Lambda Calculus with Types

Henk Barendregt
ICIS
Radboud University Nijmegen
The Netherlands
Lambda Calculus with Types (698 pp)

Authors: Henk Barendregt, Wil Dekkers, Richard Statman

Part 1. Simple Types $\lambda^A$
Gilles Dowek
Marc Bezem
Silvia Ghilezan
Michael Moortgat

Part 2. Recursive Types $\lambda^A$
Mario Coppo
Felice Cardone

Part 3 Intersection Types $\lambda^S$
Mariangiola Dezani-Ciancaglini
Fabio Alessi
Furio Honsell
Paula Severi
Pawel Urzyczyn
The fathers

Alonzo Church (1903-1995) as mathematics student at Princeton University (1922 or 1924)

Haskell B. Curry (1900-1982) as BA in mathematics at Harvard (1920)
Church’s contribution: untyped lambda terms (1933)

Lambda terms

\[
\text{var ::= } c \mid \text{var}'
\]

\[
\text{term ::= var} \mid \text{term term} \mid \lambda \text{var term}
\]

Lambda calculus

\[
(\lambda x. M) N = M[x := N]
\]

mathematical axiom

\[
\begin{align*}
M = N & \Rightarrow N = M \\
M = N & \& N = L \Rightarrow M = L \\
M = N & \Rightarrow MP = NP \\
M = N & \Rightarrow PM = PN \\
M = N & \Rightarrow \lambda x. M = \lambda x. N
\end{align*}
\]

logical axiom and rules

We write \( \vdash_\lambda M = N \) if \( M = N \) is provable by these axioms and rules

- Computations termination
- Processes continuation

Functional programming (Lisp, Scheme, ML, Clean, Haskell)
Curry’s contributions

Curry: Russell paradox as fixed point of the operator Not

Idea: For \( a \in A \) write \( Aa \)

Then as \( \{x \mid P[x]\} \) we can take \( \lambda x.P[x] \)

Indeed, we get the intended interpretation

\( a \in \{x \mid P[x]\} \) becomes \( (\lambda x.P[x])a = P[a] \)

Taking \( R = \{x \mid x \notin x\} = \lambda x.\neg(xx) \) we get

\[ \forall r. [Rr \iff \neg(rr)] \]

hence \( RR \iff \neg(RR) \)

Note that \( RR \equiv (\lambda x.\neg(xx))(\lambda x.\neg(xx)) = Y(\neg) \] □

Typing of Whitehead-Russell was transformed into ‘functionality’

\[
\frac{\Gamma \vdash FABM \quad \Gamma \vdash AN \quad \Gamma, Ax \vdash BM}{\Gamma \vdash B(MN)} \quad \frac{\Gamma \vdash FAB(\lambda x.M)}{\Gamma \vdash FAB(\lambda x.M)}
\]
Curry’s contributions

Curry: Russell paradox as fixed point of the operator Not

Idea: For \( a \in A \) write \( Aa \)

Then as \( \{ x \mid P[x] \} \) we can take \( \lambda x.P[x] \)

Indeed, we get the intended interpretation

\( a \in \{ x \mid P[x] \} \) becomes \( (\lambda x.P[x])a = P[a] \)

Taking \( R = \{ x \mid x \notin x \} = \lambda x.\neg(xx) \) we get

\[ \forall r. [Rr \iff \neg(rr)] \]

hence \( RR \iff \neg(RR) \)

Note that \( RR \equiv (\lambda x.\neg(xx))(\lambda x.\neg(xx)) = Y(\neg) \)

Typing of Whitehead-Russell was transformed into ‘functionality’

\[
\Gamma \vdash M : A \rightarrow B \quad \Gamma \vdash N : A \quad \Gamma, x : A \vdash M : B \\
\Gamma \vdash MN : B \\
\Gamma \vdash \lambda x. M : (A \rightarrow B)
\]
Curry: combinators, correspondence with logic, linguistics

| I ≡ λx.x : A→A |
| K ≡ λxy.x : A→B→A |
| S ≡ λxyz.xz(yz) : (A→B→C)→(A→B)→A→C |

From these all closed lambda terms can be defined applicatively

Also with types

Curry: “Hey, these are tautologies” ↦ Curry-Howard correspondence

Inspired by Ajdukiewicz (and indirectly by Leśniewski)

Curry gave types to syntactic categories

<table>
<thead>
<tr>
<th>n noun/subject</th>
<th>s sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>n→n</td>
<td>‘red hat’ (adjective)</td>
</tr>
<tr>
<td>(n→n)→n</td>
<td>‘redness’</td>
</tr>
<tr>
<td>n→(n→n)</td>
<td>‘(John and Henry) are brothers’</td>
</tr>
<tr>
<td>n→s</td>
<td>‘Mary sleeps’</td>
</tr>
<tr>
<td>n→n→s</td>
<td>‘Mary kisses John’</td>
</tr>
<tr>
<td>s→s</td>
<td>‘not(Mary kisses John)’</td>
</tr>
</tbody>
</table>

