Lawvere-Tierney Sheafification in Homotopy Type Theory

Kevin Quirin and Nicolas Tabareau
Inria
École des Mines de Nantes
Nantes, France

***Abstract*—Sheafification is a popular tool in (higher) topos theory which allows to extend the internal logic of a topos with new principles. One of its most famous applications is the possibility to transform a topos into a boolean (classical) topos using the dense topology, which corresponds in essence to Gödel's double negation translation. The same construction has not been considered in Martin-Löf type theory because of a mismatch between topos theory and type theory. This mismatched has been fixed recently by considering homotopy type theory, an extension of Martin-Löf type theory with new principles inspired by category theory and homotopy theory, and which corresponds closely to higher topoi. In this paper, we give a computer-checked construction of Lawvere-Tierney sheafification in homotopy type theory, which allows in particular to give a meaning to the (propositional) law of excluded middle inside homotopy type theory, without axiom, being compatible with the full type-theoretic axiom of choice.**

I. Introduction

Sheafification [1] is a very powerful geometric construction that has been initially stated in topology and has quickly been lifted to mathematical logic. In the field of topos theory, it provides a way to construct new topoi from already existing ones, allowing logical principles—that can not be proven to be true or false in the old topos—to be valid (or invalid) in the new topos. The most common use of sheafification is the construction a boolean topos, thus validating the classical principle of the law of excluded middle (EM), from an intuitionistic topos using the dense topology. An other famous application has been developed by Cohen [2] to prove that the continuum hypothesis is independent of the usual axioms of Zermelo-Frænkel set theory, even in presence of the axiom of choice (AC). The initial work of Cohen uses forcing but can be rephrased in terms of sheafification [1].

Even if similar questions have been considered around the Curry-Howard isomorphism, there is no canonical way to extend a programming language such as type theory with new logical or computational principle while keeping consistency automatically. For instance, much effort have been done to provide a computational content to the law of excluded middle in order to define a constructive version of classical logic. This has lead to various calculi, with most notably the $\lambda\mu$-calculus of Parigot [3], but this line of work has not appeared to be fruitful to define a new version of type theory with classical principles. Other works have tried to extend continuation-passing-style (CPS) transformation to type theory, but they have been faced with the difficulty that the CPS transformation is incompatible with (full) dependent sums [4], which puts emphasis on the tedious link between the axiom of choice and the law of excluded middle in type theory.

Nevertheless the axiom of choice has shown to be realizable by computational meaning in a classical setting by techniques turning around the notion of (modified) bar induction [5], Krivine's realizability [6] and even more recently with restriction on elimination of dependent sums and lazy evaluation [7].

Unfortunately, all those works do not share the generality and simplicity of sheafification in topos theory. However, type theory is known to be quite closed to topos theory so one could wonder why similar techniques have not been developed in the field of type theory. The answer to this question has been given recently by the advent of homotopy type theory [8], which is an extension of Martin-Löf type theory with principles inspired by (higher) category theory and homotopy theory, such as higher inductive types [9], [10] and Voevodsky's univalence principle [11].

This new point of view on type theory has revealed the homotopy structure of types where for instance (homotopical) propositions are just types with an irrelevant equality and (homotopical) sets are types with a propositional equality. When restricted to such propositions and sets, type theory corresponds quite closely to topos theory but the mismatch starts when considering higher homotopical types. Fortunately, an higher version of topos theory has been developed recently by mathematicians, synthesized in the monograph of Lurie on higher topos theory [12]. Lurie has developed their all the tools that have been defined in topos theory, but in an higher setting. In particular, the theory of sheaves has been lifted to higher topos theory. As the notion of higher topoi appears to correspond very closely to homotopy type theory, this provides a new hope that tackling the problem of extending the power of type theory using sheafification is actually possible.

Nevertheless, the adaptation of the sheafification in higher topos theory to homotopy type theory is not straightforward because the construction in higher topoi is restricted to the initial Grothendieck which is still very topological, and not very amenable to formalization in type theory. It seems more promising to use a generalized sheafification, called Lawvere-Tierney sheafification [1], but this construction has not been considered yet in the setting of higher topos theory.

