Internship report M2
Non-singleton Elimination

Soudant Léo, supervised by **Pierre-Marie Pédrot**, Galinette team

2025

General context

We attempt to create type theories corresponding with the properties of the internal languages of sheaf topoï. Martin Baillon succeeded for some instance of the problem.

Research problem

An element of a topological sheaf may be defined over an open whenever it is defined on a cover of that open, so long as they are compatible over the intersections. We hope to be able to do something similar in a type theory, where open set correspond to proof irrelevant propositions. This requires special attention to the compatibility condition : when a disjunction covers a proposition, it should be possible to eliminate from this disjunction to create a term in any type whenever compatible marginal terms are found.

Your contribution

I have proved using Rocq that a system T with a part of the wanted features normalises. I also have a model in Rocq where types are interpreted as sheaves, even though some other variant probably exists.

Arguments supporting its validity

The type theory devellopped by Martin Baillon, which I aim to generalise, proved continuity of all functional $(\mathbf{N} \to \mathbf{B}) \to \mathbf{B}$. I hope to find similar results at term.

Summary and future works

I will refine the type theory further as well as study other instances like the one Martin Baillon studied. I should also make a fork of the logrel Rocq project, and adapt it to my theory.

Contents

1 Introduction

1.1 The outline

2 Sheaves

2.1 Sheaves on topological spaces

We fix a topological space X, in this case presheves on that space are presheaves on the category underlying its poset of open subsets. More precisely :

Definition 2.1 (Presheaf on a topology)**.** *A* presheaf P *is given by*

- *For every open* $U \in \mathcal{O}(X)$*, a set* $P(U)$ *of* sections
- *For every open* U*, section* $s \in P(U)$*, and open subset* $V \subseteq U$*, a* restriction $s|_V \in P(V)$
- *If* $s \in P(U)$*,* $s|_U = s$
- *If* $s \in P(U)$*,* $W \subseteq V \subseteq U$*,* $(s|_V)|_W = s|_W$

The $\lambda(s \in P(U)), s|_V : P(U) \to P(V)$ function gives the $P(f)$ necessary to construct the functor $P : \mathcal{O}(X)^{op} \to$ **Set** justifying the use of the same term presheaf here and in the categorical context.

While in general presheaves are unrelated to sheaves, we can introduce what sheaves look like in the topos (see definition 2.4) of presheaves.

Definition 2.2 (Sheaf on a topology)**.** *A* sheaf $\mathcal{F}$ *is a presheaf such that given any open* U *and family of open* $(U_i)_i$ *such that* $U = \bigcup_i U_i$*, there is a bijection between :*

- *Sections* s *of* U
- *Families* $(s_i \in \mathcal{F}(U_i))_i$ *of sections where* $s_i|_{U_i \cap U_j} = s_j|_{U_i \cap U_j}$

where the forward direction is given by $s \mapsto (s|_{U_i})_i$*.*

Since the definition only acces the open subsets and not the points of the space, they can be extended to *locales*, that is, posets with finite meets and arbitrary joins satisfying the infinite distributive law. As far as we are concerned, this means $\vee, \wedge$ and $\leq$ will be used instead $\cup, \cap$ and $\subseteq$.

2.2 Kripke and Beth semantics

Presheaves give a classical tool to study intuitionistic logic through Kripke models.

Formula are relativised to an open subset (in general an object in a small category) or *forcing condition* U : $U \models \phi$, and the requirement for the formula to be verified is strengthened by asking for it to be verified on every $V \leq U$. This requirement degenerates back to the usual except in the $\Rightarrow$ and $\forall$ case :

- $U \models \phi \wedge \psi$ iff for all $V \leq U$, $V \models \phi$ and $V \models \psi$ iff $U \models \phi$ and $U \models \psi$
- $U \models \phi \Rightarrow \psi$ iff for all $V \leq U$, if $V \models \phi$ then $V \models \psi$
- $U \models \forall x : P, \phi$ iff for all $V \leq U$ and $s \in P(V)$, $V \models \phi(s/x)$

Since the key example of forcing condition is the context, it is perhaps unsurprising that the only connectors affected by are the one affecting the context (including the free variables in the context).

