TU Delft

Delft University of Technology

Bicategorical type theory
Semantics and syntax

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

DOI
10.1017/S0960129523000312

Publication date
2023

Document Version
Final published version

Published in
Mathematical Structures in Computer Science

Citation (APA)
Ahrens, B., North, P. R., & Van Der Weide, N. (2023). Bicategorical type theory: Semantics and syntax. *Mathematical Structures in Computer Science*. https://doi.org/10.1017/S0960129523000312

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Mathematical Structures in Computer Science (2023), 1–45
doi:10.1017/S0960129523000312

CAMBRIDGE UNIVERSITY PRESS

PAPER

Bicategorical type theory: semantics and syntax

Benedikt Ahrens[1,2], Paige Randall North[3] and Niels van der Weide[4]

[1]Delft University of Technology, Delft, Netherlands, [2]University of Birmingham, Birmingham, UK, [3]Utrecht University, Utrecht, Netherlands and [4]Radboud University, Nijmegen, Netherlands

Corresponding author: Niels van der Weide; Email: nweide@cs.ru.nl

(Received 28 September 2022; revised 2 September 2023; accepted 4 September 2023)

Abstract

We develop semantics and syntax for bicategorical type theory. Bicategorical type theory features contexts, types, terms, and directed reductions between terms. This type theory is naturally interpreted in a class of structured bicategories. We start by developing the semantics, in the form of *comprehension bicategories*. Examples of comprehension bicategories are plentiful; we study both specific examples as well as classes of examples constructed from other data. From the notion of comprehension bicategory, we extract the syntax of bicategorical type theory, that is, judgment forms and structural inference rules. We prove soundness of the rules by giving an interpretation in any comprehension bicategory. The semantic aspects of our work are fully checked in the Coq proof assistant, based on the UniMath library.

Keywords: Directed type theory; dependent types; comprehension bicategory; computer-checked proof

1. Introduction

In recent years, efforts have been made to develop *directed* type theory. Roughly, directed type theory should correspond to Martin-Löf type theory (MLTT) as ∞-categories correspond to ∞-groupoids. Besides theoretical interest in directed type theory, it is hoped that such a type theory can serve as a framework for synthetic directed homotopy theory and synthetic ∞-category theory. Applications of those, in turn, include reasoning about concurrent processes (Fajstrup et al. 2016).

Several proposals for *syntax* for directed type theory have been given (reviewed in Section 2.3.2), but are ad hoc and are not always semantically justified. The *semantic* aspects of directed type theory are particularly underdeveloped; a general notion of model of a directed type theory is still lacking. Indeed these proposals often provide an interpretation of their syntax in categories. They either employ preexisting 1-categorical semantical tools, thus forcing them to interpret their syntax into a 1-category of categories, as in the work by North (2019), or, in the absence of 2-categorical tools, give an ad hoc interpretation of their syntax into the 2-category of categories, as done by Licata and Harper (2011). We rectify this situation by providing a 2-categorical semantical structure.

Specifically, in this work, we introduce *comprehension bicategories* as a suitable mathematical structure for higher-dimensional (directed) type theory. Approaching the development of directed type theory from the semantic side, we extract from this the core syntax – judgment forms and structural inference rules – of a two-dimensional dependent type theory that can accommodate directed type theory. We also give a soundness proof of our structural rules. In future work, we will equip our syntax and semantics with variances and type and term formers for directed type theory.

© The Author(s), 2023. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

To motivate our approach, we analyze in Section 1.1 how higher-groupoidal structure arises in MLTT through an interplay of judgmental equality and typal identity. Our analysis thus leads to the desiderata listed in Section 1.2. In Section 1.3, we discuss the foundations we work in, and aspects of the computer formalization of some of our results.

1.1 Judgmental and typal higher dimensions

The judgment forms of traditional MLTT specify types, contexts, terms, and judgmental equality (conversion) between types and terms. There is, prima facie, nothing higher-dimensional about these judgments, and an interpretation of types as sets and terms as elements of sets seems perfectly adequate. In this sense, Martin-Löf type theory is 1-dimensional. However, MLTT is often said to be ∞-dimensional. The higher dimensions are generated by the identity type, which internalizes the judgmental equality; specifically, the well-known reflexivity rule generates a typal identity from a judgmental equality. Since the identity type can be iterated, judgmental equality then also becomes available for terms of the identity type itself. This mutual interaction between judgmental equality and typal identity provides the infrastructure to "lift" judgmental equality to higher dimensions without extending the judgmental structure of MLTT. The tower of types $(A, \mathsf{Id}_A, \mathsf{Id}_{\mathsf{Id}_A}, \dots)$ then can be given the structure of an ∞-groupoid, as shown by Van den Berg and Garner (2011) and Lumsdaine (2010).

When developing a *directed* type theory, with models in ∞-*categories*, the analogous ingredients are the following:

I1: A *judgment* of *directed reductions* between types and terms, analogous to judgmental equality;
I2: A *type former* for *homomorphisms* between terms, analogous to identity types;
I3: A notion of *model* in which to interpret the judgments and type formers.

In the present work, we propose a judgmental framework (I1), and a suitable general notion of semantics (I3), for higher-dimensional and directed type theory. In a separate work, we will expand this core by a system of *variances* suitable for accommodating a type former akin to North's hom-types (I2), to build a fully functional higher-dimensional type theory.

1.2 Deriving syntax from semantics

Previous work on higher-dimensional and directed type theory – reviewed in detail in Section 2 – has focused on syntax (I1/I2). Licata and Harper (2011, 2012) and Nuyts (2015) devise judgmental structure for higher-dimensional and directed type theory. North (2019) devises a type former for directed homomorphisms between terms, on top of the judgmental structure of MLTT. None of these works propose a general definition of *model* of directed type theory. Garner (2009) defines a notion of higher-dimensional model, but considers only *un*directed type theory.

Our approach is different from that of previous work on directed and higher-dimensional type theory. Specifically, we choose to approach the challenge from the other direction: we start by devising a suitable categorical structure for directed type theory and extract a syntax from it.

