Univalent Double Categories

Van Der Weide, Niels; Rasekh, Nima; Ahrens, Benedikt; North, Paige Randall

DOI
10.1145/3636501.3636955

Publication date
2024

Document Version
Final published version

Published in
CPP 2024 - Proceedings of the 13th ACM SIGPLAN International Conference on Certified Programs and Proofs, Co-located with

Univalent Double Categories

Niels van der Weide
Radboud University Nijmegen
Netherlands
nweide@cs.ru.nl

Nima Rasekh
Max Planck Institute for Mathematics
Germany
rasekh@mpim-bonn.mpg.de

Benedikt Ahrens
Delft University of Technology
Netherlands
University of Birmingham
UK
B.P.Ahrens@tudelft.nl

Paige Randall North
Utrecht University
Netherlands
p.r.north@uu.nl

Abstract

Category theory is a branch of mathematics that provides a formal framework for understanding the relationship between mathematical structures. To this end, a category not only incorporates the data of the desired objects, but also "morphisms", which capture how different objects interact with each other. Category theory has found many applications in mathematics and in computer science, for example in functional programming.

Double categories are a natural generalization of categories which incorporate the data of two separate classes of morphisms, allowing a more nuanced representation of relationships and interactions between objects. Similar to category theory, double categories have been successfully applied to various situations in mathematics and computer science, in which objects naturally exhibit two types of morphisms. Examples include categories themselves, but also lenses, petri nets, and spans.

While categories have already been formalized in a variety of proof assistants, double categories have received far less attention. In this paper we remedy this situation by presenting a formalization of double categories via the proof assistant Coq, relying on the Coq UniMath library. As part of this work we present two equivalent formalizations of the definition of a double category, an unfolded explicit definition and a second definition which exhibits excellent formal properties via 2-sided displayed categories. As an application of the formal approach we establish a notion of univalent double category along with a univalence principle: equivalences of univalent double categories coincide with their identities.

CCS Concepts: • **Theory of computation** ⟶ **Type theory**; **Constructive mathematics**; **Logic and verification**.

Keywords: formalization of mathematics, category theory, double categories, univalent foundations

ACM Reference Format:
Niels van der Weide, Nima Rasekh, Benedikt Ahrens, and Paige Randall North. 2024. Univalent Double Categories. In *Proceedings of the 13th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP '24), January 15–16, 2024, London, UK*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3636501.3636955

1 Introduction

Double categories [13] are a categorical concept that captures more structure than a category. They are often succinctly defined as an internal (pseudo)category in the 2-category of categories. A double category has objects, two kinds of morphisms — called *vertical* and *horizontal*, respectively — and fillers for squares formed from horizontal and vertical morphisms. As such, a double category can capture two different kinds of morphisms (and their interplay) between mathematical objects.

Many mathematical objects are better understood within a double category than within a category; for instance, the double category of sets, functions, and relations. The objects of this double category are sets X, the vertical morphisms $X \longrightarrow Y$ are functions $X \longrightarrow Y$, the horizontal morphisms $X \longrightarrow\!\!\!\!\!\!\!+\; Y$ are relations, i.e. subsets of $X \times Y$. Considering this double category allows one to generalize classical set theory (largely overlapping with the generalization given by topos theory). Similarly, one can also consider the double category of categories, functors, and profunctors, and this has been used to great success to generalize category theory [33].

Applications of double categories have become ubiquitous in mathematics and computer science; see, for instance, its applications in systems theory [5, 9, 21] and programming languages theory [12, 22].

CPP '24, January 15–16, 2024, London, UK

ACM ISBN 979-8-4007-0488-8/24/01
https://doi.org/10.1145/3636501.3636955

In the present work we develop the notion of univalent double category and a library of univalent double categories in univalent foundations. Our main result states that the bicategory of univalent double categories is univalent. As a consequence, the type of identities $A = B$ between univalent double categories A and B coincides with the type $A \simeq B$ of equivalences from A to B. The proof of this result relies crucially on Voevodsky's univalence axiom. The result entails that any construction on univalent double categories can be transported across equivalences — an instance of the *univalence principle* [3].

Double categories consist of a lot of data (see Section 2), and morphisms of double categories — and morphisms between these morphisms — need to preserve that structure suitably. In other words, the bicategory of double categories is quite complicated. For this reason, a naïve, brute-force approach to proving univalence of this bicategory would lead to difficult proofs. Instead, we develop technology to build the bicategory of double categories in layers, using displayed bicategories [1]. We then prove every layer univalent, and obtain that their "total bicategory" — which is the desired bicategory of univalent double categories — is univalent, by a result from [1]. The key layer we consider in this approach is the layer of "2-sided displayed categories". These gadgets are a simple variation on the notion of displayed category [2]. Through their use, we derive a modular proof of univalence of the bicategory of univalent double categories; in particular, we can reuse an existing proof of univalence of univalent categories from [1].

Building the bicategory of univalent double categories in a layered way also gives rise to an interesting characterization of equivalences of double categories. In Section 7, we show that a double functor between univalent double categories is an adjoint equivalence if it is a strong double functor and an adjoint equivalence on the underlying 2-sided displayed category.

1.1 UniMath

In this section we provide a brief introduction to univalent foundations and `UniMath`, and fix notations used throughout the paper. By univalent foundations, we mean Martin-Löf type theory (MLTT) plus Voevodsky's univalence axiom. We use standard notation for the type and term formers of MLTT; in particular, we write $a = b$ for the type of identifications/equalities/paths from a to b.

Crucially, we rely on the notion of *homotopy level*, and, in particular, the notions of proposition and set of univalent foundations: a type X is a proposition if $\prod_{x,y:X} x = y$ is inhabited, and a set if the type $x = y$ is a proposition for all $x, y : X$. Hence, despite working in `Coq`, we do not rely on the universes Prop or Set.

We do not rely on any inductive types other than the ones specified in the prelude of `UniMath`, such as identity types, sum types, natural numbers, and booleans.

Our key result, the univalence principle for univalent double categories, relies on the univalence principle for types, also known as Voevodsky's Univalence Axiom. This axiom is added to `Coq` as a postulate in `UniMath`.

1.2 Computer Formalization

The formalization accompanying this paper is based on the UniMath library [34], a library of computer-checked mathematics in the univalent style. UniMath itself is based on the Coq proof assistant [29].

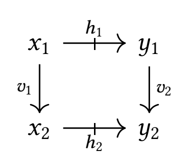
Our code has been integrated in the UniMath library in commit 5acbf27 (recorded as release v20231010), and it can be compiled with Coq 8.17.1. From this commit, we compiled an HTML documentation of UniMath; throughout this article we include links to this documentation, as in disp_bicat. The interested reader can type-check our definitions by following the compilation instructions of the UniMath library.

1.3 Synopsis

In Section 2 we informally review and motivate the notion of double category, and give an elementary, unfolded definition. The unfolded definition is easy to understand, but proving a univalence principle for it *directly* would be tough. For this reason, we introduce, in Section 3, the notion of *2-sided displayed categories*; we use these in Section 5 to build a bicategory of displayed categories that does not make use of the elementary definition. To prepare for this construction, we review the notion of displayed bicategory in Section 4. In Section 6 we construct several examples of univalent double categories. In Section 7 we give a characterization of adjoint equivalences, and of invertible 2-cells, in the bicategory of univalent double categories — that is, of equivalences of double categories and of invertible transformations between functors of univalent double categories.

2 Double Categories

In this section, we give a brief overview of the theory of double categories. Intuitively, a double category is a category with an extra class of morphisms. Morphisms in one class of morphisms are called *vertical* morphisms, and morphisms in the other class are called *horizontal* morphisms. We see horizontal morphisms as "extra" morphisms, and for those, the laws do not hold up to equality (see Remark 2.1). We denote the vertical morphisms by $x_1 \longrightarrow x_2$ and the horizontal morphisms by $x \nrightarrow y$. In addition, any double category features a collection of squares, parametrized by a boundary consisting of two horizontal and two vertical morphisms with "compatible" endpoints as follows:



Such squares are also denoted as $\begin{pmatrix} v_1 & {}^{h_1}_{h_2} & v_2 \end{pmatrix}$. For both the horizontal and the vertical morphisms we have identities and compositions. However, there is an essential difference between the two classes of morphisms: laws for the vertical morphisms hold up to equality, whereas the laws of horizontal morphisms hold up to a square. Concretely, this means that we have *unitor* and *associator* squares that witness the unitality and associativity of horizontal composition. In addition, this data is coherent: we also require the triangle and pentagon equation for this data.

Remark 2.1. The notion of double category comes in several flavors. For example, there is the notion of *strict double category*, and in those, unitality and associativity of composition holds as an equality. However, in the remainder of this paper, we look at *pseudo double categories*, and in those, composition of horizontal morphisms is only weakly unital and associative, up to *an invertible square* — see Items 10 to 12 of Definition 2.3. Pseudo double categories are a useful generalization of strict double categories. Some examples, such as spans (Example 6.3) and structured cospans (Example 6.5), are pseudo double categories, but not strict ones.

Double categories play a prominent role in applied category theory. For example, Clarke defined a double category of lenses [8], and lenses have become an important tool in the study of databases and datatypes; see, e.g., [7]. In addition, Baez and Master [6] defined a double category of Petri Nets, which are used in the study of parallel programs [16] and modeling hardware [24]. Baez and Courser defined a double category of structured cospans and of decorated cospans [4, 14], which are used to model open systems.

There are several approaches to defining the notion of double category, and each comes with their own merits and drawbacks. The most concise definition is that a double category is a pseudocategory internal to the bicategory of categories. While this definition is clean and short, its drawback is that composition is described using pullbacks, which makes it more cumbersome to work with. More concretely, let us assume we have two categories C_H and C_V together with functors $S, T : \mathsf{C}_H \longrightarrow \mathsf{C}_V$. If we were to use this definition, then a horizontal arrow from $x : \mathsf{C}_V$ to $y : \mathsf{C}_V$ would consist of an object $h : \mathsf{C}_H$ together with isomorphisms $S(h) \cong x$ and $T(h) \cong y$. In addition, the composition operation for horizontal arrows would take three objects $x, y, z : \mathsf{C}_V$, two horizontal arrows $h, k : \mathsf{C}_H$ and isomorphisms $S(h) \cong x$, $T(h) \cong y$, $S(k) \cong y$, and $T(k) \cong z$, and it returns a horizontal arrow $h \cdot k : \mathsf{C}_H$ together with isomorphisms $S(h \cdot k) \cong x$ and $T(h \cdot k) \cong z$.