More complex cases

(n→n)→(n→n)→(n→n) ‘slightly large’

((n→n)→(n→n))→(n→n)→(n→n) ‘slightly too large’
Substitution is needed

PM does not provide it: $\lambda$-calculus does
The cycle

<table>
<thead>
<tr>
<th>Category</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Untyped lambda terms (6)</td>
<td>( \lambda \langle \Lambda, \cdot \rangle )</td>
</tr>
<tr>
<td>Simple types (22)</td>
<td>Free type algebras</td>
</tr>
<tr>
<td>Recursive types (2)</td>
<td>Type algebras</td>
</tr>
<tr>
<td>Subtyping (1)</td>
<td>Type structures</td>
</tr>
<tr>
<td>Intersection types (4)</td>
<td>Intersection type structures</td>
</tr>
</tbody>
</table>

All untyped lambda terms appear again
An increasing chain of systems $\lambda^A \to \subseteq \lambda^A \subseteq \lambda^S \subseteq \lambda^S$

$$
\begin{align*}
\Gamma, x : A &\vdash x : A \\
\Gamma &\vdash M : (A \to B) & \Gamma &\vdash N : A & \Gamma, x : A &\vdash M : B \\
\hline
\Gamma &\vdash (MN) : B \\
\Gamma &\vdash (\lambda x. M) : (A \to B)
\end{align*}
$$

$$
\begin{align*}
\Gamma &\vdash M : A & A = B \\
\hline
\Gamma &\vdash M : B
\end{align*}
$$

$$
\begin{align*}
\Gamma &\vdash M : A & A \leq B \\
\hline
\Gamma &\vdash M : B
\end{align*}
$$

$$
\begin{align*}
\Gamma &\vdash M : A \cap B \\
\Gamma &\vdash M : A \cap B \\
\Gamma &\vdash M : A & \Gamma &\vdash M : B \\
\hline
\Gamma &\vdash M : A \\
\Gamma &\vdash M : B \\
\Gamma &\vdash M : A \cap B
\end{align*}
$$

$$
\Gamma \vdash M : \top
$$
An increasing chain of systems \( \lambda^A \rightarrow \subseteq \lambda^A \subseteq \lambda^S \leq \subseteq \lambda^S \cap \)

\[
\begin{align*}
\Gamma, x : A & \vdash x : A \\
\Gamma & \vdash M : (A \rightarrow B) & \Gamma & \vdash N : A & \Gamma, x : A & \vdash M : B \\
\Gamma & \vdash (MN) : B & \Gamma & \vdash (\lambda x. M) : (A \rightarrow B)
\end{align*}
\]

\[
\begin{align*}
\Gamma & \vdash M : A & A & = B \\
\Gamma & \vdash M : B
\end{align*}
\]

\[
\begin{align*}
\Gamma & \vdash M : A & A & \leq B \\
\Gamma & \vdash M : B
\end{align*}
\]

\[
\begin{align*}
\Gamma & \vdash M : A \cap B & \Gamma & \vdash M : A \cap B & \Gamma & \vdash M : A & \Gamma & \vdash M : B \\
\Gamma & \vdash M : A & \Gamma & \vdash M : B & \Gamma & \vdash M : A \cap B
\end{align*}
\]

\[
\begin{align*}
\Gamma & \vdash M : A \cap B & \Gamma & \vdash M : A \cap B & \Gamma & \vdash M : A & \Gamma & \vdash M : B \\
\Gamma & \vdash M : A & \Gamma & \vdash M : B & \Gamma & \vdash M : A \cap B
\end{align*}
\]

\[
\begin{align*}
\Gamma & \vdash M : A & \Gamma & \vdash M : B \\
\Gamma & \vdash M : \top
\end{align*}
\]
### Examples

<table>
<thead>
<tr>
<th>$\lambda^A$</th>
<th>$\lambda_{xy.yyy} : (A \to A \to B) \to A \to B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda^A$</td>
<td>$\lambda_{xx} : A$</td>
</tr>
<tr>
<td></td>
<td>$\lambda_{xx}(\lambda_{xx}) : B$</td>
</tr>
<tr>
<td>$\lambda^S$</td>
<td>$\lambda_{xx} : A \to B$</td>
</tr>
<tr>
<td></td>
<td>$\lambda_{xx} : (A \to B) \to B$</td>
</tr>
<tr>
<td>$\lambda^S$</td>
<td>$\lambda_{xx} : A \cap (A \to B) \to B$</td>
</tr>
<tr>
<td></td>
<td>$\text{KI} \Omega : A \to A$</td>
</tr>
<tr>
<td></td>
<td>$\Omega \triangleq (\lambda_{xx}(\lambda_{xx})$</td>
</tr>
</tbody>
</table>
|             | $\text{as } \Omega : T$}
Simply typed \(\lambda\)-calculus

