

# Formal Topos on Category of Sets: The Comprehensive Definition using Type Theory

Maxim Sokhatsky

<sup>1</sup> National Technical University of Ukraine  
Igor Sikorsky Kyiv Polytechnical Institute  
August 19, 2018

## Abstract

The purpose of this work is to clarify all topos definitions, and variants of its components. Not much efforts was done to give all the examples, but one example, a topos on category of sets, is constructively presented at the finale.

As this crucial example definition is used in presheaf definition, the construction of category of sets is a mandatory exercise for any constructive topos library. We propose here cubicaltt<sup>1</sup> version of elementary topos on category of sets for demonstration of categorical semantics (from logic perspective) of the fundamental notion of set theory in mathematics.

Other disputed foundations for set theory could be taken as: ZFC, NBG, ETCS. We will distinct syntetically: i) category theory; ii) set theory in univalent foundations; iii) topos theory, grothendieck topos, elementary topos. For formulation of definitions and theorems only Martin-Löf Type Theory is requested. The proofs involve cubical type checker primitives.

**Keywords:** Martin-Löf Type Theory, Topos Theory, Cubical Type Theory

## Contents

<b>1</b>	<b>Category Theory</b>	<b>2</b>
<b>2</b>	<b>Set Theory</b>	<b>3</b>
<b>3</b>	<b>Topos Theory</b>	<b>5</b>
3.1	Topological Structure . . . . .	5
3.2	Grothendieck Topos . . . . .	6
3.3	Elementary Topos . . . . .	9

---

<sup>1</sup>Cubical Type Theory, <http://github.com/mortberg/cubicaltt>

## Intro

One can admit two topos theory lineages. One lineage takes its roots from published by Jean Leray in 1945 initial work on sheaves and spectral sequences. Later this lineage was developed by Henri Paul Cartan, André Weil. The peak of lineage was settled with works by Jean-Pierre Serre, Alexander Grothendieck, and Roger Godement.

Second remarkable lineage take its root from William Lawvere and Myles Tierney. The main contribution is the reformulation of Grothendieck topology by using subobject classifier.

## 1 Category Theory

First of all very simple category theory up to pullbacks is provided. We give here all definitions only to keep the context valid.

**Definition 1.** (Category Signature). The signature of category is a  $\sum_{A:U} A \rightarrow A \rightarrow U$  where  $U$  could be any universe. The  $\text{pr}_1$  projection is called  $\text{Ob}$  and  $\text{pr}_2$  projection is called  $\text{Hom}(a, b)$ , where  $a, b : \text{Ob}$ .

$\text{cat} : U = (A : U) * (A \rightarrow A \rightarrow U)$

**Definition 2.** (Precategory). More formal, precategory  $C$  consists of the following. (i) A type  $\text{Ob}_C$ , whose elements are called objects; (ii) for each  $a, b : \text{Ob}_C$ , a set  $\text{Hom}_C(a, b)$ , whose elements are called arrows or morphisms. (iii) For each  $a : \text{Ob}_C$ , a morphism  $1_a : \text{Hom}_C(a, a)$ , called the identity morphism. (iv) For each  $a, b, c : \text{Ob}_C$ , a function  $\text{Hom}_C(b, c) \rightarrow \text{Hom}_C(a, b) \rightarrow \text{Hom}_C(a, c)$  called composition, and denoted  $g \circ f$ . (v) For each  $a, b : \text{Ob}_C$  and  $f : \text{Hom}_C(a, b)$ ,  $f = 1_b \circ f$  and  $f = f \circ 1_a$ . (vi) For each  $a, b, c, d : A$  and  $f : \text{Hom}_C(a, b)$ ,  $g : \text{Hom}_C(b, c)$ ,  $h : \text{Hom}_C(c, d)$ ,  $h \circ (g \circ f) = (h \circ g) \circ f$ .

```
isPrecategory (C: cat): U
= (id: (x: C.1) -> C.2 x x)
* (c: (x y z: C.1) -> C.2 x y -> C.2 y z -> C.2 x z)
* (homSet: (x y: C.1) -> isSet (C.2 x y))
* (left: (x y: C.1) -> (f: C.2 x y) ->
  Path (C.2 x y) (c x x y (id x) f) f)
* (right: (x y: C.1) -> (f: C.2 x y) ->
  Path (C.2 x y) (c x y y f (id y)) f)
* ((x y z w: C.1) -> (f: C.2 x y) ->
  (g: C.2 y z) -> (h: C.2 z w) ->
  Path (C.2 x w) (c x z w (c x y z f g) h)
    (c x y w f (c y z w g h)))
```

```
carrier (C: precategory) : U
hom      (C: precategory) (a b: carrier C) : U
compose  (C: precategory) (x y z: carrier C)
          (f: hom C x y) (g: hom C y z) : hom C x z
```

