A Generalized Logical Framework

András Kovács[1] and Christian Sattler[2]

Chalmers University of Technology & University of Gothenburg, Sweden
[1] andrask@chalmers.se [2] sattler@chalmers.se

Logical frameworks (LFs [3]) and the closely related two-level type theories (2LTTs [1]) let us work in a mixed syntax of a metatheory and a chosen object theory. Here, we have a second-order view on the object theory, where contexts, variables and substitutions are implicit, and binders are represented as meta-level functions. There are some well-known limitations to LFs. First, we have to pick a model of the object theory externally. Second, since we only have a second-order view on that model, many constructions cannot be expressed; for example, the induction principle for the syntax of an object theory requires a notion of first-order model, where contexts and substitutions are explicit. Various ways have been described to make logical frameworks more expressive by extending them with modalities (e.g. [9, 4, 8, 7]). In the current work we describe an LF with the following features:

- We can work with multiple models of multiple object theories at the same time. By "theory" we mean a second-order generalized algebraic theory (SOGAT [10, 6]); this includes all type theories and programming languages that only use structural binders.
- We have both an "external" first-order view and an "internal" second-order view on each model, and we can freely switch between perspectives. All models of object theories are defined internally in the LF.
- The LF is fully structural as a type theory; no substructural modalities are used.

The Generalized Logical Framework (GLF). The basic structure is as follows.

- We have a universe U closed under the type formers of extensional type theory.
- We have $\mathsf{Base} : \mathsf{U}$, $1 : \mathsf{Base}$ and $\mathsf{PSh} : \mathsf{Base} \to \mathsf{U}$, such that each PSh_i is a universe that supports ETT. We have cumulativity: $\mathsf{PSh}_i \subseteq \mathsf{U}$. We can only eliminate from PSh_i to PSh_i. *Semantically, each* PSh_i *is a universe of presheaves over some base category represented by* i*. The terminal category is* 1*.*
- For convenience, we assume type-in-type everywhere, so that $\mathsf{U} : \mathsf{U}$ and $\mathsf{PSh}_i : \mathsf{PSh}_i$.
- We define $\mathsf{Cat}_i : \mathsf{PSh}_i$ as the type of categories internally to PSh_i, where types of objects and morphisms are in PSh_i. We have $\mathsf{In} : \{i : \mathsf{Base}\} \to \mathsf{Cat}_i \to \mathsf{U}$ and $\mathsf{base} : \mathsf{In}\, C \to \mathsf{Base}$. *Informally,* $\mathsf{In}\, C$ *is a type of "permission tokens" for working in a presheaf universe.*

Let us look at a basic scenario in GLF. In the empty context, we have PSh_1 as a universe of sets. Internally to PSh_1, we can define some $C : \mathsf{Cat}_1$. Now, under the assumption of $i : \mathsf{In}\, C$, we can form $\mathsf{PSh}_{(\mathsf{base}\, i)}$ as the universe of presheaves over C. Then, we may define another category D inside $\mathsf{PSh}_{(\mathsf{base}\, i)}$, and get a new universe PSh_j under the assumption of $j : \mathsf{In}\, D$. Hence, GLF at its heart is a type theory for *iterated internal categories and presheaves.*

At this point there is no interesting interaction between presheaf universes, so we proceed to specify some. Recall that the standard semantics of 2LTTs is in presheaves over a chosen model of an object theory, where a model consists of a category of contexts plus extra structure. In GLF, if we define a model M of a theory in some PSh_i universe, we would like to have a "view"

on that model internally to presheaves over M, and also a way to move between the internal view and the external M. Hence we specify that for any first-order model M of a second-order generalized algebraic theory T, we have

1. A second-order model of T internally to PSh_i, assuming $i : \mathsf{In}\, M$[1]. We write this second-order model as S_i.
2. *Yoneda embedding* as a certain family of maps from M to S_j.