Simple types from ground type 0

\[ \Pi = 0 \mid \Pi \rightarrow \Pi \]

\(\Lambda(A)\): \(\lambda\)-terms of type \(A\). Write \(\Lambda_{\rightarrow} = \bigcup_{A \in \Pi} \Lambda(A)\)

\[
\begin{align*}
x^A &\in \Lambda(A) \\
M \in \Lambda(A \rightarrow B), \ N \in \Lambda(A) \quad &\Rightarrow \quad (MN) \in \Lambda(B) \\
M \in \Lambda(B) \quad &\Rightarrow \quad (\lambda x^A.M) \in \Lambda(A \rightarrow B)
\end{align*}
\]

Church’s version of \(\lambda_{\rightarrow}^A\)

Default equality \(\equiv_{\beta\eta}\) preserves types

\[
\begin{align*}
(\lambda x^A.M)N &= M[x^A:=N] \quad &\beta\text{-conversion} \\
\lambda x^A.Mx^A &= M \quad &\eta\text{-conversion}
\end{align*}
\]
Church vs Curry. Some results

Prop. (i) For $M \in \Lambda^0(A)$ one has

$$\vdash |M| : A$$

(ii) For $M \in \Lambda^0$ in $\beta$-nf such that $\vdash M : A$

there is a unique $M^A \in \Lambda(A)$ such that $|M^A| \equiv M$

(iii) For open $M$ not in $\beta$-nf (ii) fails: $\text{KI}_y : A \rightarrow A$

(iv) Even for closed $M$ not in $\beta$-nf (ii) fails: $(\lambda x.x)(\lambda y.1) : A \rightarrow A$

The counter-examples in (iii), (iv) are due to the presence or creation of a $K$-redex

Prop. For normal $M$ one can identify $\vdash M : A$ and $M^A \in \Lambda(A)$

preserving reduction

Prop. $\vdash M : A$ $\Rightarrow$ $M$ has a $\beta\eta$-nf (Normalization Theorem)

Prop. $\vdash M : A \& M \mapsto_{\beta\eta} M' \Rightarrow \vdash M' : A$ (Subject Reduction Theorem)
Type structures

Def. Let $\mathcal{M} = \{ \mathcal{M}(A) \}_{A \in \mathfrak{A}}$ be a family of non-empty sets

(i) $\mathcal{M}$ is called a type structure for $\lambda^A \rightarrow$ if

$$\mathcal{M}(A \rightarrow B) \subseteq \mathcal{M}(B)^{\mathcal{M}(A)}$$

Here $Y^X$ denotes the collection of set-theoretic functions

$$\{ f \mid f : X \rightarrow Y \}$$

(ii) Let $\mathcal{M}$ be provided with application operators

$$(\mathcal{M}, \cdot) = (\{ \mathcal{M}(A) \}_{A \in \mathfrak{A}}, \{ \cdot_{A,B} \}_{A,B \in \mathfrak{A}})$$

$$\cdot_{A,B} : \mathcal{M}(A \rightarrow B) \times \mathcal{M}(A) \rightarrow \mathcal{M}(B).$$

A typed applicative structure is such an $(\mathcal{M}, \cdot)$ satisfying extensionality:

$$\forall f, g \in \mathcal{M}(A \rightarrow B) \left[ \left( \forall a \in \mathcal{M}(A) \ f \cdot_{A,B} a = g \cdot_{A,B} a \right) \Rightarrow f = g \right].$$

Prop. The notions ‘type structure’ and ‘typed applicative structure’ are equivalent
**λ**-Full type structures

Def. Given a set \( X \). The *full type structure* over \( X \)

\[
\mathcal{M}_X = \{ X(A) \}_{A \in \mathbb{T}}
\]

where \( X(A) \) is defined inductively as follows

\[
\begin{align*}
X(0) & \triangleq X; \\
X(A \rightarrow B) & \triangleq X(B)^{X(A)} , \text{the set of functions from } X(A) \text{ into } X(B)
\end{align*}
\]

Def. \( \mathcal{M}_n \triangleq \mathcal{M}_{\{1, \ldots, n\}} \)

\[
\begin{array}{c|c}
\hdots & \\
\hline
\mathcal{M}(A \rightarrow B) & F \\
\hdots & \\
\mathcal{M}(A) & M \\
\hdots & \\
\mathcal{M}(B) & FM \\
\hdots & \\
\mathcal{M}(2) & \\
\mathcal{M}(1) & X^X \\
\mathcal{M}(0) & X
\end{array}
\]

Partial view of \( \mathcal{M} = \mathcal{M}_X \)