We refine the ingredients above to the following list of desiderata for our work:

D1: A system of inference rules for dependent types with directed reductions;
D2: A definition of *mathematical structures* suitable for the mathematical modelling of the syntactic rules;
D3: An *interpretation* of the inference rules in such a mathematical structure;
D4: A syntax for type and term formers on top of D1;
D5: A semantic structure for the interpretation of type and term formers.

In the present work, we achieve desiderata D1, D2, and D3. The study of variances and type and term constructors will be reported on elsewhere. In Section 2, we refer back to these desiderata for describing related work.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

The semantics we propose are described in Section 5, and the extracted syntax is described in Section 8. Both our syntax and semantics are quite general; for instance, our reductions are *proof-relevant* – like those considered by Licata and Harper (2011, 2012), and unlike judgmental equality in MLTT, which is proof-irrelevant. Syntax and semantics could reasonably be simplified or specialized. Crucially, our work provides a framework to modify syntax and semantics *in lock-step*, with a clear mechanism to analyze changes to the syntax on the semantic side and vice versa. We suggest some possible variants in Section 10.

1.3 Foundations and formalization in UniMath

The main results presented here are agnostic to foundations: they can be formalized in both set theory and type theory.

However, some of the notions we employ can economically be formulated using dependent types. In particular, we work with cloven (Grothendieck) fibrations of (bi)categories, whose formulation in set theory usually relies on postulating equality of objects. Using dependent types, a formulation of such concepts can be given that avoids any reasoning about equality of objects; instead, these concepts are formulated in terms of fibers. For this reason, we use type-theoretic language throughout the paper; see also, e.g., Remark 25.

We carefully distinguish data and property; in particular, we postulate entities in categories (e.g., limits) to be explicitly given as data rather than to merely exist. We do not rely on any choice axioms or on excluded middle.

The results of Sections 3–7 of this work are checked in `Coq` (Coq Development Team 2022), based on the `UniMath` library (Voevodsky et al. 2022) of univalent mathematics. In univalent mathematics, we distinguish *strict* (bi)categories and *univalent* (bi)categories (Ahrens et al. 2021, Section 4). Our definitions are agnostic to this difference; hence, our definition of bicategory (Definition 1) does not make a commitment to either strictness or univalence. Specific examples can then be strict (such as Example 11) or univalent (such as Example 49); see also Remark 2.

Our code has been integrated into `UniMath` in commit 840ac16. The computer-checked definitions and results are accompanied by a link (e.g., `bicat`) to the corresponding definition in an HTML version of that commit. The code written specifically for this work comprises approximately 33,600 lines of code; specifically, the `coqwc` tool counts as follows:

```
 spec    proof comments
13143    20553      449 total
```

We build upon an existing library of (bi)category theory by Ahrens et al. (2015, 2021) and use heavily the *displayed* machinery, developed for 1-categories by Ahrens and Lumsdaine (2019) and extended to bicategories by Ahrens et al. (2021). In particular, the formalized notions of cloven Grothendieck fibration we are using (in the 1-categorical case) and developing (in the bicategorical case) are based on displayed (bi)categories; we can thus discuss these notions without postulating equality of objects.

The syntax presented in Section 8 and its interpretation given in Section 9 are not computer-checked; we therefore give these constructions in more detail in the paper.

1.4 Synopsis

In Section 2, we review related work. In Section 3, we review (displayed) bicategories and functors. In Section 4, we define cloven global and local (op/iso)fibrations of bicategories, and we use these

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

notions to define comprehension bicategories. In Section 5, we discuss some examples of comprehension bicategories. In Section 6, we discuss Street (op)fibrations internal to bicategories, which form our main examples of comprehension bicategories. In Section 7, we define display map bicategories, and we show how any display map bicategory gives rise to a comprehension bicategory. In Section 8, we present structural type-theoretic rules for the syntax of a two-dimensional type theory, which we call BTT. In Section 9, we give an interpretation of BTT in any comprehension bicategory. In Section 10, we discuss variations of BTT and the semantic structures these variations correspond to. In Section 11, we explain the difference between terms in BTT and terms in other approaches to directed type theory.

1.5 Version history

A shorter version of this paper was published in the proceedings of Logic in Computer Science under a different name (Ahrens et al. 2022). Compared to that shorter version, the following material has been added in the present version:

- We provide more instantiations of Example 57. In particular, we show that several bicategories of structured categories have pullbacks (Example 19).
- We give more detail on the specific comprehension bicategory given by functors into the 1-category of strict categories, in Example 44. We also present a formalization of that example.
- We introduce display map bicategories and show that any display map bicategory gives rise to a comprehension bicategory.
- We present the type theory BTT in more detail and specify more of its type-theoretic rules.
- We give more detail in the interpretation of the rules of BTT in any comprehension bicategory.

Furthermore, compared to the short version, we have added the definition of *weak* comprehension bicategory (see Definition 38 and Remark 40), where the comprehension χ is not required to preserve (op)cartesian cells. Our interpretation of BTT works in any such weak comprehension bicategory.

2. Related Work

In this section, we review work with a similar goal to ours, as well as work we rely on. We pay particular attention to the desiderata outlined in Section 1.2 and to the difference between judgmental and typal dimensions.

2.1 Nondependent type theories

The following works satisfy a nondependent variant of D1, together with suitable adaptations of D2 and D3. However, due to the absence of type dependency, they do not immediately compare to our work.

Seely (1987) presents a syntax for a two-dimensional simply-typed lambda calculus, consisting of types, terms, and reductions between terms. Seely then constructs a 2-category out of that syntax. Tabareau (2011) frames aspect-oriented programming in a 2-categorical way, developing a lambda calculus that provides an internal language for 2-categories. Hirschowitz (2013) constructs a 2-adjunction between 2-signatures for lambda calculi (where such signatures specify types, terms, and reductions) and the category of Cartesian closed 2-categories. Fiore and Saville (2019) construct an internal language for cartesian closed bicategories; the result is a class (parametrized by a notion of signature for constants) of *simple* 2-dimensional type theories or lambda calculi.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

This last work shares one aspect with ours that the others do not: it uses (weak) *bi*categorical structure, rather than (strict) *2*-categorical structure.