Remark 2.2. Note that one could also look at categories internal to a 1-category instead of a bicategory. By looking at categories internal to the 1-category of strict categories, one obtains yet another notion of double category. This approach is taken in Lean [18, 20], where pullbacks are used directly, and in 1lab [30], where pullbacks are avoided by looking at the internal language of a presheaf category. However, this approach comes with a significant limitation: by looking at strict categories, one loses examples such as spans in Set (Example 6.3), and the square construction for univalent categories (Example 6.1). Note that the type of objects of internal categories in the 1-category of strict categories must form a set, since strict categories have a set of objects. However, we did not add such a requirement in Remark 2.1. See also our discussion of related work in Section 8 for information about formalizations of different notions of double category.

We can avoid pullbacks by going for an unfolded definition, which looks as follows:

Definition 2.3 (doublecategory)**.** A **double category** consists of

1. a category C called the **vertical category**;
2. for all objects $x : \mathsf{C}$ and $y : \mathsf{C}$, a type $x \nrightarrow y$ of **horizontal morphisms**;
3. for every object $x : \mathsf{C}$ a **horizontal identity** $\mathsf{id}_x : x \nrightarrow x$;
4. for all horizontal morphisms $h : x \nrightarrow y$ and $k : y \nrightarrow z$, a **horizontal composition** $h \odot k : x \nrightarrow z$;
5. for all horizontal morphisms $h : x_1 \nrightarrow y_1$ and $k : x_2 \nrightarrow y_2$ and vertical morphisms $v_x : x_1 \nrightarrow x_2$ and $v_y : y_1 \nrightarrow y_2$, a set $\begin{pmatrix} v_x & {}^{h}_{k} & v_y \end{pmatrix}$ of **squares**;
6. for all horizontal morphisms $h : x \nrightarrow y$, we have a **vertical identity** $\mathsf{id}^v_{\mathsf{sq}}(h) : \begin{pmatrix} \mathsf{id}_x & {}^{h}_{h} & \mathsf{id}_y \end{pmatrix}$;
7. for all squares $\tau_1 : \begin{pmatrix} v_1 & {}^{h}_{k} & w_1 \end{pmatrix}$ and $\tau_2 : \begin{pmatrix} v_2 & {}^{k}_{l} & w_2 \end{pmatrix}$, we have a **vertical composition**
$$\tau_1 \cdot_{\mathsf{sq}} \tau_2 : \begin{pmatrix} v_1 \cdot v_2 & {}^{h}_{l} & w_1 \cdot w_2 \end{pmatrix};$$
8. for all $v : x \longrightarrow y$, we have a **horizontal identity**
$$\mathsf{id}^h_{\mathsf{sq}}(v) : \begin{pmatrix} v & {}^{\mathsf{id}_x}_{\mathsf{id}_y} & v \end{pmatrix};$$
9. for all squares $\tau_1 : \begin{pmatrix} v_1 & {}^{h_1}_{k_1} & v_2 \end{pmatrix}$ and $\tau_2 : \begin{pmatrix} v_2 & {}^{h_2}_{k_2} & v_3 \end{pmatrix}$, we have a **horizontal composition**
$$\tau_1 \odot_{\mathsf{sq}} \tau_2 : \begin{pmatrix} v_1 & {}^{h_1 \odot h_2}_{k_1 \odot k_2} & v_3 \end{pmatrix};$$
10. for all $h : x \nrightarrow y$, we have a **left unitor**
$$\lambda_h : \begin{pmatrix} \mathsf{id}_x & {}^{\mathsf{id}_x \odot h}_{h} & \mathsf{id}_y \end{pmatrix};$$
11. for all $h : x \nrightarrow y$, we have a **right unitor**
$$\rho_h : \begin{pmatrix} \mathsf{id}_x & {}^{h \odot \mathsf{id}_y}_{h} & \mathsf{id}_y \end{pmatrix};$$
12. for all $h_1 : w \nrightarrow x$, $h_2 : x \nrightarrow y$, and $h_3 : y \nrightarrow z$, we have an **associator**
$$\alpha_{(h_1,h_2,h_3)} : \begin{pmatrix} \mathsf{id}_w & {}^{h_1 \odot (h_2 \odot h_3)}_{(h_1 \odot h_2) \odot h_3} & \mathsf{id}_z \end{pmatrix}.$$

This data is required to satisfy several laws, stating, in particular, that horizontal identities and horizontal composition are functorial, and that the left unitor, right unitor, and associator are natural transformations and invertible. In addition, we have the **triangle** and **pentagon** law. Their description can be found in Fig. 1 and Fig. 2.

To formulate the laws in Definition 2.3, one needs to use transports. The necessity of these transports come from the laws of the squares. For example, if we compose a square $\tau : \begin{pmatrix} v & \substack{h \\ k} & w \end{pmatrix}$ with the identity square, then we should get the original square τ back. However, the square $\tau \odot_{\mathsf{sq}} \mathsf{id}^h_{\mathsf{sq}}(w)$ has different sides than τ, because the the top and bottom sides of $\tau \odot_{\mathsf{sq}} \mathsf{id}^h_{\mathsf{sq}}(w)$ are composed with identities. As such, we need the laws for vertical composition in order to state the laws for composition of squares.

Remark 2.4. Strict double categories can also be defined in an unfolded style. One can do so by slightly modifying Definition 2.3: we add the requirement that the horizontal morphisms form a set and that the unitors and associators are identities. Such an approach is used in the Agda-categories library [15].

However, Definition 2.3 is still unsatisfactory for our purposes. Many notions from double category theory can be derived from the bicategory DoubleCat of double categories. For example, equivalences of double categories are the same as adjoint equivalences in DoubleCat [27], monoidal double categories are the same pseudomonoids in DoubleCat [4], and fibrations of double categories are the same as internal Street fibrations [10]. For this reason, DoubleCat plays a prominent role in double category theory.

Since we are working in univalent foundations, we would also like a notion of univalence for double categories and a univalence principle for them. This principle can be formulated by saying that DoubleCat is a univalent bicategory. All in all, our goals in this paper are

- to define the notion of univalent double category;
- to define the bicategory DoubleCat of double categories;
- to prove that DoubleCat is a univalent bicategory.

The unfolded definition from Definition 2.3 poses several complications for our purposes. More specifically, proving that DoubleCat is univalent would become unfeasible. This is because we are forced to consider the identity type of double categories, which is rather complicated. However, by using *displayed bicategories* [1], one can give a simpler proof that DoubleCat is univalent. Intuitively, the idea is to break up the definition into smaller layers. The identity type of each of these layers is simpler, and that simplifies the proof of univalence.

This is the basic philosophy behind the definition of double category that we describe in the remainder of this paper. More specifically, we take the following steps:

- We define the notion of 2-sided displayed categories in Section 3. With 2-sided displayed categories, we can describe categories with an additional class of morphisms and squares.
- In Section 5, we describe the bicategory of double categories. We start by defining the displayed bicategory of 2-sided displayed categories, and step-by-step we add data and properties to acquire double categories. For example, in Definition 5.5, we add horizontal identities to the structure, and in Definition 5.8, we add a horizontal composition operation. Simultaneously, we prove that the resulting bicategory is univalent.

Another advantage of our approach is that we can use it to construct adjoint equivalences and invertible 2-cells of double categories. We describe this process in Section 7.

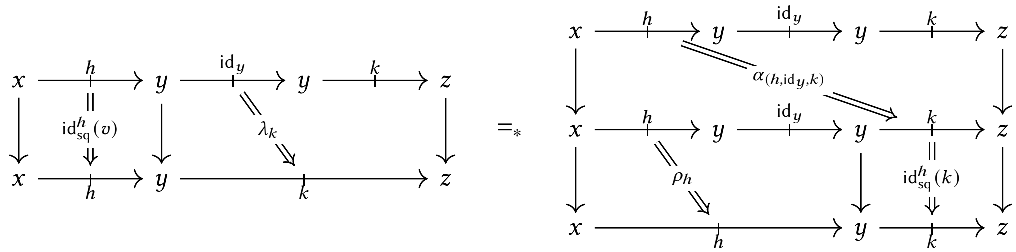
3 2-Sided Displayed Categories

The notion of displayed categories was developed by Ahrens and Lumsdaine [2]. Displayed categories are useful for various purposes, and among those are defining the notion of Grothendieck fibration and modularly defining univalent categories. Intuitively, a displayed category represents structure/property of objects and morphisms in some category C. Displayed categories consist of a type family of *displayed objects* parametrized by the objects of C, and a family of sets of *displayed morphisms* parametrized by the morphisms in C and displayed objects. For example, we have a displayed category of group structures over the category of sets. The displayed objects over a set X are group structures on X, and the set of displayed morphisms over $f : X \longrightarrow Y$ from a group structure G_X over X to a group structure G_Y over Y are proofs that f preserves the group operations.

In this section, we define 2-sided displayed categories — a variation of the notion of displayed categories. The difference between 2-sided displayed categories and displayed categories is that displayed categories depend on *one* category, whereas 2-sided displayed categories depend on *two* categories. Note that 2-sided displayed categories share many purposes with displayed categories: they can be used to define univalent categories in a modular way, and they can be used to define 2-sided fibrations [17, 28]. However, in this paper we view 2-sided displayed categories in another way, namely as an extra class of morphisms on a category.

