Issue XXVIII: Categories with Representable Maps

Максим Сохацький [1]

[1] Національний технічний університет України
Київський політехнічний інститут імені Ігоря Сікорського
23 травня 2025 р.

Зміст

Анотація

This article presents a modern categorical framework, termed Categories with Representable Maps (CwR), designed to model structures for dependent type theories. Inspired by Uemura's work, the framework unifies related models such as categories with families, categories with attributes, comprehension categories, and natural models. We provide a comprehensive set of classical mathematical definitions and theorems, focusing on specialized categorical structures like fibrations, indexed categories, and representable maps, while establishing their properties and equivalences.

As example we present a categorical model of Martin-Löf Type Theory (MLTT-75) with dependent products (Π-types), dependent sums (Σ-types), and identity types (Id-types). The model is based on Grothendieck fibrations and Uemura's categories with representable maps, generalizing Awodey's natural models. Formal definitions are provided, with pullback diagrams resembling Awodey's style.

1 Categories with Representable Maps

The Categories with Representable Maps (CwR) framework offers a robust foundation for categorical semantics, generalizing prior models used in type

theory. Assuming a base category $\mathcal{C}$ with all pullbacks, this framework builds on specialized structures to define representable maps and their properties, ensuring flexibility and unification across related categorical models. This article delineates the core definitions and theorems of the CwR framework, providing a concise yet complete theory.

Martin-Löf Type Theory (MLTT-75) is a dependent type theory with Π-types, Σ-types, and Id-types. We model its categorical semantics using a *category with representable maps* (CwR), starting from Grothendieck fibrations, as described in [1].

1.1 Definitions

Definition 1 (Fiber Category)**.** For a functor $p : \mathcal{E} \to \mathcal{C}$ and an object $c \in \mathcal{C}$, the *fiber category* $\mathcal{E}_c$ has:

- Objects: $e \in \mathcal{E}$ such that $p(e) = c$.
- Morphisms: $f : e' \to e$ in $\mathcal{E}$ such that $p(f) = \mathrm{id}_c$.

Definition 2 (Cartesian Morphism)**.** For a functor $p : \mathcal{E} \to \mathcal{C}$, a morphism $\phi : e' \to e$ in $\mathcal{E}$ is *Cartesian* if, for any $g : e'' \to e$ in $\mathcal{E}$ and $h : p(e'') \to p(e')$ in $\mathcal{C}$ with $p(g) = p(\phi) \circ h$, there exists a unique $k : e'' \to e'$ in $\mathcal{E}$ such that $p(k) = h$ and $g = \phi \circ k$.

Definition 3 (Grothendieck Fibration)**.** A functor $p : \mathcal{E} \to \mathcal{C}$ is a *Grothendieck fibration* if, for every $e \in \mathcal{E}$ and $f : c' \to p(e)$ in $\mathcal{C}$, there exists a Cartesian morphism $\phi : e' \to e$ in $\mathcal{E}$ such that $p(\phi) = f$.

Definition 4 (Grothendieck Construction)**.** For an indexed category $\Phi : \mathcal{C}^{\mathrm{op}} \to \mathbf{Cat}$, the *Grothendieck construction* produces a category $\int \Phi$ with:

- Objects: Pairs (c, x), where $c \in \mathcal{C}$, $x \in \Phi(c)$.
- Morphisms: From $(c', x') \to (c, x)$, pairs (f, α), where $f : c' \to c$ in $\mathcal{C}$, $\alpha : x' \to \Phi(f)(x)$ in $\Phi(c')$.
- Composition: For $(g, \beta) : (c'', x'') \to (c', x')$ and $(f, \alpha) : (c', x') \to (c, x)$, the composite is $(f \circ g, \Phi(g)(\alpha) \circ \beta)$.

The functor $p : \int \Phi \to \mathcal{C}$, mapping $(c, x) \mapsto c$, $(f, \alpha) \mapsto f$, is a Grothendieck fibration.

Definition 5 (Discrete Fibration)**.** A functor $p : \mathcal{E} \to \mathcal{C}$ is a *discrete fibration* if, for every $e \in \mathcal{E}$ and $f : c' \to p(e)$ in $\mathcal{C}$, there exists a unique $\tilde{f} : e' \to e$ in $\mathcal{E}$ such that $p(\tilde{f}) = f$.

Definition 6 (Indexed Category)**.** An *indexed category* over $\mathcal{C}$ is a functor $\Phi : \mathcal{C}^{\mathrm{op}} \to \mathbf{Cat}$. For each $c \in \mathcal{C}$, $\Phi(c)$ is a category, and for each $f : c' \to c$, $\Phi(f) : \Phi(c) \to \Phi(c')$ is a functor.

