Case Study on Displayed Monoidal Categories

Kobe Wullaert
Collaboration: B. Ahrens, R. Matthes

Delft University of Technology

June 21, 2022

Introduction

What

Formalization in UniMath:

1. Monoidal categories and functors,
2. displayed monoidal categories and monoidal sections.

[1] B. Ahrens, R. Matthes, A. Mörtberg:
https://doi.org/10.1145/3497775.3503678

Introduction

What

Formalization in UniMath:

1. Monoidal categories and functors,
2. displayed monoidal categories and monoidal sections.

Why

1. Semantics of linear type theories and logics.
2. Model of computation and string diagrams.
3. Our reason is technically rather complex, needed to describe substitution[1].

[1] B. Ahrens, R. Matthes, A. Mörtberg: https://doi.org/10.1145/3497775.3503678

Problem

Monoidal category has tensor product $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$.

Problem

Monoidal category has tensor product $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$.
When formalizing:

```
Lemma disp_binprod_transportf_pr2 (a b : C) (f g : a --> b)
      (a' b' : C') (f' g' : a' --> b')
      (x : D a) (y : D b) (x' : D' a') (y' : D' b')
      (ff : x -->[f] y) (ff' : x' -->[f'] y')
      (e : catbinprodmor f f' = catbinprodmor g g')
  : transportf (mor_disp _ _) (maponpaths (dirprod_pr2) e) ff'
    =
    pr2 (transportf (@mor_disp _ disp_binprod_data (a,,a') _ (x,, x') (_,, _)) e (ff,, ff')) .
```

Problem

Monoidal category has tensor product $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$.
When formalizing:

```
Lemma disp_binprod_transportf_pr2 (a b : C) (f g : a --> b)
      (a' b' : C') (f' g' : a' --> b')
      (x : D a) (y : D b) (x' : D' a') (y' : D' b')
      (ff : x -->[f] y) (ff' : x' -->[f'] y')
      (e : catbinprodmor f f' = catbinprodmor g g')
  : transportf (mor_disp _ _) (maponpaths (dirprod_pr2) e) ff'
    =
    pr2 (transportf (@mor_disp _ disp_binprod_data (a,,a') _ (x,, x') (_,, _)) e (ff,, ff')) .
```

$\rightsquigarrow$ Difficulties with constructing a bijective correspondence between two categorical notions.

Solution

Reformalize tensor product:

1. $\mathcal{C} \times \mathcal{C} \to \mathcal{C}$.

Solution

Reformalize tensor product:

1. $\mathcal{C} \times \mathcal{C} \to \mathcal{C}$.
2. $\mathcal{C} \to [\mathcal{C}, \mathcal{C}]$

Displayed categories

Formal framework to build categories from other categories.

Displayed categories

Formal framework to build categories from other categories.

Example

Category **PtSet** of pointed sets.

1. Objects: $(X \in \mathbf{Set}, x \in X)$
2. Morphisms: $(f \in \mathbf{Set}(X, Y), f(x) = y)$

Displayed categories

Formal framework to build categories from other categories.

Example

Category **PtSet** of pointed sets.

1. Objects: $(X \in \mathbf{Set}, x \in X)$
2. Morphisms: $(f \in \mathbf{Set}(X, Y), f(x) = y)$

Displayed category over $\mathcal{C}$

1. $X \in Ob(\mathcal{C}) \mapsto$ a type, e.g. structures on X.
2. $f \in \mathcal{C}(X, Y) \mapsto$ a type, e.g. preservation of the structure by f.

Displayed categories

Formal framework to build categories from other categories.

Example

Category **PtSet** of pointed sets.

1. Objects: $(X \in \mathbf{Set}, x \in X)$
2. Morphisms: $(f \in \mathbf{Set}(X, Y), f(x) = y)$

Displayed category over $\mathcal{C}$

1. $X \in Ob(\mathcal{C}) \mapsto$ a type, e.g. structures on X.
2. $f \in \mathcal{C}(X, Y) \mapsto$ a type, e.g. preservation of the structure by f.
3. Category $\mathcal{C}$ + displayed category $\mathcal{D} \to$ *total category* $\int_{\mathcal{C}} \mathcal{D}$.

Displayed categories

Formal framework to build categories from other categories.

Example

Category **PtSet** of pointed sets.

