The Limits of the Curry-Howard Isomorphism

Reinhard Kahle[1] and Anton Setzer[2]

[1]DM and CENTRIA, FCT
Universidade Nova de Lisboa, Portugal

[2]Dept. of Computer Science, Swansea University, Swansea, UK

Tübingen, FRC workshop, 21 February 2014

Martin-Löf Type Theory and Curry Howard Isomorphism

- Martin-Löf Type Theory (MLTT) can be considered as "**radical** formalisation **Curry-Howard Isomorphism**"
- **Propositions as types**
  - No distinction between data types and propositions.
- Propositions are true if they are inhabited (have a **proof**).
- Because of the last two items, elements of types (Set = collection of types) must be total:
  - Otherwise we can prove

$$\begin{aligned} p &: A \\ p &= p \end{aligned}$$

Function Type in MLTT

- An element of $A \to B$ is a **program** which for $a : A$ returns $b : B$.
- Implicitly contains an implication.
  So **implication explained by an implication**.
- In order to overcome this, Martin-Löf refers to that we
  we know what a program is that takes input $a : A$ and returns $b : B$.
- Doesn't mean that we know what an arbitrary program is but
  - when we introduce a program we need to explain that it is a program of its type, and
  - we know how to apply a program.
- Therefore **programs are always typed**.

Inductive Data Types

- As in other axiomatic systems proof theoretic strength obtained by adding data types and their introduction/elimination/equality rules.
- Inductive data types – in Agda notation

$$
\begin{array}{l}
\mathrm{data}\ \mathbb{N} : \mathrm{Set\ where} \\
\quad \begin{array}{lcl} 0 & : & \mathbb{N} \\ \mathrm{S} & : & \mathbb{N} \to \mathbb{N} \end{array}
\end{array}
$$

- Elimination rule is higher type primitive recursion.

Universes

- Universes = collection of sets.
- Formulated if using the logical framework as:

$$\mathrm{U}_0 : \mathrm{Set} \qquad \mathrm{T}_0 : \mathrm{U}_0 \to \mathrm{Set}$$

- U_0 = set of codes for sets.
- T_o = decoding function.

Universe closed under W

$$
\begin{array}{l}
\text{mutual} \\
\quad \text{data } \mathrm{U}_0 : \text{Set where} \\
\quad\quad \widehat{\mathbb{N}} \quad : \quad \mathrm{U}_0 \\
\quad\quad \widehat{\mathrm{W}} \quad : \quad (x : \mathrm{U}_0) \to (\mathrm{T}_0\ x \to \mathrm{U}_0) \to \mathrm{U}_0 \\
\quad\quad \cdots \\
\\
\quad \mathrm{T}_0 : \mathrm{U}_0 \to \text{Set} \\
\quad \mathrm{T}_0\ \widehat{\mathbb{N}} \qquad\quad = \quad \mathbb{N} \\
\quad \mathrm{T}_0\ (\widehat{\mathrm{W}}\ a\ b) \quad = \quad \mathrm{W} x : \mathrm{T}_0\ a . \mathrm{T}_0\ (b\ x) \\
\quad \cdots
\end{array}
$$

Universe Operator (Palmgren)

- If $A : \mathrm{Fam}(\mathrm{Set})$ where the type of families of sets is

$$\mathrm{Fam}(\mathrm{Set}) = \Sigma X : \mathrm{Set}.X \to \mathrm{Set}$$

  then

$$\mathrm{U}^{+}\ A : Set$$

  is a universe containing (codes for) A.

- $\mathrm{U}^{+}\ A$ can be defined as well as a universe closed under

$$\begin{array}{l} f\ : \mathrm{Fam}(\mathrm{Set}) \to \mathrm{Fam}(\mathrm{Set}) \\ f\ X = A \end{array}$$

- (Usage of $\mathrm{Fam}(\mathrm{Set})$ can be avoided by Currying)

Super Universe Operator (Palmgren)

- If $A : \mathrm{Fam}(\mathrm{Set})$ then
$$\mathrm{SU}\ A : \mathit{Set}$$
is a super universe containing A, i.e. a universe closed under U^{+}.
- $\mathrm{SU}\ A$ can be defined as well as a universe closed under
$$\begin{array}{l} f : \mathrm{Fam}(\mathrm{Set}) \to \mathrm{Fam}(\mathrm{Set}) \\ f\ X = A \cup (\mathrm{U}^{+}\ X) \end{array}$$

External Mahlo Universe

- Generalise the above to allow formation of universes closed under arbitrary operators:
  - If $f : \mathrm{Fam}(\mathrm{Set}) \to \mathrm{Fam}(\mathrm{Set})$ then

$$\mathrm{U}_f : Set$$

  is a universe closed under f.