**Definition 3.** (Terminal Object). Is such object  $\text{Ob}_C$ , that

$$\prod_{x,y:\text{Ob}_C} \text{isContr}(\text{Hom}_C(y,x)).$$

```
isTerminal (C: precategory) (y: carrier C): U
  = (x: carrier C) -> isContr (hom C x y)
terminal (C: precategory): U
  = (y: carrier C) * isTerminal C y
```

**Definition 4.** (Categorical Pullback). The pullback of the cospan  $A \xrightarrow{f} C \xleftarrow{g} B$  is a object  $A \times_C B$  with morphisms  $pb_1 : \times_C \rightarrow A$ ,  $pb_2 : \times_C \rightarrow B$ , such that diagram commutes:

$$\begin{array}{ccc} A \times_C B & \xrightarrow{pb_2} & B \\ f \downarrow & \searrow & \downarrow pb_1 \\ A & \xrightarrow{g} & C \end{array}$$

Pullback  $(\times_C, pb_1, pb_2)$  must be universal, means for any  $(D, q_1, q_2)$  for which diagram also commutes there must exists a unique  $u : D \rightarrow \times_C$ , such that  $pb_1 \circ u = q_1$  and  $pb_2 \circ u = q_2$ .

```
homTo (C: precategory) (X: carrier C): U
  = (Y: carrier C) * hom C Y X
cospan (C: precategory): U
  = (X: carrier C) * ( _: homTo C X) * homTo C X
cospanCone (C: precategory) (D: cospan C): U
  = (W: carrier C) * hasCospanCone C D W
cospanConeHom (C: precategory) (D: cospan C)
  (E1 E2: cospanCone C D) : U
  = (h: hom C E1.1 E2.1) * isCospanConeHom C D E1 E2 h
isPullback (C: precategory) (D: cospan C) (E: cospanCone C D) : U
  = (h: cospanCone C D) -> isContr (cospanConeHom C D h E)
hasPullback (C: precategory) (D: cospan C) : U
  = (E: cospanCone C D) * isPullback C D E
```

## 2 Set Theory

Here is given the  $\infty$ -groupoid model of sets.

**Definition 5.** (Mere proposition, PROP). A type  $P$  is a mere proposition if for all  $x, y : P$  we have  $x = y$ :

$$\text{isProp}(P) = \prod_{x,y:P} (x = y).$$

**Definition 6.** (0-type). A type  $A$  is a 0-type if for all  $x, y : A$  and  $p, q : x =_A y$  we have  $p = q$ .

**Definition 7.** (1-type). A type  $A$  is a 1-type if for all  $x, y : A$  and  $p, q : x =_A y$  and  $r, s : p =_{=A} q$ , we have  $r = s$ .

**Definition 8.** (A set of elements, SET). A type  $A$  is a SET if for all  $x, y : A$  and  $p, q : x = y$ , we have  $p = q$ :

$$\text{isSet}(A) = \prod_{x, y : A} \prod_{p, q : x = y} (p = q).$$

**Definition 9.** `data N = Z | S (n : N)`

```
n_grpd (A: U) (n: N): U = (a b: A) -> rec A a b n where
  rec (A: U) (a b: A) : (k: N) -> U
    = split { Z -> Path A a b ; S n -> n_grpd (Path A a b) n }
```

```
isContr (A: U): U = (x: A) * ((y: A) -> Path A x y)
isProp (A: U): U = n_grpd A Z
isSet (A: U): U = n_grpd A (S Z)
PROP : U = (X:U) * isProp X
SET : U = (X:U) * isSet X
```

**Definition 10.** (II-Contractability). If fiber is set thene path space between any sections is contractible.

```
setPi (A: U) (B: A -> U) (h: (x: A) -> isSet (B x)) (f g: Pi A B)
  (p q: Path (Pi A B) f g)
  : Path (Path (Pi A B) f g) p q
```