A. Lawvere-Tierney Topology

B. Overview of the Result

The main contribution of this article is presents a computer check

- modalities in HoTT
- Giraud's axiom

Plan of the paper :

- In section II, we define the foundations on which our worked is based.
- In section III, we explain the construction of the sheafification functor.
- Finally, we discuss related works and present future works.

II. Homotopy Type Theory

A. Homotopy Type Theory

NT ► *Il faut au minimum introduire proprement les n-Types, le squash sur les props* ◄

Our work is mainly based on the stratification by Type_n :

Definition 1. $\text{Is-}n\text{-type}$ *is defined by induction on* $n \geqslant -2$ *:*

- $\text{Is-}(-2)\text{-type}X$ *if* X *is a contractible type, i.e,* X *is pointed by* $c : X$*, and every other point in* X *is connected to* c*.*
- $\text{Is-}(n+1)\text{-type}X = \prod_{x,y:X} \text{Is-}n\text{-type}(x = y)$.

Then, $\mathrm{Type}_n = \sum_{X:\mathrm{Type}} \text{Is-}n\text{-type}X$.

For a type X, to be in Type_n means that path spaces of X are trivial after $n+1$ iteration. We define some shortcuts

- $\mathrm{IsContr} := \text{Is-}(-2)\text{-type}$ and $\mathrm{Contr} := \mathrm{Type}_{-2}$
- $\mathrm{IsHProp} := \text{Is-}(-1)\text{-type}$ and $\mathrm{HProp} := \mathrm{Type}_{-1}$

Using higher inductive types, one can define for any type X $\|X\| : \mathrm{HProp}$ generated by

- a function $|\cdot|_X : X \to \|X\|$,
- paths $x = y$ for all $x, y : \|X\|$.

$\|X\|$ is called the *(propositional) truncation of* X. The recursion principle asserts :

Lemma 2. *For any* $A : \mathrm{Type}$ *and* $B : \mathrm{HProp}$*, if* $f : A \to B$ *then there is an induced* $g : \|A\| \to B$ *such that* $g(|a|) = f(a)$ *for any* $a : A$*.*

We refer to [8] for more details.

Working in Type_n allows to use the subobjects classification as in [13] : for all type B, Type_n classifies subobjects of B with n-truncated fibers ; we have an equivalence

$$\left(\sum_{A:\mathrm{Type}} \sum_{f:A\to B} \prod_{b\in B} \text{Is-}n\text{-type } \mathrm{fib}_f(b)\right) \simeq (B \to \mathrm{Type}_n)$$

and a pullback (with $\mathrm{Type}_n^\bullet$ the universe of pointed n-truncated types) :

$$\begin{array}{ccc} A & \xrightarrow{t_f} & \mathrm{Type}_n^\bullet \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle \pi_1} \\ B & \xrightarrow[\chi_f]{} & \mathrm{Type}_n \end{array}$$

forall f with n-truncated fibers, with

$$t_f = \lambda a,\ (\mathrm{fib}_f(f(a)), (a, \mathrm{idpath})).$$

B. Left exact Modalities

NT ► *To be checked: Condition iv of Thm 7.7.5 of HoTT* ◄

We use the following definition of truncated left exact modalities :

Definition 3. *Let* p *be a truncation index. Then a left exact modality is the data of*

(i) A predicate $P : \mathrm{Type}_p \to \mathrm{HProp}$
(ii) For every p*-truncated type* A*, a* p*-truncated type* $\bigcirc A$ *such that* $P(\bigcirc A)$
(iii) For every p*-truncated type* A*, a map* $\eta_A : A \to \bigcirc A$
such that
(iv) For every p*-truncated types* A *and* B*, if* $P(B)$ *then*

$$\begin{cases} (\bigcirc A \to B) & \to & (A \to B) \\ f & \mapsto & f \circ \eta_A \end{cases}$$

is an equivalence.
(v) For all $A : \mathrm{Type}_p$ *and* $B : A \to \mathrm{Type}_p$ *such that* $P(A)$ *and* $\prod_{x:A} P(Bx)$*, then* $P\left(\sum_{x:A} B(x)\right)$
(vi) For all $A : \mathrm{Type}_p$ *and* $x, y : A$*, if* $\bigcirc A$ *is contractible, then* $\bigcirc(x = y)$ *is contractible.*

Conditions (i) to (iv) define a reflective subuniverse*, (i) to (v) a* modality*.*

The type of all p-types such that P will be noted $\mathrm{Type}_p^\bigcirc$

Since basic operations (dependent products, products, sigma types) are stable for truncation levels, all theorem in [8], chapter 7.7 remains true. In particular, if $X : \mathrm{Type}_p$ verifies $P(X)$, then for any $x, y : X$, $P(x = y)$.