Internal Mahlo Universe

- The internal Mahlo universe V is a universe internalising closure under $\lambda f.\mathrm{U}_f$: If
$$f : \mathrm{Fam}(\mathrm{V}) \to \mathrm{Fam}(\mathrm{V})$$
then
$$\widehat{\mathrm{U}}_f : \mathrm{V}$$
is a code for a subuniverse
$$\mathrm{U}_f : \mathrm{Set}$$
of V closed under f

Illustration of the Mahlo Universe

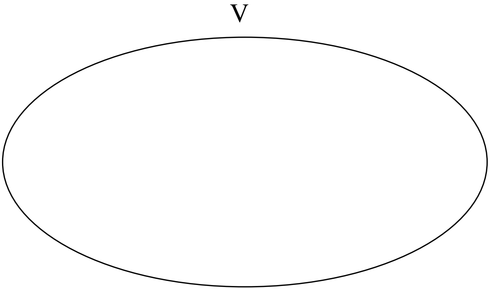


Illustration of the Mahlo Universe

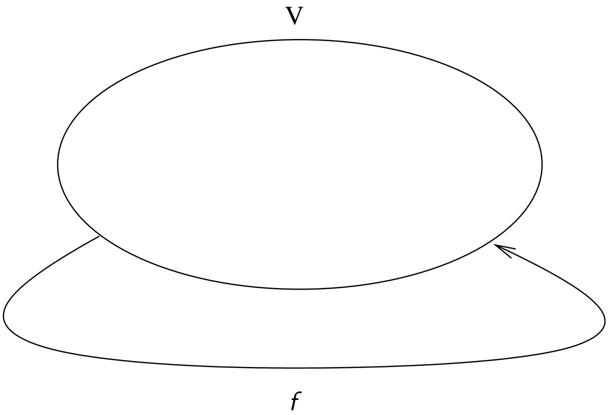


Illustration of the Mahlo Universe

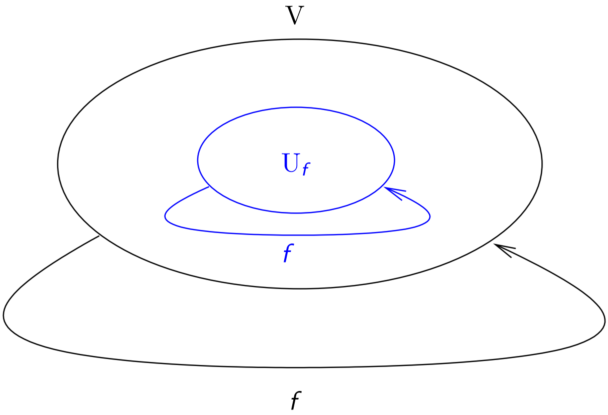
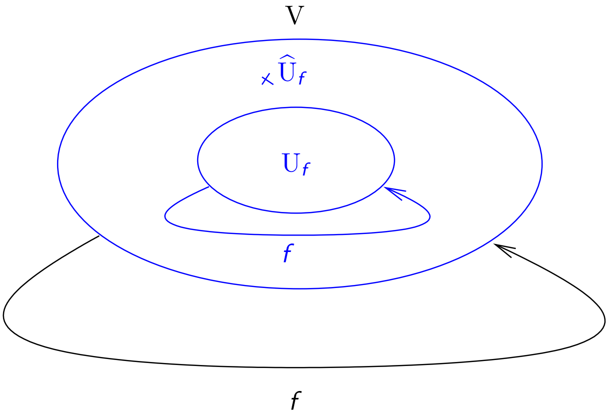


Illustration of the Mahlo Universe



Problems of Mahlo Universe

- Constructor
  $$\widehat{\mathrm{U}} : (\mathrm{Fam}(\mathrm{V}) \to \mathrm{Fam}(\mathrm{V})) \to \mathrm{V}$$
  refers to
  - the set of total functions
    $$\mathrm{Fam}(\mathrm{V}) \to \mathrm{Fam}(\mathrm{V})$$
  - which depends on the totality of V.
- So the reason for defining an element of V depends on the totality of V.