**Definition 11.** ( $\Sigma$ -Contractability). If fiber is set then  $\Sigma$  is set.

```
setSig (A:U) (B: A -> U) (base: isSet A)
  (fiber: (x:A) -> isSet (B x)) : isSet (Sigma A B)
```

**Definition 12.** (Unit type, 1). The unit 1 is a type with one element.

```
data unit = tt
unitRec (C: U) (x: C): unit -> C = split tt -> x
unitInd (C: unit -> U) (x: C tt): (z:unit) -> C z
  = split tt -> x
```

**Theorem 1.** (Category of Sets, **Set**). Sets forms a Category. All compositional theorems proved by using reflection rule of internal language. The proof that Hom forms a set is taken through  $\Pi$ -contractability.

```

Set: precategory = ((Ob, Hom), id, c, HomSet, L, R, Q) where
  Ob: U = SET
  Hom (A B: Ob): U = A.1 -> B.1
  id (A: Ob): Hom A A = idfun A.1
  c (A B C: Ob) (f: Hom A B) (g: Hom B C): Hom A C
    = o A.1 B.1 C.1 g f
  HomSet (A B: Ob): isSet (Hom A B) = setFun A.1 B.1 B.2
  L (A B: Ob) (f: Hom A B): Path (Hom A B)(c A A B (id A) f) f
    = refl (Hom A B) f
  R (A B: Ob) (f: Hom A B): Path (Hom A B)(c A B B f (id B)) f
    = refl (Hom A B) f
  Q (A B C D: Ob) (f: Hom A B) (g: Hom B C) (h: Hom C D)
    : Path (Hom A D) (c A C D (c A B C f g) h)
      (c A B D f (c B C D g h))
    = refl (Hom A D) (c A B D f (c B C D g h))

```

### 3 Topos Theory

Topos theory extends category theory with notion of topological structure but reformulated in a categorical way as a category of sheaves on a site or as one that has cartesian closure and subobject classifier. We give here two definitions.

#### 3.1 Topological Structure

**Definition 13.** (Topology). The topological structure on A (or topology) is a subset  $S \in A$  with following properties: i) any finite union of subsets of S is belong to S; ii) any finite intersection of subsets of S is belong to S. Subsets of S are called open sets of family S.

```

Structure topology (A : Type) := {
  open :> (A -> Prop) -> Prop;
  empty_open: open (empty _);
  full_open: open (full _);
  inter_open: forall u,
    open u -> forall v, open v
      -> open (inter A u v) ;
  union_open: forall s, (subset _ s open)
    -> open (union A s) }.

```

For fully functional general topology theorems and Zorn lemma you can refer to the Coq library <sup>2</sup>topology by Daniel Schepler.

---

<sup>2</sup><https://github.com/verimath/topology>

```

Definition subset (A: Type) (u v : A → Prop)
:= (forall x : A, u x → v x).
Definition disjoint (A: Type) (u v : A → Prop)
:= forall x, ~ (u x /\ v x).
Definition union (A: Type) (B : (A → Prop) → Prop)
:= fun x : A => exists U, B U /\ U x.
Definition inter (A: Type) (u v : A → Prop)
:= fun x : A => u x /\ v x.
Definition empty (A: Type)
:= fun x : A => False.
Definition full (A: Type)
:= fun x : A => True.

```

### 3.2 Grothendieck Topos

Grothendieck Topology is a calculus of coverings which generalizes the algebra of open covers of a topological space, and can exist on much more general categories. There are three variants of Grothendieck topology definition: i) sieves; ii) coverage; iii) covering families. A category have one of these three is called a Grothendieck site.

**Examples:** Zariski, flat, étale, Nisnevich topologies.

A sheaf is a presheaf (functor from opposite category to category of sets) which satisfies patching conditions arising from Grothendieck topology, and applying the associated sheaf functor to presheaf forces compliance with these conditions.

The notion of Grothendieck topos is a geometric flavour of topos theory, where topos is defined as category of sheaves on a Grothendieck site with geometric morphisms as adjoint pairs of functors between topoi, that satisfy exactness properties. [2]

As this flavour of topos theory uses category of sets as a prerequisite, the formal construction of set topos is crucial in doing sheaf topos theory.