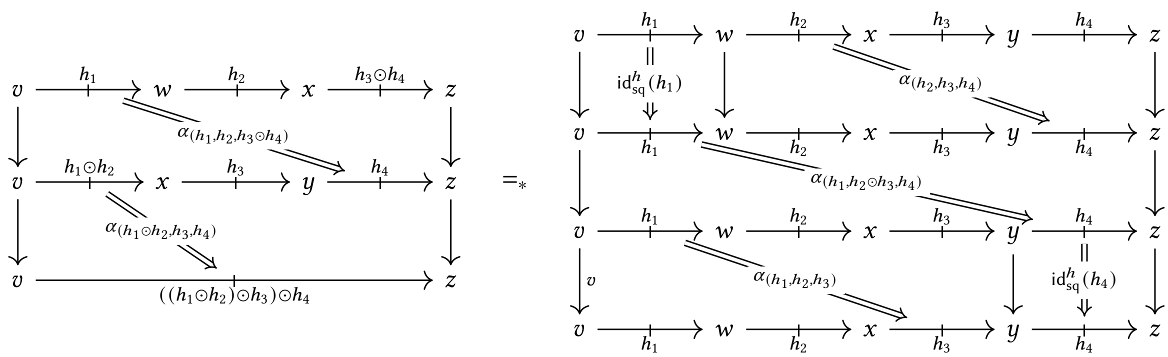
Definition 3.1 (twosided_disp_cat)**.** Let C_1 and C_2 be categories. A **2-sided displayed category** D over C_1 and C_2 consists of

1. for all objects $x_1 : C_1$ and $x_2 : C_2$ a type $D_{(x_1,x_2)}$ of **objects over** x_1 **and** x_2
2. for all objects $\overline{x} : D_{(x_1,x_2)}$ and $\overline{y} : D_{(y_1,y_2)}$ and morphisms $f_1 : x_1 \longrightarrow y_1$ in C_1 and $f_2 : x_2 \longrightarrow y_2$ in C_2, a set $\overline{x} \longrightarrow_{(f_1,f_2)} \overline{y}$ of **morphisms over** f_1 **and** f_2



We use $=_*$ to denote dependent equality, see Remark 3.2.

Figure 1. The triangle equation



We use $=_*$ to denote dependent equality, see Remark 3.2.

Figure 2. The pentagon equation

3. for every object $\overline{x} : D_{(x_1,x_2)}$ a morphism $\overline{\mathsf{id}}_{\overline{x}}$ over id_{x_1} and id_{x_2}
4. for all $\overline{f} : \overline{x} \longrightarrow_{(f_1,f_2)} \overline{y}$ and $\overline{g} : \overline{y} \longrightarrow_{(g_1,g_2)} \overline{z}$, a morphism $\overline{f} \cdot \overline{g} : \overline{x} \longrightarrow_{(f_1 \cdot g_1, f_2 \cdot g_2)} \overline{z}$

such that the following equations hold.

5. for all $\overline{f} : \overline{x} \longrightarrow_{(f_1,f_2)} \overline{y}$, we have $\overline{f} \cdot \overline{\mathsf{id}}_{\overline{y}} =_* \overline{f}$ and $\overline{\mathsf{id}}_{\overline{x}} \cdot \overline{f} =_* \overline{f}$;
6. for all $\overline{f} : \overline{w} \longrightarrow_{(f_1,f_2)} \overline{x}$, $\overline{g} : \overline{x} \longrightarrow_{(g_1,g_2)} \overline{y}$, and $\overline{h} : \overline{y} \longrightarrow_{(h_1,h_2)} \overline{z}$, we have $\overline{f} \cdot (\overline{g} \cdot \overline{h}) =_* (\overline{f} \cdot \overline{g}) \cdot \overline{h}$.

Remark 3.2. Here we use the notation $=_*$ to represent a *dependent* equality, i.e., a path between an element $y_1 : Y(x_1)$ and $y_2 : Y(x_2)$ such that $x_1 = x_2$.

Note that the laws in Items 5 and 6 in Definition 3.1 are actually dependent equalities. For examples, if $\overline{f} : \overline{x} \longrightarrow_{(f_1,f_2)} \overline{y}$, then the left-hand side of $\overline{f} \cdot \overline{\mathsf{id}}_{\overline{y}} =_* \overline{f}$ is a morphism that lives over $f_1 \cdot \mathsf{id}_{y_1}$ and $f_2 \cdot \mathsf{id}_{y_2}$, respectively. However, the right-hand side lives over f_1 and f_2, and thus their types are not equal. We can solve this by properly using a transport. Note that the type of 2-sided displayed categories over C_1 and C_2 is equivalent to the type of displayed categories over the product $\mathsf{C}_1 \times \mathsf{C}_2$. We use 2-sided displayed categories instead of displayed categories because the 2-sided variant is closer to Definition 2.3.

Every displayed category D over C gives rise to a total category $\int \mathsf{D}$ and a functor $\int \mathsf{D} \longrightarrow \mathsf{C}$. For 2-sided displayed categories, we can do the same.

Definition 3.3 (total_twosided_disp_category)**.** Let D be a 2-sided displayed category over C_1 and C_2. Then we define the **total category** $\int \mathsf{D}$ to be the category whose objects consists of triples $x_1 : \mathsf{C}_1$, $x_2 : \mathsf{C}_2$, and $\overline{x} : \mathsf{D}_{(x_1,x_2)}$. We also define the **projection functors** $\pi_1^{\mathsf{D}} : \int \mathsf{D} \longrightarrow \mathsf{C}_1$ and $\pi_2^{\mathsf{D}} : \int \mathsf{D} \longrightarrow \mathsf{C}_2$ to be the functors that take the first and second coordinate of a triple, respectively.

Note that every 2-sided displayed category D over C_1 and C_2 gives rise to a span $\mathsf{C}_1 \xleftarrow{\pi_1^{\mathsf{D}}} \int \mathsf{D} \xrightarrow{\pi_2^{\mathsf{D}}} \mathsf{C}_2$ of categories. Now let us consider some examples of 2-sided displayed categories.

Example 3.4 (arrow_twosided_disp_cat)**.** Let C be a category. We define the 2-sided displayed category $\mathsf{Arr}(\mathsf{C})$ over C and C as follows.

- The objects over x and y are morphisms $\varphi : x \longrightarrow y$.
- Suppose that we have morphisms $f : x_1 \longrightarrow x_2$, $g : y_1 \longrightarrow y_2$, $\varphi_1 : x_1 \longrightarrow y_1$, and $\varphi_2 : x_2 \longrightarrow y_2$, then

the set $\varphi_1 \longrightarrow_{(f,g)} \varphi_2$ is defined to be the collection of proofs that $f \cdot \varphi_2 = \varphi_1 \cdot g$.

The total category $\int \mathsf{Arr}(\mathsf{C})$ is equivalent to the arrow category of C.

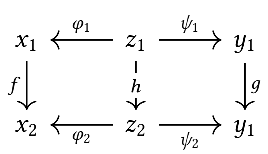
Example 3.5 (comma_twosided_disp_cat)**.** Given functors $F : \mathsf{C}_1 \longrightarrow \mathsf{C}_3$ and $G : \mathsf{C}_2 \longrightarrow \mathsf{C}_3$, we define the 2-sided displayed category $\mathsf{Comma}(F, G)$ over C_1 and C_2:

- The objects over $x : \mathsf{C}_1$ and $y : \mathsf{C}_2$ are morphisms $\varphi : F(x) \longrightarrow G(y)$.
- Given morphisms $f : x_1 \longrightarrow x_2$, $g : y_1 \longrightarrow y_2$, $\varphi_1 : F(x_1) \longrightarrow G(y_1)$, and $\varphi_2 : F(x_2) \longrightarrow G(y_2)$, the set $\varphi_1 \longrightarrow_{(f,g)} \varphi_2$ is defined to be the collection of proofs that $F(f) \cdot \varphi_2 = \varphi_1 \cdot G(g)$.

The category $\int \mathsf{Comma}(F, G)$ is equivalent to the comma category of F and G.

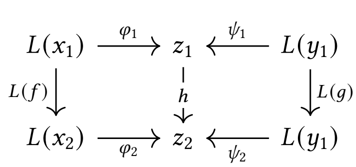
Example 3.6 (twosided_disp_cat_of_spans)**.** Let C be a category. We define the 2-sided displayed category $\mathsf{Span}(\mathsf{C})$ over C and C:

- The objects over x and y are **spans** from $x : \mathsf{C}$ to $y : \mathsf{C}$. More concretely, they consist of an object $z : \mathsf{C}$ and two morphisms $\varphi : z \longrightarrow x$ and $\psi : z \longrightarrow y$.
- Suppose that we have $f : x_1 \longrightarrow x_2$ and $g : y_1 \longrightarrow y_2$. A morphism from $x_1 \xleftarrow{\varphi_1} z_1 \xrightarrow{\psi_1} y_1$ to $x_2 \xleftarrow{\varphi_2} z_2 \xrightarrow{\psi_2} y_2$ over f and g consists of a morphism $h : z_1 \longrightarrow z_2$ such that the following diagrams commute.



Example 3.7 (twosided_disp_cat_of_struct_cospans)**.** Suppose that we have a functor $L : \mathsf{C}_1 \longrightarrow \mathsf{C}_2$. We define the 2-sided displayed category $\mathsf{StructCospan}(L)$ over C_1 and C_1:

- The objects over x and y are **structured cospans** from $x : \mathsf{C}_1$ to $y : \mathsf{C}_1$, that is to say, an object $z : \mathsf{C}_2$ together with morphisms $L(x) \xrightarrow{\varphi} z \xleftarrow{\psi} L(y)$.
- Given two structured cospans $L(x_1) \xrightarrow{\varphi_1} z_1 \xleftarrow{\psi_1} L(y_1)$ and $L(x_2) \xrightarrow{\varphi_2} z_2 \xleftarrow{\psi_2} L(y_2)$, and two morphisms $f : x_1 \longrightarrow x_2$ and $g : y_1 \longrightarrow y_2$, a displayed morphism consists of a morphism $h : z_1 \longrightarrow z_2$ such that the following diagram commutes



Example 3.8 (twosided_disp_cat_of_lenses)**.** Let C be a category with chosen binary products. A **lens** l from s to v consists of a **get**-morphism $\mathsf{get}_l : s \longrightarrow v$ and a **put**-morphism $\mathsf{put}_l : v \times s \longrightarrow s$ such that

- $\mathsf{put}_l \cdot \mathsf{get}_l = \pi_1$;
- $\mathsf{get}_l \times \mathsf{id}_s \cdot \mathsf{put}_l = \mathsf{id}_s$;
- $\mathsf{id}_v \times \mathsf{put}_l \cdot \mathsf{put}_l = \pi_1 \times (\pi_2 \cdot \pi_2) \cdot \mathsf{put}_l$.