We look at the example where T is pure lambda calculus. A second-order model of pure LC in PSh_i is simply $\mathsf{Tm} : \mathsf{PSh}_i$ together with an isomorphism $\mathsf{Tm} \simeq (\mathsf{Tm} \to \mathsf{Tm})$ for abstraction and application. We write lam for the former and $-\$-$ for the latter. A first-order model is a unityped category with families [2], where we write $\mathsf{Con} : \mathsf{PSh}_i$ for the type of contexts, $\mathsf{Sub} : \mathsf{Con} \to \mathsf{Con} \to \mathsf{PSh}_i$ for substitutions, $\mathsf{Tm} : \mathsf{Con} \to \mathsf{PSh}_i$ for terms, $\Gamma\triangleright : \mathsf{Con}$ for the extension of $\Gamma : \mathsf{Con}$ with a binding, and we have a natural isomorphism $\mathsf{Tm}\,\Gamma \simeq \mathsf{Tm}\,(\Gamma\triangleright)$ to represent abstraction and application. Now, assuming a first-order model M in PSh_i and $j : \mathsf{PSh}_{(\mathsf{base}\, i)}$, we can use the second-order view when working inside PSh_j. Let us define the Y-combinator as an example:

$$\begin{aligned}&\mathsf{YC} : \mathsf{Tm}_{\mathsf{S}_j}\\&\mathsf{YC} := \mathsf{lam}_{\mathsf{S}_j}(\lambda\, f.\,(\mathsf{lam}_{\mathsf{S}_j}(\lambda x.\, x\,\$_{\mathsf{S}_j}\, x))\,\$_{\mathsf{S}_j}\,(\mathsf{lam}_{\mathsf{S}_j}(\lambda x.\, f\,\$_{\mathsf{S}_j}\,(x\,\$_{\mathsf{S}_j}\, x))))\end{aligned}$$

With a reasonable amount of sugar, we may write $\mathsf{YC} := \mathsf{lam}\, f.\,(\mathsf{lam}\, x.\, x\, x)\,(\mathsf{lam}\, x.\, f\,(x\, x))$. In other words, PSh_j now is effectively a presentation of a two-level type theory over pure LC where S_j constitutes the inner level and the ETT type formers in PSh_j constitute the outer level. Since we specify S for every second-order algebraic theory, all 2LTTs are syntactic fragments of GLF. Next, Yoneda embedding for pure LC is as follows:

$$\begin{aligned}&\mathsf{Y} : \mathsf{Con}_M &&\to ((j : \mathsf{In}_M) \to \mathsf{PSh}_j)\\&\mathsf{Y} : \mathsf{Sub}_M\,\Gamma\,\Delta &&\simeq ((j : \mathsf{In}_M) \to \mathsf{Y}\,\Gamma\, j \to \mathsf{Y}\,\Delta\, j)\\&\mathsf{Y} : \mathsf{Tm}_M\,\Gamma &&\simeq ((j : \mathsf{In}_M) \to \mathsf{Y}\,\Gamma\, j \to \mathsf{Tm}_{\mathsf{S}_j})\end{aligned}$$

such that Y preserves empty context and context extension, so $\mathsf{Y}\bullet j \simeq \top$ and $\mathsf{Y}\,(\Gamma\triangleright)\, j \simeq \mathsf{Y}\,\Gamma\, j \times \mathsf{Tm}_{\mathsf{S}_j}$, and Y preserves all other structure strictly. *Notation:* we write Λ for inverses of Y. Now, Y and Λ allow ad-hoc switching between perspectives. Let's redefine some operations in M:

$$\begin{aligned}&\mathsf{id} : \mathsf{Sub}_M\,\Gamma\,\Gamma && \mathsf{comp} : \mathsf{Sub}_M\,\Delta\,\Theta \to \mathsf{Sub}_M\,\Gamma\,\Delta \to \mathsf{Sub}_M\,\Gamma\,\Theta\\&\mathsf{id} := \Lambda\,(\lambda\, j\,\gamma.\,\gamma) && \mathsf{comp}\,\sigma\,\delta := \Lambda\,(\lambda\, j\,\gamma.\,\mathsf{Y}\,\sigma\,(\mathsf{Y}\,\delta\,\gamma\, j)\, j)\end{aligned}$$