Another useful property is that we can extend property *(iii)* to dependent product.

Moreover, left exactness implies in particular fibers preservations :

Proposition 4. *For any* n*-truncated types* X *and* Y*, and any map* $f : X \to Y$*, the modalisation of fiber of* f *above any element* $y : Y$ *is the fiber of* $\bigcirc f$ *above* $\eta_Y y$ *:*

$$\bigcirc\left(\sum_{x:X} (fx = y)\right) = \sum_{x:\bigcirc X} (\bigcirc fx = \eta_Y y).$$

Moreover the following diagram commute

$$\begin{array}{ccc} \sum_{x:X}(fx = y) & \xrightarrow{\eta} & \bigcirc\left(\sum_{x:X}(fx = y)\right) \\ {\scriptstyle \gamma}\downarrow & \diagup\!\!\diagup & \\ \sum_{x:\bigcirc X}(\bigcirc fx = \eta_Y y) & & \end{array}$$

where $\pi_1 \circ \gamma = \eta_X$*, and* $\pi_2 \circ \gamma$ *is the usual modalisation of paths.*

As we want, a left exact modality preserves HProp :

Proposition 5. *If* $P : \mathrm{HProp}$*, then* $\bigcirc\widehat{P} : \mathrm{HProp}$*, where* $\widehat{P}$ *is* P *seen as a* p*-type.*

A left exact modality leads to a new theory with new principles. A basic example is the *open modality of* P for a P : HProp, $\bigcirc_P T = P \to T$. When one works in the subuniverse defined by this modality, defining an inhabitant of a type T is actually giving an inhabitant of $P \to T$. Thus, being in this subuniverse is just adding the axiom P to the theory. Sadly, adding an axiom this way does not add any computational content to P, thus is not completely satisfactory.

Another example is the $\neg\neg$-modality, $\bigcirc_{\neg\neg} T = (T \to \bot) \to \bot$. Although we might think it enables us to use classical reasonings in the corresponding reflective subuniverse, the main problem is that every type in the new universe is an HProp, which is very restrictive. We will use this modality only in the HProp layer, and try to extend it to every Type_n without collapsing every type.

C. Generalized pullbacks

$\boxed{\text{KQ}}$ ►*Do we keep this name ?*◄

Definition 6. *Let* $f : X \to Y$ *be a map, and* p : nat. *The* p*-pullback of* f*, noted* $X \times_Y \cdots \times_Y X$ *is the limit of the diagram*

$$\begin{tikzcd} X \arrow[dr, "f"'] & \cdots & X \arrow[dl, "f"] \\ & Y & \end{tikzcd}$$

with p *copies of* X. *The* 0*-pullback of* f *is* Unit.

In homotopy type theory, we have

$$X \times_Y \cdots \times_Y X = \sum_{x : X^n} (fx_1 = fx_2) \wedge \cdots \wedge (fx_{n-1} = fx_n).$$

Definition 7. *If* $f : X \to Y$ *is a map, then the* Čech nerve *of* f *is the diagram*

$$C(f) := \cdots \; X \times_Y X \times_Y X \rightrightarrows\!\!\!\!\rightarrow X \times_Y X \rightrightarrows X$$

with canonical projections.

Axiom 8 (Giraud)**.** *If* $f : X \to Y$ *is a surjection, then the colimit of* $C(f)$ *is* Y.

Viewing homotopy type theory as a type-theoristic version of higher topos theory, [12] suggests that this axiom can be proved, although we admit it in this paper.

Formalization of colimits is based on the formalization of limits in [14].

III. Sheaves

We will define sheaves by induction on the homotopical truncation level of types : the base case will be any left exact modality on HProp, and we give in this section some definitions useful for the inductive case.

In this section, we suppose given a truncation index $n \geqslant -1$, and a left exact modality $\bigcirc$ on Type_n.

Definition 9. *Let* E *be a type. The* closure *of a subobject of* E *classified by* χ *is the subobject of* E *classified by* $\bigcirc \circ \chi$.