2.2 Theories for higher categories

There is a body of work on designing type theories for ω-groupoids and ω-categories. In these type theories, one works, semantically speaking, *within one fixed* ∞-groupoid (or ω-category). Compare this to, e.g., Martin-Löf type theory, where one manipulates ∞-groupoids (types and identity types) and ∞-functors (functions) between them. Analogously, in our type theory, each type can be thought of as a category. Despite these different goals, we mention some of the work in this area.

Brunerie (2016) constructs a type theory whose models are weak ∞-groupoids. Benjamin et al. (2021) (see also the work by Finster and Mimram 2017) design a type theory whose models are precisely ω-categories à la Grothendieck–Maltsiniotis. Finster et al. (2022) study meta-theoretic properties of a language for strictly unital ∞-categories. There are also computer tools implementing such type theories, see, e.g., the work by Bar et al. (2018) and Reutter and Vicary (2019).

2.3 Theories with dependent types

In this section, we review work on higher-dimensional and directed type theory with dependent types. We start with a review of work on *un*directed type theory.

2.3.1 Undirected type theory

The idea of considering higher-dimensional interpretations of type theory stems from Hofmann and Streicher's groupoid interpretation of Martin-Löf type theory by Hofmann and Streicher (1994). This interpretation is generalized to stacks (poset-indexed groupoids satisfying a sheaf condition) in order to prove the independence of several logical principles by Coquand et al. (2017). It is furthermore generalized, from different angles, to higher dimensions, see, e.g., work by Van den Berg and Garner (2011), Lumsdaine (2010), and Kapulkin and Lumsdaine (2021). All of this work considers the Martin-Löf identity type, which is undirected.

Licata and Harper (2012) develop a two-dimensional dependent type theory with a judgment for *equivalences* $\Gamma \vdash \alpha : M \simeq_A N$ between terms $M, N : A$. These equivalences are postulated to have (strict) inverses. The authors give an interpretation of types as groupoids: terms are (interpreted as) objects in the interpreting groupoid, and equivalences are morphisms, necessarily invertible. No general notion of semantic structure is discussed; this work hence satisfies an *un*directed version of D1, but not D2.

Coraglia and Di Liberti (2021) introduce judgmental theories and calculi for them as a general framework to present and study deductive systems. They instantiate their framework to obtain a type theory that describes an internal language of 2-toposes (Coraglia and Di Liberti 2021, Section 5.3); however, they only consider a one-dimensional type theory à la Martin-Löf, and thus their work does not satisfy D1.

Garner (2009) studies a typal two-dimensional type theory à la Martin-Löf: the forms of judgment are the same as in Martin-Löf type theory. Garner calls a type X "discrete" if it satisfies identity reflection (that is, if any identity $p : x = y$ between elements $x, y : X$ induces a judgmental equality $x \equiv y$. Rules are then added that turn any identity type into a discrete type, effectively making every type into a 1-type, in the sense of the h-levels of homotopy type theory. Garner defines a notion of two-dimensional model based on (strict) *comprehension 2-categories*. Exploiting the restriction to 1-truncated types, a sound and complete interpretation of that two-dimensional type theory in any model is then given. The Martin-Löf identity type is undirected; correspondingly, Garner defines their comprehension 2-categories to consist of *locally groupoidal*

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

2-categories. Thus, Garner's work satisfies D1 for *undirected* reductions, using the identity type for this purpose. Garner also considers type constructors such as dependent pair types and dependent product types, thus satisfying D4 and D5 in this case.[1]

2.3.2 Directed type theory

Licata and Harper (2011) (see also Licata 2011, Chapter 7) also design a *directed* two-dimensional type theory and give an interpretation for it in the strict 2-category of categories. Their syntax has a judgment for *substitutions* between contexts, written $\Gamma \vdash \theta : \Delta$, and *transformations* between parallel substitutions. An important aspect of their work is *variance* of contexts/types, built into the judgments. The type formers there have a certain variance – *co*variance or *contra*variance – in each of the arguments. They do not define a general notion of model for their theory; this work hence satisfies D1, but not D2.

Nuyts (Nuyts 2015, Section 1.3.1) observes that the type theory developed by Licata and Harper (2011) does not allow for a nontrivial Martin-Löf identity type – any such type would coincide with the directed transformations. Nuyts thus attempts to generalize the treatment of variance by Licata and Harper, and designs a directed type theory with additional variances, such as isovariance and invariance. Nuyts does not provide any interpretation of their syntax, and thus no proof of (relative) consistency; the work hence does not satisfy D2.

North (2019) develops a type former for *directed* types of morphisms, resulting in a typal higher-dimensional directed type theory based on the judgments of MLTT. North's work thus does not satisfy D1. The model given by North is in the 2-category of categories, similar to the model by Licata and Harper (2011).

Shulman (2012), in unfinished work, aims to develop 2-categorical logic, including a two-dimensional notion of topos and a suitable internal language for such toposes. Specifically, Shulman sketches two internal languages for 2-toposes. The first language (Shulman 2011) is undirected, consisting only of types and terms. The second language (Shulman 2010) is only described in a short sketch; it is a kind of directed type theory featuring, in particular, variances. Our work is similar to Shulman's in the sense that both start from a (bi)categorical notion and extract a language from it, with the goal of developing a precise correspondence between extensions of the syntax and additional structure on the semantics. Unfortunately, Shulman's work is unfinished, which makes a more complete evaluation difficult. However, it contains several ideas that have influenced the present work. For instance, Shulman (2019) emphasizes the usefulness of restricting to (op)fibrations instead of considering all 1-cells when constructing bicategories of arrows – we do this in our main examples of comprehension bicategory, Examples 45 and 57.