Then we define a 2-sided displayed category $\mathsf{Lens}(C)$ over C and C as follows.

- The displayed objects over s and v are lenses from s to v.
- Given morphisms $f_1 : s_1 \longrightarrow s_2$ and $f_2 : v_1 \longrightarrow v_2$ and lenses l_1 from s_1 to v_1 and l_2 from s_2 to v_2, the displayed morphisms from l_1 to l_2 over f_1 and f_2 are proofs that $g_1 \cdot f_2 = f_1 \cdot g_2$ and $p_1 \cdot f_1 = f_2 \times f_1 \cdot p_2$.

Our next goal is to define *univalent* 2-sided displayed categories. To do so, we take the same approach as for categories and for displayed categories. We first define the notion of isomorphism, and we prove that the identity is an isomorphism. With that in place, we obtain a map that sends equalities of displayed objects to isomorphisms, and univalence is formulated by saying that this map is an equivalence of types.

Definition 3.9 (is_iso_twosided_disp)**.** Let D be a 2-sided displayed category over C_1 and C_2, and let $f_1 : x_1 \longrightarrow y_1$ and $f_2 : x_2 \longrightarrow y_2$ be isomorphisms in C_1 and C_2 respectively. In addition, suppose that we have objects $\overline{x} : \mathsf{D}_{(x_1,y_1)}$ and $\overline{y} : \mathsf{D}_{(x_2,y_2)}$. Then we say that $\overline{f} : \overline{x} \longrightarrow_{(f_1,f_2)} \overline{y}$ is an **isomorphism** if we have a morphism $\overline{f^{-1}} : \overline{y} \longrightarrow_{(f_1^{-1},f_2^{-1})} \overline{x}$ such that $\overline{f} \cdot \overline{f^{-1}} =_* \overline{\mathsf{id}}_{\overline{x}}$ and $\overline{f^{-1}} \cdot \overline{f} =_* \overline{\mathsf{id}}_{\overline{y}}$.

Proposition 3.10 (isaprop_is_iso_twosided_disp)**.** *For every morphism $\overline{f} : \overline{x} \longrightarrow_{(f_1,f_2)} \overline{y}$ over isomorphisms f_1 and f_2, the type that $\overline{f}$ is an isomorphism is a proposition.*

Proposition 3.11 (id_iso_twosided_disp)**.** *For all displayed objects $\overline{x} : \mathsf{D}_{(x_1,x_2)}$, the identity $\overline{\mathsf{id}}_{\overline{x}}$ is an isomorphism.*

Definition 3.12 (is_univalent_twosided_disp_cat)**.** Let D be a 2-sided displayed category over C_1 and C_2.

- For all objects $x_1 : \mathsf{C}_1$ and $x_2 : \mathsf{C}_2$ and displayed objects $\overline{x}, \overline{y} : \mathsf{D}_{(x_1,x_2)}$, we have a map that sends identities $p : \overline{x} = \overline{y}$ to isomorphisms $\mathsf{idtoiso}_{\overline{x},\overline{y}}(p) : \overline{x} \longrightarrow_{(\mathsf{id}_{x_1},\mathsf{id}_{x_2})} \overline{y}$.
- We say that D is **univalent** if for all $\overline{x}, \overline{y} : \mathsf{D}_{(x_1,x_2)}$, the map $\mathsf{idtoiso}_{\overline{x},\overline{y}}$ is an equivalence of types.

Note that in the formalization, the definition of univalence is equivalent, but formulated slightly differently. In Definition 3.12, we only look at paths $p : \overline{x} = \overline{y}$ between displayed objects lying over the same objects in the base, whereas in the formalization, we also take paths in the base into account. Each of the 2-sided displayed categories from Examples 3.4 to 3.8 is univalent.

Proposition 3.13 (is_univalent_total)**.** *If* D *is a univalent 2-sided displayed category over* C_1 *and* C_2*, and* C_1 *and* C_2 *are univalent, then* $\int$ D *is univalent as well.*

Recall that every 2-sided displayed category gives rise to a span of categories. Hence, by Proposition 3.13, every univalent 2-sided displayed category D over C_1 and C_2 gives rise to a span $C_1 \xleftarrow{\pi_1^D} \int D \xrightarrow{\pi_2^D} C_2$ of univalent categories. To end this section, we define the notions of *2-sided displayed functors* and *2-sided displayed natural transformations*. These play a prominent role when we define the bicategory of double categories in Section 5.

Definition 3.14 (twosided_disp_functor)**.** Suppose that we have 2-sided displayed categories D over C_1 and C_2, and D′ over C_3 and C_4. In addition, suppose that we have functors $F_1 : C_1 \longrightarrow C_3$ and $F_2 : C_2 \longrightarrow C_4$. A **2-sided displayed functor** $\overline{F}$ over F_1 and F_2 from D to D′ consists of

- a map that assigns to every object $\overline{x} : D_{(x_1,x_2)}$ an object $\overline{F}(\overline{x}) : D'_{(F_1(x_1),F_2(x_2))}$;
- a map that assigns to every morphism $\overline{f} : \overline{x} \longrightarrow_{(f_1,f_2)} \overline{y}$ a morphism $\overline{F}(\overline{f}) : \overline{F}(\overline{x}) \longrightarrow_{(F_1(f_1),F_2(f_2))} \overline{F}(\overline{y})$

such that $\overline{F}(\overline{\mathrm{id}}_{\overline{x}}) =_* \overline{\mathrm{id}}_{\overline{F}(\overline{x})}$ and $\overline{F}(\overline{f} \cdot \overline{g}) =_* \overline{F}(\overline{f}) \cdot \overline{F}(\overline{g})$.

Definition 3.15 (twosided_disp_nat_trans)**.** Suppose that we have 2-sided displayed categories D over C_1 and C_2, and D′ over C_3 and C_4. In addition, suppose that we have functors $F_1, G_1 : C_1 \longrightarrow C_3$ and $F_2, G_2 : C_2 \longrightarrow C_4$, 2-sided displayed functors $\overline{F}$ over F_1 and F_2 and $\overline{G}$ over G_1 and G_2, and natural transformations $\tau_1 : F_1 \Rightarrow G_1$ and $\tau_2 : F_2 \Rightarrow G_2$. A **2-sided displayed natural transformation** $\overline{\tau}$ over τ_1 and τ_2 from $\overline{F}$ to $\overline{G}$ consists of a map that assigns to every $\overline{x} : D_{(x_1,x_2)}$ a morphism $\overline{F}(\overline{x}) \longrightarrow_{(\tau_1(x_1),\tau_2(x_2))} \overline{G}(\overline{x})$ such that the usual naturality condition holds.

4 A Recap on (Displayed) Bicategories

Our next goal is to construct the bicategory of double categories. To do so, we recall in this section the definitions and propositions that we use in the remainder of this paper. These definitions were originally introduced in [1], and full definitions can be found there. Recall that a bicategory not only has objects and morphisms, but also 2-cells. The notion of *displayed bicategory* is similar to that of displayed category.

Definition 4.1 (disp_bicat)**.** Let B be a bicategory. A **displayed bicategory** D over B consists of

- for each object x : B, a type D_x of objects over x;
- for all 1-cells $f : x \longrightarrow y$ and displayed objects $\overline{x} : D_x$ and $\overline{y} : D_y$, a type $\overline{x} \longrightarrow_f \overline{y}$ of 1-cells over f;
- for all 2-cells $\tau : f \Rightarrow g$ and displayed 1-cells $\overline{f} : \overline{x} \longrightarrow_f \overline{y}$ and $\overline{g} : \overline{x} \longrightarrow_g \overline{y}$, a set $\overline{f} \Rightarrow_\tau \overline{g}$ of 2-cells over τ.

In addition, there should be suitable identities, composition, unitors, and associators, and the usual coherence laws should be satisfied.

There are numerous examples of displayed bicategories and they are discussed in [1], and we quickly recall the ones that we need in Section 5. If we have displayed bicategories D_1 and D_2 over B, then we have a displayed bicategory $D_1 \times D_2$ over D whose displayed objects, 1-cells, and 2-cells are pairs of displayed objects, 1-cells, and 2-cells of D_1 and D_2 respectively. The full subbicategory can also be defined using a displayed bicategory: if we have a predicate P on the objects of a bicategory B, then we define a displayed bicategory over B whose displayed objects over x are proofs of $P(x)$, and whose displayed 1-cells and 2-cells are inhabitants of the unit type.

Every displayed bicategory gives rise to a *total bicategory*.

Definition 4.2 (total_bicat)**.** Given a displayed bicategory D over B, we define its **total bicategory** as the bicategory whose objects are given by pairs of objects x : B and $\overline{x} : D_x$. The 1-cells and 2-cells are defined similarly.

Univalent bicategories are defined in a similar way as univalent categories, but there is a slight difference. For categories, univalence is expressed by saying that identity of objects is equivalent to isomorphisms between objects. For bicategories on the other hand, we formulate univalence in two steps. First of all, we say that identity of 1-cells is equivalent to invertible 2-cells between them. This is called *local univalence* in [1]. Secondly, we say that identity of objects is equivalent to adjoint equivalences between them. In [1], this is called *global univalence*. Then a *univalent bicategory* is a bicategory that is both locally and globally univalent. Similarly, we define *univalent displayed bicategories*. The key theorem for univalent displayed bicategories is the following.

Proposition 4.3 (total_is_univalent_2)**.** *Let* D *be a univalent displayed bicategory over a univalent bicategory* B*. Then* $\int$ D *is univalent.*

One key application of univalence for bicategories is *equivalence induction*. More specifically, to prove some property for every invertible 2-cell, it suffices to only consider identity 2-cells. Similarly, to prove some property for every adjoint equivalence, one only has to show it for identity equivalences. This is similar to path induction in homotopy type theory [25, 31].

