Category theory in dedukti/lambdapi

Luc Chabassier

Wednesday 27th September, 2023

Categories: definitions

Partie 1

One collection of morphisms

A category $\mathscr{C}$ is a pair of sets $\mathscr{C}, \mathrm{Hom}(\mathscr{C})$, called the *objects* and *morphisms* of $\mathscr{C}$, along with :

- Two functions $s, d : \mathrm{Hom}(\mathscr{C}) \to \mathscr{C}$ called the *source* and *target*
- A function $i : \mathscr{C} \to \mathrm{Hom}(\mathscr{C})$ called the *identities*
- A function $\circ : \mathrm{Hom}(\mathscr{C})\ {}_s{\times}_d\ \mathrm{Hom}(\mathscr{C}) \to \mathrm{Hom}(\mathscr{C})$ called the *composition*

One collection of morphisms

A category $\mathscr{C}$ is a pair of sets $\mathscr{C}, \mathrm{Hom}(\mathscr{C})$, called the *objects* and *morphisms* of $\mathscr{C}$, along with :

- Two functions $s, d : \mathrm{Hom}(\mathscr{C}) \to \mathscr{C}$ called the *source* and *target*
- A function $i : \mathscr{C} \to \mathrm{Hom}(\mathscr{C})$ called the *identities*
- A function $\circ : \mathrm{Hom}(\mathscr{C})\,{}_s\!\times_d \mathrm{Hom}(\mathscr{C}) \to \mathrm{Hom}(\mathscr{C})$ called the *composition*

Such that :

- $\forall x \in \mathscr{C}, s(i(x)) = x \wedge d(i(x)) = x$

One collection of morphisms

A category $\mathscr{C}$ is a pair of sets $\mathscr{C}, \mathrm{Hom}(\mathscr{C})$, called the *objects* and *morphisms* of $\mathscr{C}$, along with :

- Two functions $s, d : \mathrm{Hom}(\mathscr{C}) \to \mathscr{C}$ called the *source* and *target*
- A function $i : \mathscr{C} \to \mathrm{Hom}(\mathscr{C})$ called the *identities*
- A function $\circ : \mathrm{Hom}(\mathscr{C})\ {}_s\!\times_d\ \mathrm{Hom}(\mathscr{C}) \to \mathrm{Hom}(\mathscr{C})$ called the *composition*

Such that :

- $\forall x \in \mathscr{C}, s(i(x)) = x \wedge d(i(x)) = x$
- $\forall m_1, m_2 \in \mathrm{Hom}(\mathscr{C}), s(m_2 \circ m_1) = s(m_1) \wedge d(m_2 \circ m_1) = d(m_2)$

One collection of morphisms

A category $\mathscr{C}$ is a pair of sets $\mathscr{C}, \mathrm{Hom}(\mathscr{C})$, called the *objects* and *morphisms* of $\mathscr{C}$, along with :

- Two functions $s, d : \mathrm{Hom}(\mathscr{C}) \to \mathscr{C}$ called the *source* and *target*
- A function $i : \mathscr{C} \to \mathrm{Hom}(\mathscr{C})$ called the *identities*
- A function $\circ : \mathrm{Hom}(\mathscr{C})\, {}_s\!\times_d\, \mathrm{Hom}(\mathscr{C}) \to \mathrm{Hom}(\mathscr{C})$ called the *composition*

Such that :

- $\forall x \in \mathscr{C}, s(i(x)) = x \wedge d(i(x)) = x$
- $\forall m_1, m_2 \in \mathrm{Hom}(\mathscr{C}), s(m_2 \circ m_1) = s(m_1) \wedge d(m_2 \circ m_1) = d(m_2)$
- $\forall m \in \mathrm{Hom}(\mathscr{C}), m \circ i(s(m)) = m \wedge i(d(m)) \circ m = m$

One collection of morphisms

A category $\mathscr{C}$ is a pair of sets $\mathscr{C}, \mathrm{Hom}(\mathscr{C})$, called the *objects* and *morphisms* of $\mathscr{C}$, along with :