Riehl and Shulman (2017) design a *simplicial* type theory (STT) featuring a directed interval type, as a synthetic theory of $(\infty, 1)$-categories. As a notion of model, they introduce "comprehension categories with shapes" (Riehl and Shulman 2017, Def. A.5). These are (1-categorical) towers of fibrations accounting for several layers of contexts. Further work on STT was done, among others, by Weaver and Licata (2020) and Buchholtz and Weinberger (2021). STT is not higher dimensional in the sense of Licata and Harper (2011) or the present work; in particular, reductions, both in the tope layer and in the type layer, are undirected. This work thus does not satisfy D1.

Summary In the present work, we define bicategorical semantics for the interpretation of types, terms, and reductions and derive from it a system of inference rules; our work thus satisfies D1, D2, and D3. We do not handle D4 and D5 in this work.

Among the described related work, our work is closest to work by Licata and Harper (2011) and Garner (2009). Compared to Licata and Harper (2011), we add a general definition of "model" of a *directed* two-dimensional type theory and provide many examples of models. Compared to Garner (2009), we cover *directed* reductions and provide many instances of our general definition of model. Compared to both works, we do *not* handle type and term formers.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

3. Preliminaries

Here, we sketch some definitions used later on. Many would be very long if given in full; instead, we try to convey some intuition while pointing to the formalized definitions. As a reference for bicategory theory, see Bénabou (1967). We use here the vocabulary and notation introduced by Ahrens et al. (2021).

Definition 1 (`bicat`). *A bicategory consists of a type* B_0 *of 0-cells (or objects), a type* $a \to b$ *of* 1-cells from a to b *for every* $a, b : \mathsf{B}_0$*, and a* set $f \Rightarrow g$ *of* 2-cells from f to g *for every* $a, b : \mathsf{B}_0$ *and* $f, g : a \to b$*. We have identity* $\mathrm{id}_1(a) : a \to a$ *and composition of 1-cells* $f \cdot g : a \to c$ *(also written* fg*), which we write in diagrammatical order. These operations do* not *satisfy the axioms for a 1-category. Instead, we have, for instance, the* left unitors*, that is, invertible 2-cells* $\ell_f : \mathrm{id}_1(a) \cdot f \Rightarrow f$ *for any 1-cell* f*, and similarly right unitors* $r_f : f \cdot \mathrm{id}_1(b) \Rightarrow f$*. Analogously, we have the* associators*, a family of invertible 2-cells* $\alpha(f, g, h) : f \cdot (g \cdot h) \to (f \cdot g) \cdot h$*. For 2-cells* $\theta : f \Rightarrow g$ *and* $\tau : g \Rightarrow h$ *(where* $f, g, h : a \to b$ *for some* $a, b : \mathsf{B}_0$*), we have a* vertical composition $\theta \bullet \tau : f \Rightarrow h$*. For any 1-cell* $f : a \to b$*, we have an* identity 2-cell $\mathrm{id}_2(f) : f \Rightarrow f$*, which is neutral with respect to vertical composition:* $\mathrm{id}_2(f) \bullet \theta = \theta$*. For any two objects* a *and* b*, the 1-cells from* a *to* b*, and 2-cells between them, form the objects and morphisms of the hom-category* $\underline{\mathsf{B}(a, b)}$*, with composition given by vertical composition of* B*. We also have* left *and* right whiskering*; given a 2-cell* $\theta : f \Rightarrow g : b \to c$ *and a 1-cell* $e : a \to b$*, we have the* left whiskering $e \triangleleft \theta : e \cdot f \Rightarrow e \cdot g$*, and, similarly, the* right whiskering $\theta \triangleright h : f \cdot h \Rightarrow g \cdot h$ *for* $h : c \to d$*. We do not list the axioms that these operations satisfy; the interested reader can consult, e.g., Def. 2.1 of Ahrens et al. (2021).*

We occasionally write 1_a *for* $\mathrm{id}_1(a)$ *and* 1_f *for* $\mathrm{id}_2(f)$*.*

Remark 2. We do *not* generally require that our bicategories (and displayed bicategories, see Definition 4) are univalent in the sense of Ahrens et al. (2021).

Sometimes it is still interesting to assume that a given (displayed) bicategory is univalent; in such cases, fibrations are better behaved, since lifts can be shown to be actually unique rather than just essentially unique (see, for instance, Proposition 26). Whenever we state such a result, the assumption on the (displayed) bicategory to be univalent is stated explicitly.

We denote by Cat the bicategory of categories, and by Grpd the bicategory of groupoids. The bicategory B^{op} has the same objects as B, but 1-cells from x to y in B^{op} are 1-cells $f : y \to x$ in B. The 2-cells in B^{op} are 2-cells in B. The bicategory B^{co} has the same objects and 1-cells as B, but 2-cells from f to g in B^{co} are the same as 2-cells from g to f in B.

Definition 3 (`psfunctor`). *Given two bicategories* B *and* B'*, a pseudofunctor* $F : \mathsf{B} \to \mathsf{B}'$ *is given by maps* $F_0 : \mathsf{B}_0 \to \mathsf{B}'_0$*,* $F_1 : (a \to b) \to (F_0 a \to F_0 b)$*,*[2] *and* $p_2 : (f \Rightarrow g) \to (F_1 f \Rightarrow F_1 g)$*, preserving structure on 1-cells up to invertible 2-cells in* B' *(specified as part of the functor F) and preserving structure on 2-cells up to equality.*

We build complicated bicategories from simpler ones by adding structure at all dimensions. The additional structure should come with its own composition and identity, which should lie suitably over composition and identity of the original bicategory. This idea is formalized in the notion of *displayed bicategory* – a layer of data over a base bicategory – and the resulting *total bicategory* – the bicategory of pairs $(b, \overline{b})$ of a cell b in the base and a cell $\overline{b}$ "over" b. We also obtain a pseudofunctor from the total bicategory into the base, given at all dimensions by the first projection.