Definition 7 (Representable Functor)**.** A functor $F : \mathcal{C}^{op} \to \mathbf{Set}$ is *representable* if there exists $c \in \mathcal{C}$ such that $F \cong \mathrm{Hom}_{\mathcal{C}}(-, c)$.

Definition 8 (Representable Map)**.** In a category $\mathcal{C}$ with pullbacks, a morphism $f : A \to B$ is *representable* if it belongs to a class Rep(f) satisfying:

- *Pullback stability*: For every $g : C \to B$, the pullback $P = C \times_B A$ exists with projections $h_1 : P \to A$, $h_2 : P \to C$, and Rep(h_2).
- *Universality*: For any Q with $q_1 : Q \to A$, $q_2 : Q \to C$ such that $f \circ q_1 = g \circ q_2$, there exists a unique $u : Q \to P$ such that $h_1 \circ u = q_1$, $h_2 \circ u = q_2$.

Definition 9 (CwR)**.** A category with representable maps (CwR) is a category with a class of morphisms (representable maps) that are pullback-stable and exponentiable, generalizing Awodey's natural models. A *category with representable maps* (CwR) is a structure with:

- A category $\mathcal{C}$.
- A predicate $\mathrm{Rep} : \mathcal{C}.\mathrm{Hom}(A, B) \to \mathrm{Prop}$ for representable maps.
- *Pullback stability*: For every $f : A \to B$ with Rep(f) and $g : C \to B$, there exists a pullback P with morphisms $h_1 : P \to A$, $h_2 : P \to C$ such that $f \circ h_1 = g \circ h_2$, Rep($h_2$), and P is universal.
- *Exponentiability*: For every $f : A \to B$ with Rep(f), there exists $\Pi_f : \mathrm{Ob}$ and $\pi : \Pi_f \to B$ with Rep(π), such that for any $g : C \to A$, there exists $h : C \to \Pi_f$ with $\pi \circ h = f \circ g$.

```
structure CwR where
  cat : Category
  Rep : ∀ {A B : cat.Ob}, cat.Hom A B → Prop
  pullback : ∀ {A B C : cat.Ob} {f : cat.Hom A B},
    Rep f → (g : cat.Hom C B) →
    ∃ (P : cat.Ob) (∃ (h1 : cat.Hom P A) (∃ (h2 : cat.Hom P C)
      (cat.comp f h1 = cat.comp g h2 ∧
       Rep h2 ∧
       ∀ (Q : cat.Ob) (q1 : cat.Hom Q A) (q2 : cat.Hom Q C),
       cat.comp f q1 = cat.comp g q2 →
       ∃ (u : cat.Hom Q P)
         (cat.comp h1 u = q1 ∧ cat.comp h2 u =q2))))
  exponentiable : ∀ {A B : cat.Ob} {f : cat.Hom A B},
    Rep f →
    ∃ (Pi_f : cat.Ob) (∃ (pi : cat.Hom Pi_f B)
      (Rep pi ∧
       ∀ (C : cat.Ob) (g : cat.Hom C A),
       ∃ (h : cat.Hom C Pi_f) (cat.comp pi h =cat.comp f g)))
```

1.2 Theorems

The CwR framework is supported by five theorems that establish its properties and connections to related categorical structures.

Theorem 1 (Fibration-Indexed Category Equivalence)**.** For any indexed category $\Phi : \mathcal{C}^{\mathrm{op}} \to \mathbf{Cat}$, the Grothendieck construction produces a Grothendieck fibration $\mathtt{p} : \int \Phi \to \mathcal{C}$, and every Grothendieck fibration arises as the Grothendieck construction of some indexed category.

Theorem 2 (Representable Map Stability)**.** In a CwR $(\mathcal{C}, \mathrm{Rep}, \Pi)$, the class of representable maps is closed under pullback stability, and every representable map $\mathtt{f} : \mathtt{A} \to \mathtt{B}$ induces a representable morphism $\pi_{\mathtt{f}} : \Pi_{\mathtt{f}} \to \mathtt{B}$.

Theorem 3 (Discrete Fibration Representation)**.** Every discrete fibration $\mathtt{p} : \mathcal{E} \to \mathcal{C}$ corresponds to a representable map in the slice category $\mathcal{C}/\mathtt{c}$ for some $\mathtt{c} \in \mathcal{C}$, and every representable map induces a discrete fibration in a suitable slice category.

Theorem 4 (Framework Equivalence)**.** Every CwR $(\mathcal{C}, \mathrm{Rep}, \Pi)$ can be equipped with a structure equivalent to a category with families, or natural model under the existence of terminal objects.

1.3 Example MLTT-75 Model

We model MLTT-75 in a CwR, interpreting contexts, types, terms, and type formers.

