The Strength of Martin-Löf Type Theory with the Logical Framework

(Work in Progress)

Anton Setzer
Swansea University, UK

1. Motivation.
2. Models of $ML_1^W$ without LF.
3. Models of $ML_1^W$ with LF.
4. Conclusion.
1. Motivation

Logical framework (LF) added to Martin-Löf Type Theory (MLTT) in order to provide an infrastructure for defining set constructions.

LF obtained by adding

- one type level Type on top of the standard type level Set,
- s.t. Set ∪ {Set} ⊆ Type,
- and by closing both Set and Type under the dependent function type

\[(x : A) \rightarrow B\]

and (possibly) the dependent product

\[(x : A) \times B\]
Logical Framework

Type

$\text{Set} \rightarrow \text{Set}$

$\text{Set} \times \rightarrow \text{Set}$

$\text{N} \rightarrow \text{N}$

$\text{N} \rightarrow \text{Set}$

Anton Setzer (Swansea): The Strength of Martin-Löf Type Theory with the Logical Framework
Simplification by LF

Without the LF, elimination for \( \mathbb{N} \) is given by

\[
\begin{align*}
\Gamma, x : \mathbb{N} &\Rightarrow C[x] : \text{Set} \\
\Gamma &\Rightarrow \text{step}0 : C[0] \\
\Gamma, x : \mathbb{N}, y : C[x] &\Rightarrow \text{step}S[x, y] : C[S(x)] \\
\Gamma &\Rightarrow n : \mathbb{N} \\
\Gamma &\Rightarrow P(\text{step}0, (x, y)\text{step}S[x, y], n) : C[n]
\end{align*}
\]

together with an equality version of it,

With the LF, it is given by

\[
\begin{align*}
P : (C : \mathbb{N} \rightarrow \text{Set}) &\rightarrow (\text{step}0 : C \ 0) \\
&\rightarrow (\text{step}S : (n : \mathbb{N}) \rightarrow C \ n \rightarrow C \ (S \ n)) \\
&\rightarrow (n : \mathbb{N}) \rightarrow C \ n
\end{align*}
\]
Most theorem provers for dependent type theory based on the LF.

In order to simplify our interpretation in $\text{KPI}^+$ we use a version where we have

$$
\frac{A : \text{Set}}{\mathcal{E}l(A) : \text{Type}}
$$

rather than

$$
\frac{A : \text{Set}}{A : \text{Type}}
$$
Problem

- LF amounts to **adding a universe** (namely $\text{Set}$) to type theory.
  - Why doesn’t this increase its strength?
- Because of this we **avoided** until now the LF in proof theoretic analyses of extensions of MLTT.
- **Goal**: Extend the methodology of proof theoretic analyses so that LF is included.
  - **Aim**: show $|\text{ML}_1 W + \text{LF}| = |\text{ML}_1 W|$ similarly for other variants of MLTT.
2. Models of $\text{ML}_1 W$ without LF

Let $\text{CTerm} = \text{set of closed terms}$.

Environment $\eta = \text{finite functions } \text{Var} \rightarrow \text{CTerm}$.

Model of $\text{ML}_1 W$ without LF introduced by defining a PER model in $\text{KPI}^+ := \text{KPI}^r + \exists I. \text{"I inaccessible"}$.

For certain terms $A$ corresponding to set expressions we define for environments $\eta$ s.t. $\text{FV}(A) \subseteq \text{dom}(\eta)$

$$[A]_\eta \subseteq \text{CTerm}^2$$

Then we show by induction on derivations that, if

$$\text{ML}_1 W \vdash \Gamma \Rightarrow \theta$$

then

$$\text{KPI}^+ \vdash \text{Correct}(\Gamma \Rightarrow \theta)$$
Models of $\text{ML}_W$ (no LF)

- For simplicity we treat $[A]$ as a set of terms rather than a set of pairs of terms.
- For instance

\[
\begin{align*}
\text{Correct}(x : A \Rightarrow B : \text{Set}) & := \\
& \quad \text{Correct}(\emptyset \Rightarrow A : \text{Set}) \\
& \quad \land \forall r \in [A].\text{PER}(\lbrack B \rbrack_{x \leftarrow r}) \land \text{Closure}(\lbrack B \rbrack_{x \leftarrow r})
\end{align*}
\]

\[
\begin{align*}
\text{Correct}(x : A \Rightarrow b : B) & := \\
& \quad \text{Correct}(x : A \Rightarrow B : \text{Set}) \\
& \quad \land \forall r \in [A].b[x := r] \in [B]_{x \leftarrow r}
\end{align*}
\]
3. Models of $\text{ML}_1 W + LF$

- With the LF the judgement $A : \text{Set}$ is no longer special.
  - $A : \text{Set}$ has the same status as $a : A$.
  - Instead “$A : \text{Type}$” is special.