Definition 4 (Ahrens et al. 2021, Def. 6.1, `disp_bicat`). *Let* B *be a bicategory. A displayed bicategory* D *over* B *consists of*

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

(1) for any $b : \mathsf{B}_0$*, a type* D_b *of objects over b;*

(2) for any $f : a \to b$ *and* $x : \mathsf{D}_a$ *and* $y : \mathsf{D}_b$*, a type* $x \xrightarrow{f} y$ *of 1-cells over f;*

(3) for any $\theta : f \Rightarrow g$ *and* $\overline{f} : x \xrightarrow{f} y$ *and* $\overline{g} : x \xrightarrow{g} y$*, a set* $\overline{f} \overset{\theta}{\Rightarrow} \overline{g}$ *of 2-cells over* θ*;*

together with suitably typed composition (over composition in B*) and identity (over identity in* B*) for both 1- and 2-cells. These operations are subject to "axioms over axioms in* B*".*

Definition 5 (Ahrens et al. 2021, Def. 6.2, `total_bicat`). *Given a displayed bicategory* D *over* B*, we define the total bicategory* $\int \mathsf{D}$ *to have, as cells at dimension i, pairs* $(b, \overline{b})$ *where b is a cell of* B *at dimension i and* $\overline{b}$ *is a cell of* D *over b, with the obvious source and target.*

We define the projection pseudofunctor $\pi : \int \mathsf{D} \to \mathsf{B}$ *to be given, on any cell, by* $(b, \overline{b}) \mapsto b$*.*

Definition 6 (`disp_subbicat`). *Suppose that we have a bicategory* B*, a predicate* P_0 *on the 0-cells (by which we mean a proposition* $P_0(a)$ *for every 0-cell a), and a predicate* P_1 *on the 1-cells. Furthermore, we assume that* P_1 *is closed under identity and composition; that is,* $P_1(\mathrm{id}\,(x))$ *holds for every x satisfying* P_0 *and that for all* $f : x \to y$ *and* $g : y \to z$ *between 0-cells satisfying* P_0 *we have* $P_1(f \cdot g)$ *if we have both* $P_1(f)$ *and* $P_1(g)$*. Then we define a displayed bicategory* $\mathsf{SubBicat}(P_0, P_1)$ *on* B *such that*

- *the type of displayed objects over x is* $P_0(x)$*;*
- *the type of displayed 1-cells over* $f : x \to y$ *is* $P_1(f)$*; and*
- *the type of displayed 2-cells over* $\theta : f \Rightarrow g$ *is the unit type.*

The total bicategory of this bicategory selects 0-cells and 1-cells from the original bicategory B. Its objects are objects $x : \mathsf{B}$ such that $P_0(x)$, its 1-cells are 1-cells $f : x \to y$ in B such that $P_1(f)$, and its 2-cells are the same as 2-cells $\tau : f \Rightarrow g$ in B. The projection pseudofunctor $\pi : \mathsf{SubBicat}(P_0, P_1) \to \mathsf{B}$ is the inclusion.

Remark 7. Note that Definition 6 is defined slightly differently in the formalization. In `disp_subbicat`, we do not require that P_0 and P_1 are propositions. Instead, we only require this in the theorem `is_univalent_2_subbicat` that shows the univalence of this displayed bicategory.

We instantiate Definition 6 to define bicategories of categories with a certain structure.

Example 8 (`StructuredCategories.v`). We define the following displayed bicategories over Cat:

- $\mathsf{D}_{\mathsf{Terminal}}$ where $P_0(\mathsf{C})$ says that C has a terminal object and $P_1(F)$ says that F preserves terminal objects. We denote its total bicategory by $\mathsf{Cat}_{\mathsf{Terminal}}$.
- $\mathsf{D}_{\mathsf{BinProd}}$ where $P_0(\mathsf{C})$ says that C has binary products and $P_1(F)$ says that F preserves binary products. We denote its total bicategory by $\mathsf{Cat}_{\mathsf{BinProd}}$.
- $\mathsf{D}_{\mathsf{Pullback}}$ where $P_0(\mathsf{C})$ says that C has pullbacks and $P_1(F)$ says that F preserves pullbacks. We denote its total bicategory by $\mathsf{Cat}_{\mathsf{Pullback}}$.
- $\mathsf{D}_{\mathsf{FinLim}}$ where $P_0(\mathsf{C})$ says that C has finite limits and $P_1(F)$ says that F preserves finite limits. We denote its total bicategory by $\mathsf{Cat}_{\mathsf{FinLim}}$.
- $\mathsf{D}_{\mathsf{Initial}}$ where $P_0(\mathsf{C})$ says that C has an initial object and $P_1(F)$ says that F preserves initial objects. We denote its total bicategory by $\mathsf{Cat}_{\mathsf{Initial}}$.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

- $\mathsf{D}_{\mathsf{BinSum}}$ where $P_0(\mathsf{C})$ says that C has binary coproducts and $P_1(F)$ says that F preserves binary coproducts. We denote its total bicategory by $\mathsf{Cat}_{\mathsf{BinSum}}$.

The above examples can be defined using Definition 6 assuming that the categories involved are univalent, since then the type families P_0 and P_1 stating a choice of limits are indeed predicates. For nonunivalent categories, one could consider instead the truncated predicates stating mere existence of limits.

The total bicategories of the displayed bicategories in Example 8 are bicategories of categories with certain (co)limits. In addition, we can combine the structure of these by taking the product of the suitable displayed bicategories. Note that we could use similar methods to construct bicategories of extensive categories, regular categories, exact categories, and (pre)toposes.

