Category Theory for Anders

Groupoid Infinity

May 2026

Abstract

This article describes the design and implementation of the categorical library in the Anders proof assistant. We focus on incorporating the design principles from Jacques Carette et al (2021) formalization in Agda, specifically regarding definitional duality and the use of redundant properties for free op-theorems in non-cubical foundations, with native cubical Rezk competion for univalent categories.

Contents

1 Introduction

The Anders [1] categorical library is designed to be a minimal yet powerful implementation of category theory in a pure Martin-Löf Type Theory (MLTT) with cubical CCHM extensions. The library [2] prioritizes proof-relevance (even for units), univalence, and duality-friendliness.

2 Definitional Duality

In Anders, leveraging its native cubical foundations (such as CCHM), the opposite category operation $(\cdot)^{\text{op}}$ is a definitional involution:

$$(\mathcal{C}^{\text{op}})^{\text{op}} \equiv \mathcal{C} \tag{1}$$

This property provides "duality for free," where any result established for a category $\mathcal{C}$ is immediately applicable to its dual without the need for explicit transport along an isomorphism.

We have retained the precategory signature with additional (redundant in cubical setting) properties — specifically symmetric laws like `assoc-inv` and `id-id` — primarily to ensure compatibility for proof extraction to Coq/HoTT and other formalizations [3] that lack a multidimensional interval. While the Agda formalization [1] introduces these properties to achieve definitional duality in systems where it is not native, in Anders they are not strictly required for duality purposes. Instead, their inclusion is a deliberate design choice to facilitate interoperability and extraction to systems without native cubical support.

[1] `https://github.com/groupoid/anders`

[2] `./library/mathematics/categories`

[3] `https://christine.groupoid.space`

3 Cubical Foundations

Unlike the Agda formalization which often relies on setoids to maintain compatibility with different Agda modes, Anders is natively cubical. This allows us to use Path types for equality between morphisms, which are proof-relevant by default.

Working with *hSet* in HoTT offers several significant advantages over a setoid-based approach:

- **Definitional Simplification**: By using *hSets*, we avoid the overhead of manually managing equivalence relations and congruence proofs.
- **Path-Based Equality**: Native Path types allow us to leverage cubical primitives for transport and composition, which is more direct than setoid-based rewriting.
- **Univalence Integration**: Univalent categories require an equivalence between paths and isomorphisms; this is a natural statement in HoTT/Cubical but becomes highly complex when using setoids to represent the underlying "sets".
- **Higher Structures**: HoTT allows for a more uniform treatment of higher categories, where the 0-type (hSet) restriction on morphisms can be naturally generalized to higher groupoids.

A univalent category in Anders is a precategory where the map from paths between objects to isomorphisms is an equivalence.

4 Core Library Structure

The core library consists of the following modules:

- `category.anders`: Basic definitions of precategories and categories.
- `functor.anders`: Functors and their compositions.
- `natural.anders`: Natural transformations and isomorphisms.
- `adjunction.anders`: Adjunctions between categories.
- `yoneda.anders`: Yoneda lemma.
- `universal.anders`: Universal Mapping Property.

5 Natural Transformations and Isomorphisms

Natural transformations are implemented as Sigma types consisting of the component map η and a naturality condition. We have formalized the definitional duality for natural transformations: if $\alpha : F \Rightarrow G$ is a natural transformation between functors $F, G : \mathcal{C} \to \mathcal{D}$, then α also defines a natural transformation $\alpha^{\mathrm{op}} : G^{\mathrm{op}} \Rightarrow F^{\mathrm{op}}$ between the dual functors. In Anders, this duality is definitional, meaning $(\alpha^{\mathrm{op}})^{\mathrm{op}} \equiv \alpha$ (after normalization process).

6 Adjunctions

Adjunctions $L \dashv R$ between categories $\mathcal{C}$ and $\mathcal{D}$ are defined via the unit-counit formulation:

- A unit natural transformation $\eta : \mathrm{id}_{\mathcal{C}} \Rightarrow R \circ L$.
- A counit natural transformation $\varepsilon : L \circ R \Rightarrow \mathrm{id}_{\mathcal{D}}$.
- Two triangle identities (zig and zag).

Due to the current limitations of the Anders compiler regarding named projections within nested Sigma types, the implementation uses numeric indices (e.g., `u.1` for η) to ensure stable typechecking.

7 Equivalences of Categories

A strong equivalence between two categories $\mathcal{C}$ and $\mathcal{D}$ is a pair of functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ equipped with natural isomorphisms:

- $\eta : \mathrm{id}_{\mathcal{C}} \cong G \circ F$
- $\varepsilon : F \circ G \cong \mathrm{id}_{\mathcal{D}}$

We have implemented `opEquivalence`, which shows that an equivalence $e : \mathcal{C} \cong \mathcal{D}$ induces an equivalence $e^{\mathrm{op}} : \mathcal{C}^{\mathrm{op}} \cong \mathcal{D}^{\mathrm{op}}$. To ensure definitional duality, we utilize `symNatIso` to adjust the directions of the unit and counit in the dualized structure.

8 The Category of Categories

The category of small precategories, denoted $\mathcal{CAT}$, is formalized as a large category (of type U_2). The objects are precategories of type U_1, and morphisms are functors of type U. While $\mathcal{CAT}$ is naturally a 2-category, we provide its 1-categorical structure by specifying identity functors and functor composition. The operation of taking the opposite category is currently provided as a definitional involution on the objects of $\mathcal{CAT}$.

9 The Yoneda Lemma

The Yoneda Lemma is a fundamental result in category theory that relates natural transformations between a hom-functor and an arbitrary functor to the values of that functor. For a small precategory $\mathcal{C}$, an object $A \in \mathcal{C}$, and a functor $F : \mathcal{C} \to \mathcal{SET}$, the lemma states:

$$\mathrm{Nat}(\mathcal{C}(A, -), F) \cong F(A) \tag{2}$$

Our implementation addresses several technical challenges inherent in formalizing the Yoneda Lemma:

- **Large Target Category**: Standard functors in our library are restricted to small precategories. Since $\mathcal{SET}$ is a large category, we manually define `catfunSET` and `nattransSET` using Sigma types that accommodate objects from higher universes.

- **Cubical Isomorphism**: The isomorphism is established via the projection of `YonedaMap` and its inverse `YonedaInv`. The computability property of beta reduction `Yoneda-`β holds definitionally, while uniqueness property of isomorphism inverse `Yoneda-`η is proved by establishing a path between natural transformations.

- **Proof Relevance**: Proof Irrelevance is disabled even for Units, Path proofs are relevant by design.

- **Identification of Transformations**: To prove that two natural transformations are equal, we utilize `nattransSET-isProp`, which shows that the naturality condition for functors into $\mathcal{SET}$ is a mere proposition. This allows us to use `lemSig'` from the foundation library to establish the equality by only checking the component map η.

10 Conclusion

The Anders categorical library continues to expand by incorporating higher-level structures while maintaining strict adherence to definitional duality and cubical principles. The use of symmetric laws and explicit path-based proofs allows for a clean and performant formalization that avoids the overhead of traditional setoid-based approaches preserving their compatibility with this categorical base library.

References

[1] Jason Z. S. Hu and Jacques Carette. *Formalizing Category Theory in Agda*. CPP 2021.