The subobject of E *classified by* χ *is said* closed *in* E *if its closure is itself :* $\chi = \bigcirc \circ \chi$.

Definition 10. *Let* E *be a type, and* $\chi : E \to \mathrm{Type}_n$. *The subobject* A *of* E *classified by* χ *is* dense *in* E *if its* $\bigcirc$*-closure is* E *seen as a subobject of* E*, i.e,*

$$\forall e : E, \; \left(\sum_{e' : E} e = e' \right) \simeq (\bigcirc(\chi \; e)).$$

Moreover, for all $x : A$ *we need the following coherence diagram to commute*

$$\begin{tikzcd} \sum_{e' : A} x = e' \arrow[r, equal] \arrow[d, "\iota"'] & (\chi \; x) \arrow[d, "\eta_{(\chi \; x)}"] \\ \sum_{e' : E} x = e' \arrow[r, equal] & \bigcirc(\chi \; x) \end{tikzcd}$$

where $\iota : x \mapsto (x_{11}; x_2)$.

It follows from fibers preservation that any n-subobject of a type seen as a n-subobject of its closure is closed.

Definition 11 (Restriction)**.** *For any* A, E : Type *and* F : Type_{n+1}*, we define :*

- *if* $A \hookrightarrow E$, M_E^χ *is the map sending an arrow* $f : E \to F$ *to* $\restriction fA$ *the restriction of* f *to* A.
- *if* $A \to E$ *with* n*-truncated fibers,* Φ_E^χ *is the map sending* $f : E \to f$ *to* $\restriction fA$.

Definition 12 (Separated Type)**.** *A type* F *in* Type_{n+1} *is* separated *if for all type* E*, and all dense subobject of* E *classified by* $\chi : E \to \mathrm{Type}_n$, Φ_E^χ *is a monomorphism. In other words, the dotted arrow, if exists, is unique.*

$$\begin{tikzcd} \sum_{e : E} (\chi \; e) \arrow[r, "f"] \arrow[d, "\pi_1"'] & F \\ E \arrow[ur, dashed, "!"'] & \end{tikzcd}$$

Definition 13 (Sheaves)**.** *A type* F *of* Type_{n+1} *is a* $(n+1)$-sheaf *if it is separated, and for all type* E *and all dense subobject of* E *classified by* $\chi : E \to \mathrm{Type}_{-1}$, M_E^χ *is an equivalence. In other words, the dotted arrow exists and is unique.*

$$\begin{tikzcd} \sum_{e : E} (\chi \; e) \arrow[r, "f"] \arrow[d, "\pi_1"'] & F \\ E \arrow[ur, dashed, "\exists!"'] & \end{tikzcd}$$

Note that these definitions are almost the same as the ones in [1]. The main difference is that *separated* is defined for n-subobjects, while *sheaf* only for -1-subobject.

From these definitions, one can show that

Proposition 14.
- $\mathrm{Type}_n^\bigcirc$ *is a sheaf.*
- *If* $A : \mathrm{Type}_{n+1}$ *and* $B : A \to \mathrm{Type}_{n+1}$ *such that for all* $a : A$, $(B \; a)$ *is a sheaf, then* $\prod_{a:A} B \; a$ *is a sheaf.*

IV. Construction of sheafification

We mimic the construction in [1]. From any left exact modality on HProp, we extend the new principles it gives to every Type_n, $n \geqslant 0$. Actually, if n_0 is a fixed truncation index, and $\bigcirc$ a left exact modality on Type_{n_0}, then we can in the same way extend the properties of the modality to any Type_n, for $n > n_0$.

Once the first modality is defined, the extension to all Type_n is automatic. It means that the new principles we want to add in the new theory must be introduced ine the first modality.

We have to be careful : if we want to propagate the properties of the first modality $\bigcirc_{-1}$, all the higher modalities $\bigcirc_n$ must be compatible with $\bigcirc_{-1}$, i.e, we want the property

Proposition 15. *If $P : \mathrm{HProp}$, and $\widehat{P}$ is P seen as a Type_n, then $\bigcirc_n \widehat{P} = \bigcirc_{-1} P$, and the following coherence diagram commute*

$$\begin{tikzcd} \bigcirc_{-1} P \arrow[r, equal] & \bigcirc_n \widehat{P} \\ P \arrow[u, "\eta_{-1}"] \arrow[r, equal] & \widehat{P} \arrow[u, "\eta_n"'] \end{tikzcd}$$

The sheafification process works exactly that way in [1] : from a left exact modality on the internal logic (a Lawvere-Tierney topology), we define a new left exact modality on the whole topos. We view that as the induction step between HProp and Type_0, and we want to extend it.