Where presheaves and their Kripke semantics strengthened the requirements (not unlike intuitionistic logic has less tools), sheaves and their Beth semantics lower them again.

Again, this weakened requirements degenerates in most cases.

- $U \models \phi \wedge \psi$ iff there exists $(U_i)_i$ with $U = \bigwedge_i U_i$, such that for all i, $U_i \models \phi$ and $U_i \models \psi$ iff $U \models \phi$ and $U \models \psi$.
- $U \models \phi \vee \psi$ iff there exists $(U_i)_i$ with $U = \bigwedge_i U_i$, such that for all i, $U_i \models \phi$ or $U_i \models \psi$.
- $U \models \exists x : \mathcal{F}, \phi$ iff there exists $(U_i)_i$ with $U = \bigwedge_i U_i$, such that for all i, there exists $s_i \in \mathcal{F}(U_i)$, $U_i \models \phi(s_i/x)$.

We note that $U \models \phi$ iff for all $V \leq U$, $V \models \phi$ in the case of kripke semantics and iff there exists $(U_i)_i$ with $U = \bigwedge_i U_i$, such that for all i, $U_i \models \phi$ in the case of Beth semantics, as can be shown by induction on ϕ in both cases.

The goal of this stage may be formulated as bringing this weakening of condition into type theory.

2.3 Topoï and sheaves in topoï

Proofs, results and details for this section can often be found in *Sheaves in geometry and logic: A first introduction to topos theory* by Saunders Maclane and Ieke Moerdijk [1].

Definition 2.3 (Subobject). *In a category, a* subobject *of X is an equivalence class of monomorphism $m : A \rightarrowtail X$, where the equivalence comes from the preorder where $m : A \rightarrowtail X$ is smaller than $m' : A' \rightarrowtail X$ when there is a map $f : A \to A'$ with $m' \circ f = m$.*

We deduce a presheaf **Sub** where $\mathbf{Sub}(X)$ is the the set of subobjects of X, and $\mathbf{Sub}(f) : \mathbf{Sub}(Y) \to \mathbf{Sub}(X)$ sends $m : A \to Y$ to its pullback by $f : X \to Y$.

Definition 2.4 (Topos). *A* topos *is a cartesian closed category with all finite limits and a suboject* classifier Ω *and an isomorphism $\mathbf{Sub}(X) \cong \mathbf{Hom}(X, \Omega)$ natural in X.*

We note that topoï also have finite colimits.

A topos serves to give models of intuitionistic logic in classical mathematical language. It has an internal logic which is higher order.

For example, **Set** is a topos, and given a topos $\mathcal{E}$, $\mathcal{E}/X$, the category of maps with codomain X and commuting triangles, as well as $\mathcal{E}^{\mathbf{C}^{op}}$, the category of contravariant functors from a small category $\mathbf{C}$ and natural transformation, are all topoï. In particular categories of presheaves are topoï, and correspond to Kripke models.

The subobject classifier Ω is equipped with an internal meet-semilattice structure inherited from the meet-semilattice structure on each $\mathbf{Sub}(X)$, which is natural in X.

Definition 2.5 (Lawvere-Tierney topology). *A* Lawvere-Tierney topology *is a left exact idempotent monad j on the internal meet-semilattice on Ω.*

- $id_\Omega \leq j$,
- $j \circ j \leq j$
- $j \circ \wedge = \wedge \circ j \times j$

From a topology j we extract a closure operator J_X of $\mathbf{Sub}(X)$ for any X.

Definition 2.6 (Dense subobject). *A suboject U of X is dense if $J_X U = X$*

A topology can be lifted to a left exact idempotent monad on the entirety of the topos, the sheafification monad.