\( F, M, \Lambda_{ch} \)

finite at each level \( A \) if \( X \) is finite
Full type structures

Def. Given a set $X$. The **full type structure** over $X$

$$\mathcal{M}_X = \{X(A)\}_{A \in \mathbb{T}}$$

where $X(A)$ is defined inductively as follows

$$X(0) \triangleq X;$$
$$X(A \rightarrow B) \triangleq X(B)^{X(A)}$$

the set of functions from $X(A)$ into $X(B)$

Def. $\mathcal{M}_n \triangleq \mathcal{M}\{1, \ldots, n\}$

<table>
<thead>
<tr>
<th>$\mathcal{M}(A \rightarrow B)$</th>
<th>$[F]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{M}(A)$</td>
<td>$[M]$</td>
</tr>
<tr>
<td>$\mathcal{M}(B)$</td>
<td>$[FM]$</td>
</tr>
<tr>
<td>$\mathcal{M}(2)$</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{M}(1) = X^X$</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{M}(0) = X$</td>
<td></td>
</tr>
</tbody>
</table>

Partial view of $\mathcal{M} = \mathcal{M}_X$

$F, M \in \Lambda^{ch}$

finite at each level $A$ if $X$ is finite
Let \( \rho \) be a valuation in \( \mathcal{M}_X \): we require \( \rho(x^A) \in \mathcal{M}(A) \)

For \( M \in \Lambda \rightarrow (A) \) we define \( \llbracket M \rrbracket_\rho \in \mathcal{M}(A) \)

\[
\begin{align*}
\llbracket x^A \rrbracket_\rho & \triangleq \rho(x^A) \\
\llbracket MN \rrbracket_\rho & \triangleq \llbracket M \rrbracket_\rho \llbracket N \rrbracket_\rho \\
\llbracket \lambda x^A . M \rrbracket_\rho & \triangleq \lambda d \in X(A). \llbracket M \rrbracket_\rho(x^A := d)
\end{align*}
\]

where \( \rho(x^A := d) = \rho' \) with

\[
\begin{align*}
\rho'(x^A) & \triangleq d \\
\rho'(y^B) & \triangleq \rho(y^B) \quad \text{if } y^B \not\equiv x^A
\end{align*}
\]

Define

\[
\mathcal{M}_X \models M = N \iff \forall \rho \; \llbracket M \rrbracket_\rho = \llbracket N \rrbracket_\rho
\]

\[
\iff \llbracket M \rrbracket \equiv \llbracket N \rrbracket \quad \text{if } M, N \in \Lambda^0
\]
We know that $\mathbb{N} = \langle \mathbb{N}, +, \times, 0, 1 \rangle$ can be ‘$\lambda$-defined’

Prop. The rational numbers

$$\mathbb{Q} = \langle \mathbb{Q}, +, \times, -, :, 0, 1 \rangle$$

cannot be $\lambda$-defined: there is no type $\mathbb{Q}$ and terms

$$\vdash M_+ : \mathbb{Q} \rightarrow \mathbb{Q} \rightarrow \mathbb{Q}$$
$$\vdash M_\times : \mathbb{Q} \rightarrow \mathbb{Q} \rightarrow \mathbb{Q}$$
$$\vdash M_- : \mathbb{Q} \rightarrow \mathbb{Q} \rightarrow \mathbb{Q}$$
$$\vdash M_0 : \mathbb{Q}$$
$$\vdash M_1 : \mathbb{Q}$$

such that the usual laws hold

Proof. The homomorphic image of a field $K$ is $K$ itself;
but $\mathcal{M}_X(\mathbb{Q})$ is finite for $X$ finite
\[ \lambda \to M_1, M_2 \]

**Def.** \( \text{Th}(M) = \{ M = N \mid M, N \in \Lambda^0 \land M \models M = N \} \)

\( \text{Th}(M_1) \) is inconsistent: all terms of the same type are equated

\[ M_2 \models c_1 = c_3 : 1 \to 0 \to 0 \]
\[ M_2 \not\models c_1 = c_2 : 1 \to 0 \to 0 \]

**Exercises**

\[ M_2 \models c_2 = c_4 = c_6 = \cdots \]
\[ M_2 \not\models c_1 = c_3 = c_5 = \cdots \]
\[ M_2 \not\models c_0 = c_1 \]
\[ M_2 \not\models c_0 = c_2 \]
\[ M_5 \models c_4 = c_{64} \quad (64 = \text{lcm}\{1, 2, 3, 4, 5\}) \]
\[ M_6 \not\models c_4 = c_{64} \quad \text{even if } (64 = \text{lcm}\{1, 2, 3, 4, 5, 6\}) \]
\[ M_6 \models c_5 = c_{65} \]
\( \lambda \rightarrow \) Partial semantics (Friedman [1975])

Let \( M \) be a typed applicative structure

A partial valuation in \( M \) is a family \( \rho = \{ \rho_A \}_{A \in T} \) of partial maps

\[
\rho_A : \text{Var}(A) \not\rightarrow M(A)
\]