With reasonable amount of sugar, this might look like

$$\mathsf{id} := \Lambda\,\gamma.\,\gamma \qquad \mathsf{comp}\,\sigma\,\delta := \Lambda\,\gamma.\,\mathsf{Y}\,\sigma\,(\mathsf{Y}\,\delta\,\gamma)$$

Or, making Y implicit, we may even write $\mathsf{comp}\,\sigma\,\delta := \Lambda\,\gamma.\,\sigma\,(\delta\,\gamma)$. We can develop this into a "second-order notation" for object theories, which is nicely readable and can be rigorously elaborated into annotated GLF operations. We only give here a glimpse of what this notation could look like. The example below comes from a model construction involving models of MLTT

[1] Here we implicitly cast M to its underlying category.

as CwFs, which looks fairly obtuse with explicit substitutions and De Bruijn indices [5, Section 5]:

$$\begin{aligned}
&\mathsf{Con}^\circ\,\Gamma && := \mathsf{Ty}\,(F\,\Gamma)\\
&\mathsf{Ty}^\circ\,\Gamma^\circ\,A && := \mathsf{Ty}\,(F\,\Gamma \triangleright \Gamma^\circ \triangleright F\,A[\mathsf{p}])\\
&\mathsf{Tm}^\circ\,\Gamma^\circ\,A^\circ\,t && := \mathsf{Tm}\,(F\,\Gamma \triangleright \Gamma^\circ)\,(A^\circ[\mathsf{id},\, F\,t[\mathsf{p}]))\\
&\Gamma^\circ \triangleright^\circ A^\circ && := \Sigma(\Gamma^\circ[\mathsf{p} \circ F_{\triangleright.1}])(A^\circ[\mathsf{p} \circ F_{\triangleright.1} \circ \mathsf{p},\, \mathsf{q},\, \mathsf{q}[F_{\triangleright.1} \circ \mathsf{p}]])\\
&\ldots
\end{aligned}$$

but which looks reasonable in sugary GLF notation:

$$\begin{aligned}
&\mathsf{Con}^\circ\,\Gamma && := \mathsf{Ty}\,(\gamma : F\,\Gamma)\\
&\mathsf{Ty}^\circ\,\Gamma^\circ\,A && := \mathsf{Ty}\,(\gamma : F\,\Gamma,\, \gamma^\circ : \Gamma^\circ\,\gamma,\, \alpha : F\,A\,\gamma)\\
&\mathsf{Tm}^\circ\,\Gamma^\circ\,A^\circ\,t && := \mathsf{Tm}\,(\gamma : F\,\Gamma,\, \gamma^\circ : \Gamma^\circ\,\gamma)\,(A^\circ\,(\gamma,\, \gamma^\circ,\, F\,t\,\gamma))\\
&\Gamma^\circ \triangleright^\circ A^\circ && := \Lambda\,(F_{\triangleright.2}(\gamma,\, \alpha)).\,\Sigma(\gamma^\circ : \Gamma^\circ\,\gamma) \times A^\circ\,(\gamma,\, \gamma^\circ,\, \alpha)\\
&\ldots
\end{aligned}$$

Sketch of the semantics. First, we give a short motivation. In the semantics, each PSh_i should be an universe of internal presheaves over an internal category. Clearly the semantics should involve categories, but there are well-known complications with the category of categories: a) there is no general Π type b) Π-types of presheaves and universes of presheaves are not stable under reindexing by arbitrary functors. The former issue could be addressed by having a "directed type theory", while the latter could be addressed with modalities. In the case of GLF we don't need either of these solutions. The reason is that we can't do any interesting categorical reasoning in GLF, and Base and In are used purely for managing internal/external languages, and it suffices to have enough semantic structure to represent the internal/external shifts.