Definition 2.7 (j-Sheaf in topos). *An object F is a j-*sheaf *in a topos if for any dense subobject U of any object X, the morphism $\mathbf{Hom}(X, F) \to \mathbf{Hom}(U, F)$ obtained by precomposition is an isomorphism.*

A j-Sheaf is up to isomorphism the result of sheafifying an object.

j-Sheaves form a topos. The sheaves on a presheaf topos correspond to Beth semantics.

2.4 Sheaves in type theory

Consider a type theory with a notion of proof irrelevant propositions Prop, *e.g.* book-HoTT with mere propositions, or Rocq with SProp.

In this case, a Lawvere-Tierney topology may be similarily defined, as a monad:

- $\mathsf{J} : \mathsf{Prop} \to \mathsf{Prop}$
- $\eta : \Pi(P : \mathsf{Prop}).P \to \mathsf{J}\ P$
- $\mathsf{bind} : \forall(PQ : \mathsf{Prop}).\mathsf{J}\ P \to (P \to \mathsf{J}\ Q) \to \mathsf{J}\ Q$

Then a sheaf is just a type T with

- A map $\mathsf{ask}_T : \Pi(P : \mathsf{Prop}).\ \mathsf{J}\ P \to (P \to T) \to T$
- A coherence $\varepsilon_T : \Pi(P : \mathsf{Prop})\ (j : \mathsf{J}\ P)\ (x : T).\ \mathsf{ask}_T\ \ P\ j\ (\lambda p : P.x) = x$

Now, the sheafified of a type doesn't exists in general, if the theory admits quotient inductive types, it can then be defined as follow :

$$\begin{aligned}
&\mathsf{Inductive}\ \mathcal{S}_{\mathsf{J}}\ T : \mathsf{Type} := \\
&|\ \mathsf{ret} : T \to \mathcal{S}_{\mathsf{J}}\ T \\
&|\ \mathsf{ask} : \Pi(P : \mathsf{Prop}), \mathsf{J}\ P \to (P \to \mathcal{S}_{\mathsf{J}}\ T) \to \mathcal{S}_{\mathsf{J}}\ T \\
&|\ \varepsilon : \Pi(P : \mathsf{Prop})\ (j : J\ P)\ (x : \mathcal{S}_{\mathsf{J}}\ T).\, \mathsf{ask}\ \ P\ j\ (\lambda p : P.x) = x
\end{aligned}$$

We note that by taking $I := \Sigma(P : \mathsf{Prop}).\mathsf{J}\ P$ and $O\ (P, j) : P$, a sheaf is then a

- A type T
- A map $\mathsf{ask}_T : \Pi(i : I), (O\ i \to T) \to T$
- A coherence map $\varepsilon_T : \Pi(i : I)\ (x : T), \mathsf{ask}_T\ \ i\ (\lambda o : O\ i.x) = x$

and simimlarily for sheafification. This is marginally simpler, and make sheaves appear as quotient dialogue trees, hence why we will henceforth consider (I, O)-*sheaves* instead of J-*sheaves*.

2.5 Geometric formulas

Definition 2.8. *A* geometric formula *is a formula of the form* $\forall \overrightarrow{x}, \bigwedge_{i \in I} O_i(\overrightarrow{x}) \to \bigvee_{j \in J} \exists \overrightarrow{y} \bigwedge_{k \in K_j} Q_{j,k}(\overrightarrow{x}, \overrightarrow{y})$

Where O_i and $Q_{j,k}$ are atomic formulas.

3 Models

A significant part of my internship was dedicated to contructing models of type theory in Rocq.

1. A model of a variant of Baclofen TT using dialogue trees. Predicates must be linearized before eliminating an inductive into them.
2. An exceptional model, with a type of exceptions E. A special type of dialogue trees where $I = E$ and $Oi = \mathbf{0}$, the resulting theory is inconsistent (when E is inhabited), as always when $Oi \to \mathbf{0}$ for some i.
3. A model using sheaves, which requires univalence, and quotient inductive types to model positive types.
4. A incomplete model using presheaves.

4 System T and MLTT

4.1 MLTT

We sought to extend the following variant of MLTT.