Idea for an Extended Predicative Mahlo Universe

- However, for defining U_f, only the restriction of f to $\mathrm{Fam}(\mathrm{U}_f)$ is required to be total.
  - Only local knowledge of V is needed.
- Idea: For f partial object, we try to define a subuniverse

$$\mathrm{Pre}\ \mathrm{V}\ f$$

  of V closed under f.
- If we succeed then add a code $\widehat{\mathrm{U}}_f$ for $\mathrm{Pre}\ \mathrm{V}\ f$ to V.
- Requires that we have the notion of a partial object f.

Explicit Mathematics (EM)

- Problem: In MLTT we have no reference of the set of partial objects ("potential programs", collection of terms of our language).
- In Feferman's explicit mathematics (EM) this exist.
- We will work in EM, but use syntax borrowed from type theory,
  - however write $a \in B$ instead of $a : B$.

Basics of EM

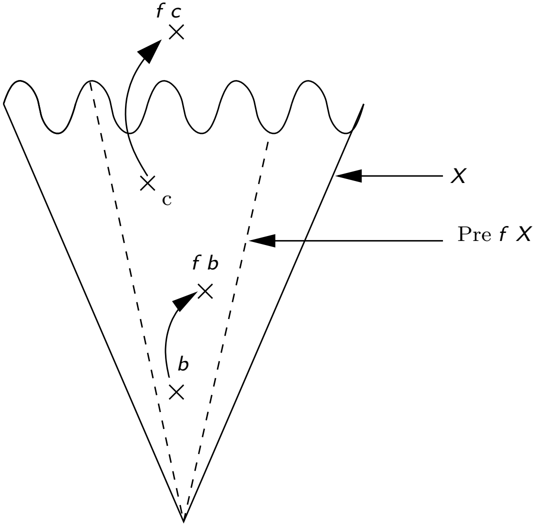
- EM more Russell-style, therefore we can have
  - $\mathrm{V} \in \mathrm{Set}$,
  - $\mathrm{V} \subset \mathrm{Set}$,
  - no need to distinguish between $\widehat{\mathrm{U}}$ and U.
- We can encode $\mathrm{Fam}(\mathrm{V})$ into V, therefore need only to consider functions

$$f : \mathrm{V} \to \mathrm{V}$$

- We define now f, $X \in \mathrm{Set}$, $X \subseteq \mathrm{Set}$

$$\mathrm{Pre}\ f\ X \in \mathrm{Set} \qquad \mathrm{Pre}\ f\ X \subseteq X$$

Pre f X



Closure of Pre f X

- $\mathrm{Pre}\ f\ X$ is closed under universe constructions, if result is in X.
- Closure under Σ (called join in EM):

$$\forall a \in \mathrm{Pre}\ f\ X.\ \forall b \in a \to \mathrm{Pre}\ f\ X.\ \Sigma\ a\ b \in X \to \Sigma\ a\ b \in \mathrm{Pre}\ f\ X$$

- $\mathrm{Pre}\ f\ X$ is closed under f, if result is in X:

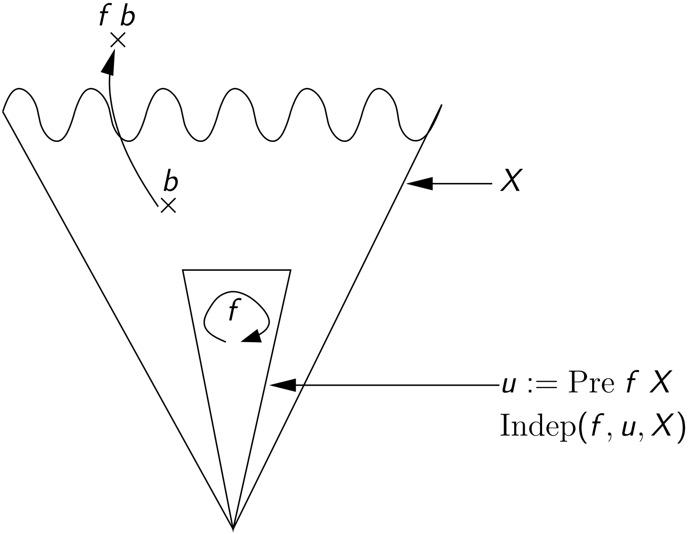
$$\forall a \in \mathrm{Pre}\ f\ X.\ f\ a \in X \to f\ a \in \mathrm{Pre}\ f\ X$$

Independence of Pre f X

- If, whenever a universe construction or f is applied to elements of $\mathrm{Pre}\ f\ X$ we get elements in X, then $\mathrm{Pre}\ f\ X$ is independent of future extensions of X.