The following (displayed) bicategories are used:

Example 9 (`trivial_displayed_bicat`). Given bicategories B_1 and B_2, we define a displayed bicategory ${\mathsf{B}_1}^{+\mathsf{B}_2}$ over B_1 as follows:

- The displayed 0-cells over $x : \mathsf{B}_1$ are 0-cells $y : \mathsf{B}_2$.
- The displayed 1-cells over $f : x_1 \to x_2$ from $y_1 : \mathsf{B}_2$ to $y_2 : \mathsf{B}_2$ are 1-cells $g : y_1 \to y_2$ in B_2.
- The displayed 2-cells over $\theta : f \Rightarrow g$ from $g_1 : y_1 \to y_2$ to $g_2 : y_1 \to y_2$ are 2-cells $\tau : g_1 \Rightarrow g_2$ in B_2.

The total bicategory is $\int {\mathsf{B}_1}^{+\mathsf{B}_2} \simeq \mathsf{B}_1 \times \mathsf{B}_2$ with projection $\pi : \mathsf{B}_1 \times \mathsf{B}_2 \to \mathsf{B}_1$.

Example 10 (`cod_disp_bicat`). Let B be a bicategory. Define a displayed bicategory $\mathsf{B}^{\downarrow}$ over B as follows:

- The displayed objects over $y : \mathsf{B}$ are 1-cells $x \to y$.
- The displayed 1-cells over $g : y_1 \to y_2$ from $h_1 : x_1 \to y_1$ to $h_2 : x_2 \to y_2$ are pairs consisting of a 1-cell $f : x_1 \to x_2$ and an invertible 2-cell $\gamma : g \cdot h_2 \Rightarrow h_1 \cdot f$.
- Given displayed 1-cells $f_1 : x_1 \to x_2$ with $\gamma_1 : g_1 \cdot h_2 \Rightarrow h_1 \cdot f_1$, and $f_2 : x_1 \to x_2$ with $\gamma_2 : g_2 \cdot h_2 \Rightarrow h_1 \cdot f_2$, we define the displayed 2-cells over $\theta : g_1 \Rightarrow g_2$ from (f_1, γ_1) to (f_2, γ_2) as 2-cells $\tau : f_1 \Rightarrow f_2$ such that $\gamma_1 \bullet (h_1 \lhd \tau) = (\theta \rhd h_2) \bullet \gamma_2$.

The generated total bicategory is the *arrow bicategory*, $\int \mathsf{B}^{\downarrow} = \mathsf{B}^{\to}$ with projection given by the codomain, $\mathsf{cod} : \mathsf{B}^{\to} \to \mathsf{B}$.

In the next example, we write StrictCat for the bicategory of strict categories and $\underline{\mathsf{StrictCat}}$ for the category of strict categories.

Example 11 (`disp_bicat_of_functors_into_cat`). We define a displayed bicategory IndexedCat over StrictCat as follows:

- The displayed objects over $\mathsf{C} : \mathsf{StrictCat}$ are functors $G : \mathsf{C} \to \underline{\mathsf{StrictCat}}$.
- The displayed 1-cells over $F : \mathsf{C}_1 \to \mathsf{C}_2$ from $G_1 : \mathsf{C}_1 \to \underline{\mathsf{StrictCat}}$ to $G_2 : \mathsf{C}_2 \to \underline{\mathsf{StrictCat}}$ are natural transformations $\gamma : G_1 \Rightarrow F \cdot G_2$.
- The displayed 2-cells over $n : F_1 \Rightarrow F_2$ from $\gamma_1 : G_1 \Rightarrow F_1 \cdot G_2$ to $\gamma_2 : G_1 \Rightarrow F_2 \cdot G_2$ are proofs that for all $x : \mathsf{C}$ we have $\gamma_1(x) \cdot G_2(n(x)) = \gamma_2(x)$.

The associated projection pseudofunctor $\pi : \int \mathsf{IndexedCat} \to \mathsf{StrictCat}$ maps functors $\mathsf{C} \to \underline{\mathsf{StrictCat}}$ to their domain.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

Example 12 (`disp_bicat_of_opcleaving`). We define a displayed bicategory $\mathsf{OpCleav}$ over Cat as follows:

- The displayed objects over $C : \mathsf{Cat}$ are displayed categories D over C together with an opcleaving.
- The displayed 1-cells over $F : C_1 \to C_2$ from D_1 to D_2 are displayed functors $\overline{F}$ from D_1 to D_2 that preserve opcartesian morphisms.
- The displayed 2-cells over $\theta : F \Rightarrow G$ from $\overline{F} : D_1 \xrightarrow{F} D_2$ to $\overline{G} : D_1 \xrightarrow{G} D_2$ are displayed natural transformations from $\overline{F}$ to $\overline{G}$ over θ.

The associated projection pseudofunctor $\pi : \int\mathsf{OpCleav} \to \mathsf{Cat}$ maps any opcleaving to its codomain category.

Similarly, we define displayed bicategories Cleav and IsoFib of cleavings and isocleavings, respectively.

The idea of displayed (bi)categories transfers to functors:

Definition 13 (Ahrens et al. 2021, Def. 8.2, `disp_psfunctor`). *Given $F : B \to B'$ and D and D' displayed bicategories over B and B', respectively, a displayed pseudofunctor $\overline{F}$ over F consists of*

- *for all objects $x : B$ and $\overline{x} : D_x$ an object $\overline{F}(\overline{x}) : D'_{F(x)}$;*
- *for all displayed morphisms $\overline{f} : \overline{x} \xrightarrow{f} \overline{y}$, a displayed 1-cell $\overline{F}(\overline{f}) : \overline{F}(\overline{x}) \xrightarrow{F(f)} \overline{F}(\overline{y})$;*
- *for all displayed 2-cells $\overline{\theta} : \overline{f} \overset{\theta}{\Rightarrow} \overline{g}$, a displayed 2-cell $\overline{F}(\overline{\theta}) : \overline{F}(\overline{f}) \overset{F(\theta)}{\Longrightarrow} \overline{F}(\overline{g})$;*
- *coherence isomorphisms for identity and composition of displayed 1-cells, over the analogous isomorphisms in the base;*
- *satisfying suitable equations over the corresponding equations in the base.*

We denote by $\int\overline{F} : \int D \to \int D'$ the induced total pseudofunctor.