- Two functions $s, d : \mathrm{Hom}(\mathscr{C}) \to \mathscr{C}$ called the *source* and *target*
- A function $i : \mathscr{C} \to \mathrm{Hom}(\mathscr{C})$ called the *identities*
- A function $\circ : \mathrm{Hom}(\mathscr{C}) \,{}_s{\times}_d\, \mathrm{Hom}(\mathscr{C}) \to \mathrm{Hom}(\mathscr{C})$ called the *composition*

Such that :

- $\forall x \in \mathscr{C}, s(i(x)) = x \wedge d(i(x)) = x$
- $\forall m_1, m_2 \in \mathrm{Hom}(\mathscr{C}), s(m_2 \circ m_1) = s(m_1) \wedge d(m_2 \circ m_1) = d(m_2)$
- $\forall m \in \mathrm{Hom}(\mathscr{C}), m \circ i(s(m)) = m \wedge i(d(m)) \circ m = m$
- $\forall m_1, m_2, m_3 \in \mathrm{Hom}(\mathscr{C}), (m_3 \circ m_2) \circ m_1 = m_3 \circ (m_2 \circ m_1)$

A family of morphisms

A category $\mathscr{C}$ is a set $\mathscr{C}$ of *objects*, along with :

- A collection of sets $(\mathscr{C}(a, b))_{a,b \in \mathscr{C}}$ called the *morphisms*
- A collection of morphisms $(i_a)_{a \in \mathscr{C}}$ called the *identities*
- A collection of functions $(\circ_{a,b,c} : \mathscr{C}(b, c) \times \mathscr{C}(a, b) \to \mathscr{C}(a, c))_{a,b,c \in \mathscr{C}}$ called the *composition*

A family of morphisms

A category $\mathscr{C}$ is a set $\mathscr{C}$ of *objects*, along with :

- A collection of sets $(\mathscr{C}(a,b))_{a,b\in\mathscr{C}}$ called the *morphisms*
- A collection of morphisms $(i_a)_{a\in\mathscr{C}}$ called the *identities*
- A collection of functions $(\circ_{a,b,c} : \mathscr{C}(b,c)\times\mathscr{C}(a,b)\to\mathscr{C}(a,c))_{a,b,c\in\mathscr{C}}$ called the *composition*

Such that :

- $\forall a,b\in\mathscr{C}, \forall m\in\mathscr{C}(a,b), m\circ_{a,a,b} i_a = m \wedge i_b \circ_{a,b,b} m = m$

A family of morphisms

A category $\mathscr{C}$ is a set $\mathscr{C}$ of *objects*, along with :

- A collection of sets $(\mathscr{C}(a,b))_{a,b\in\mathscr{C}}$ called the *morphisms*
- A collection of morphisms $(i_a)_{a\in\mathscr{C}}$ called the *identities*
- A collection of functions $(\circ_{a,b,c} : \mathscr{C}(b,c) \times \mathscr{C}(a,b) \to \mathscr{C}(a,c))_{a,b,c\in\mathscr{C}}$ called the *composition*

Such that :

- $\forall a, b \in \mathscr{C}, \forall m \in \mathscr{C}(a,b), m \circ_{a,a,b} i_a = m \wedge i_b \circ_{a,b,b} m = m$
- the composition is associative

Intuition

- A generalisation of functions and sets

Intuition

- A generalisation of functions and sets
- A transitive/reflexive closure of a multigraph

Intuition

- A generalisation of functions and sets
- A transitive/reflexive closure of a multigraph
- A generalized monoid with partial operation

Intuition

- A generalisation of functions and sets
- A transitive/reflexive closure of a multigraph
- A generalized monoid with partial operation
- A proof-relevant poset

Usual formalisations

Partie 2

In HOL

Paper

"Category Theory with Adjunctions and Limits"[4]

- Uses a variation of the first definition, but without the object set.
- Uses a *null* morphism for invalid compositions.

In HOL

Paper

"Category Theory with Adjunctions and Limits"[4]

- Uses a variation of the first definition, but without the object set.
- Uses a *null* morphism for invalid compositions.

Advantages

Compositions can always be written.