We consider a type of levels containing two elements $\mathbf{s}$ and $\mathbf{l}$ for small and large, with generic ℓ

With terms :

$$M, N ::= x|\lambda x.M|MN|0|S|\mathbf{N}_{\text{rec}}|\bot_{\text{rec}}|\mathbf{N}|\bot|\Pi x : A.B|\square_{\mathbf{s}}|\square_{\mathbf{l}}$$

Contexts :

$$\Gamma ::= \Gamma, x : A|\cdot$$

And conversion rules :

$$\textsc{Wf-Empty}\frac{}{\cdot \vdash \mathsf{well\text{-}formed}} \qquad \textsc{Wf-Ext}\frac{\Gamma \vdash A \equiv A \qquad \Gamma \vdash \mathsf{well\text{-}formed}}{\Gamma, x : A \vdash \mathsf{well\text{-}formed}}$$

$$\textsc{Int-Typ}\frac{\Gamma \vdash \mathsf{well\text{-}formed}}{\Gamma \vdash \mathbf{N} \equiv \mathbf{N} : \square_\ell} \qquad \textsc{Emp-Typ}\frac{\Gamma \vdash \mathsf{well\text{-}formed}}{\Gamma \vdash \bot \equiv \bot : \square_\ell}$$

$$\textsc{Fun-Typ}\frac{\Gamma \vdash A \equiv A' : \square_\ell \qquad \Gamma, x : A \vdash B \equiv B' : \square_\ell}{\Gamma \vdash \Pi x : A, B \equiv \Pi x : A', B' : \square_\ell} \qquad \textsc{Typ-Typ}\frac{\Gamma \vdash \mathsf{well\text{-}formed}}{\Gamma \vdash \square_{\mathbf{s}} : \square_{\mathbf{l}}}$$

$$\textsc{Fun-Intro}\ \frac{\Gamma, x : A \vdash M \equiv M' : B \qquad \Gamma \vdash A \equiv A : \Box_\ell}{\Gamma \vdash \lambda x.M \equiv \lambda x.M' : \Pi x : A, B} \qquad \textsc{Fun-Elim}\ \frac{\Gamma \vdash M \equiv M' : \Pi x : A, B \qquad \Gamma \vdash N \equiv N' : A}{\Gamma \vdash MN \equiv M'N' : B(N/x)}$$

$$\textsc{Axiom}\ \frac{\Gamma \vdash \mathsf{well\text{-}formed} \qquad x : A \in \Gamma}{\Gamma \vdash x \equiv x : A} \qquad \textsc{Beta}\ \frac{\Gamma, x : A \vdash M \equiv M' : B \qquad \Gamma \vdash N \equiv N' : A}{\Gamma \vdash (\lambda x.M)N \equiv M'(N'/x) : B(N/x)}$$

$$\textsc{Int-Zero}\ \frac{\Gamma \vdash \mathsf{well\text{-}formed}}{\Gamma \vdash 0 \equiv 0 : \mathbf{N}} \qquad \textsc{Int-Succ}\ \frac{\Gamma \vdash \mathsf{well\text{-}formed}}{\Gamma \vdash S \equiv S : \mathbf{N} \to \mathbf{N}}$$

$$\textsc{Int-Rec}\ \frac{\Gamma \vdash \mathsf{well\text{-}formed}}{\Gamma \vdash \mathbf{N}_{\mathrm{rec}} \equiv \mathbf{N}_{\mathrm{rec}} : \Pi A : \mathbf{N} \to \Box_{\mathbf{s}}, A0 \to (\Pi n : \mathbf{N}, An \to A(Sn)) \to \Pi n : \mathbf{N}, An}$$

$$\textsc{Int-Rec-Zero}\ \frac{\begin{array}{c}\Gamma \vdash A \equiv A : \mathbf{N} \to square_{\mathbf{s}} \qquad \Gamma \vdash N_0 \equiv N_0' \equiv A0 \\ \Gamma \vdash N_S \equiv N_S : \Pi n : \mathbf{N}, An \to A(Sn)\end{array}}{\Gamma \vdash \mathbf{N}_{\mathrm{rec}} A N_0 N_S 0 \equiv N_0' : A0}$$