Definition 10 (MLTT-75 Model)**.** Given a CwR $\mathcal{C}$, the model of MLTT-75 is defined as:

- *Contexts*: Objects $\Gamma \in \mathcal{C}.\mathrm{Ob}$.
- *Types*: Pairs $(A, f : A \to \Gamma)$ with $\mathrm{Rep}(f)$, representing A in context Γ.
- *Terms*: Morphisms $t : \Gamma \to A$ such that $f \circ t = \mathrm{id}_\Gamma$, i.e., sections of f.
- *Context extension*: For $\Gamma \vdash A$, the context $\Gamma, x : A$ is the pullback of $f : A \to \Gamma$ along id_Γ.
- *Type formers*: Π-types, Σ-types, and Id-types, defined via exponentials, pullbacks, and diagonals.

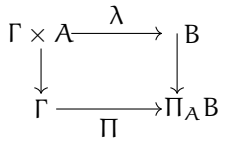
```
structure MLTT75 (cwr : CwR) where
  Context : Type
  Context := cwr.cat.Ob

  Type : Context → Type
  Type Γ := ∃ (A : cwr.cat.Ob)
    (∃ (f : cwr.cat.Hom A Γ) (cwr.Rep f))

  Term : ∀ (Γ : Context), Type Γ → Type
  Term Γ (∃ A (∃ f _))
    := ∃ (t : cwr.cat.Hom Γ A)
         (cwr.cat.comp f t = cwr.cat.id)

  ContextExt : ∀ (Γ : Context), Type Γ → Context
  ContextExt Γ (∃ A (∃ f rf)) := (cwr.pullback rf cwr.cat.id).fst
```

1.4 Π-Types

For $\Gamma \vdash A : \text{Type}$ and $\Gamma, x : A \vdash B : \text{Type}$, the Π-type $\Pi_{x:A} B$ is formed using the exponential in the slice category.

```
PiType : ∀ (Γ : Context) (A : Type Γ), Type (ContextExt Γ A) → Type Γ
PiType Γ (∃ A (∃ f rf)) (∃ B (∃ g rg)) :=
  let exp := cwr.exponentiable rf
  ∃ exp.fst (∃ exp.snd.fst exp.snd.snd.fst)
```

The constructor λ forms terms of $\Pi_{x:A} B$. The pullback diagram is:

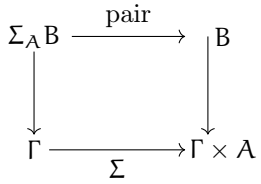


1.5 Σ-Types

For $\Gamma \vdash A : \text{Type}$ and $\Gamma, x : A \vdash B : \text{Type}$, the Σ-type $\Sigma_{x:A} B$ is the composition via pullback.

```
SigmaType : ∀ (Γ : Context) (A : Type Γ), Type (ContextExt Γ A) → Type Γ
SigmaType Γ (∃ A (∃ f rf)) (∃ B (∃ g rg)) :=
  let pull := cwr.pullback rg (cwr.cat.id)
  ∃ pull.fst (∃ pull.snd.fst pull.snd.snd.snd.fst)
```

The constructor pair forms terms of $\Sigma_{x:A} B$. The pullback diagram is:

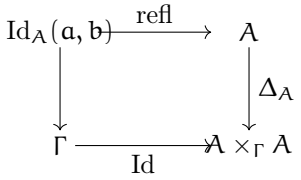


1.6 Id-Types

For $\Gamma \vdash A : \text{Type}$ and $a, b : A$, the identity type $\text{Id}_A(a, b)$ is formed using the diagonal map.

```
Diagonal : ∀ (Γ : Context) (A : Type Γ),
  cwr.cat.Hom (A.fst) (cwr.pullback A.snd.fst cwr.cat.id).fst
Diagonal Γ (∃ A (∃ f _))
  := (cwr.cat.id, cwr.cat.id, rfl)

IdType : ∀ (Γ : Context) (A : Type Γ) (a b : Term Γ A), Type Γ
IdType Γ (∃ A (∃ f rf)) (∃ a _) (∃ b _) :=
  let pull := cwr.pullback rf (Diagonal Γ (∃ A (∃ f rf)))
  ∃ pull.fst (∃ pull.snd.fst pull.snd.snd.snd.fst)
```

The constructor refl forms terms of $\text{Id}_A(a, a)$. The pullback diagram is:



1.7 Conclusion

The CwR framework provides a unified and flexible foundation for categorical semantics, integrating fibrations, indexed categories, and representable maps. Its definitions and theorems ensure robustness and connectivity to related categorical models, making it a powerful tool for theoretical and applied category theory.

Література

[1] Uemura, T., "A general framework for the semantics of type theory," arXiv:1904.04097.

[2] Awodey, S., "Natural models of homotopy type theory," Mathematical Structures in Computer Science, 2018.