A Generalized Logical Framework

András Kovács[1], Christian Sattler[1]

[1]University of Gothenburg & Chalmers University of Technology

17 Apr 2025, EuroProofNet WG6 meeting, Genoa

Overview

1. Two-level type theories (2LTT):
   - metaprogramming over a **single model** of a **single type theory**.

Overview

1. Two-level type theories (2LTT):
   - metaprogramming over a **single model** of a **single type theory**.
   - the chosen model is defined **outside the system**.

Overview

1. Two-level type theories (2LTT):
   - metaprogramming over a **single model** of a **single type theory**.
   - the chosen model is defined **outside the system**.
   - **only a second-order** view on the model.

Overview

1. Two-level type theories (2LTT):
   - metaprogramming over a **single model** of a **single type theory**.
   - the chosen model is defined **outside the system**.
   - **only a second-order** view on the model.
2. Generalized logical framework (GLF):
   - metaprogramming over **any number of models** of **any number of type theories**.
   - models are defined **inside the system**.
   - both a **first-order** and a **second-order** view on each model.

Overview

1. Two-level type theories (2LTT):
   - metaprogramming over a **single model** of a **single type theory**.
   - the chosen model is defined **outside the system**.
   - **only a second-order** view on the model.
2. Generalized logical framework (GLF):
   - metaprogramming over **any number of models** of **any number of type theories**.
   - models are defined **inside the system**.
   - both a **first-order** and a **second-order** view on each model.

In this talk:

1. A syntax of GLF + examples + increasing amount of syntactic sugar.
2. A short overview of semantics.

GLF basic universes & type formers

Set		An universe that supports ETT.
Base	: **Set**	Sort of "base categories".
1	: **Base**	The terminal category as a base category.
PSh	: **Base** $\to$ **Set**	Universes of presheaves. Cumulativity: $\mathsf{PSh}_i \subseteq \mathsf{Set}$. Supports ETT.
		We can only eliminate from PSh_i to PSh_i.
Cat$_i$	: **PSh**$_i$	$:=$ *type of categories in* PSh_i
In	: **Cat**$_i$ $\to$ **Set**	"Permission token" for working in presheaves over some $C : \mathsf{Cat}_i$.
base	: **In** $C \to$ **Base**	"Using the permission".

We use type-in-type everywhere for simplicity, i.e. $\mathsf{Set} : \mathsf{Set}$ and $\mathsf{PSh}_i : \mathsf{PSh}_i$.

Basic things we can do

$$\mathsf{Set} : \mathsf{Set} \qquad \mathsf{Base} : \mathsf{Set} \qquad \mathbf{1} : \mathsf{Base} \qquad \mathsf{PSh} : \mathsf{Base} \to \mathsf{Set}$$

$$\mathsf{Cat}_i : \mathsf{PSh}_i := \textit{type of cats in } \mathsf{PSh}_i \qquad \mathsf{In} : \mathsf{Cat}_i \to \mathsf{Set} \qquad \mathsf{base} : \mathsf{In}\,\mathbb{C} \to \mathsf{Base}$$

PSh_1 is a universe supporting ETT, semantically a universe of sets.

We can define some $\mathbb{C} : \mathsf{Cat}_1$, where $\mathsf{Obj}(\mathbb{C}) : \mathsf{PSh}_1$.

Now, under the assumption of $i : \mathsf{In}\,\mathbb{C}$, we can form the universe $\mathsf{PSh}_{(\mathsf{base}\,i)}$, which is semantically the universe of presheaves over $\mathbb{C}$.

Syntax sugar: we'll omit base in the following.

At this point, we have no interesting interaction between PSh_1 and PSh_i.

Example: embedding pure lambda calculus

A **second-order model of pure LC** in PSh_i consists of:

$$
\begin{aligned}
\mathsf{Tm} &: \mathsf{PSh}_i \\
\mathsf{lam} &: (\mathsf{Tm} \to \mathsf{Tm}) \to \mathsf{Tm} \\
\_\$\_ &: \mathsf{Tm} \to \mathsf{Tm} \to \mathsf{Tm} \\
\beta \quad &: \mathsf{lam}\, f \,\$\, t = f\ t \\
\eta \quad &: \mathsf{lam}\,(\lambda x.\, t \,\$\, x) = t
\end{aligned}
$$

We define $\mathsf{SMod}_i : \mathsf{PSh}_i$ as the above Σ-type.

Example: embedding pure lambda calculus

A **first-order model of pure LC** consists of:

- A category of contexts and substitutions with $\mathsf{Con} : \mathsf{PSh}_i$, $\mathsf{Sub} : \mathsf{Con} \to \mathsf{Con} \to \mathsf{PSh}_i$ and terminal object $\bullet$.
- $\mathsf{Tm} : \mathsf{Con} \to \mathsf{PSh}_i$, plus a term substitution operation.
- A context extension operation $-\triangleright : \mathsf{Con} \to \mathsf{Con}$ such that $\mathsf{Sub}\,\Gamma\,(\Delta\triangleright) \simeq \mathsf{Sub}\,\Gamma \times \mathsf{Tm}\,\Gamma$.
- A natural isomorphism $\mathsf{Tm}\,(\Gamma\triangleright) \simeq \mathsf{Tm}\,\Gamma$ whose components are λ and application.

We define $\mathsf{FMod}_i : \mathsf{PSh}_i$ as the above Σ-type.

FMod is mechanically derivable from SMod.[1]

[1] Ambrus Kaposi & Szumi Xie: *Second-Order Generalised Algebraic Theories*.

Example: embedding pure lambda calculus