In HOL

Paper

"Category Theory with Adjunctions and Limits"[4]

- Uses a variation of the first definition, but without the object set.
- Uses a *null* morphism for invalid compositions.

Advantages

Compositions can always be written.

Limitations

Proofs are littered with checking for the null morphism.

In dependently typed theory

Papers

- "Experience Implementing a Performant Category-Theory Library in Coq"[2](Coq)
- "Univalent categories and the Rezk completion"[1](Coq)
- "Formalizing of Category Theory in Agda"[3](Agda)

In dependently typed theory

```
Record Category := Category {
  object: Type;
  morphism: object -> object -> Type;
  identity: forall x, morphism x x;
  compose: forall s d d',
    C[d, d] -> C[s, d] -> C[s, d'];
  associativity: forall x1 x2 x3 x4
    (m1: C[x1,x2])(m2: C[x2,x3])(m3: C[x3,x4]),
    (m3 o m2) o m1 = m3 o (m2 o m1);
  left_identity: forall a b (f: C[a,b]), 1 o f = f;
  right_identity:
    forall a b (f: C[a,b]), f o 1 = f;
}.
```

In dependently typed theory

Advantages

- Intuitive to work with

In dependently typed theory

Advantages

- Intuitive to work with
- Closer to the usual formulation

In dependently typed theory

Advantages

- Intuitive to work with
- Closer to the usual formulation

In dependently typed theory

Advantages

- Intuitive to work with
- Closer to the usual formulation

Limitations

- Dependently typed

In dependently typed theory

Advantages

- Intuitive to work with
- Closer to the usual formulation

Limitations

- Dependently typed
- Higher structure

Higher structure

Remark

$(C(a,b), \cdot =_{C(a,b)} \cdot)$ is itself a category.

Higher structure

Remark

$(C(a,b), \cdot =_{C(a,b)} \cdot)$ is itself a category.

Problem

C is a weak 2-category.

Higher structure

Remark

$(C(a,b), \cdot =_{C(a,b)} \cdot)$ is itself a category.

Problem

C is a weak 2-category.

Need for additional hypothesis :

```
morphism_hset:
  forall a b (m1 m2: C[a,b])
             (p1 p2: m1 = m2),
    p1 = p2.
```

Lambdapi approach

Partie 3

Objective

Strict categories

Define categories in *LambdaPi* such that $m_1 = m_2$ implies that $m_1 \equiv m_2$ in lambdapi.

Objective

Strict categories

Define categories in *LambdaPi* such that $m_1 = m_2$ implies that $m_1 \equiv m_2$ in lambdapi.

Limitations

This can only work for categories where morphism equality is decidable.

Successes

Ad-hoc approach for :

- Discrete categories

Successes

Ad-hoc approach for :

- Discrete categories
- Linear categories

Successes

Ad-hoc approach for :

- Discrete categories
- Linear categories
- Product of strict categories

Successes

Ad-hoc approach for :

- Discrete categories
- Linear categories
- Product of strict categories
- *Coproduct of strict categories*

Successes

Ad-hoc approach for :

- Discrete categories
- Linear categories
- Product of strict categories
- *Coproduct of strict categories*
- Free categories

Successes

Ad-hoc approach for :

- Discrete categories
- Linear categories
- Product of strict categories
- *Coproduct of strict categories*
- Free categories
- Simplex category

Foundamental limitation

C strict category, C' category.

Objective

$F : C \to C'$ functor (think cubical sets. . .)

Foundamental limitation

C strict category, C' category.

Objective

$F : C \to C'$ functor (think cubical sets...)

Problem

$f \equiv g$ means that $F\, f \equiv F\, g$ for *confluence*.

Conclusion

Partie 4

Conclusion

- Interesting for simple categories

Conclusion

- Interesting for simple categories
- But limited for real developments

Conclusion

- Interesting for simple categories
- But limited for real developments

Conclusion

- Interesting for simple categories
- But limited for real developments

Other problems

- Size issues

Conclusion

- Interesting for simple categories
- But limited for real developments

Other problems

- Size issues
- Diagrammatic reasoning