**Definition 14.** (Sieves). Sieves are a family of subfunctors

$$R \subset \text{Hom}_C(\_, U), U \in C,$$

such that following axioms hold: i) (base change) If  $R \subset \text{Hom}_C(\_, U)$  is covering and  $\phi : V \rightarrow U$  is a morphism of  $C$ , then the subfunctor

$$\phi^{-1}(R) = \{\gamma : W \rightarrow V \mid \phi \cdot \gamma \in R\}$$

is covering for  $V$ ; ii) (local character) Suppose that  $R, R' \subset \text{Hom}_C(\_, U)$  are subfunctors and  $R$  is covering. If  $\phi^{-1}(R')$  is covering for all  $\phi : V \rightarrow U$  in  $R$ , then  $R'$  is covering; iii)  $\text{Hom}_C(\_, U)$  is covering for all  $U \in C$ .

**Definition 15.** (Coverage). A coverage is a function assigning to each  $\text{Ob}_C$  the family of morphisms  $\{f_i : U_i \rightarrow U\}_{i \in I}$  called covering families, such that for any  $g : V \rightarrow U$  exist a covering family  $\{h : V_j \rightarrow V\}_{j \in J}$  such that each composite

$$h_j \circ g \text{ factors some } f_i:$$

$$\begin{array}{ccc} V_j & \xrightarrow{k} & U_i \\ \downarrow h & & \downarrow f_i \\ V & \xrightarrow{g} & U \end{array}$$

Co (C: precategory) (cod: carrier C) : U  
 = (dom: carrier C)  
 \* (hom C dom cod)

Delta (C: precategory) (d: carrier C) : U  
 = (index: U)  
 \* (index  $\rightarrow$  Co C d)

Coverage (C: precategory): U  
 = (cod: carrier C)  
 \* (fam: Delta C cod)  
 \* (coverings: carrier C  $\rightarrow$  Delta C cod  $\rightarrow$  U)  
 \* (coverings cod fam)

**Definition 16.** (Grothendieck Topology). Suppose category C has all pull-backs. Since C is small, a pretopology on C consists of families of sets of morphisms

$$\{\phi_\alpha : U_\alpha \rightarrow U\}, U \in C,$$

called covering families, such that following axioms hold: i) suppose that  $\phi_\alpha : U_\alpha \rightarrow U$  is a covering family and that  $\psi : V \rightarrow U$  is a morphism of C. Then the collection  $V \times_U U_\alpha \rightarrow V$  is a covering family for V. ii) If  $\{\phi_\alpha : U_\alpha \rightarrow U\}$  is covering, and  $\{\gamma_{\alpha,\beta} : W_{\alpha,\beta} \rightarrow U_\alpha\}$  is covering for all  $\alpha$ , then the family of composites

$$W_{\alpha,\beta} \xrightarrow{\gamma_{\alpha,\beta}} U_\alpha \xrightarrow{\phi_\alpha} U$$

is covering; iii) The family  $\{1 : U \rightarrow U\}$  is covering for all  $U \in C$ .

**Definition 17.** (Site). Site is a category having either a coverage, grothendieck topology, or sieves.

site (C: precategory): U  
 = (C: precategory) \* Coverage C

**Definition 18.** (Presheaf). Presheaf of a category C is a functor from opposite category to category of sets:  $C^{op} \rightarrow \text{Set}$ .

presheaf (C: precategory): U  
 = catfunctor (opCat C) Set

**Definition 19.** (Presheaf Category, **PSh**). Presheaf category **PSh** for a site  $\mathcal{C}$  is category where objects are presheaves and morphisms are natural transformations of presheaf functors.

**Definition 20.** (Sheaf). Sheaf is a presheaf on a site. In other words a presheaf  $F : \mathcal{C}^{op} \rightarrow \mathbf{Set}$  such that the canonical map of inverse limit

$$F(U) \rightarrow \varprojlim_{V \rightarrow U \in R} F(V)$$

is an isomorphism for each covering sieve  $R \subset \text{Hom}_{\mathcal{C}}(-, U)$ . Equivalently, all induced functions

$$\text{Hom}_{\mathcal{C}}(\text{Hom}_{\mathcal{C}}(-, U), F) \rightarrow \text{Hom}_{\mathcal{C}}(R, F)$$

should be bijections.

```
sheaf (C: precategory): U
= (S: site C)
  * presheaf S.1
```

**Definition 21.** (Sheaf Category, **Sh**). Sheaf category **Sh** is a category where objects are sheaves and morphisms are natural transformation of sheaves. Sheaf category is a full subcategory of category of presheaves **PSh**.