Remark 14. The square of pseudofunctors

$$\begin{array}{ccc} \int D & \xrightarrow{\int\overline{F}} & \int D' \\ {\scriptstyle \pi_D}\downarrow & & \downarrow{\scriptstyle \pi_{D'}} \\ B & \xrightarrow{F} & B' \end{array}$$

induced by $\overline{F}$ over F commutes *up to judgmental equality*.

Furthermore, we need pullbacks and products in bicategories.

Definition 15 (`has_pb`). *Let B be a bicategory, and suppose we have two 1-cells $f : a \to c$ and $g : b \to c$. A pullback structure for f and g on an object $x : B$ together with two 1-cells $\pi_1 : x \to a$ and $\pi_2 : x \to b$ and an invertible 2-cell $\gamma : p \cdot f \Rightarrow q \cdot g$ is given by the following data:*

- *for all 1-cells $p' : z \to a$ and $q' : z \to b$ and invertible 2-cells $\gamma' : p' \cdot f \Rightarrow q' \cdot g$, we have a 1-cell $u : z \to x$ together with invertible 2-cells $\theta : u \cdot p \Rightarrow p'$ and $\tau : u \cdot q \Rightarrow q'$ such that*

$$\alpha \bullet \theta \triangleright f \bullet \gamma' = u \triangleleft \gamma \bullet \alpha^{-1} \bullet \tau \triangleright g.$$

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

- *for all 1-cells* $u_1, u_2 : z \to x$ *and 2-cells* $\theta : u_1 \bullet p \Rightarrow u_2 \bullet p$ *and* $\tau : u_1 \bullet q \Rightarrow u_2 \bullet q$ *such that*

$$u_1 \triangleleft \gamma \bullet \alpha \bullet \tau \triangleright q \bullet \alpha^{-1} = \alpha \bullet \theta \triangleright f \bullet \alpha^{-1} \bullet u_2 \triangleleft \gamma,$$

we have a unique 2-cell $v : u_1 \Rightarrow u_2$ *such that* $v \triangleright p = \theta$ *and* $v \triangleright q = \tau$.

In Definition 15, it is irrelevant if we postulate data to be given explicitly or to merely exist, provided the bicategory B is univalent:

Proposition 16 (`isaprop_has_pb_ump`). *If* B *is univalent, then the type of pullback structures on* $(x, \pi_1, \pi_2, \gamma)$ *is a proposition.*

Remark 17. There are different notions of pullback in bicategories depending on whether $p \cdot f$ and $q \cdot g$ are postulated to be related up to an equality, invertible 2-cell or even just a 2-cell. In Definition 15, the square commutes up to invertible 2-cell. One could also define *strict pullbacks*: this is done similarly to Definition 15, but all involved squares must commute up to equality rather than just up to invertible 2-cell.

Example 18 (`one_types_has_pb`, `has_pb_bicat_of_univ_cats`). The bicategory of groupoids has pullbacks.

The bicategory Cat also has pullbacks, and they are given by iso-comma categories. These are defined as follows: given categories C_1, C_2, and C_3 and functors $F : \mathsf{C}_1 \to \mathsf{C}_3$ and $G : \mathsf{C}_2 \to \mathsf{C}_3$, we define the iso-comma category $F /_{\simeq} G$

- Its objects consist of objects $x : \mathsf{C}_1$ and $y : \mathsf{C}_2$ together with an isomorphism $f : F(x) \to G(y)$
- The morphisms from (x_1, y_1, f_1) to (x_2, y_2, f_2) consists of morphisms $g : x_1 \to x_2$ and $h : y_1 \to y_2$ such that the following diagram commutes:

$$\begin{array}{ccc} F(x_1) & \xrightarrow{f_1} & G(y_1) \\ {\scriptstyle F(g)}\downarrow & & \downarrow{\scriptstyle G(h)} \\ F(x_2) & \xrightarrow[f_2]{} & G(y_2) \end{array}$$

We define functors $\pi_1^{\simeq} : F /_{\simeq} G \to \mathsf{C}_1$ and $\pi_2^{\simeq} : F /_{\simeq} G \to \mathsf{C}_2$ and a natural isomorphism $n^{\simeq} : \pi_1^{\simeq} \cdot F \Rightarrow \pi_2^{\simeq} \cdot G$. The category $F /_{\simeq} G$ together with the functors $\pi_1^{\simeq}, \pi_2^{\simeq}$ and natural isomorphism $n^{\simeq}$ is universal among such data, making it the pullback of F and G.

Example 19 (`LimitsStructuredCategories.v`). The bicategories defined in Example 8 all have pullbacks as well. This is because taking the iso-comma category preserves the desired (co)limits and the projections and pairing functors preserve them.

As a special case of pullbacks in the presence of terminal objects, we can define products in bicategories (`has_binprod_ump`). If B has chosen products, we write $x \times y$ for the product of x and y, and we denote the projections by $\pi_1 : x \times y \to x$ and $\pi_2 : x \times y \to y$.

Notation 20. *In the following, given a cloven fibration, we notate the pullback (or reindexing) of* A *along* f *by* f^*A. *Given a cloven opfibration, we notate the pushforward of* A *along* f *by* $f_!A$.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

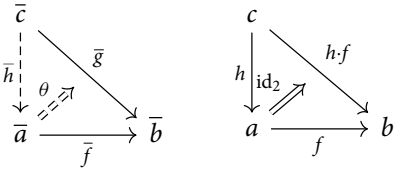
4. Comprehension Bicategories

In this section, we define the notion of global cleaving and local (op)cleavings of bicategories. Afterward, we use these notions to define comprehension bicategories. We are guided by Buckley (2014), where local and global fibrations are defined, and we add definitions for local cloven iso- and opfibrations. However, there is an important difference: while Buckley works in a set-theoretic setting, we reformulate the definitions in a type-theoretic setting using the displayed technology developed by Ahrens et al. (2021) and reviewed in Section 3 – see also Remark 25.

Throughout this section, we assume that B is a bicategory and D is a displayed bicategory over B.