The partial semantics \( \sem{}{\rho} : \lambda \rightarrow (A) \not\rightarrow M(A) \) under \( \rho \) is

\[
\begin{align*}
\sem{x^A}{\rho} &= \rho_A(x) \\
\sem{PQ}{\rho} &= \sem{P}{\rho} \sem{Q}{\rho} \\
\sem{\lambda x^A.P}{\rho} &= \lambda d \in M(A). \sem{P}{\rho[x:=d]}
\end{align*}
\]

Often we write \( \sem{M}{\rho} \) for \( \sem{M}{\rho} \)

The expression \( \sem{M}{\rho} \) may not always be defined, even if \( \rho \) is total

The problem arises with \( \sem{\lambda x.P}{\rho} \) when

\[
\lambda d \in M(A). \sem{P}{\rho[x:=d]} \in M(B)_{M(A)} - M(A \rightarrow B)
\]

If \( \sem{\lambda x.P}{\rho} \) exists it is uniquely defined
A typed $\lambda$-model is a type structure $\mathcal{M}$ such that $$\llbracket M \rrbracket_\rho$$ is defined for all $A \in \mathbb{T}$, $M \in \Lambda(A)$, and $\rho$ such that $\text{FV}(M) \subseteq \text{dom}(\rho)$

Examples of typed $\lambda$-models

- $\mathcal{M}_X$ : full type structures
- $\mathcal{M}_{\beta\eta}$ : open typed terms modulo $\beta\eta$-equality
- $\mathcal{M}[C]$ : closed typed term models modulo extensionality

where $C$ is a set of typed constants such that $\mathcal{M}[C](0) \neq \emptyset$

In $\mathcal{M}[C]$ one can define $$\llbracket M \rrbracket_\rho = [M[\bar{x} := \rho(\bar{x})]]$$

and show it works
There are only five $\text{Th}(\mathcal{M}[C])$ coming from

- $C_1 = \{c^0, d^0\}$
- $C_2 = \{c^0, f^1\}$
- $C_3 = \{c^0, f^1, g^1\}$
- $C_4 = \{c^0, \Phi^{3\rightarrow0\rightarrow0}\}$
- $C_5 = \{c^0, b^{0\rightarrow0\rightarrow0}\}$

There is a sixth, the inconsistent theory, coming from $C_0 = \{c^0\}$

One has

$\text{Th}(\mathcal{M}[C_5]) \subseteq \text{Th}(\mathcal{M}[C_4]) \subseteq \text{Th}(\mathcal{M}[C_3]) \subseteq \text{Th}(\mathcal{M}[C_2]) \subseteq \text{Th}(\mathcal{M}[C_1]) \subseteq \text{Th}(\mathcal{M}[C_0])$

$\text{Th}(\mathcal{M}[C_5]) = \{M = N \mid M =_{\beta\eta} N\}$ minimal theory

$\text{Th}(\mathcal{M}[C_1])$ is the unique maximally consistent theory consisting of all consistent equations together

$\mathcal{M}[C_1]$ is the minimal model, with decidable equality (Loader)
The model $\mathcal{M}[C]$

Let $C$ be a set of typed constants

Examples $C_0 = \{c^0\}, \ C_1 = \{c^0, d^0\}$

Define $\Lambda^0_\to[C]$ as the set of closed typed $\lambda$-terms built up from $C$

Now $\Lambda^0_\to[C]$ modulo 'extensionality' will be considered as a term model

For $M, N \in \mathcal{M}[C](A)$ define

$$M \equiv^\text{ext}_C N \iff M =_{\beta\eta} N \quad \text{if } A = 0$$

$$\iff \forall P \in \mathcal{M}[C](B). [MP \equiv^\text{ext}_C NP] \quad \text{if } A = B \to C$$

Then $M \equiv^\text{ext}_C N \iff \forall \tilde{P} \in \mathcal{M}[C]. [M\tilde{P} =_{\beta\eta} N\tilde{P}]$

Thm. $\mathcal{M}[C] = \Lambda^0_\to[C]/\equiv^\text{ext}_C$ is an extensional $\lambda$-model

not trivial at all, we need

$$\mathcal{M}[C] \models M = N \Rightarrow \mathcal{M}[C] \models FM = FN \& \mathcal{M}[C] \models \lambda x.M = \lambda x.N$$

Exercises 1. $\mathcal{M}(c^0) \cong \mathcal{M}_1$, where $\mathcal{M}[c^0] = \mathcal{M}[\{c^0\}]$

2. $\mathcal{M}[c^0, d^0] \models c_1 = c_2$

3. $\mathcal{M}[c^0, d^0] \not\cong \mathcal{M}_2$
For $M, N \in \Lambda^0 \rightarrow [C](A)$ define

$$M \equiv_{\text{obs}}^C N \iff \forall F : (A \rightarrow 0). FM =_{\beta\eta} FN$$

Remark

$$M \equiv_{\text{obs}}^C N \implies FM \equiv_{\text{obs}}^C FN$$

$$M \equiv_{\text{obs}}^C N \implies M \equiv_{\text{ext}}^C N$$

$$M \equiv_{\text{ext}}^C N \iff \forall Z. MZ \equiv_{\text{ext}}^C NZ$$

Thm. $\forall M, N. [M \equiv_{\text{obs}}^C N \iff M \equiv_{\text{ext}}^C N]$ non-trivial

Cor. $M[C]$ is an (extensional) $\lambda$-model
A relation on $\mathcal{M}_C$ is a family $R = \{R_A\}_{A \in \mathbb{T}}$ with $R_A \subseteq \mathcal{M}_C(A)^n$.