The model of GLF is constructed in two steps. First, we give a model for the theory that has PSh, Base and In as sorts but does not support U, and then take presheaves over that model to obtain a model of a 2LTT where U represents the outer layer. In the inner model, we start with an inductive definition of certain *trees of categories*:

$$\begin{aligned}
&\mathbf{data}\,\mathsf{Tree}\,(B : \mathsf{Cat}) : \mathsf{Set}\,\mathbf{where}\\
&\quad \mathsf{node} : (\Gamma : \mathsf{PSh}\,B)(n : \mathbb{N})(C : \mathsf{Fin}\,n \to \mathsf{Fib}\,(B \triangleright \mathsf{Disc}\,\Gamma))\\
&\qquad\quad \to ((i : \mathsf{Fin}\,n) \to \mathsf{Tree}\,(B \triangleright \mathsf{Disc}\,\Gamma \triangleright C\,i))\\
&\qquad\quad \to \mathsf{Tree}\,B
\end{aligned}$$

Here, PSh means presheaves in sets, Fib is cartesian fibrations, Disc creates a discrete fibration from a presheaf and $-\triangleright-$ takes the total category of a fibration. Now, the objects of the semantic base category are elements of $\mathsf{Tree}\,1$, and morphisms between trees are level-wise natural transformations between the Γ components together with $\mathsf{Fin}\,n \to \mathsf{Fin}\,m$ renamings of subtree indices. The non-discrete Fib components are preserved by morphisms.

In a nutshell, each node represents a presheaf universe and each edge represents an internal/external switch. A semantic element of Base selects a node of a tree, while an In is an index that points to a subtree of a node. A semantic PSh is a dependent presheaf over a Γ in a given node. Extending a context with an In binding adds a new empty subtree to a given node, and extending a context with a presheaf variable extends the Γ presheaf in a node with a dependent presheaf.

References

[1] Danil Annenkov, Paolo Capriotti, Nicolai Kraus, and Christian Sattler. Two-level type theory and applications. *ArXiv e-prints*, may 2019.

[2] Simon Castellan, Pierre Clairambault, and Peter Dybjer. Categories with families: Unityped, simply typed, and dependently typed. *CoRR*, abs/1904.00827, 2019.

[3] Robert Harper, Furio Honsell, and Gordon Plotkin. A framework for defining logics. *J. ACM*, 40(1):143–184, January 1993.

[4] Martin Hofmann. Semantical analysis of higher-order abstract syntax. In *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2-5, 1999*, pages 204–213. IEEE Computer Society, 1999.

[5] Ambrus Kaposi, Simon Huber, and Christian Sattler. Gluing for type theory. In Herman Geuvers, editor, *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24-30, 2019, Dortmund, Germany*, volume 131 of *LIPIcs*, pages 25:1–25:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[6] Ambrus Kaposi and Szumi Xie. Second-order generalised algebraic theories: Signatures and first-order semantics. In Jakob Rehof, editor, *9th International Conference on Formal Structures for Computation and Deduction, FSCD 2024, July 10-13, 2024, Tallinn, Estonia*, volume 299 of *LIPIcs*, pages 10:1–10:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[7] Ian Orton and Andrew M. Pitts. Axioms for Modelling Cubical Type Theory in a Topos. In Jean-Marc Talbot and Laurent Regnier, editors, *25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*, volume 62 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:19, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[8] Brigitte Pientka, Andreas Abel, Francisco Ferreira, David Thibodeau, and Rébecca Zucchini. Cocon: Computation in contextual type theory. *CoRR*, abs/1901.03378, 2019.

[9] Jonathan Sterling. *First Steps in Synthetic Tait Computability.* PhD thesis, Carnegie Mellon University Pittsburgh, PA, 2021.

[10] Taichi Uemura. A general framework for the semantics of type theory. *CoRR*, abs/1904.04097, 2019.