KQ ►*Bof ; maybe rewrite this*◄

As in most of sheafification process, it will be done in two steps :

(i) *separation :* In this step, we *remove* the lack of sheafness from the type we are considering.
(ii) *completion :* In this step, we add what lacks now to the separated type to be a sheaf.

A. For h-propositions

For the case $n = -1$, one can take any left exact modality on HProp. Here we took the $\neg\neg$ modality :

$$\forall P : \mathrm{HProp}, \quad \bigcirc_j P = \neg\neg P.$$

One can easily show that it defines a left exact modality.

The $\neg\neg$-modality allows to work with a propositional law of excluded middle.

B. From Type to Separated Type

In this section, we suppose given a truncation index $n \geqslant -1$, and a left exact modality $\bigcirc$ on Type_n, compatible with the modality on HProp in the sense of proposition 15.

Let T be in Type_{n+1}. We define $\Box T$ as the image of $\bigcirc^T \circ \{\cdot\}_T$, as in

$$\begin{tikzcd} T \arrow[r, "\{\cdot\}_T"] \arrow[d, "\mu_T"'] & \mathrm{Type}_n^T \arrow[d, "\bigcirc^T"] \\ \Box T \arrow[r] & \left(\mathrm{Type}_n^{\bigcirc}\right)^T \end{tikzcd} ,$$

where $\{\cdot\}_T$ is the singleton map $\lambda(t : T),\ \lambda(t' : T),\ t = t'$. In type theory words,

$$\begin{aligned} \Box T &= \mathrm{Im}(\lambda\ t : T,\ \lambda\ t',\ \bigcirc(t = t')) \\ &= \sum_{u : T \to \mathrm{Type}_n^{\bigcirc}} \left\| \sum_{a:X} (\lambda t,\ \bigcirc(a = t)) = u \right\| \end{aligned}$$

There again, the separation step has the same definition as in [1], using $\mathrm{Type}_n^{\bigcirc}$ instead of the j-subobject classifier.

Proposition 16. *For all $T : \mathrm{Type}_{n+1}$, $\Box T$ is separated.*

Proof. We will use the following lemma :

Lemma 17. *A $(n+1)$-truncated type T with an embedding $f : T \to U$ into a separated $(n+1)$-truncated type U is itself separated.*

As $\Box T$ embeds in $\left(\mathrm{Type}_n^{\bigcirc}\right)^T$, we only have to show that the latter is separated. But it is the case with both parts proposition 14. ☐

We now need to show that $\Box$ is universal. The following lemma is central in the proof :

Lemma 18. *Let $T : \mathrm{Type}_{n+1}$. Then $\Box T$ is the colimit of the closed diagonal diagram*

$$\cdots \overline{\Delta_3} \rightrightarrows\!\!\!\rightarrow \overline{\Delta_2} \rightrightarrows \overline{\Delta_1}$$

where Δ_k is the k-pullback of $\mathrm{id} : T \to T$.

This lemma is an adaptation of the sheafification process in [1], where they consider only the kernel pair of μ_T instead of the Čech nerve.

Proof. As μ_T is an surjection, we know by Giraud axiom that $\Box T$ is the colimit of $C(\mu_T)$. If we can show that $C(\mu_T) = C(\mathrm{id})$, the the result will follow.

Let $k : \mathrm{nat}$, let's show that

$$\begin{aligned} &\sum_{t : T^k} (\mu_T t_1 = \mu_T t_2 \wedge \cdots \wedge \mu_T t_{k-1} \mu_T t_k) \\ &= \sum_{t : T^k} \bigcirc(t_1 = t_2 \wedge \cdots \wedge t_{k-1} = t_k) \end{aligned}$$

By induction on k, and the preservation of products by $\bigcirc$, it suffices to show that for all $a, b : T$, $\bigcirc(a = b) = (\mu_T a = \mu_T b)$. By univalence, we want arrows in both ways, forming an equivalence.