$$\begin{aligned}\mathrm{Indep}(f, \mathrm{Pre}\ f\ X, X) := {} & (\forall a \in \mathrm{Pre}\ f\ X.\ \forall b \in a \to \mathrm{Pre}\ f\ X.\ \Sigma\ a\ b \in X) \\ & \wedge\ \cdots \\ & \wedge\ \forall a \in \mathrm{Pre}\ f\ X.\ f\ a \in X\end{aligned}$$

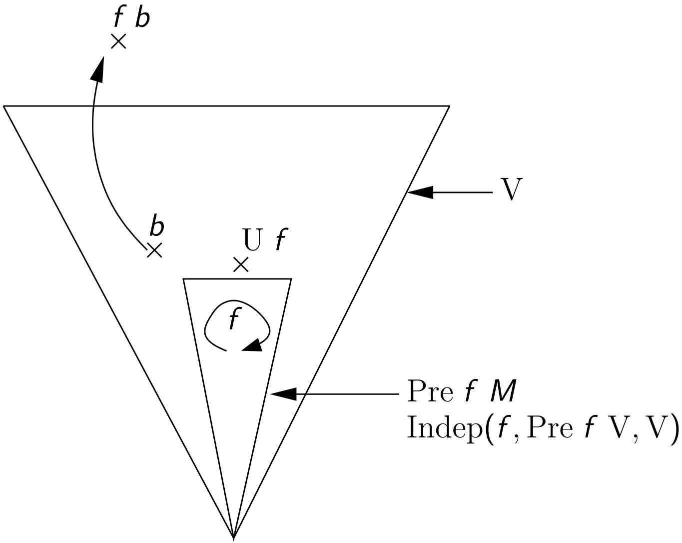
Indep A f



Introduction Rule for V

- $$\begin{aligned}\forall f.\, \mathrm{Indep}(f, \mathrm{Pre}\ f\ \mathrm{V}, \mathrm{V}) \to (&\mathrm{U}_f \in \mathrm{Set} \\ &\wedge \mathrm{U}_f =_{\mathrm{ext}} \mathrm{Pre}\ f\ \mathrm{V} \\ &\wedge \mathrm{U}_f \in \mathrm{V})\end{aligned}$$
- V admits an **elimination rule** expressing that V is the smallest universe closed under universe constructions and introduction of U_f.

Introduction Rule for V



Interpretation of Axiomatic Mahlo

- It easily follows:

$$\forall f \in \mathrm{V} \to \mathrm{V}.\ \mathrm{Indep}(f, \mathrm{Pre}\ f\ \mathrm{V}, \mathrm{V})$$

therefore

$$\forall f \in \mathrm{V} \to \mathrm{V}.\ \mathrm{U}_f \in \mathrm{V}\ \wedge\ \mathrm{Univ}(f)\ \wedge\ f \in \mathrm{U}_f \to \mathrm{U}_f$$

- So V closed under axiomatic Mahlo constructions.
- Therefore extended predicative Mahlo has at least strength of axiomatic Mahlo.

Curry-Howard isomorphism

Curry-Howard isomorphism

Intuitionistic implicational natural deduction	Lambda calculus type assignment rules
$\dfrac{}{\Gamma_1, \alpha, \Gamma_2 \vdash \alpha}\,\mathrm{Ax}$	$\dfrac{}{\Gamma_1, x : \alpha, \Gamma_2 \vdash x : \alpha}$
$\dfrac{\Gamma, \alpha \vdash \beta}{\Gamma \vdash \alpha \to \beta} \to I$	$\dfrac{\Gamma, x : \alpha \vdash t : \beta}{\Gamma \vdash \lambda x.t : \alpha \to \beta}$
$\dfrac{\Gamma \vdash \alpha \to \beta \qquad \Gamma \vdash \alpha}{\Gamma \vdash \beta} \to E$	$\dfrac{\Gamma \vdash t : \alpha \to \beta \qquad \Gamma \vdash u : \alpha}{\Gamma \vdash t\ u : \beta}$

Total Functions

- Traditionally, (mathematical) functions were always considered as total (maybe on a restricted domain).

Total Functions

- Traditionally, (mathematical) functions were always considered as total (maybe on a restricted domain).
- It was an essential feature of the Ackermann Function, the first example of a intuitively computable functions which is not primitive recursive, that it is total.

Total Functions

- Traditionally, (mathematical) functions were always considered as total (maybe on a restricted domain).
- It was an essential feature of the Ackermann Function, the first example of a intuitively computable functions which is not primitive recursive, that it is total.

- The distinguished logician David Kaplan once expressed his conviction:

"A function carries its domain."

