The Type Theoretic Concept of Set

Set Theory, Classical and Constructive

Amsterdam, May 6-7, 2010

Peter Aczel
petera@cs.man.ac.uk

Manchester University,
visiting LMU, Munich
Plan of the talk

- Introduction
- The classical conception of set
- The type theoretic conception of set
- A quick look at the type theory
- Some references
Some set theories

\[ CZF \Rightarrow IZF \Rightarrow ZF \]

\[ CZF_{R,E} \Rightarrow IZF_R \quad (CZF_{R,E} + EM) \]

\[ ZF \leq \neg \neg IZF \]

\[ CZF \sim ID_1 \sim KP \ll IZF \sim ZF \]
The type-theoretic interpretations

\[ CZF \leq_{tt} (ML + \cup + \forall) \]

\[ ZF \leq_{tt} (ML + \cup + \forall + EM) \]

\[ ZF \leq \neg \neg IZF \leq_{tt} (ML + \Omega + \cup + \forall) \]
The cumulative hierarchy in ZF

- \( V = \bigcup_{\alpha \in \text{On}} V_\alpha, \)
  where \( V_\alpha = \bigcup_{\beta < \alpha} \text{Pow}(V_\beta) \) for \( \alpha \in \text{On}. \)

- \( V_0 \subseteq V_1 \subseteq \cdots V_\alpha \subseteq V_{\alpha+1} \subseteq \cdots \)

- \( V_0 = \emptyset, V_1 = \{\emptyset\}, V_{\alpha+1} = \text{Pow}(V_\alpha) \)

- \( V_\alpha = \text{the set of sets formed by stage } \alpha \)

- \( \{x \in V_\beta \mid \cdots x \cdots\} \in V_\alpha \text{ if } \beta < \alpha \)
The classical iterative conception

Zermelo, Scott, Schoenfield, Boolos, Parsons,...

- Sets are extensional.
- Sets are formed in stages, $\alpha$, out of elements formed at earlier stages.
- A set is formed by collecting together (into a whole) its elements.
- There are lots of stages:
  1. There is a stage.
  2. For each stage there is a later stage.
  3. There is a stage, $\omega$, reflecting 1,2.
  4. If $\{\alpha_i\}_{i \in I}$ is a family of stages indexed by a set $I$ then there is a stage later than all the $\alpha_i$. 
The formation of powersets

Suppose $X$ is an infinite set formed at some stage $\alpha$.
- Then each element of $X$ will have been formed at some stage before $\alpha$.
- So each subset of $X$ will have been formed at or before stage $\alpha$. But can the elements of each subset of $X$, however the subset is formed, really be collected into a whole at or before stage $\alpha$?

- So the powerset, $\text{Pow}(X)$, will be formed at any stage after $\alpha$. But can the subsets of $X$ be collected into a whole so easily?
Types, sets and classes

Any discussion concerning the concept of set must distinguish between the three distinct notions of **type**, **set** and **class**.

For example the intended universe of $ZF$ set theory is a **type**, the objects in that universe are **sets** and $\{ x \mid x \not\in x \}$ is a **class**.

The notion of **type** is perhaps best taken as a pre-mathematical philosophical notion.

The notion of **set** of $ZF$ is an iterative combinatorial notion.

The notion of **class** is a logical notion - the extension of a predicate.
A mathematical **object** is always given as an object of some **type**.

We write $a : A$ for the **judgement** that $a$ is an object of type $A$.

A **class** on a type is the extension of a **propositional function** on the type.

If $B$ is a propositional function on the type $A$ then its **extension** is the class $C = \{ x : A \mid B(x) \}$.

For $a : A$ the **proposition** that $a$ is in the class $C$ is $B(a)$.

If also $C' = \{ x : A \mid B'(x) \}$ then $(C = C')$ is the proposition

$$(\forall x : A)[B(x) \leftrightarrow B'(x)].$$
What is a set of elements of a type?

It is a collection into a whole of objects chosen from the type.

- e.g. given the type $\mathbb{N}$ of natural numbers we have sets of natural numbers such as $\{0\}, \{0, 1\}, \{0, 3, 18\}, \{\}$
- and sets $\{0, 2, 4, 6, \ldots, 92\}, \{2i \mid i < n\}$ for $n : \mathbb{N}$,
- and infinite sets such as $\{0, 2, 4, 6, \ldots\} = \{2i \mid i : \mathbb{N}\}$.
- In general we can form sets of natural numbers $\{a_i \mid i \in I\}$ with $a_i : \mathbb{N}$ for $i : I$, where $I$ is an index-type.
Index-types

I use the word index-type for something like - Bishop’s constructive notion of set, which I think is also something like - Martin-Löf’s type-theoretic notion of set or data-type and something like - the category theorists’ notion of set when they talk about a category of sets.

I need a distinct word in order to avoid confusion with the combinatorial notion of set, which is what axiomatic set theory is about.

A set is formed out of its elements. But an index-type is an object that is conceptually prior to its elements.