- We need to define $[A]_\eta$ for type expressions rather than set-expressions.

- Correctness statements as before, but with $\text{Set}$ replaced by $\text{Type}$.

- Need to define $[\text{Set}]$. 
Interpretation of Elements of Type

- **Idea:** \([\text{Set}] = \bigcup_{\alpha \in \text{Ord}} \text{Set}^\alpha\) which is a proper class.

- **Problem:** If we interpret

\[ [\mathbb{N} \to \text{Set}] := \{a \mid \forall n \in [\mathbb{N}]. a \ n \in [\text{Set}]\} \]

we will interpret large elimination, which increases the proof theoretical strength.

- Large elimination means that for \(C := \text{W}x : A.B\) or \(C := \mathbb{N}\) we can define \(f : C \to D\) by induction over \(C\) for any \(D : \text{type}\).

- Small elimination means that we require \(D : \text{Set}\).
Interpretation of Elements of Type

We need to make sure that

\[ [\mathbb{N} \to \text{Set}] = \bigcup_{n \in \mathbb{N}} ([\mathbb{N}] \to \text{Set}^{\kappa_n}) \]

(where \( \kappa_n \) = \( n \)th admissible above \( I \)).

For this we define

\[ [\mathbb{N} \to \text{Set}]^n = [\mathbb{N}] \to \text{Set}^{\kappa_n} \]
Interpretation of Elements of Type

What is $[\text{Set} \to \text{Set}]$?

Cannot restrict it to $\text{Set}^\kappa_n \to \text{Set}^\kappa_n$.

E.g. for any $n \in \mathbb{N}$ we have

$$\lambda x. (Wy : E l(x).x) \in \text{Set}^\kappa_n \xrightarrow{} \text{Set}^{\kappa_n+1}.$$

We can define $[\text{Set} \to \text{Set}]^e$ for any $e :: \text{nat} \to \text{nat}$ e.g.

$$\lambda x. (Wy : E l(x).x) \in [\text{Set} \to \text{Set}]^{\lambda n.n+1}$$

$[(\text{Set} \to \text{Set}) \to \text{Set}]^e$ defined for $e :: (\text{nat} \to \text{nat}) \to \text{nat}$. 
Functionals of Finite Types

Let the finite types be $\epsilon, \text{nat}, \alpha \rightarrow \beta, \alpha \times \beta$.

Let $e :: \alpha$ mean that $e$ is a Kleene index for a functional of finite type $\alpha$.

$\epsilon$ is the trivial type (contains only element 0).

We can contract $\epsilon \times \alpha, \alpha \times \epsilon, \epsilon \rightarrow \alpha$ to $\alpha$ and $\alpha \rightarrow \epsilon$ to $\epsilon$.

$\text{Btype}(A)$ is defined as a finite type as follows:

$\text{Btype}(\text{Set}) := \text{nat}$.

$\text{Btype}(\text{El}(t)) := \epsilon$.

$\text{Btype}((x : A) \rightarrow B) := \text{Btype}(A) \times \text{Btype}(B)$. 

Anton Setzer (Swansea): The Strength of Martin-Löf Type Theory with the Logical Framework
We need to guarantee as well that if e.g.

\[ \text{ML}_1 W \vdash x : A, y : B \Rightarrow \text{Context} \]

then \( \llbracket A \rrbracket \downarrow \land \forall a \in \llbracket A \rrbracket. \llbracket B \rrbracket_{[x \mapsto a]} \downarrow \).

This will require that certain \( \alpha = \kappa_n \) do exist.

E.g. \( \llbracket \mathcal{E}l(t) \rrbracket \downarrow \) if \( t \in \text{Set}^{\kappa_n} \).

Ctype\((A)\) is defined as a sequence of finite types:

- \( \text{Ctype(Set)} := \emptyset \).
- \( \text{Ctype}(\mathcal{E}l(t)) := \text{nat} \).
- \( \text{Ctype}((x : A) \rightarrow \times B) \)
  \[ := \text{Ctype}(A) \rightarrow \text{Ctype}(B) \). \]
We define for $\vec{g} :: \text{Ctype}(A)$ whether $\llbracket A \rrbracket \vec{g} \downarrow$:

- $\llbracket \text{Set} \rrbracket^\emptyset \downarrow := \top$.
- $\llbracket \mathcal{E}l(t) \rrbracket^n \downarrow := \exists \alpha. (\alpha = \kappa_n \land t \in \text{Set}^\alpha)$.
- $\llbracket (x : A) \rightarrow \times B \rrbracket^f;\vec{g} \downarrow := \llbracket A \rrbracket^f \downarrow \land \forall h :: \text{Btype}(A). \forall a \in \llbracket A \rrbracket^f;h. \llbracket B \rrbracket^{\vec{g}(h)}_{x \mapsto a} \downarrow$. 
We define $[A]^{\vec{g},h}$ for $\vec{g} :: Ctype(A)$, $h :: Btype(A)$:

- $[\text{Set}]^{\emptyset;n}$
  $$:= \{a \mid \exists \alpha. \alpha = \kappa_n \land a \in \text{Set}^\alpha\}.$$

- $[\text{El}(t)]^{n;e}$
  $$:= \{a \mid \exists \alpha. \alpha = \kappa_n \land a \in \text{El}^\alpha(t)\}.$$

- $[(x : A) \to B]^{\vec{f},\vec{g};h}$
  $$:= \{a \mid \forall k :: Btype(A). \forall b \in [A]^{\vec{f};k}. a \ b \in [B]^{\vec{g}(k);h \ k}\}.$$

- $[(x : A) \times B]^{\vec{f},\vec{g};h}$
  $$:= \{a \mid \pi_0(a) \in [A]^{\vec{f};\pi_0(h)} \land \pi_1(a) \in [B]^{\vec{g}(\pi_0(h));\pi_1(h)}\}.$$

Example

\[ [\text{Set} \to \text{Set}]^0; f \]
\[
:= \{ a \mid \forall k :: \text{nat}. \forall b.(\exists \alpha. \alpha = \kappa_n \land b \in \text{Set}^\alpha) \
\quad \to (\exists \alpha. \alpha = \kappa_f n \land a b \in \text{Set}^\alpha) \} \]

Especially

\[
\lambda x. (W y : \text{El}(x).x) \in [\text{Set} \to \text{Set}]^0; \lambda n. n+1
\]
Correct($\Gamma \Rightarrow \theta$)

- Btype($x_1 : A_1, \ldots, x_n : A_n \Rightarrow A : \text{Type}$)
  
  := Btype((x_1 : A_1) \to \cdots \to (x_n : A_n) \to A : \text{Type}).

- Similarly for Ctype.

- For $\vec{f}, \vec{g} :: Ctype(\Gamma \Rightarrow A : \text{Type})$, we define
  
  Correct($\Gamma \Rightarrow A : \text{Type}$)$^{\vec{f}, \vec{g}}$ :=
  
  Correct($\Gamma \Rightarrow \text{Context}$)$^{\vec{f}}$
  
  $\land \forall \vec{k} :: \text{Btype}(\Gamma). \forall \vec{r} \in [\Gamma]^{\vec{f}, \vec{k}}.$
  
  $\lbrack A \rbrack^{\vec{g}(\vec{k})} \downarrow$
  
  $\land \forall l :: \text{Btype}(A). \text{PER}([A]^{\vec{g}(\vec{k})}; l) \land \text{Closure}([A]^{\vec{g}(\vec{k})}; l)$. 
For $\vec{f}, \vec{g} :: \text{Ctype}(\Gamma \Rightarrow A : \text{Type})$, we define

\[
\text{Correct}(\Gamma \Rightarrow a : A)_{\vec{f}, \vec{g}; l} := \\
\text{Correct}(\Gamma \Rightarrow A : \text{Type})_{\vec{f}}
\]

and

\[
\land \forall \vec{k} :: \text{Btype}(\Gamma). \forall \vec{r} \in [\Gamma]_{\vec{f}; \vec{k}}.
\]

\[a[\vec{x} \mapsto \vec{r}] \in [A]_{\vec{g}(\vec{k}); l \vec{k}}.\]

Now prove by Meta-induction on the derivation that if

\[\text{ML}_1 W \vdash \Gamma \Rightarrow \theta\]

then there Meta-exist $\vec{f}, g$ s.t.

\[\text{KPI}^+ \vdash \text{Correct}(\Gamma \Rightarrow \theta)_{\vec{f}; g}\]
Conclusion

- LF doesn’t add strength, but very difficult to deal with it (unless one treats it as a proper universe).
- From a foundational point of view this means that the logical framework adds a lot of syntactic complexity to type theory (meaning explanation).
- $\Rightarrow$ LF is too “strong” for just providing an infrastructure for defining type theories.
- Approach by P. Aczel to provide a “weaker” form of the LF.
- Methodology for upper bounds seems to work for many variants of MLTT.