Proposition 4.4 (J_2_0)**.** *Let* B *be a univalent bicategory, and suppose that for all objects* x, y : B*, we have a predicate* P *on adjoint equivalences* $x \simeq y$*. Then* P *holds for every adjoint equivalence if* P *holds for* $\mathrm{id}_x : x \simeq x$ *for every* x : B.

Proposition 4.5 (J_2_1). *Let* B *be a univalent bicategory, and suppose that for all objects x, y :* B *and 1-cells $f, g : x \longrightarrow y$, we have a predicate P on invertible 2-cells $f \cong g$. Then P holds for every invertible 2-cell if P holds for* $\mathsf{id}_f : f \cong f$ *for every $f : x \longrightarrow y$.*

5 The Bicategory of Double Categories

In this section, we define the bicategory of univalent double categories, and we prove that this bicategory is univalent. The notion of *displayed bicategory* plays a key role in this construction [1].

The construction proceeds in several steps. We start in Definition 5.1 by defining a displayed bicategory $\mathsf{TwoSidedDisp_d}$ over the bicategory UnivCat of univalent categories, and the objects over C are 2-sided displayed categories D over C and C. If we look at the total bicategory TwoSidedDisp of this displayed bicategory, then the objects consists of a category C and a 2-sided displayed D over C and C. This means that we have a category with an extra class of morphisms and a class of squares.

To obtain a the bicategory of double categories, we need to add more structure. We define two displayed bicategories $\mathsf{HorId_d}$ and $\mathsf{HorComp_d}$ over TwoSidedDisp in Definition 5.5 and Definition 5.8. The displayed bicategory $\mathsf{HorId_d}$ adds horizontal identities to the structure, and $\mathsf{HorComp_d}$ adds horizontal compositions. By taking their product and the total bicategory, we obtain the bicategory HorIdComp, of which the objects consists of a category, horizontal morphisms, squares, horizontal identities, and compositions.

Next we define displayed bicategories $\mathsf{Lun_d}$, $\mathsf{Run_d}$, and $\mathsf{Assoc_d}$ over HorIdComp. These add the left unitor, the right unitor, and the associator to the structure. Again we take their product and the total bicategory to obtain the bicategory UnAssoc. Finally, we define DoubleCat as a full subbicategory of UnAssoc: the predicate we use, expresses the triangle and pentagon coherence.

At each step, we prove that the relevant displayed bicategories are univalent. The machinery of displayed bicategories allows us to combine all of this to conclude that DoubleCat is univalent. The advantage of using displayed bicategories over a direct approach is that the proof of univalence becomes simpler and more modular. This is because the displayed approach allows us to consider the identity of each part individually, and we are able to reuse results (e.g., the bicategory of univalent categories is univalent).

The main idea behind this construction is that we can split up the definition of a double category into several layers. Instead of looking at the whole, we look at these layers separately, and that allows for reusability and modularity. This is also why the notion of 2-sided displayed category plays an important role in this construction: it is one of the layers to define double categories.

Definition 5.1 (disp_bicat_twosided_disp_cat). The displayed bicategory $\mathsf{TwoSidedDisp_d}$ over UnivCat is defined as follows:

- The displayed objects over C are univalent 2-sided displayed categories D over C and C.
- The displayed morphisms from D_1 to D_2 over $F : \mathsf{C}_1 \longrightarrow \mathsf{C}_2$ are 2-sided displayed functors $\overline{F}$ over F and F from D_1 to D_2.
- The displayed 2-cells from $\overline{F}$ to $\overline{G}$ over $\tau : F \Rightarrow G$ are 2-sided displayed natural transformations over τ and τ from $\overline{F}$ to $\overline{G}$.

We define TwoSidedDisp to be $\int \mathsf{TwoSidedDisp_d}$.

An object of TwoSidedDisp consists of a univalent category C and a univalent 2-sided displayed category D over C and C. If we compare this to Definition 2.3, then we already got the data from Items 1, 2 and 5 to 7. The vertical category is given by C, the horizontal morphisms from x to y are given by the displayed objects $\mathsf{D}_{(x,y)}$, and the squares $\left(v_1 \ {}^{h}_{k} \ v_2\right)$ are given by displayed morphisms $h \longrightarrow_{(v_1,v_2)} k$. The vertical identity and composition for squares is given by the identity and composition in D, respectively, and similarly for the laws involving vertical composition of squares.

Proposition 5.2 (univalent_2_twosided_disp_cat). *The displayed bicategory* $\mathsf{TwoSidedDisp_d}$ *is univalent.*

5.1 Identities and Composition

Next we add horizontal identities (Items 3 and 8 in Definition 2.3 and composition (Items 4 and 9 in Definition 2.3), in the form of two displayed bicategories over TwoSidedDisp. To define the first one, we define when a 2-sided displayed category supports *horizontal identities.*

Definition 5.3 (hor_id). Let C be a category and let D be a 2-sided displayed category over C and C. Then we say that D has **horizontal identities** if

1. for all x : C, we have a displayed object $\mathsf{id}^{\mathsf{D}}_{\mathsf{h}}(x) : \mathsf{D}_{(x,x)}$;
2. for all morphisms $v : x \longrightarrow y$, we have a displayed morphism $\mathsf{id}^{\mathsf{D}}_{\mathsf{sq}}(v) : \mathsf{id}^{\mathsf{D}}_{\mathsf{h}}(x) \longrightarrow_{(v,v)} \mathsf{id}^{\mathsf{D}}_{\mathsf{h}}(y)$;

such that $\mathsf{id}^{\mathsf{D}}_{\mathsf{sq}}(\mathsf{id}_x) = \mathsf{id}_{\mathsf{id}^{\mathsf{D}}_{\mathsf{h}}(x)}$ and $\mathsf{id}^{\mathsf{D}}_{\mathsf{sq}}(v_1 \cdot v_2) = \mathsf{id}^{\mathsf{D}}_{\mathsf{sq}}(v_1) \cdot \mathsf{id}^{\mathsf{D}}_{\mathsf{sq}}(v_2)$.

We also define when a 2-sided displayed functor *preserves horizontal identities.*

Definition 5.4 (double_functor_hor_id). Let D be a 2-sided displayed category over C and C and let D′ be a 2-sided displayed category over C′ and C′. Suppose that we have a functor $F : \mathsf{C} \longrightarrow \mathsf{C}'$ and a 2-sided displayed functor $\overline{F}$ from D to D′ over F and F, and that D and D′ have horizontal identities. Then we say that $\overline{F}$ **preserves horizontal identities** if for all $x : \mathsf{C}_1$ we have a natural square $F_{\mathsf{id}}(x) : \mathsf{id}^{\mathsf{D}'}_{\mathsf{h}}(F(x)) \longrightarrow_{(\mathsf{id}_{F(x)},\mathsf{id}_{F(x)})} F(\mathsf{id}^{\mathsf{D}}_{\mathsf{h}}(x))$.

The precise naturality condition for the square can be found in the formalization. In addition, note that we consider *lax* double functors: we do not require $F_{\mathsf{id}}(x)$ to be invertible.

Definition 5.5 (disp_bicat_twosided_disp_cat_hor_id)**.** We define the displayed bicategory $\mathsf{HorId_d}$ over $\mathsf{TwoSidedDisp}$ as follows:

- the displayed objects over a pair of a univalent category C and a univalent 2-sided displayed category D are horizontal identities for D (Definition 5.3);
- the displayed 1-cells over a functor $F : \mathsf{C}_1 \longrightarrow \mathsf{C}_2$ and 2-sided displayed functor $\overline{F}$ from D_1 to D_2 that preserve horizontal identities (Definition 5.4);
- the displayed 2-cells over a natural transformations $\tau : F \Rightarrow G$ and a 2-sided displayed natural transformation $\overline{\tau}$ are proofs that $\overline{\tau}$ preserves horizontal identities. The precise formulation can be found in the formalization.

Next we look at horizontal compositions.

Definition 5.6 (hor_comp)**.** Let C be a category and let D be a 2-sided displayed category over C and C. Then we say that D has **horizontal composition** if

- for all $h : \mathsf{D}_{(x,y)}$ and $k : \mathsf{D}_{(y,z)}$, we have a displayed object $h \odot k : \mathsf{D}_{(x,z)}$;
- for all displayed morphisms $s_1 : h_1 \longrightarrow_{(v_1,v_2)} h_2$ and $s_2 : k_1 \longrightarrow_{(v_2,v_3)} k_2$, we have a displayed morphism $s_1 \odot_{\mathsf{sq}} s_2 : h_1 \odot k_1 \longrightarrow_{(v_1,v_3)} h_2 \odot k_2$;

such that

- $\mathsf{id}_h \odot_{\mathsf{sq}} \mathsf{id}_k = \mathsf{id}_{h \odot k}$.
- $(s_1 \cdot t_1) \odot_{\mathsf{sq}} (s_2 \cdot t_2) = (s_1 \odot_{\mathsf{sq}} s_2) \cdot (t_1 \odot_{\mathsf{sq}} t_2)$.

Definition 5.7 (double_functor_hor_comp)**.** Let D be a 2-sided displayed categories over C and C and let D' be 2-sided displayed categories over C' and C'. Suppose that we have a functor $F : \mathsf{C} \longrightarrow \mathsf{C}'$ and a 2-sided displayed functor $\overline{F}$ from D to D' over F and F, and that D_1 and D_2 have horizontal identities. Then we say that $\overline{F}$ **preserves horizontal compositions** if for all $h : \mathsf{D}_{(x,y)}$ and $k : \mathsf{D}_{(y,k)}$ we have a natural square $F_{\mathsf{comp}}(h,k) : \left(F(h) \odot F(k) \;{}^{\mathsf{id}_{F(x)}}_{\mathsf{id}_{F(z)}}\; F(h \odot k)\right)$.

Definition 5.8 (disp_bicat_twosided_disp_cat_hor_comp)**.** The displayed bicategory $\mathsf{HorComp_d}$ over $\mathsf{TwoSidedDisp}$ is defined as follows:

- the displayed objects over a pair of a univalent category C and a univalent 2-sided displayed category D are horizontal composition for D (Definition 5.6);
- the displayed 1-cells over a functor $F : \mathsf{C}_1 \longrightarrow \mathsf{C}_2$ and 2-sided displayed functor $\overline{F}$ from D_1 to D_2 that preserve horizontal composition (Definition 5.7);
- the displayed 2-cells over a natural transformations $\tau : F \Rightarrow G$ and a 2-sided displayed natural transformation $\overline{\tau}$ are proofs that $\overline{\tau}$ preserves horizontal composition. The precise formulation can be found in the formalization.