**Definition 22.** (Grothendieck Topos). Topos is the category of sheaves **Sh**( $\mathcal{C}, J$ ) on a site  $\mathcal{C}$  with topology  $J$ .

**Theorem 2.** (Giraud). A category  $\mathcal{C}$  is a Grothendieck topos iff it has following properties: i) has all finite limits; ii) has small disjoint coproducts stable under pullbacks; iii) any epimorphism is coequalizer; iv) any equivalence relation  $R \rightarrow E$  is a kernel pair and has a quotient; v) any coequalizer  $R \rightarrow E \rightarrow Q$  is stably exact; vi) there is a set of objects that generates  $\mathcal{C}$ .

**Definition 23.** (Geometric Morphism). Suppose that  $\mathcal{C}$  and  $\mathcal{D}$  are Grothendieck sites. A geometric morphism

$$f : \mathbf{Sh}(\mathcal{C}) \rightarrow \mathbf{Sh}(\mathcal{D})$$

consist of functors  $f_* : \mathbf{Sh}(\mathcal{C}) \rightarrow \mathbf{Sh}(\mathcal{D})$  and  $f^* : \mathbf{Sh}(\mathcal{D}) \rightarrow \mathbf{Sh}(\mathcal{C})$  such that  $f^*$  is left adjoint to  $f_*$  and  $f^*$  preserves finite limits. The left adjoint  $f^*$  is called the inverse image functor, while  $f_*$  is called the direct image. The inverse image functor  $f^*$  is left and right exact in the sense that it preserves all finite colimits and limits, respectively.



### 3.3 Elementary Topos

Giraud theorem was a synonymical topos definition involved only topos properties but not a site properties. That was step forward on predicative definition. The other step was made by Lawvere and Tierney, by removing explicit dependance on categorical model of set theory (as category of set is used in definition of presheaf). This information was hidden into subobject classifier which was well defined through categorical pullback and property of being cartesian closed (having lambda calculus as internal language).

Elementary topos doesn't involve 2-categorical modeling, so we can construct set topos without using functors and natural transformations (what we need in geometrical topos theory flavour). This flavour of topos theory more suited for logic needs rather than geometry, as its set properties are hidden under the predicative pullback definition of subobject classifier rather than functorial notation of presheaf functor. So we can simplify proofs at the homotopy levels, not to lift everything to 2-categorical model.

**Definition 24.** (Monomorphism). An morphism  $f : Y \rightarrow Z$  is a monic or mono if for any object  $X$  and every pair of parallel morphisms  $g_1, g_2 : X \rightarrow Y$  the

$$f \circ g_1 = f \circ g_2 \rightarrow g_1 = g_2.$$

More abstractly,  $f$  is mono if for any  $X$  the  $\text{Hom}(X, -)$  takes it to an injective function between hom sets  $\text{Hom}(X, Y) \rightarrow \text{Hom}(X, Z)$ .

```
mono (P: precategory) (Y Z: carrier P) (f: hom P Y Z): U
  = (X: carrier P) (g1 g2: hom P X Y)
  -> Path (hom P X Z) (compose P X Y Z g1 f)
              (compose P X Y Z g2 f)
  -> Path (hom P X Y) g1 g2
```

**Definition 25.** (Subobject Classifier[3]). In category  $C$  with finite limits, a subobject classifier is a monomorphism  $true : 1 \rightarrow \Omega$  out of terminal object  $1$ , such that for any mono  $U \rightarrow X$  there is a unique morphism  $\chi_U : X \rightarrow \Omega$  and

$$\begin{array}{ccc} U & \xrightarrow{k} & 1 \\ \downarrow & & \downarrow true \\ X & \xrightarrow{\chi_U} & \Omega \end{array}$$

pullback diagram:

```
subobjectClassifier (C: precategory): U
  = (omega: carrier C)
  * (end: terminal C)
  * (trueHom: hom C end.1 omega)
  * (chi: (V X: carrier C) (j: hom C V X) -> hom C X omega)
  * (square: (V X: carrier C) (j: hom C V X) -> mono C V X j
    -> hasPullback C (omega, (end.1, trueHom), (X, chi V X j)))
  * ((V X: carrier C) (j: hom C V X) (k: hom C X omega)
    -> mono C V X j
    -> hasPullback C (omega, (end.1, trueHom), (X, k))
    -> Path (hom C X omega) (chi V X j) k)
```