Total Functions

- Traditionally, (mathematical) functions were always considered as total (maybe on a restricted domain).
- It was an essential feature of the Ackermann Function, the first example of a intuitively computable functions which is not primitive recursive, that it is total.

- The distinguished logician David Kaplan once expressed his conviction:

  "A function carries its domain."

- Do you agree?

Total Functions

- Traditionally, (mathematical) functions were always considered as total (maybe on a restricted domain).
- It was an essential feature of the Ackermann Function, the first example of a intuitively computable functions which is not primitive recursive, that it is total.

- The distinguished logician David Kaplan once expressed his conviction:

"A function carries its domain."

- Do you agree?

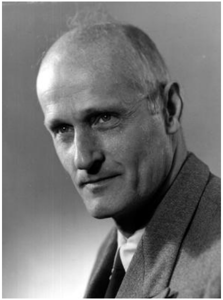
"Generic Ackermann"

If there is an inductive scheme to generate "all" total functions, one can always diagonalize over it to construct a new Ackermann-style total function outside of this class.

Partial Functions

Kleene, 1981

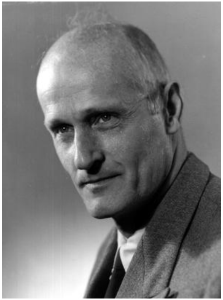
When Church proposed this thesis, I sat down to disprove it by diagonalizing out of the class of the λ-definable functions. But, quickly realizing that the diagonalisation cannot be done effectively, I became overnight a supporter of the thesis.



Partial Functions

Kleene, 1981

When Church proposed this thesis, I sat down to disprove it by diagonalizing out of the class of the λ-definable functions. But, quickly realizing that the diagonalisation cannot be done effectively, I became overnight a supporter of the thesis.



Partiality blocks Diagonalization

Partial Functions

- Partiality blocks Diagonalization ...

Partial Functions

- Partiality blocks Diagonalization ...
  - technical note: we assume a strict evaluation strategy of functions.

Partial Functions

- Partiality blocks Diagonalization ...
  - technical note: we assume a strict evaluation strategy of functions.
- ... but it comes for the price of Undecidability.

Partial Functions

Nachum Dershowitz (2005): full 'one-minute proof' of undecidability

Consider any programming language supporting programs as data [...], which has some sort of conditional (**if** . . . **then** . . . **else** . . .) and includes at least one non-terminating program (which we denote **loop**). Consider the decision problem of determining whether a program X diverges on itself, that is, $X(X) = \perp$, where $\perp$ denotes a non-halting computation. Suppose A were a program that purported to return true (T) for (exactly) all such X. Then A would perforce fail to answer correctly regarding the behavior of the following (Lisp-ish) program:

$$C(Y) := \textbf{if } A(Y) \textbf{ then } T \textbf{ else loop}(),$$

since we would be faced with the following contradiction:

$$C(C) \text{ returns } T \quad \Leftrightarrow \quad A(C) \text{ returns } T \quad \Leftrightarrow \quad C(C) \text{ diverges.}$$

Partial Functions

Nachum Dershowitz (2005): full 'one-minute proof' of undecidability

Consider any programming language supporting programs as data [...], which has some sort of conditional (**if** . . . **then** . . . **else** . . .) and includes at least one non-terminating program (which we denote **loop**). Consider the decision problem of determining whether a program X diverges on itself, that is, $X(X) = \perp$, where $\perp$ denotes a non-halting computation. Suppose A were a program that purported to return true (T) for (exactly) all such X. Then A would perforce fail to answer correctly regarding the behavior of the following (Lisp-ish) program:

$$C(Y) := \textbf{if } A(Y) \textbf{ then } T \textbf{ else } \textbf{loop}(),$$

since we would be faced with the following contradiction:

$$C(C) \text{ returns } T \quad \Leftrightarrow \quad A(C) \text{ returns } T \quad \Leftrightarrow \quad C(C) \text{ diverges.}$$

Discussion

- The introduction of partial functions sould be seen as one of the major conceptional advances in Mathematics in the XXth century.

Discussion

- The introduction of partial functions sould be seen as one of the major conceptional advances in Mathematics in the XXth century.
- As shown, the idea of partial functions is fundamental for the extended predicative construction of a Mahlo Universe.

Discussion

- The introduction of partial functions sould be seen as one of the major conceptional advances in Mathematics in the XXth century.
- As shown, the idea of partial functions is fundamental for the extended predicative construction of a Mahlo Universe.

Question

How could the Curry-Howard isomorphism be extended to partial functions?

Discussion

Question

What could be **partial proofs**?