1. Objects: $(X \in \mathbf{Set}, x \in X)$
2. Morphisms: $(f \in \mathbf{Set}(X, Y), f(x) = y)$

Displayed category over $\mathcal{C}$

1. $X \in Ob(\mathcal{C}) \mapsto$ a type, e.g. structures on X.
2. $f \in \mathcal{C}(X, Y) \mapsto$ a type, e.g. preservation of the structure by f.
3. Category $\mathcal{C}$ + displayed category $\mathcal{D}$ $\to$ *total category* $\int_{\mathcal{C}} \mathcal{D}$.
4. Forgetful functor $U : \int_{\mathcal{C}} \mathcal{D} \to \mathcal{C}$.

Displayed categories

Formal framework to build categories from other categories.

Example

Category **PtSet** of pointed sets.

1. Objects: $(X \in \mathbf{Set}, x \in X)$
2. Morphisms: $(f \in \mathbf{Set}(X, Y), f(x) = y)$

Displayed category over $\mathcal{C}$

1. $X \in Ob(\mathcal{C}) \mapsto$ a type, e.g. structures on X.
2. $f \in \mathcal{C}(X, Y) \mapsto$ a type, e.g. preservation of the structure by f.
3. Category $\mathcal{C}$ + displayed category $\mathcal{D} \rightarrow$ *total category* $\int_{\mathcal{C}} \mathcal{D}$.
4. Forgetful functor $U : \int_{\mathcal{C}} \mathcal{D} \rightarrow \mathcal{C}$.

Displayed monoidal categories

Formal framework to build monoidal categories from other monoidal categories.

1. **PtSet** monoidal with $(X, x) \otimes (Y, y) :\equiv (X \times Y, (x, y))$.

Displayed monoidal categories

Formal framework to build monoidal categories from other monoidal categories.

1. **PtSet** monoidal with $(X, x) \otimes (Y, y) :\equiv (X \times Y, (x, y))$.
2. $X \times Y =: X \otimes_{\mathbf{Set}} Y$.

Displayed monoidal categories

Formal framework to build monoidal categories from other monoidal categories.

1. **PtSet** monoidal with $(X, x) \otimes (Y, y) :\equiv (X \times Y, (x, y))$.
2. $X \times Y =: X \otimes_{\mathbf{Set}} Y$.
3. Displayed tensor: Tensor product carries over to added structure and properties.

Displayed monoidal categories

Formal framework to build monoidal categories from other monoidal categories.

1. **PtSet** monoidal with $(X, x) \otimes (Y, y) :\equiv (X \times Y, (x, y))$.
2. $X \times Y =: X \otimes_{\mathbf{Set}} Y$.
3. Displayed tensor: Tensor product carries over to added structure and properties.
4. Monoidal category $\mathcal{C}$ + displayed monoidal category $\mathcal{D}$ $\to$ *total category* $\int_{\mathcal{C}} \mathcal{D}$ is monoidal.

Displayed monoidal categories

Formal framework to build monoidal categories from other monoidal categories.

1. **PtSet** monoidal with $(X, x) \otimes (Y, y) :\equiv (X \times Y, (x, y))$.
2. $X \times Y =: X \otimes_{\mathbf{Set}} Y$.
3. Displayed tensor: Tensor product carries over to added structure and properties.
4. Monoidal category $\mathcal{C}$ + displayed monoidal category $\mathcal{D}$ $\to$ *total category* $\int_{\mathcal{C}} \mathcal{D}$ is monoidal.
5. Forgetful functor $U : \int_{\mathcal{C}} \mathcal{D} \to \mathcal{C}$ is strict monoidal.

Displayed monoidal categories

Formal framework to build monoidal categories from other monoidal categories.

1. **PtSet** monoidal with $(X, x) \otimes (Y, y) :\equiv (X \times Y, (x, y))$.
2. $X \times Y =: X \otimes_{\mathbf{Set}} Y$.
3. Displayed tensor: Tensor product carries over to added structure and properties.
4. Monoidal category $\mathcal{C}$ + displayed monoidal category $\mathcal{D}$ $\rightarrow$ *total category* $\int_{\mathcal{C}} \mathcal{D}$ is monoidal.
5. Forgetful functor $U : \int_{\mathcal{C}} \mathcal{D} \rightarrow \mathcal{C}$ is strict monoidal.

Conclusion

We formalized in UniMath:

1. Displayed monoidal categories induce a strict monoidal functor from the total category to the base category.
2. Constructed cartesian monoidal categories.
3. Constructed **PtSet** and **BinOp** as total displayed monoidal categories.
4. Constructed category of pointed endofunctors as total displayed monoidal categories.
5. Introduced monoidal sections.
6. Correspondence between monoidal categories and one-object bicategories.
7. Solved the open conjecture/technical problem.
8. Currently in progress: Bicategory of univalent monoidal categories is univalent.