**Theorem 3.** (Category of Sets has Subobject Classifier).

**Definition 26.** (Cartesian Closed Categories). The category  $C$  is called cartesian closed if exists all: i) terminals; ii) products; iii) exponentials. Note that this definition lacks beta and eta rules which could be found in embedding **MLTT**.

```
isCCC (C: precategory): U
= (Exp: (A B: carrier C) -> carrier C)
  * (Prod: (A B: carrier C) -> carrier C)
  * (Apply: (A B: carrier C) -> hom C (Prod (Exp A B) A) B)
  * (P1: (A B: carrier C) -> hom C (Prod A B) A)
  * (P2: (A B: carrier C) -> hom C (Prod A B) B)
  * (Term: terminal C)
  * unit
```

**Theorem 4.** (Category of Sets has Cartesian Closure). As you can see from exp and pro we internalize  $\Pi$  and  $\Sigma$  types as SET instances, the isSet predicates are provided with contractability. Existence of terminals is proved by propPi. The same technique you can find in **MLTT** embedding.

```
cartesianClosure : isCCC Set
= (expo, prod, appli, proj1, proj2, term, tt) where
  exp (A B: SET): SET = (A.1 -> B.1, setFun A.1 B.1 B.2)
  pro (A B: SET): SET = (prod A.1 B.1, setSig A.1 (\(- : A.1)
    -> B.1) A.2 (\(- : A.1) -> B.2))
  expo: (A B: SET) -> SET = \ (A B: SET) -> exp A B
  prod: (A B: SET) -> SET = \ (A B: SET) -> pro A B
  appli: (A B: SET) -> hom Set (pro (exp A B) A) B
    = \ (A B: SET) -> \ (x: (pro (exp A B) A).1) -> x.1 x.2
  proj1: (A B: SET) -> hom Set (pro A B) A
    = \ (A B: SET) (x: (pro A B).1) -> x.1
  proj2: (A B: SET) -> hom Set (pro A B) B
    = \ (A B: SET) (x: (pro A B).1) -> x.2
  unitContr (x: SET) (f: x.1 -> unit) : isContr (x.1 -> unit)
    = (f, \ (z: x.1 -> unit) -> propPi x.1 (\(-:x.1)->unit)
      (\(x:x.1) -> propUnit) f z)
  term: terminal Set = ((unit, setUnit),
    \ (x: SET) -> unitContr x (\(z: x.1) -> tt))
```

Note that rules of cartesian closure forms a type theoretical language called lambda calculus.

**Definition 27.** (Elementary Topos). Topos is a precategory which is cartesian closed and has subobject classifier.

```
Topos (cat: precategory) : U
= (cartesianClosure: isCCC cat)
  * subobjectClassifier cat
```

**Theorem 5.** (Topos Definitions). Any Grothendieck topos is an elementary topos too. The proof is slightly based on results of Giraud theorem.

**Theorem 6.** (Category of Sets forms a Topos). There is a cartesian closure and subobject classifier for a category of sets.

```
internal : Topos Set
          = ( cartesianClosure , hasSubobject )
```

**Theorem 7.** (Freyd). Main theorem of topos theory[1]. For any topos  $C$  and any  $b : \text{Ob}_C$  relative category  $C \downarrow b$  is also a topos. And for any arrow  $f : a \rightarrow b$  inverse image functor  $f^* : C \downarrow b \rightarrow C \downarrow a$  has left adjoint  $\sum_f$  and right adjoint  $\prod_f$ .

## Conclusion

We gave here constructive definition of topology as finite unions and intersections of open subsets. Then make this definition categorically compatible by introducing Grothendieck topology in three different forms: sieves, coverage, and covering families. Then we defined an elementary topos and introduce category of sets, and proved that **Set** is cartesian closed, has object classifier and thus a topos.

This intro could be considered as a formal introduction to topos theory (at least of the level of first chapter) and you may evolve this library to your needs or ask to help porting or developing your application of topos theory to a particular formal construction.

## References

- [1] R. Goldblatt. *Topoi: The Categorical Analysis of Logic*. Studies in Logic and the Foundations of Mathematics. Elsevier Science, 2014.
- [2] J.F. Jardine. *Local Homotopy Theory*. Springer Monographs in Mathematics. Springer New York, 2015.
- [3] P.T. Johnstone. *Topos Theory*. Dover Books on Mathematics. Dover Publications, 2014.