We define $\mathsf{HorIdComp_d}$ to be $\mathsf{HorId_d} \times \mathsf{HorComp_d}$, and we define $\mathsf{HorIdComp}$ to be the total bicategory of $\mathsf{HorIdComp_d}$.

Proposition 5.9 (disp_univalent_2_disp_bicat_twosided_disp_cat_id_hor_comp)**.** *The displayed bicategory* $\mathsf{HorIdComp_d}$ *is univalent.*

5.2 Unitors and Associators

At this point, we obtained the bicategory $\mathsf{HorIdComp}$, and the objects of that bicategory consists of a univalent category C, a univalent displayed D over C and C, together with horizontal identities (Definition 5.3) and horizontal compositions (Definition 5.6). This corresponds to Items 1 to 9 in Definition 2.3 and now we look at Items 10 to 12 from Definition 2.3. For each of these, we define a displayed bicategory over $\mathsf{HorIdComp}$. Due to space constraints, we only say how the displayed objects of those displayed bicategories are defined.

Definition 5.10 (disp_bicat_lunitor)**.** We define a displayed bicategory $\mathsf{Lun_d}$ over $\mathsf{HorIdComp}$ whose displayed objects over a univalent category C and a univalent 2-sided displayed category D with horizontal identities and compositions consists of a natural isomorphism $\mathsf{id}^{\mathsf{D}}_{\mathsf{h}}(x) \odot h \longrightarrow_{(\mathsf{id}_x, \mathsf{id}_y)} h$ for each $h : \mathsf{D}_{(x,y)}$.

Definition 5.11 (disp_bicat_runitor)**.** We define a displayed bicategory $\mathsf{Run_d}$ over $\mathsf{HorIdComp}$ whose displayed objects over a univalent category C and a univalent 2-sided displayed category D with horizontal identities and compositions consists of a natural isomorphism $h \odot \mathsf{id}^{\mathsf{D}}_{\mathsf{h}}(y) \longrightarrow_{(\mathsf{id}_x, \mathsf{id}_y)} h$ for each $h : \mathsf{D}_{(x,y)}$.

Definition 5.12 (disp_bicat_lassociator)**.** We define a displayed bicategory $\mathsf{Assoc_d}$ over $\mathsf{HorIdComp}$ whose displayed objects over a univalent category C and a univalent 2-sided displayed category D with horizontal identities and compositions consists of a natural isomorphism

$$h_1 \odot (h_2 \odot h_3) \longrightarrow_{(\mathsf{id}_w, \mathsf{id}_z)} (h_1 \odot h_2) \odot h_3$$

for all $h_1 : \mathsf{D}_{(w,x)}$, $h_2 : \mathsf{D}_{(x,y)}$, and $h_3 : \mathsf{D}_{(y,z)}$.

We define $\mathsf{UnAssoc_d}$ to be $\mathsf{Lun_d} \times \mathsf{Run_d} \times \mathsf{Assoc_d}$, and we define $\mathsf{UnAssoc}$ to be the total bicategory of $\mathsf{UnAssoc_d}$.

Proposition 5.13 (is_univalent_2_bicat_unitors_and_associator)**.** *The displayed bicategory* $\mathsf{UnAssoc_d}$ *is univalent.*

Definition 5.14 (bicat_of_double_cats)**.** We define the bicategory $\mathsf{DoubleCat}$ of double categories as the full subbicategory where the predicate expresses that the triangle and pentagon laws are satisfied.

Theorem 5.15 (is_univalent_2_bicat_of_double_cats)**.** *The bicategory* DoubleCat *is univalent.*

The proof of Theorem 5.15 uses Voevodsky's univalence axiom.

The objects of DoubleCat collect all the data and properties mentioned in this section. Each of these data and properties correspond to some part of Definition 2.3. However, the only thing missing in Definition 2.3 is a univalence condition. To define this notion, we first note that every double category as defined in Definition 2.3 gives rise to a category of objects and vertical morphisms, and to a 2-sided displayed category of horizontal morphisms and squares.

Definition 5.16 (is_double_univalent)**.** A double category as defined in Definition 2.3 is called **univalent** if

- its underlying category of objects and vertical morphisms is univalent; and
- its corresponding 2-sided displayed category of horizontal morphisms and squares is univalent.

Theorem 5.17 (double_cat_weq_univalent_doublecategory)**.** *The type of objects of* DoubleCat *is equivalent to the type of univalent double categories as defined in Definitions 2.3 and 5.16.*

Proof. This theorem is proved by inspecting what objects of DoubleCat consist of, and comparing it to Definition 2.3. Such an object consists of

- a category C (Item 1 in Definition 2.3);
- a 2-sided displayed category over C and C (Items 2 and 5 to 7 in Definition 2.3);
- horizontal identities (Items 3 and 8 in Definition 2.3);
- a horizontal composition operator (Items 4 and 9 in Definition 2.3);
- left unitors (Item 10 in Definition 2.3);
- right unitors (Item 11 in Definition 2.3);
- associators (Item 12 in Definition 2.3).

In addition, the laws and the univalence that are satisfied by objects of DoubleCat correspond to those in Definitions 2.3 and 5.16. □

The 1-cells in DoubleCat are *lax double functors*. They consist of an underlying functor and 2-sided displayed functor that preserve horizontal identities and compositions as described in Definitions 5.4 and 5.7. Finally, every 2-cell in DoubleCat has an underlying natural transformation and 2-sided displayed natural transformation.

Note that there are several notions of morphisms between double categories, namely lax, oplax, and pseudo double functors. To obtain a univalent bicategory, the choice does not matter: one could define variants of DoubleCat where oplax or pseudo double functors are used. Each variant leads to a univalent bicategory, since in all cases the adjoint equivalences are the same.

Remark 5.18. Note that the double categories in DoubleCat are univalent, and this univalence condition means that the underlying category and 2-sided displayed category are univalent. From this, we see that objects in DoubleCat are the same as pseudocategories internal to the bicategory of univalent categories.

In [3, Example 9.3], a notion of univalent double bicategory is defined such that identities correspond to *gregarious equivalences* of double bicategories. In the particular case where the underlying bicategory given by objects, vertical morphisms and squares is a category (meaning the 1-morphisms form a set and the assignment from identities of 1-morphisms to 2-morphisms is an equivalence), our notion of univalence coincides with the notion introduced in [3].

6 Examples of Double Categories

Now we construct several examples of double categories using Definition 5.14. All of the double categories considered here are univalent.

Example 6.1 (square_double_cat)**.** Let C be a univalent category. In Example 3.4, we defined the 2-sided displayed category Arr(C) over C and C. This gives rise to a double category as follows.

- the horizontal identities are given by the identity morphism;
- horizontal composition is given by the composition of morphisms.

The unitality and associativity of horizontal composition reduce to the ordinary laws of composition for morphisms. All laws involving squares hold because the type of morphisms in a category is a set.

Example 6.2 (kleisli_double_cat)**.** Let T be a monad on a univalent category C. In Example 3.5, we defined the 2-sided displayed category Comma(F, G) for arbitrary functors $F : \mathsf{C}_1 \longrightarrow \mathsf{C}_3$ and $G : \mathsf{C}_2 \longrightarrow \mathsf{C}_3$. We take F to be the identity on C and G to be the endofunctor underlying T. Concretely, we look at Comma$(\mathrm{id}_\mathsf{C}, T)$, meaning that the horizontal morphisms are morphisms in the Kleisli category of T. We obtain the following double category.

- the horizontal identities are given by the unit of T;
- given morphisms $h : x \longrightarrow T(y)$ and $k : y \longrightarrow T(z)$, their horizontal composition is defined as the following composition

$$x \xrightarrow{h} T(y) \xrightarrow{Tk} T(T(z)) \xrightarrow{\mu_z} T(z)$$

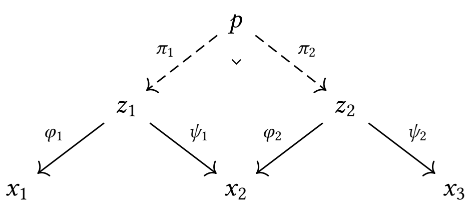
The construction of the unitors and associators for this double category reduces to proving unitality and associativity of composition in the Kleisli category.

One way to instantiate Example 6.2 is by taking C to be Set and T to be the power set monad. Note that morphisms from X to Y in the Kleisli category of the power set monad are the same as relations between X and Y. Hence, the resulting double category has functions as vertical morphisms, and relations as horizontal morphisms.

Note that both Examples 6.1 and 6.2 are strict double categories. In both cases, the type of horizontal morphisms is a set and unitality and associativity for horizontal composition holds up to equality.

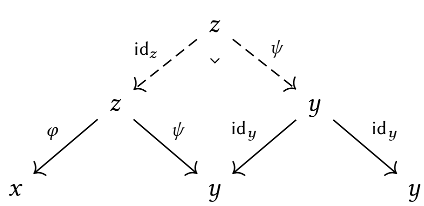
Example 6.3 (spans_double_cat)**.** Let C be a univalent category with pullbacks. We defined the 2-sided displayed category Span(C) in Example 3.6. This gives rise to a double category.

- The horizontal identity on an object x : C is given by the span $x \xleftarrow{\mathrm{id}_x} x \xrightarrow{\mathrm{id}_x} x$.
- Suppose that we have spans $x_1 \xleftarrow{\varphi_1} z_1 \xrightarrow{\psi_1} y_1$ and $x_2 \xleftarrow{\varphi_2} z_2 \xrightarrow{\psi_2} y_2$. Their composition is given by the following span



Here p is the pullback of ψ_1 and φ_2.

To construct the right unitor of this double category, we consider the following diagram



The square in this diagram is a pullback, and from this, we get the desired isomorphism for the right unitor. Similarly, we can define the left unitor and the associator. The proofs of the triangle and pentagon laws follow by diagram chasing, and details can be found in the formalization.