The index-types form a type $\mathbb{U}$. 
Sets, of elements of a type, 1

- Given a type \( A \), a set of elements of \( A \) is given by:
  1. an index-type \( I \), the index-type of the set,
  2. a function \( f : I \to A \), also thought of as a family of elements of \( A \), \( \{a_i\}_{i : I} \), where \( a_i = f(i) : A \) for \( i : I \).

- We may write the set as \((\text{set } i : I) f(i)\) or \([a_i \mid i : I]\).

- The chosen elements of the set are the \( a_i \) for \( i : I \).

- The sets of elements of \( A \) form a type \( \text{Sub}(A) \).

- \( A \) itself may be an index-type, so that we may form the set \([x \mid x : A] : \text{Sub}(A)\) of all the objects of \( A \).

- But, in general, the type \( A \) need not be an index-type.
An equality relation, \( =_A \), on a type \( A \) is an assignment of a proposition \( (b =_A c) \) to \( b, c : A \) so that the laws for an equivalence relation hold; i.e.

\[
(\forall x : A)[x =_A x], \quad (\forall x, y : A)[x =_A y \rightarrow y =_A x], \\
(\forall x, y, z : A)[x =_A y \rightarrow (y =_A z \rightarrow x =_A z)].
\]

Given an equality relation \( =_A \) on a type \( A \) we may define the membership relation \( \in_A \) and extensional equality relation \( =_{Sub(A)} \) as follows:

- If \( \alpha : \text{Sub}(A) \) is \([a_i | i : I]\) then, for \( a : A \), \((a \in_A \alpha)\) is the proposition \((\exists i : I)[a =_A a_i]\).
- If also \( \beta : \text{Sub}(A) \) is \([b_j | j : J]\) then \((\alpha =_{\text{Sub}(A)} \beta)\) is the proposition

\[
(\forall i : I)(\exists j : J)[a_i =_A b_j] \land (\forall j : J)(\exists i : I)[a_i =_A b_j].
\]
The type of iterative sets

The type $\forall$ of iterative sets is the inductive type obtained by iterating the set-of operation.

The iterative sets are generated using the following rule.

Any set-of objects in $\forall$ is an object in $\forall$

In Constructive Type Theory $\forall$ is the inductive type having the introduction rule

$$
\frac{I : \exists \ f : I \rightarrow \forall}{(\text{set } i : I) f(i) : \forall}
$$

So we have $\text{Sub}(\forall) = \forall$. 
Equality and membership on $\forall$

- We can recursively define $(\alpha =_{\forall} \beta)$ for $\alpha, \beta : V$ using the rule

$$\frac{(\forall i : I)(\exists j : J)[a_i =_{\forall} b_j] \land (\forall j : J)(\exists i : I)[a_i =_{\forall} b_j]}{\alpha =_{\forall} \beta}$$

where $\alpha = [a_i \mid i : I]$ and $\beta = [b_j \mid j : J]$.

- Also

$$\alpha \in_{\forall} \beta := (\exists j : J)(\alpha =_{\forall} b_j).$$
ZF justified in type theory

The logic of Type theory is obtained using the Propositions-as-types paradigm.

The logic using this paradigm is intuitionistic logic.

Also the type-theoretic axiom of choice holds:

$$(\forall x : A)(\exists y : B(x))R(x, y)$$

$$\rightarrow (\exists f : (\prod x : A)B(x))(\forall x : A)R(x, f(x))$$

The type $\forall$ of iterative sets provides an interpretation of CZF (Constructive ZF).

If the logic of type theory is made classical then $\forall$ provides an interpretation of ZF.

But this last step is not constructive.
The basic type theory $ML, 1$

The primitive forms of type

$$\begin{align*}
0, 1, B, \mathbb{N} & \text{ type} \\
\frac{\text{A type} \quad B(x) \text{ type } (x:A) \quad \Pi x:A)B(x), (\Sigma x:A)B(x) \text{ type}}{}
\end{align*}$$

The introduction rules

$$\begin{align*}
* & : 1 \\
t, f & : B \\
\frac{b(x):B(x) \quad (x:A)}{(\lambda x:A)b(x):(\Pi x:A)B(x)}
\end{align*}$$

$$\begin{align*}
0 & : \mathbb{N} \\
n & : \mathbb{N} \\
\frac{\text{suc}(n):\mathbb{N}}{}
\end{align*}$$

$$\begin{align*}
a & : A \\
b & : B(a) \\
\frac{\text{pair}(a,b):(\Sigma x:A)B(x)}{}
\end{align*}$$

Defined types

$$\begin{align*}
(A_1 \to A_2) & := (\Pi x:A_1)A_2 \\
(A_1 \times A_2) & := (\Sigma x:A_1)A_2
\end{align*}$$