- Suppose $p : (\mu_T a = \mu_T b)$. Then projecting p along first components yields $q : \prod_{t:T} \bigcirc(a = t) = \bigcirc(b = t)$. Taking for example $t = b$, we deduce $\bigcirc(a = b) = \bigcirc(a = a)$, and the latter is inhabited by $\eta_{a=a} 1$.
- Suppose now $p : \bigcirc(a = b)$. Let ι be the first projection from $\Box T \to (T \to \mathrm{Type}_n^{\bigcirc})$. ι is an embedding, thus it suffices to prove $\iota(\mu_T a) = \iota(\mu_T b)$, i.e, $\prod_{t:T} \bigcirc(a = t) = \bigcirc(b = t)$. The latter remains true by univalence.

The fact that these two form an equivalence is technical, we refer to the formalization for an explicit proof. ☐

Now, let Q be a separated Type_{n+1}, and $f : T \to Q$. Then the following diagram commutes

$$\begin{tikzcd}
\cdots \overline{\Delta_3} \arrow[r, shift left=2] \arrow[r] \arrow[r, shift right=2] \arrow[dr, "f\circ\pi_1"'] & \overline{\Delta_2} \arrow[r, shift left] \arrow[r, shift right] \arrow[d, "f\circ\pi_1"] & \overline{\Delta_1} = T \arrow[dl, "f\circ\pi_1"] \\
& Q &
\end{tikzcd}$$

But we know (lemma 18) that $\Box T$ is the colimit of the closed diagonals diagram, thus there is an universal arrow $\Box T \to Q$.

This proves the following proposition

Proposition 19. *$(\Box, \mu)$ defines a reflective subuniverse on* Type_{n+1}.

We now have to prove point *(v)* in the definition of modality. Let $A : \mathrm{Type}_{n+1}$ be a sheaf and $B : A \to \mathrm{Type}_{n+1}$ be a sheaf family. We want to show that $\sum_{x:A}(Bx)$ is separated. Let E be a type, and $\sum_{e:E}(\chi\, e)$ a dense subobject of E.

Let f, g be two maps from $\sum_{e:E}(\chi\, e)$ to $\sum_{x:A}(Bx)$, equal when precomposed with π_1.

$$\begin{tikzcd}
\sum_{e:E}(\chi\, e) \arrow[r, shift left, "f\circ\pi_1"] \arrow[r, shift right, "g\circ\pi_1"'] \arrow[d, "\text{dense}"'] & \sum_{x:A}(Bx) \\
E \arrow[ur, shift left, "f"] \arrow[ur, shift right, "g"'] &
\end{tikzcd}$$

We can restrict the previous diagram to

$$\begin{tikzcd}
\sum_{e:E}(\chi\, e) \arrow[r, shift left, "\pi_1\circ f\circ\pi_1"] \arrow[r, shift right, "\pi_1\circ g\circ\pi_1"'] \arrow[d, "\text{dense}"'] & \sum_{x:A}(Bx) \\
E \arrow[ur, shift left, "\pi_1\circ f"] \arrow[ur, shift right, "\pi_1\circ g"'] &
\end{tikzcd}$$

and as A is separated, $\pi_1 \circ f = \pi_2 \circ g$. For the second components, let $x : E$. Notice that for all $x : E$, $\sum_{y:E} x = y$ has a dense subobject, $\sum_{y:\sum_{e:E}(\chi\, e)} x = y_1$:

$$\begin{tikzcd}
\sum_{y:\sum_{e:E}(\chi\, e)} x = y_1 \arrow[r, shift left, "\pi_2\circ f\circ\pi_1\circ\pi_1"] \arrow[r, shift right, "\pi_2\circ g\circ\pi_1\circ\pi_1"'] \arrow[d, "\text{dense}"'] & B\, x \\
\sum_{y:E} x = y \arrow[ur, shift left, "\pi_2\circ f\circ\pi_1"] \arrow[ur, shift right, "\pi_2\circ g\circ\pi_1"'] &
\end{tikzcd}$$

Using the sepatation property of $B\, x$, one can show that second components, transported correctly along the first components equality, are equal. The complete proof can be found in the formalisation. This proves the following proposition

Proposition 20. *$(\Box, \mu)$ defines a modality on* Type_{n+1}.

