is Church-Rosser.**

Any term has (at most) a unique NF.

**$\beta$-reduction is not normalizing.**

Some terms have no NF ($\Omega$).



## Paradoxes





## Self application

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

- Stop considering anything can be applied to anything
- A function and its argument have different behaviors

## Church vs. Curry





## Simple Types

- A set of type variables
  $\alpha, \beta, \ldots$
- A symbol $\to$ for functions
  $\alpha \to \alpha$, $\alpha \to (\beta \to \gamma)$, $(\alpha \to \beta) \to \gamma$, $\ldots$
- Possibly constants for "primitive" types
  $\iota$ for integers, etc.



## Simple Types

By convention $\to$ is right-associative:

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

This matches the right-associativity of $\lambda$:

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



## Alonzo Style, or Haskell Way?

**Church:**
**Typed $\lambda$-calculus**

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

**Curry:**
**$\lambda$-calculus with Types**

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

## $\lambda^{\to}$: Type Assignments





## Types





## Simple Types

$\mathcal{TV}$ a set of type variables $\alpha, \beta, \ldots$

### Simple Types

The set $\mathcal{T}$ of types $\sigma, \tau, \ldots$:

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



## Type Contexts

### Statement

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

### Type Context, Basis

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

### Assignment

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

## Type Contexts

### Type Context Restrictions

$\Gamma - x$ is the $\Gamma$ with all assignment $x : \sigma$ removed.
$\Gamma \restriction M$ is $\Gamma - \mathrm{FV}(M)$.



## Type Deductions





## A Natural Presentation

Type derivations are trees built from the following nodes.

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



## Type Statement

### Type Statement

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

$$\Gamma \vdash M : \sigma$$

if there is a derivation of $M : \sigma$ whose non-canceled assumptions are in $\Gamma$.

## Type Statements

Prove

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

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

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



## Type Statements

Prove

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

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



## Alternative Presentation of Type Derivations

### Type Derivations

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

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

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

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



## Type Statements

Prove

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

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

## Type Statements

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

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



## Typability

**Typability**

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

- $\omega, \Omega$ are not typable.
- $S, K, I$ are typable.
- $Y$ is not typable!



## Subject Construction Lemma I

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

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

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

- If $M = NL$, then

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



## Subject Construction Lemma I

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

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

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

  - If $x \notin \mathrm{FV}(N)$

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

## Derivations are not Unique

They are for $\beta$-normal forms.



## Subject Reduction Theorem





## Conversions and Types

**$\alpha$-Invariance**

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

**Substitution**

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

then

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



## Subject Reduction Theorem

**Subject Reduction Theorem**

If $\Gamma \vdash M : \sigma$ and $M \to_\beta N$ then $\Gamma \vdash N : \sigma$.

What about the converse?

**Subject Expansion**

If $\Gamma \vdash N : \sigma$ and $M \to_\beta N$ then $\Gamma \vdash M : \sigma$.

$\Omega$ is not typable, but $KI\Omega \to I$.

## Reducibility





## Types and Normalization

**$\lambda^{\to}$ is Strongly Normalizing**

All typable terms are $\beta$-strongly normalizing.



## Typability





## $\vdash M : \sigma$ suffices

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

$$\Gamma \vdash M : \tau$$

is equivalent to

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

## Decidability of Type Assignment

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

$$\vdash M : \sigma?$$

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

$$\vdash M :?$$

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

$$\vdash ? : \sigma$$



## Decidability of Type Assignment

**Type Checking is Decidable**

It is decidable whether a statement of $\lambda^{\to}$ is provable.

**Typability is Decidable**

It is decidable whether a term of $\lambda^{\to}$ has a type.



## Types are not Unique...

Typable terms do not have unique types.

$$\dfrac{\dfrac{\dfrac{}{\{x : \sigma\} \mapsto x : \sigma}}{\{x : \sigma\} \mapsto \lambda y \cdot x : \tau \to \sigma}}{\mapsto K : \sigma \to \tau \to \sigma}$$

is valid for *any* $\sigma, \tau$, including $\sigma = \sigma' \to \sigma'', \tau = (\tau' \to \tau'') \to \tau'$ etc.



## Types are not Unique. . .

but some are more Unique than others ...

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

## Bibliography Notes

[Ker, 2005a] Complete and readable lecture notes on $\lambda$-calculus. Uses conventions different from ours.

[Ker, 2005b] Additional information, including slides.

[Barendregt and Barendsen, 2000] A classical introduction to $\lambda$-calculus.



## Bibliography I



Barendregt, H. and Barendsen, E. (2000).
Introduction to lambda calculus.
http://www.cs.ru.nl/~erikb/onderwijs/T3/materiaal/lambda.pdf.


Ker, A. D. (2005a).
Lambda calculus and types.
http://web.comlab.ox.ac.uk/oucl/work/andrew.ker/lambda-calculus-notes-full-v3.pdf.


Ker, A. D. (2005b).
Lambda calculus notes.
http://web.comlab.ox.ac.uk/oucl/work/andrew.ker/.