No type dependency so far!
There are standard elimination rules for each form of type, e.g., the standard elimination rule for $\mathbb{B}$ is

$$
\begin{array}{c}
A(x) \text{ type } (x : \mathbb{B}) \quad c: \mathbb{B} \quad a_1 : A(\text{t}) \quad a_2 : A(\text{f}) \\
\text{cases}(c, a_1, a_2) : A(c) \\
\text{cases}(\text{t}, a_1, a_2) = a_1 : A(\text{t}) \\
\text{cases}(\text{f}, a_1, a_2) = a_2 : A(\text{f})
\end{array}
$$

In order to have dependent types we also use

$$
\begin{array}{c}
c: \mathbb{B} \quad A_1, A_2 \text{ type} \\
\text{Cases}(c, A_1, A_2) \text{ type} \\
\text{Cases}(\text{t}, A_1, A_2) = A_1 \\
\text{Cases}(\text{f}, A_1, A_2) = A_2
\end{array}
$$
More defined types

\[(A_1 + A_2) := (\Sigma x : \mathbb{B}) \text{Cases}(x, A_1, A_2) \quad (A_1, A_2 \text{ type})\]

\[\text{True}(c) := \text{Cases}(c, 1, 0) \quad (c : \mathbb{B})\]

So we have the derived introduction rules

\[
\begin{align*}
\text{pair}(\mathbb{t}, a) & : A_1 + A_2 \\
\text{pair}(\mathbb{f}, a) & : A_1 + A_2
\end{align*}
\]

and the equalities

\[
\text{True}(\mathbb{t}) = 1 \quad \text{True}(\mathbb{f}) = 0
\]
The basic type theory $ML_4$

Propositions-as-Types (à la Curry-Howard )
Proposition = Type

<table>
<thead>
<tr>
<th>$Prop$</th>
<th>$\bot$</th>
<th>$\top$</th>
<th>$A_1 \supset A_2$</th>
<th>$A_1 \land A_2$</th>
<th>$A_1 \lor A_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Type$</td>
<td>$0$</td>
<td>$1$</td>
<td>$A_1 \rightarrow A_2$</td>
<td>$A_1 \times A_2$</td>
<td>$A_1 + A_2$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$Prop$</th>
<th>$(\forall x : A)B(x)$</th>
<th>$(\exists x : A)B(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Type$</td>
<td>$(\Pi x : A)B(x)$</td>
<td>$(\Sigma x : A)B(x)$</td>
</tr>
</tbody>
</table>
Adding Excluded Middle

\[
\sigma(A) : (A + (A \to \emptyset))
\]

or equivalently

\[
\begin{align*}
\tau(A) & : \mathbb{B} \\
t(A) & : (\text{True}(\tau(A)) \leftrightarrow A)
\end{align*}
\]

Note: \((B \to A) : = (B \to A) \times (A \to B)\)
Adding a type \(\Omega\)

\[
\begin{align*}
\Omega \text{ type} & \quad a : \Omega \\
\Rightarrow & \quad \mathbb{T}(a) \text{ type} \\
\Omega \text{ type} & \quad \frac{}{A \text{ type}} \\
& \quad \left\{ \begin{array}{l}
\tau(A) : \Omega \\
t(A) : (\mathbb{T}(\tau(A)) \leftrightarrow A)
\end{array} \right.
\end{align*}
\]

- Think of \(\Omega\) as a type of predicative ‘propositions’, or perhaps truth values. \(\mathbb{T}(a)\) is the proposition that expresses that \(a\) is a true predicative ‘proposition’. These rules are analogous to Bertrand Russell’s Axiom of Reducibility and express that each proposition \(A\) is logically equivalent to the proposition \(\mathbb{T}(\tau(A))\).

- Excluded Middle is the extreme version of \(\Omega\) where \(\Omega = \mathbb{B}\) and \(\mathbb{T}(a) = \text{true}(a)\).
Adding a type universe $\mathbb{U}$

- The rules of $\mathbb{U}$ express that $\mathbb{U}$ is a type of types that reflects the previous type forming rules.
- Reflecting the rules of $ML$ we get

\[
\begin{align*}
  \mathbb{U} \text{ type} & \quad \frac{A : \mathbb{U}}{A \text{ type}} \quad 0, 1, B, N : \mathbb{U} & \quad \frac{A : \mathbb{U}}{B(x) : \mathbb{U}, (\Sigma x : A) B(x) : \mathbb{U}} \\
  (\Pi x : A) B(x), (\Sigma x : A) B(x) : \mathbb{U}
\end{align*}
\]

- Reflecting the $\mathbb{Ω}$ rules we get

\[
\begin{align*}
  \mathbb{Ω} : \mathbb{U} & \quad \frac{a : \mathbb{Ω}}{T(a) : \mathbb{U}}
\end{align*}
\]
The type-theoretic interpretations

\[ \text{CZF} \leq_{tt} (ML + \cup + \forall) \]

\[ \text{ZF} \leq_{tt} (ML + \cup + \forall + EM) \]

\[ \text{ZF} \leq_{	ext{not}} \text{IZF} \leq_{tt} (ML + \emptyset + \cup + \forall) \]
Martin-Löf’s Philosophy

The type-theoretic interpretation of CST