A relation is *logical* if for all $A, B \in \mathbb{T}$ and all $\vec{M} \in \mathcal{M}_C(A \rightarrow B)^n$

$$R_{A \rightarrow B}(M_1, \cdots, M_n) \iff \forall N_1 \in \mathcal{M}_C(A) \cdots N_n \in \mathcal{M}_C(A) [R_A(N_1, \cdots, N_n) \Rightarrow R_B(M_1N_1, \cdots, M_nN_n)]$$

Thus a logical relation is fully determined by $R_0$

Prop. Suppose $\approx_{\text{ext}}^C$ is logical on $\mathcal{M}_C$. Then for all $M, N \in \mathcal{M}_C$

$$M \approx_{\text{ext}}^C N \iff M \approx_{\text{obs}}^C N$$

Proof. Only ($\Rightarrow$) is interesting

Assume $M \approx_{\text{ext}}^C N$ and $F \in \mathcal{M}_C(A \rightarrow 0)$ towards $FM =_{\beta\eta} FN$

Trivially $F \approx_{\text{ext}}^C F$

$$\Rightarrow FM \approx_{\text{ext}}^C FN, \quad \text{as } \approx_{\text{ext}}^C \text{ is logical}$$

$$\Rightarrow FM =_{\beta\eta} FN, \quad \text{as the type is } 0 \blacksquare$$
It remains to show that the $\approx_{C_i}^{ext}$ are logical

Def. $BE$ is the logical relation on $\mathcal{M}[C]$ determined by

$$BE_0(M, N) \triangleq M =_{\beta\eta} N$$

Lemma 1. Suppose $BE(c, c)$ for $c \in C$. Then

$$\forall M \in \Lambda[C]. BE(M, M)$$

Proof. By the usual arguments for logical relations. ■
Lemma 2. Suppose $BE(c, c)$ for all $c \in C$. Then $\approx^\text{ext}_C$ is $BE$ and hence logical

Proof. By Lemma 1 one has for all $M \in M[C]$

$$BE(M, M)$$

(0)

It follows that $BE$ is an equivalence relation on $M[C]$. We claim that for all $F, G \in M[C](A)$

$$BE_A(F, G) \iff F \approx^\text{ext}_C G,$$

By induction on the structure of $A$. Case $A = 0$. By definition. Case $A = B \to C$, then

$$(\Rightarrow) \quad BE_{B \to C}(F, G) \Rightarrow BE_C(FP, GP), \quad \text{for all } P \in M[C](B),$$

since $P \approx^\text{ext}_C P$ and hence by the IH $BE_B(P, P)$

$$\Rightarrow FP \approx^\text{ext}_C GP, \quad \text{for all } P \in M[C] \text{ by the IH}$$

$$\Rightarrow F \approx^\text{ext}_C G,$$

by definition.

$$(\Leftarrow) \quad F \approx^\text{ext}_C G \Rightarrow FP \approx^\text{ext}_C GP, \quad \text{for all } P \in M[C],$$

$$\Rightarrow BE_C(FP, GP)$$

(1)

by the induction hypothesis. In order to prove $BE_{B \to C}(F, G)$, assume $BE_B(P, Q)$ towards $BE_C(FP, GQ)$. Well, since also $BE_B(G, G)$, by (0), we have

$$BE_C(GP, GQ).$$

(2)

It follows from (1) and (2) and the transitivity of $BE$ (which on this type is the same as $\approx^\text{ext}_C$ by the IH) that $BE_C(FP, GQ)$ indeed.