$$\textsc{Int-Rec-Succ}\ \frac{\begin{array}{c}\Gamma \vdash A \equiv A' : \mathbf{N} \to square_{\mathbf{s}} \qquad \Gamma \vdash N_0 \equiv N_0' \equiv A0 \\ \Gamma \vdash N_S \equiv N_S' : \Pi n : \mathbf{N}, An \to A(Sn) \qquad \Gamma \vdash N \equiv N' : \mathbf{N}\end{array}}{\Gamma \vdash \mathbf{N}_{\mathrm{rec}} A N_0 N_S (SN) \equiv N_S' N' (\mathbf{N}_{\mathrm{rec}} A' N_0' N_S' N') : A(SN)}$$

$$\textsc{Emp-Rec}\ \frac{\Gamma \vdash \mathsf{well\text{-}formed}}{\Gamma \vdash \bot_{\mathrm{rec}} \equiv \bot_{\mathrm{rec}} : \Pi A : \bot \to \Box_{\mathbf{s}}, \Pi e : \bot, Ae}$$

$$\textsc{Sym}\ \frac{\Gamma \vdash M \equiv M' : A}{\Gamma \vdash M' \equiv M : A} \qquad \textsc{Trans}\ \frac{\Gamma \vdash M \equiv M' : A \qquad \Gamma \vdash M' \equiv M'' : A}{\Gamma \vdash M \equiv M'' : A}$$

$$\textsc{Conv}\ \frac{\Gamma \vdash M \equiv M' : A \qquad \Gamma \vdash A \equiv A' : \Box_\ell}{\Gamma M \equiv M' : A'}$$

But it is useful to consider the extension, with new terms : $M, N ::= \ldots |\mathbf{B}|\mathbf{B}_{\mathrm{rec}}|\overline{\mathsf{tt}}|\overline{\mathsf{ff}}$
And conversion rules

$$\textsc{Bool-True}\ \frac{\Gamma \vdash \mathsf{well\text{-}formed}}{\Gamma \vdash \overline{\mathsf{tt}} \equiv \overline{\mathsf{tt}} : \mathbf{B}} \qquad \textsc{Bool-False}\ \frac{\Gamma \vdash \mathsf{well\text{-}formed}}{\Gamma \vdash \overline{\mathsf{ff}} \equiv \overline{\mathsf{ff}} : \mathbf{B}}$$

$$\textsc{Bool-Rec}\ \frac{\Gamma \vdash \mathsf{well\text{-}formed}}{\Gamma \vdash \mathbf{B}_{\mathrm{rec}} \equiv \mathbf{B}_{\mathrm{rec}} : \Pi A : \mathbf{B} \to \Box_{\mathbf{s}}, A\overline{\mathsf{tt}} \to A\overline{\mathsf{ff}} \to \Pi b : \mathbf{B}, Ab}$$

$$\textsc{Bool-Rec-True}\ \frac{\begin{array}{c}\Gamma \vdash A \equiv A : \mathbf{B} \to \Box_{\mathbf{s}} \\ \Gamma \vdash M_{\overline{\mathsf{tt}}} \equiv M_{\overline{\mathsf{tt}}}' \equiv A\overline{\mathsf{tt}} \qquad \Gamma \vdash M_{\overline{\mathsf{ff}}} \equiv M_{\overline{\mathsf{ff}}} : A\overline{\mathsf{ff}}\end{array}}{\Gamma \vdash \mathbf{B}_{\mathrm{rec}} A M_{\mathsf{tt}} M_{\mathsf{ff}} \overline{\mathsf{tt}} \equiv M_{\overline{\mathsf{tt}}}' : A\overline{\mathsf{tt}}}$$