Example 6.4. Using the examples we already introduced in this section, we define a double category of sets and relations valued in propositions and a double category of sets and relations valued in sets. To define the first, we instantiate Example 6.2 using the power set monad. In the resulting double category, horizontal morphisms from X to Y are given by morphisms $X \longrightarrow \mathcal{P}(Y)$, which are the same as relations on X and Y. Furthermore, since the category of sets has pullbacks, we can instantiate Example 6.3 to sets as well. The horizontal morphisms in the resulting double category are given by spans $X \xleftarrow{f} R \xrightarrow{g} Y$, which are the same as set-valued relations on X and Y.

Example 6.5 (structured_cospans_double_cat)**.** Suppose that we have a functor $L : \mathsf{C}_1 \longrightarrow \mathsf{C}_2$ between univalent categories and suppose that C_2 has pushouts. In Example 3.7 we defined the 2-sided displayed category StructCospan(L) over C_1 and C_1 of structured cospans. This gives rise to a double category.

- The horizontal identity on an object $x : \mathsf{C}_1$ is given by the cospan
$$L(x) \xrightarrow{\mathrm{id}_{L(x)}} L(x) \xleftarrow{\mathrm{id}_{L(x)}} L(x).$$
- The construction of the horizontal composition is dual to how horizontal composition is defined in Example 6.3.

Example 6.6 ([26], lenses_double_cat)**.** Let C be a univalent category with chosen binary products. Then we define the double category of lenses of C as follows. In Example 3.8, we defined the 2-sided displayed category Lens(C). We construct a double category from it as follows.

- The identity lens from x : C to x : C is given by $\mathrm{id}_x : x \longrightarrow x$ and $\pi_1 : x \times x \longrightarrow x$.
- Suppose we have lenses l_1 from x to y and l_2 from y to z. Then we have a lens $l_1 \odot l_2$ from x to z such that $\mathrm{get}_{l_1 \odot l_2} = \mathrm{get}_{l_1} \cdot \mathrm{get}_{l_2}$, and such that $\mathrm{put}_{l_1 \odot l_2}$ is the following composition.

$$z \times x \xrightarrow{\langle \mathrm{id}, \pi_2 \rangle} (z \times x) \times x \xrightarrow{((\mathrm{id} \times \mathrm{get}_{l_1}) \cdot \mathrm{put}_{l_2}) \times \mathrm{id}} y \times x \xrightarrow{\mathrm{put}_{l_1}} x$$

Note that there are different gadgets called "lenses" in the literature. The lenses by Clarke [8, Def. 3.20] are, more specifically, "delta lenses". The double category of delta lenses has, as objects, (small) categories, as horizontal morphisms functors between categories, and vertical morphisms delta-lenses, that is, functors equipped with an extra "lifting operation" — see [8, Def. 2.1] for details. Squares are suitable commutative squares of functors.

7 Equivalences of Double Categories

In this section, we give sufficient conditions to show that a 1-cell in DoubleCat is an adjoint equivalence (Theorem 7.3), and that a 2-cell in DoubleCat is invertible (Theorem 7.1). Since these proofs are similar, we only discuss how Theorem 7.3 is proven. Let us first give conditions for when a 2-cell in DoubleCat is invertible.

Theorem 7.1 (invertible_double_nat_trans_weq)**.** *Let τ be a 2-cell in* DoubleCat. *Then τ is an invertible 2-cell if and only if its underlying natural transformation and 2-sided displayed natural transformation are pointwise isomorphisms.*

To characterize adjoint equivalences, we need the notion of a *strong double functor*.

Definition 7.2 (is_strong_double_functor)**.** Let F be a lax double functor. We say that F is a **strong double functor**

if $F_{\mathsf{id}}(x)$ and $F_{\mathsf{comp}}(h,k)$ are isomorphisms for all suitably typed x, h, and k.

Theorem 7.3 (adjoint_equivalence_double_functor_weq)**.** *Let* $L : \mathsf{C}_1 \longrightarrow \mathsf{C}_2$ *be a 1-cell in* DoubleCat*. Then* L *is an adjoint equivalence if and only if* L *is a strong double functor and* L *is an adjoint equivalence in* TwoSidedDisp.

We give a sketch of our proof of Theorem 7.3; it follows the construction of TwoSidedDisp in Section 5. As such, we first show that L lifts to an adjoint equivalence in HorIdComp, then we show that L lifts to an adjoint equivalence in UnAssoc, and finally, we conclude that L gives rise to an adjoint equivalence in DoubleCat.

Next we show that L lifts to an adjoint equivalence in HorIdComp, and to do so, we construct a displayed adjoint equivalence over L in both $\mathsf{HorId}_{\mathsf{d}}$ and $\mathsf{HorComp}_{\mathsf{d}}$. We simplify this construction by using induction over adjoint equivalences (Proposition 4.4) for which we use that the bicategory $\mathsf{HorComp}_{\mathsf{d}}$ is univalent. Intuitively, this allows us to assume that L is the identity equivalence. More concretely, we show the following.

Lemma 7.4. *Let* D : TwoSidedDisp*. Suppose that* I_1 *and* I_2 *are objects over* D *in* $\mathsf{HorId}_{\mathsf{d}}$*, and that* $f : I_1 \longrightarrow_{\mathsf{id}_{\mathsf{D}}} I_2$*. Note that* f *consists of a natural square* $\tau : \left(\mathsf{id}_x \; {}^{\mathsf{id}_{\mathsf{h}}^{\mathsf{D}'}(x)}_{\mathsf{id}_{\mathsf{h}}^{\mathsf{D}}(x)} \; \mathsf{id}_x\right)$ *for each* x*. Then* f *is a displayed adjoint equivalence if* $\tau(x)$ *is an isomorphism for every* x*.*

In our situation, the assumption in Lemma 7.4 follows from the fact that L preserves the identity up to isomorphism. Similarly, we can construct a displayed adjoint equivalence over L in $\mathsf{HorComp}_{\mathsf{d}}$, and this gives us the adjoint equivalence in HorIdComp.

To lift L to an adjoint equivalence in UnAssoc, we need to construct displayed adjoint equivalences over L in $\mathsf{Lun}_{\mathsf{d}}$, $\mathsf{Run}_{\mathsf{d}}$, and $\mathsf{Assoc}_{\mathsf{d}}$. Note that each of these displayed bicategories live over HorIdComp. Again we use Proposition 4.4, so we assume that L is the identity. Constructing the displayed adjoint equivalences then follows from diagram chasing, and the precise proof can be found in the formalization.

To conclude Theorem 7.3, we note that DoubleCat is defined as a full subbicategory of UnAssoc. Since adjoint equivalences in full subbicategories of some bicategory B are the same as adjoint equivalences in B, we get the desired adjoint equivalence in DoubleCat.

For the converse, we first note that whenever L is an adjoint equivalence DoubleCat, then L is an adjoint equivalence in TwoSidedDisp. This is because pseudofunctors preserve adjoint equivalence. To show that L is a strong double functor, we use Proposition 4.4, so it suffices to show that the identity is a strong double functor. This follows from the fact that the identity is an isomorphism.

Note that Shulman proves Theorem 7.3 for framed bicategories in a different way [27, Corollary 7.9]. Whereas our proof follows the construction of DoubleCat via displayed bicategories and makes use of induction over adjoint equivalences, Shulman's proof makes use of fully faithful and essentially surjective strong double functors.

8 Related Work

Variants of double categories have been formalized in several computer proof assistants.

Murray, Pronk, and Szyld [20] worked towards defining double categories in the Lean proof assistant.[1] The chosen approach is to define double categories as category objects in the category of categories (see also Remark 2.2). To this end, the authors start from the notion of "quiver internal to a category", and add a composition operation via suitable pullbacks, as discussed in Section 2. This approach is orthogonal to ours, as it allows one to consider not just a version of double categories, but also category objects in other categories. Note that in Lean, due to the assumption of uniqueness of identity proofs, all categories are "strict" in the sense that their objects (and morphisms) form a homotopy set. The categories internal to the category of categories in Lean, correspond most closely to what we call "strict double categories" in Section 2 — see Remark 2.1; in particular, the associativity and unitality laws for both horizontal and vertical morphisms hold up to equality.

In 1lab [30], double categories are also defined as category objects in a category of categories. There, pullbacks are avoided by looking at the internal language of a presheaf category.

Hu and Carette [15] started a library of category theory in Agda. At the time of writing that article, "[...] double categories [...] are still awaiting" formalization. In the meantime, the definition of *strict* (see Remark 2.1) double categories, as well as the construction of the dual of a double category (swapping horizontal and vertical morphisms), have been implemented.[2] In particular, Hu and Carette's double categories are symmetric, that is, horizontal and vertical morphisms play the same role. The library is based on E-category theory [23]. Accordingly, the type family of squares in a double category is indexed by setoids; to avoid "transport modulo setoid equality" in the statement of composition laws for squares, a custom equality for squares, called `FrameEquality`, is introduced. We instead do state composition laws for squares modulo transport (along identities), see the discussion in Section 2.

Displayed (bi)categories [1, 2] play a prominent role in our work. Firstly, we use 2-sided displayed categories to represent an extra class of morphisms on a category. Note that 2-sided displayed categories are the same as displayed categories over a cartesian product of categories. Secondly,

[1] https://github.com/leanprover-community/mathlib/pull/18204
[2] https://github.com/agda/agda-categories/blob/36abe6bff98be027bd4fcc3306d6dac8b2140079/src/Categories/Double/Core.agda

the bicategory of double categories is constructed in layers using the technique of displayed bicategories.

Displayed categories are the same as type refinement systems as studied by Melliès and Zeilberger [19]. Melliès and Zeilberger [19, Section 6.4] also consider a 2-sided displayed category, in the form of a displayed category over the category **Dom** × **Dom**, to formalize a logical relations theorem by Reynolds.

9 Conclusion

In this paper, we constructed the univalent bicategory of univalent double categories. The main tool in the construction is the notion of 2-sided displayed categories, which represent categories with an extra class of morphisms and squares. We also characterized the adjoint equivalences and invertible 2-cells in the bicategory of univalent double categories, and in that characterization, we made use of univalence at several points. Finally, we gave numerous examples of univalent double categories. Among our examples are the double categories of lenses and of structured cospans.