By the claim $\approx^\text{ext}_C$ is $BE$ and therefore $\approx^\text{ext}_C$ is logical. ■
Lemma 3. \( \text{BE}(M, M) \) holds for \( M \in \mathcal{M}[C] \) of types 0, 1, 0→0→0. Proof. Easy. ■

Lemma 4. Let \( c = c^3 \in \mathcal{M}[C] \). Suppose

\[
\forall F, G \in \mathcal{M}[C](2) \left[ F \approx^c_{\text{ext}} G \Rightarrow F = \beta \eta G \right]
\]

Then \( \text{BE}_{A \rightarrow 0}(c, c) \)

Proof. Let \( c \) be given. Then for \( F, G \in \mathcal{M}[C](2), P \in \mathcal{M}[C](1) \) one has

\[
\begin{align*}
\text{BE}(F, G) & \Rightarrow FP = \beta \eta GP & \text{by Lemma 3} \\
& \Rightarrow F \approx^c_{\text{ext}} G \\
& \Rightarrow F = \beta \eta G & \text{by assumption} \\
& \Rightarrow cF = \beta \eta cG
\end{align*}
\]

Therefore we have by definition \( \text{BE}(c, c) \) ■

Last mortgage

For every \( F, G \in \mathcal{M}[C](2) \) one has

\[
F \approx^c_{\text{ext}} G \Rightarrow F = \beta \eta G.
\]

We must show

\[
[\forall h \in \mathcal{M}[C](1). Fh = \beta \eta Gh] \Rightarrow F = \beta \eta G. \quad (1)
\]
Analysis of terms of given type

\[3 \rightarrow 0 \rightarrow 0\]
\[\lambda \Phi \ x. x, \lambda \Phi x. \Phi (\lambda f . x), \lambda \Phi x. \Phi (\lambda f . f x), \lambda \Phi x. \Phi (\lambda f . (\Phi (\lambda g. g(f x))))], \ldots\]
\[\lambda \Phi x. \Phi (\lambda f_1 . w_{\{f_1\}} x), \lambda \Phi x . \Phi (\lambda f_1 . w_{\{f_1\}} \Phi (\lambda f_2 . w_{\{f_1, f_2\}} x)), \ldots ;\]
\[\lambda \Phi x. \Phi (\lambda f_1 . w_{\{f_1\}} \Phi (\lambda f_2 . w_{\{f_1, f_2\}} \cdots \Phi (\lambda f_n . w_{\{f_1, \ldots, f_n\}} x) \cdots)) \triangleq \]
\[\langle w_{\{f_1\}}, w_{\{f_1, f_2\}}, \ldots, w_{\{f_1, \ldots, f_n\}} \rangle^3 \rightarrow 0 \rightarrow 0\]

\[\lambda \Phi^3 \lambda x^o\]

\[f \quad o \quad 2\]

Let \( h_m \triangleq \lambda x . \Phi (\lambda f . f^m x) = \langle f^m \rangle : \mathcal{M}[C](1)\)

Claim

\[\forall F, G \in \mathcal{M}[C](2) \exists m \in \mathbb{N}. [F h_m = G h_m \Rightarrow F = \beta_n G]\]
Reducibility of types

That the \( \text{Th}(M[C_i]) \) form a chain follows from type reducibility

Def. \( A \leq_{\beta\eta} B \iff \exists F : A \rightarrow B \)

\[
\forall M_1 M_2 : A. [M_1 =_{\beta\eta} M_2 \iff FM_1 =_{\beta\eta} FM_2]
\]

(“there is a \( \lambda \)-definable injection from \( A \) to \( B \)”)

Thm. [Hierarchy Theorem (Statman [1980])] Inhabited members of \( \Pi \) can be partitioned in decidable classes \( \Pi_0, \Pi_1, \cdots, \Pi_5 \) such that

\[
\begin{align*}
0 & <_{\beta\eta} 0 \rightarrow 0 \quad \text{and all } A, B \in \Pi_i \text{ are } \beta\eta\text{-equivalent} \\
<_{\beta\eta} 0^2 & \rightarrow 0 \\
<_{\beta\eta} \cdots \\
<_{\beta\eta} 0^k & \rightarrow 0 \\
<_{\beta\eta} \cdots \\
<_{\beta\eta} 1 & \rightarrow 0 \rightarrow 0 \\
<_{\beta\eta} 1 & \rightarrow 1 \rightarrow 0 \rightarrow 0 \\
<_{\beta\eta} 3 & \rightarrow 0 \rightarrow 0 \\
<_{\beta\eta} (0^2 & \rightarrow 0) \rightarrow 0 \rightarrow 0
\end{align*}
\]

\( \in \Pi_0 \)

\( \in \Pi_1 \)

\( \in \Pi_2 \)

\( \in \Pi_3 \)

\( \in \Pi_4 \)

\( \in \Pi_5 \)

All not-inhabited types are equivalent to \( 0 \)
Results and open problems

Thm. Each $\mathcal{M}[\mathcal{C}]$ is a term model provided $\mathcal{M}[\mathcal{C}](0)$ is inhabited.

Thm. There are only five (six) resulting theories.

Open problems

- Can this be proved more directly?
- Are $\text{Th}(\mathcal{M}[\mathcal{C}_2])$, $\text{Th}(\mathcal{M}[\mathcal{C}_3])$, $\text{Th}(\mathcal{M}[\mathcal{C}_4])$ decidable?
Recursive types via $\mu$-abstraction

If we want $A = A \rightarrow B$ we simply work in $\mathcal{T}$ modulo $A = A \rightarrow B$.