$$\textsc{Bool-Rec-False}\ \frac{\begin{array}{c}\Gamma \vdash A \equiv A : \mathbf{B} \to \Box_{\mathbf{s}} \\ \Gamma \vdash M_{\overline{\mathsf{tt}}} \equiv M_{\overline{\mathsf{tt}}} \equiv A\overline{\mathsf{tt}} \qquad \Gamma \vdash M_{\overline{\mathsf{ff}}} \equiv M_{\overline{\mathsf{ff}}}' : A\overline{\mathsf{ff}}\end{array}}{\Gamma \vdash \mathbf{B}_{\mathrm{rec}} A M_{\overline{\mathsf{tt}}} M_{\overline{\mathsf{ff}}} \overline{\mathsf{ff}} \equiv M_{\overline{\mathsf{ff}}}' : A\overline{\mathsf{ff}}}$$

4.2 System T

To identify and solve problems in a simpler envirronement, we studied a modified System T before, based on the following variant.

With types :

$$A ::= A \to A|\mathbf{N}|\bot$$

Terms :

$$M, N ::= x|\lambda x.M|MN|0|S|\mathbf{N}_{\mathrm{rec}}|\bot_{\mathrm{rec}}$$

Contexts :

$$\Gamma ::= \Gamma, x : A|\cdot$$

And conversion rules :

$$\text{Fun-Intro}\ \frac{\Gamma, x : A \vdash M \equiv M' : B}{\Gamma \vdash \lambda x.M \equiv \lambda x.M' : A \to B} \qquad \text{Fun-Elim}\ \frac{\Gamma \vdash M \equiv M' : A \to B \quad \Gamma \vdash N \equiv N' : A}{\Gamma \vdash MN \equiv M'N' : B}$$

$$\text{Axiom}\ \frac{x : A \in \Gamma}{\Gamma \vdash x \equiv x : A} \qquad \text{Beta}\ \frac{\Gamma, x : A \vdash M \equiv M' : B \quad \Gamma \vdash N \equiv N' : A}{\Gamma \vdash (\lambda x.M)N \equiv M' : B}$$

$$\text{Int-Zero}\ \frac{\Gamma \vdash \mathsf{well\text{-}formed}}{\Gamma \vdash 0 \equiv 0 : \mathbf{N}} \qquad \text{Int-Succ}\ \frac{\Gamma \vdash \mathsf{well\text{-}formed}}{\Gamma \vdash S \equiv S : \mathbf{N} \to \mathbf{N}}$$

$$\text{Int-Rec}\ \frac{\Gamma \vdash \mathsf{well\text{-}formed}}{\Gamma \vdash \mathbf{N}_{\text{rec}} \equiv \mathbf{N}_{\text{rec}} : A \to (\mathbf{N} \to A \to A) \to \mathbf{N} \to A}$$

$$\text{Int-Rec-Zero}\ \frac{\Gamma \vdash N_0 \equiv N_0' \equiv A \qquad \Gamma \vdash N_S \equiv N_S : \mathbf{N} \to A \to A}{\Gamma \vdash \mathbf{N}_{\text{rec}} N_0 N_S 0 \equiv N_0' : A}$$

$$\text{Int-Rec-Succ}\ \frac{\Gamma \vdash N_0 \equiv N_0' \equiv A \qquad \Gamma \vdash N_S \equiv N_S' : \mathbf{N} \to A \to A \quad \Gamma \vdash N \equiv N' : \mathbf{N}}{\Gamma \vdash \mathbf{N}_{\text{rec}} N_0 N_S (SN) \equiv N_S' N' (\mathbf{N}_{\text{rec}} N_0' N_S' N') : A}$$

$$\text{Emp-Rec}\ \frac{}{\Gamma \vdash \bot_{\text{rec}} \equiv \bot_{\text{rec}} : \bot \to A}$$

$$\text{Sym}\ \frac{\Gamma \vdash M \equiv M' : A}{\Gamma \vdash M' \equiv M : A} \qquad \text{Trans}\ \frac{\Gamma \vdash M \equiv M' : A \quad \Gamma \vdash M' \equiv M'' : A}{\Gamma \vdash M \equiv M'' : A}$$