There are numerous ways to extend this work. An interesting special case of double categories is given by *framed bicategories* [27]. We can obtain a univalent bicategory of univalent framed bicategories by extending the work in Section 5: we take a full subbicategory of DoubleCat that expresses that the the double category is framed (i.e., some functor is a fibration). However, currently framed bicategories are not considered in our formalization. Furthermore, in many applications, one would like to have more structure on a double category, such as a (symmetric) monoidal structure. Such structures can conveniently be defined by looking at pseudomonoids in DoubleCat. To construct a univalent bicategory of (symmetric) monoidal double categories, one would need to combine ideas from [1, 32] and [35]. The methods in this paper are also applicable to define virtual double categories [11]. To do so, one would need to modify the notion of a 2-sided displayed category so that the source of a square is not given by a morphism, but by a sequence of composable morphisms.

In addition, our notion of univalent double category is unable to capture univalent categories with profunctors. This is because we do not have a category of univalent categories, but only a bicategory. This is another situation where the right solution is to pursue a formalization of double bicategories [33] and its suitable notion of univalence [3, Example 9.3].

Acknowledgments

We gratefully acknowledge the work by the Coq development team in providing the Coq proof assistant and surrounding infrastructure, as well as their support in keeping UniMath compatible with Coq. We are very grateful to Mike Shulman for answering our questions about profunctors, and to Noam Zeilberger for pointers on type refinement systems and their links to (2-sided) displayed categories. We also thank Jurriaan Rot and Dario Stein for pointers to the literature on applications of double categories to computer science. We finally thank the anonymous referees of CPP 2024 for their helpful comments.

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-21-1-0334. This research was supported by the NWO project "The Power of Equality" OCENW.M20.380, which is financed by the Dutch Research Council (NWO).

References

[1] Benedikt Ahrens, Dan Frumin, Marco Maggesi, Niccolò Veltri, and Niels van der Weide. 2021. Bicategories in univalent foundations. *Math. Struct. Comput. Sci.* 31, 10 (2021), 1232–1269. https://doi.org/10.1017/S0960129522000032

[2] Benedikt Ahrens and Peter LeFanu Lumsdaine. 2019. Displayed Categories. *Log. Methods Comput. Sci.* 15, 1 (2019). https://doi.org/10.23638/LMCS-15(1:20)2019

[3] Benedikt Ahrens, Paige Randall North, Michael Shulman, and Dimitris Tsementzis. 2022. The Univalence Principle. arXiv:2102.06275v3 [math.CT] To be published in Mem. AMS.

[4] John C. Baez and Kenny Courser. 2020. Structured cospans. *Theory Appl. Categ.* 35 (2020), Paper No. 48, 1771–1822.

[5] John C. Baez, Kenny Courser, and Christina Vasilakopoulou. 2022. Structured versus Decorated Cospans. *Compositionality* 4 (Sept. 2022). Issue 3. https://doi.org/10.32408/compositionality-4-3

[6] John C. Baez and Jade Master. 2020. Open Petri nets. *Math. Struct. Comput. Sci.* 30, 3 (2020), 314–341. https://doi.org/10.1017/S0960129520000043

[7] Aaron Bohannon, Benjamin C. Pierce, and Jeffrey A. Vaughan. 2006. Relational lenses: a language for updatable views. In *Proceedings of the Twenty-Fifth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 26-28, 2006, Chicago, Illinois, USA*, Stijn Vansummeren (Ed.). ACM, 338–347. https://doi.org/10.1145/1142351.1142399

[8] Bryce Clarke. 2023. *The double category of lenses.* Ph. D. Dissertation. Macquarie University. https://doi.org/10.25949/22045073.v1

[9] Kenny Courser. 2020. *Open Systems: A Double Categorical Perspective.* Ph. D. Dissertation. University of California, Riverside. https://doi.org/10.48550/arXiv.2008.02394 arXiv:2008.02394 [math.CT]

[10] G. S. H. Cruttwell, M. J. Lambert, D. A. Pronk, and M. Szyld. 2022. Double fibrations. *Theory Appl. Categ.* 38 (2022), Paper No. 35, 1326–1394.

[11] G. S. H. Cruttwell and Michael A. Shulman. 2010. A unified framework for generalized multicategories. *Theory Appl. Categ.* 24 (2010), No. 21, 580–655.

[12] Pierre-Évariste Dagand and Conor McBride. 2013. A Categorical Treatment of Ornaments. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013.* IEEE Computer Society, 530–539. https://doi.org/10.1109/LICS.2013.60

[13] Charles Ehresmann. 1963. Catégories structurées. In *Annales scientifiques de l'École Normale Supérieure*, Vol. 80. 349–426.

[14] Brendan Fong. 2015. Decorated cospans. *Theory Appl. Categ.* 30 (2015), Paper No. 33, 1096–1120.

[15] Jason Z. S. Hu and Jacques Carette. 2021. Formalizing category theory in Agda. In *CPP '21: 10th ACM SIGPLAN International Conference on Certified Programs and Proofs, Virtual Event, Denmark, January 17-19, 2021*, Catalin Hritcu and Andrei Popescu (Eds.). ACM, 327–342. https://doi.org/10.1145/3437992.3439922

[16] Vadim E. Kotov. 1978. An Algebra for Parallelism Based on Petri Nets. In *Mathematical Foundations of Computer Science 1978, Proceedings, 7th Symposium, Zakopane, Poland, September 4-8, 1978 (Lecture Notes in Computer Science, Vol. 64)*, Józef Winkowski (Ed.). Springer, 39–55. https://doi.org/10.1007/3-540-08921-7_55

[17] Fosco Loregian and Emily Riehl. 2020. Categorical notions of fibration. *Expo. Math.* 38, 4 (2020), 496–514. https://doi.org/10.1016/j.exmath.2019.02.004

[18] The mathlib Community. 2020. The Lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2020, New Orleans, LA, USA, January 20-21, 2020*, Jasmin Blanchette and Catalin Hritcu (Eds.). ACM, 367–381. https://doi.org/10.1145/3372885.3373824

[19] Paul-André Melliès and Noam Zeilberger. 2015. Functors are Type Refinement Systems. In *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2015, Mumbai, India, January 15-17, 2015*, Sriram K. Rajamani and David Walker (Eds.). ACM, 3–16. https://doi.org/10.1145/2676726.2676970

[20] Zach Murray, Dorette Pronk, and Martin Szyld. 2022. Implementing Double Categories in the Lean Proof Assistant. https://www.mathstat.dal.ca/~mszyld/Zach_slides.pdf Talk at Science Atlantic Mathematics, Statistics, and Computer Science Conference, October 15, 2022.

[21] David Jaz Myers. 2021. Double Categories of Open Dynamical Systems (Extended Abstract). *Electronic Proceedings in Theoretical Computer Science* 333 (feb 2021), 154–167. https://doi.org/10.4204/eptcs.333.11

[22] Max S. New and Daniel R. Licata. 2023. A Formal Logic for Formal Category Theory. In *Foundations of Software Science and Computation Structures - 26th International Conference, FoSSaCS 2023, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2023, Paris, France, April 22-27, 2023, Proceedings (Lecture Notes in Computer Science, Vol. 13992)*, Orna Kupferman and Pawel Sobocinski (Eds.). Springer, 113–134. https://doi.org/10.1007/978-3-031-30829-1_6

[23] Erik Palmgren. 2017. On Equality of Objects in Categories in Constructive Type Theory. In *23rd International Conference on Types for Proofs and Programs, TYPES 2017, May 29-June 1, 2017, Budapest, Hungary (LIPIcs, Vol. 104)*, Andreas Abel, Fredrik Nordvall Forsberg, and Ambrus Kaposi (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 7:1–7:7. https://doi.org/10.4230/LIPIcs.TYPES.2017.7

[24] James L. Peterson. 1977. Petri Nets. *ACM Comput. Surv.* 9, 3 (1977), 223–252. https://doi.org/10.1145/356698.356702

[25] Egbert Rijke. 2022. Introduction to homotopy type theory. *arXiv preprint arXiv:2212.11082* (2022).

[26] Mitchell Riley. 2018. Categories of Optics. arXiv:1809.00738 [math.CT]

[27] Michael Shulman. 2008. Framed bicategories and monoidal fibrations. *Theory Appl. Categ.* 20 (2008), No. 18, 650–738.

[28] Ross Street. 1980. Fibrations in bicategories. *Cahiers Topologie Géom. Différentielle* 21, 2 (1980), 111–160.

[29] The Coq Development Team. 2023. *The Coq Proof Assistant.* https://doi.org/10.5281/zenodo.8161141

[30] The 1Lab Development Team. [n. d.]. The 1Lab. https://1lab.dev

[31] The Univalent Foundations Program. 2013. *Homotopy Type Theory: Univalent Foundations of Mathematics.* https://homotopytypetheory.org/book, Institute for Advanced Study.

[32] Niccolò Veltri and Niels van der Weide. 2021. Constructing Higher Inductive Types as Groupoid Quotients. *Log. Methods Comput. Sci.* 17, 2 (2021). https://lmcs.episciences.org/7391

[33] Dominic Verity. 2011. Enriched categories, internal categories and change of base. *Repr. Theory Appl. Categ.* 20 (2011), 1–266.

[34] Vladimir Voevodsky, Benedikt Ahrens, Daniel Grayson, et al. [n. d.]. UniMath — a computer-checked library of univalent mathematics. available at http://unimath.org. https://doi.org/10.5281/zenodo.8427604

[35] Kobe Wullaert, Ralph Matthes, and Benedikt Ahrens. 2022. Univalent Monoidal Categories. In *28th International Conference on Types for Proofs and Programs, TYPES 2022, June 20-25, 2022, LS2N, University of Nantes, France (LIPIcs, Vol. 269)*, Delia Kesner and Pierre-Marie Pédrot (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 15:1–15:21. https://doi.org/10.4230/LIPIcs.TYPES.2022.15

Received 2023-09-19; accepted 2023-11-25