As this modality is just a step in the construction, we will not show that it is left exact, we will do it only for the sheafification modality.

C. From Separated Type to Sheaf

If T is already a separated type, the following lemma alllows us to build a sheaf :

Lemma 21. *Let $T : \mathrm{Type}_{n+1}$ be separated, and U be a sheaf. If T embeds in U, and is closed in U, then T is a sheaf.*

As any separated type T embeds in $\left(\mathrm{Type}_n^{\bigcirc}\right)^T$, it suffices to take the closure of T to get a sheaf, and the unit $\nu_T : T \to \star T$ is obvious.

Proposition 22. *$(\star, \nu)$ defines a reflective subuniverse.*

Proof. Let $T, Q : \mathrm{Type}_{n+1}$ such that Q is a sheaf. Let $f : T \to Q$. Because Q is a sheaf, it is in particular separated ; thus we can extend f to $\Box f : \Box T \to Q$.

But as $\star T$ is the closure of $\Box T$, $\Box T$ is dense into $\star T$, so the sheaf property of Q allows to extend $\Box f$ to $\star f : \star T \to Q$.

As all these steps are universal, the composition is. ☐

D. Summary

If $T : \mathrm{Type}_{n+1}$, the sheafification of T is

$$\star T = \sum_{u:T\to\mathrm{Type}_n^{\bigcirc}} \neg\neg \left\| \sum_{a:X} (\lambda t,\ \bigcirc (a = t)) = u \right\|$$

We already know that $\star$ defines a modality. It remains to show that this modality is left exact.

The last thing to prove is the cumulativity of our sheafifications :

Proposition 23. *If $T : \mathrm{Type}_n$, then $\bigcirc T = \star\widehat{T}$ (where $\widehat{T}$ is T seen as a* Type_{n+1}.

It will imply that $\star$ is compatible with HProp in the sense of proposition 15.

V. Future Works

Prove that the Giraud axiom holds in homotopy type theory.

Extend the sheafification to all types, even if non truncated.

VI. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

References

[1] S. MacLane and I. Moerdijk, *Sheaves in Geometry and Logic*. Springer-Verlag, 1992.
[2] P. Cohen and M. Davis, *Set theory and the continuum hypothesis*. WA Benjamin New York, 1966.
[3] M. Parigot, "Classical proofs as programs," in *Computational logic and proof theory*. Springer, 1993, pp. 263–276.
[4] G. Barthe and T. Uustalu, "Cps translating inductive and coinductive types," *ACM SIGPLAN Notices*, vol. 37, no. 3, pp. 131–142, 2002.
[5] S. Berardi, M. Bezem, and T. Coquand, "On the computational content of the axiom of choice," *Journal of Symbolic Logic*, pp. 600–622, 1998.
[6] J.-L. Krivine, "Dependent choice,'quote'and the clock," *Theoretical Computer Science*, vol. 308, no. 1, pp. 259–276, 2003.
[7] H. Herbelin, "A constructive proof of dependent choice, compatible with classical logic," in *Proceedings of the 2012 27th Annual IEEE/ACM Symposium on Logic in Computer Science*. IEEE Computer Society, 2012, pp. 365–374.

[8] Univalent Foundations Project, *Homotopy Type Theory: Univalent Foundations for Mathematics*. http://homotopytypetheory.org/book, 2013.
[9] P. Lumsdaine, "Higher inductive types: a tour of the menagerie," *Blog post available at http://homotopytypetheory. org/2011/04/24/higher-inductive-types-a-tour-of-the-menagerie*, 2011.
[10] P. Lumsdaine and M. Shulman, "Higher inductive types," *preparation*, 2013.
[11] C. Kapulkin, P. L. Lumsdaine, and V. Voevodsky, "The simplicial model of univalent foundations," *arXiv preprint arXiv:1211.2851*, 2012.
[12] J. Lurie, *Higher topos theory*, ser. Annals of mathematics studies. Princeton, N.J., Oxford: Princeton University Press, 2009. [Online]. Available: http://opac.inria.fr/record=b1128872
[13] E. Rijke and B. Spitters, "Sets in homotopy type theory," 2013.
[14] J. Avigad, C. Kapulkin, and P. L. Lumsdaine, "Homotopy limits in type theory," 2014.