5 ShTT

5.1 Martin Baillon's ShTT

My work was meant to generalise the work of my supervisor past PhD student, Martin Baillon. He worked on a roughly similar MLTT extended with boolean and a generic function α.

$$M, N ::= \ldots | \alpha$$

With *forcing contexts*, with n and b an integer and boolean respectively

$$\mathcal{L} ::= \mathcal{L}, n \mapsto b | \cdot$$

We write $n \mapsto_{\mathcal{L}} b$ when $n \mapsto b$ appears in $\mathcal{L}$, and $n \not\mapsto_{\mathcal{L}}$ when neither $n \mapsto \mathsf{tt}$ nor $n \mapsto \mathsf{ff}$ do.

And conversion rules :

$$\text{name}\ \frac{\mathcal{L}, \Gamma \vdash ?_0 \quad \cdots \quad \mathcal{L}, \Gamma \vdash ?_n}{\mathcal{L}, \Gamma \vdash ?}$$

whenever the following is a rule of MLTT (with booleans)

$$\text{name}\ \frac{\Gamma \vdash ?_0 \quad \cdots \quad \Gamma \vdash ?_n}{\Gamma \vdash ?}$$

Exception made of Wf-Empty, which becomes

$$\text{Wf-Empty}\ \frac{}{\cdot, \cdot \vdash \mathsf{well\text{-}formed}}$$

The new conversion rules are :

$$\text{Wf-Ext-Forc}\ \frac{\mathcal{L}, \Gamma \vdash \mathsf{well\text{-}formed} \quad n \not\mapsto_{\mathcal{L}}}{\mathcal{L}, n \mapsto b, \Gamma \vdash \mathsf{well\text{-}formed}}$$

$$\text{Gen}\ \frac{\mathcal{L}, \Gamma \vdash \mathsf{well\text{-}formed}}{\mathcal{L}, \Gamma \vdash \alpha \equiv \alpha : \mathbf{N} \to \mathbf{B}} \qquad \text{Ask}\ \frac{\mathcal{L}, \Gamma \vdash \mathsf{well\text{-}formed} \quad n \mapsto_{\mathcal{L}} b}{\mathcal{L}, \Gamma \vdash \alpha \overline{n} \equiv \overline{b}}$$

$$\text{Split}\ \frac{\mathcal{L}, n \mapsto \mathsf{tt}, \Gamma \vdash M \equiv M' : A \quad \mathcal{L}, n \mapsto \mathsf{ff}, \Gamma \vdash M \equiv M' : A \quad n \not\mapsto_{\mathcal{L}}}{\mathcal{L}, \Gamma \vdash M \equiv M' : A}$$

Amongst other thing, this theory can be used to show that any term $\cdot \vdash M \equiv M : (\mathbf{N} \to \mathbf{B}) \to \mathbf{N}$ of MLTT has a continuity proof.

5.2 ShTT

The prototype of theory that I defined and hoped to study is as follows :

We imagine we have a set Ω of *atoms*. We first set a set I with decidable equality, and a (morally finite) set Ai for each $i \in I$ standing for arity, and finally a family $((O_{i,\alpha})_{\alpha \in Ai})_{i \in I}$ of elements of Ω. The term extend those of MLTT as follows :

$$M, N ::= \ldots | \digamma_i (M_\alpha)_{\alpha \in Ai}$$

When instanciating with finite Ai, it would rather be $\digamma_i M_1 \cdots M_n$.

The forcing contexts $\mathcal{L}$ are now subset of Ω, and never cause ill-formation.

The conversion rules also copy those from MLTT by adding a forcing context as in 5.1, without the WF-Empty exception.