This leads to recursive types via *simultaneous recursion*.

Alternatively we can write $A \triangleq \mu \alpha . \alpha \rightarrow B$ and postulate $A = A \rightarrow B$.

Consider

$$\mathcal{T}_\mu^A ::= A \mid \mathcal{T}_\mu^A \rightarrow \mathcal{T}_\mu^A \mid \mu A \mathcal{T}_\mu^A$$

where we work modulo $\mu$-reduction

$$\mu \alpha . A \Rightarrow_\mu A[\alpha := \mu \alpha . A]$$

We must be careful not to create confusion of variables.

The $\mu$-type $\mu \alpha . (\beta \rightarrow \mu \beta . (\alpha \rightarrow \beta))$ is not safe:

‘naively contracting’ $\mu \alpha$ leads to a clash.
Thm. [V. van Oostrom] By contrast to λ-terms for every $A \in \Pi^A_\mu$ one can make a renaming $A'$ such that never a variable clash occurs with the $\mu$-reducts of $A'$

Proof-sketch. Avoid configurations like

This is hereditary ■
Prop. (A. Polonsky)

For every $M \in \Lambda^\emptyset$ there exists a ‘principal pair’ $(S_M, a_M)$ such that

\[
\vdash_{S_M} M : a \\
\vdash_S M : a \iff \exists h : S_M \to S. h(a_M) \leq a
\]

for all type structures $S = \langle S, \to, \leq \rangle$ and $a \in S$

Def. Given $M, N \in \Lambda^\emptyset$ define

$M \lesssim N \iff \Delta$ there exists a morphism $h : S_M \to S_N$

Open problems

- Is $\lesssim$ on $\Lambda^\emptyset$ decidable?
- Study the unsolvables under $\lesssim$
\( \lambda^S \) Subject Reduction

In \( \lambda^o \) one has

\[
\begin{align*}
\Gamma \vdash M : A \\
M \rightarrow^\beta N
\end{align*}
\] \Rightarrow \quad \Gamma \vdash N : A

This also holds for \( \lambda^S \), for many intersection type structures \( S \)

The converse, \textit{subject expansion}, does not hold for \( \lambda^o \)
Subject expansion

Suppose

\[ \vdash P[x := Q] : A \]

where \( P \equiv \cdots x \cdots x \cdots x \cdots \)

so \( \cdots Q \cdots Q \cdots Q \cdots : A \)

Each of these occurrences of \( Q \) may need another type \( B_1, B_2, B_3 \)

But then we can give \( \lambda x. P \) the type \( B_1 \cap B_2 \cap B_3 \Rightarrow A \)

Hence the \( \beta \)-expansion \( (\lambda x. P)Q \) also the type \( A \)

If the number of occurrences of \( x \) in \( P \) is 0,

then we may give to \( \lambda x. P \) the type \( U \Rightarrow A \)

which is consistent as the empty intersection
again

\[ \vdash (\lambda x. P)Q : A \]
A model for $\lambda\beta$ (Barendregt, Coppo, Dezani)

Therefore

$$\begin{align*}
\Gamma \vdash M : A \\
M =_{\beta} N
\end{align*} \Rightarrow \Gamma \vdash N : A$$

so (for closed $M$)

$$X_M = \{ A \mid \vdash M : A \}$$

which looks like a $\lambda$-model. Indeed, such a set $X$ is a filter of types

- $X \neq \emptyset$
- $A, B \in X \Rightarrow (A \cap B) \in X$
- $B \geq A \in X \Rightarrow B \in X$

For filters $X, Y$ one can define application

$$XY = \{ B \mid \exists A \in Y (A \rightarrow B) \in X \}$$

is well defined and one has (for many intersection type structures)

$$X_M X_N = X_{MN}$$

Given an intersection type structure $S$, then $\mathcal{F}^S = \{ X \subseteq S \mid X \text{ is a filter} \}$ is the filter structure over $S$. If $S$ is natural it is a $\lambda$-model.
Open problems

Investigate equivalence of categories involved

Recent result relating $\lambda^A$ and $\lambda^S$

Sylvain Salvati: Elements of $M_n$ can be described as intersection types

relation between results of Loader and Urzyczyn concerning respectively

- undecidability of $\lambda$-definability in $M_n$
- undecidability of inhabitation in $\lambda^S$

\[
d \in M_X(A) \leadsto \xi^A_d \in \prod^X^\cap
\]

\[
x_0^d = d
\]

\[
x_{A \to B}^d = \bigcap_{e \in X(B)} \xi^B_e \to \xi^A_{de}
\]

\[
[M] = d \iff \vdash M : \xi^A_d, \quad \text{for } M \in \Lambda(A)
\]
Marketing Strategy (sine qua non)

Summary in $\leq 20$ words of 698 pages

*This handbook with exercises reveals in formalisms hitherto mainly used for designing and verifying software and hardware unexpected mathematical beauty*