The new conversion rules are as follows :

$$\textsc{Dig-}i\,\frac{\forall \alpha\alpha'. \mathcal{L}, O_{i,\alpha}, O_{i,\alpha'}, \Gamma \vdash M_\alpha \equiv M'_{\alpha'} : A}{\mathcal{L}, \Gamma \vdash \digamma_i (M_\alpha)_\alpha \equiv \digamma_i (M'_\alpha)_\alpha : A}$$

This rule is both the expected congruence rule for conversion and also a compatibility rule for typing that would be stated separately in a system with a purre typing judgement.

$$\textsc{Ask-}i\,\frac{\forall \alpha\alpha'. \mathcal{L}, O_{i,\alpha}, O_{i,\alpha'}, \Gamma \vdash M_\alpha \equiv M'_{\alpha'} : A}{\mathcal{L}, O_{i,\alpha}, \Gamma \vdash \digamma_i (M_{\alpha'})_{\alpha'} \equiv M'_\alpha : A}$$

$$\textsc{Dig-Ev}\,\frac{\mathcal{L}, \Gamma \vdash N \equiv N' : A \qquad \forall \alpha\alpha', \mathcal{L}, O_{i,\alpha}, O_{i,\alpha}, \Gamma \vdash M_\alpha \equiv M_{\alpha'} : \Pi x : A, B}{\mathcal{L}, \Gamma \vdash (\digamma_i (M_\alpha)_\alpha) N \equiv \digamma (M_\alpha N)_\alpha : B(N/x)}$$

$$\textsc{Int-Rec-Dig}\,\frac{\begin{array}{c}\mathcal{L}, \Gamma \vdash A \equiv A : \mathbf{N} \to \square_{\mathsf{s}} \qquad \mathcal{L}, \Gamma \vdash M_0 \equiv M'_0 : \mathbf{N} \qquad \mathcal{L}, \Gamma \vdash M_S \equiv M'_S : \Pi n : \mathbf{N}, An \to A(Sn) \\ \forall \alpha\alpha', \mathcal{L}, O_{i,\alpha}, O_{i,\alpha}, \Gamma \vdash M_\alpha \equiv M_{\alpha'} : \mathbf{N}\end{array}}{\mathcal{L}, \Gamma \vdash \mathbf{N}_{\mathrm{rec}} A M_0 M_S (\digamma_i (M_\alpha)_\alpha) \equiv \digamma_i (\mathbf{N}_{\mathrm{rec}} M_0 M_S M_\alpha)_\alpha : A \digamma_i (M_\alpha)_\alpha}$$

$$\textsc{Emp-Rec-Dig}\,\frac{\mathcal{L}, \Gamma \vdash A \equiv A : \bot \to \square_{\mathsf{s}} \qquad \forall \alpha\alpha', \mathcal{L}, O_{i,\alpha}, O_{i,\alpha}, \Gamma \vdash M_\alpha \equiv M_{\alpha'} : \bot}{\mathcal{L}, \Gamma \vdash \bot_{\mathrm{rec}} A (\digamma_i (M_\alpha)_\alpha) \equiv \digamma_i (\bot_{\mathrm{rec}} M_\alpha)_\alpha : A \digamma_i (M_\alpha)_\alpha}$$

This theory is still incomplete. For example, to instantiate it with 5.1, even adding the necessary generic function α, and setting $I := \mathbf{N}$, $Ai = \mathbf{B}$ and $O_{n,b} := n \mapsto b$ so that Dig-i may play the role of Split, we have no way to provide the compatibility proof $n \mapsto \mathsf{tt}, n \mapsto \mathsf{ff} \vdash M_{\mathsf{tt}} \equiv M_{\mathsf{ff}}$, which must be some form of ex-falso. Ask-i cannot be used in stead of Ask either.

6 Logical relations

6.1 System T extension

7 Meta-informations

7.1 Time expenditure

7.2 Difficulties

The subject is rather vast and a bit unclear.

Doing logical relations for MLTT is a large task, even before adding sheaves. My next attempt will probably start from logrel rocq insted of nothing

8 Conclusion

Appendix

A Dummy

References

[1] Saunders MacLane and Ieke Moerdijk. *Sheaves in geometry and logic: A first introduction to topos theory.* Springer Science & Business Media, 2012.