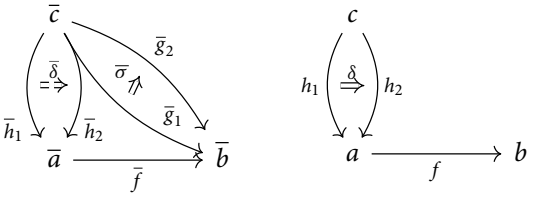
Definition 21 (Buckley 2014, Def. 3.1.1, `cartesian_1cell`). *Let $f : a \to b$ be a 1-cell in* B*, and let $\overline{f} : \overline{a} \xrightarrow{f} \overline{b}$ be a displayed 1-cell over f in* D*. A cartesian structure on $\overline{f}$ consists of the following data. Note that we draw diagrams in the displayed category on the left side and diagrams in the base category on the right side.*

(1) For any $\overline{g} : \overline{c} \xrightarrow{h \cdot f} \overline{b}$, a choice of a displayed morphism $\overline{h} : \overline{c} \xrightarrow{h} \overline{a}$ and a displayed isomorphism θ over the identity isomorphism on $h \cdot f$ in B*.*



We call $(\overline{h}, \theta)$ the lift of $(h, \overline{g})$.

(2) Given lifts $(\overline{h}_1, \theta_1)$ and $(\overline{h}_2, \theta_2)$ of $(h_1, \overline{g}_1)$ and $(h_2, \overline{g}_2)$, respectively, and $\delta : h_1 \Rightarrow h_2$, and a 2-cell $\overline{\sigma} : \overline{g}_1 \Rightarrow \overline{g}_2$ over $\delta \triangleright f$, we have a unique 2-cell $\overline{\delta} : \overline{h}_1 \Rightarrow \overline{h}_2$ over δ such that $\overline{\delta} \triangleright \overline{f} \bullet \theta_2 = \theta_1 \bullet \overline{\sigma}$.



Given a cartesian structure on $\overline{f}$, we call $\mathsf{F}_1(\overline{f}, \overline{g}) := \overline{h}$ and $\mathsf{F}_2(\overline{f}, \overline{g}) := \theta$. We also call $\mathsf{E}(\delta, \overline{\sigma}) := \overline{\delta}$.

Proposition 22. *We have the following cartesian 1-cells:*

- *(`cartesian_1cell_id`) The identity $\overline{\mathrm{id}_1} : \overline{a} \to \overline{a}$ is cartesian.*
- *(`comp_cartesian_1cell`) If $\overline{f} : \overline{a} \to \overline{b}$ and $\overline{g} : \overline{b} \to \overline{c}$ are cartesian, then so is $\overline{f} \cdot \overline{g}$.*

Problem 23. Given two cartesian 1-cells $\overline{f} : \overline{a} \to \overline{b}$ and $\overline{f}' : \overline{a}' \to \overline{b}$, to construct an adjoint equivalence $e : \overline{a} \to \overline{a}'$ over the identity and an invertible 2-cell from $\overline{f}$ to $e \cdot \overline{f}'$.

Construction 24 for Problem 23; (`EquivalenceBetweenCartesian.v`). The 1-cells of the adjoint equivalence are constructed as cartesian factorizations (first item of Definition 21). In that way, we also obtain the desired invertible 2-cell. □

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

Remark 25. Recall that a displayed 1-cell $\overline{f} : \overline{a} \xrightarrow{f} \overline{b}$ in D gives rise to the 1-cell $(f, \overline{f})$ in the total bicategory $\int\mathsf{D}$. The definition of cartesian structure on $\overline{f}$ in D of Definition 21 gives rise to a notion of cartesian structure for $(f, \overline{f})$ in $\int\mathsf{D}$. By expressing the definition of cartesian 1-cell in the displayed bicategory (instead of in the resulting projection $\pi : \int\mathsf{D} \to \mathsf{B}$), we can postulate that a lift in $\int\mathsf{D}$ lies *definitionally* over a given cell in B, not just modulo an invertible 2-cell.

Buckley (2014) shows, in Remark 3.1.6, that these two definitions are equivalent.

Proposition 26 (`isaprop_cartesian_1cell`). *Suppose that* D *is a univalent displayed bicategory over* B*, and let* $\overline{f}$ *be a displayed 1-cell in* D *over* f *in* B*. Then the type of cartesian 1-cell structures on* $\overline{f}$ *is a proposition.*

Definition 27 (`global_cleaving`). *A global cleaving on* D *is a choice, for any* $f : a \to b$ *in* B *and* $\overline{b} : \mathsf{D}_b$*, of*

(1) a displayed object $\overline{a}$ *over* a*;*
(2) a displayed 1-cell $\overline{f} : \overline{a} \xrightarrow{f} \overline{b}$*;*
(3) a cartesian structure on $\overline{f}$*.*

Given a global cleaving on D*, we use the notation* $\overline{b}[f] := \overline{a}$ *and* $L_f(\overline{b}) := \overline{f}$ *to denote the choice given by the cleaving.*

Remark 28. The notion of global cleaving as in Definition 27 gives rise to a notion of cloven fibration on the total bicategory $\int\mathsf{D}$.

Next we look at local cleavings and opcleavings. A 2-cell is opcartesian if and only if it is opcartesian in the 1-categorical sense for the hom-functor. However, in the formalization, we give the following direct definition not relying on hom-categories and prove the characterization via hom-categories afterward (`opcartesian_2cell_weq_opcartesian`). Similarly, we give an unfolded definition of local opcleaving.

Definition 29 (`is_opcartesian_2cell`). *Suppose we have 1-cells* $f, g : x \to y$*, a 2-cell* $\theta : f \Rightarrow g$*, and displayed objects* $\overline{x}$ *and* $\overline{y}$ *over* x *and* y*, respectively. Given displayed 1-cells* $\overline{f} : \overline{x} \xrightarrow{f} \overline{y}$ *and* $\overline{g} : \overline{x} \xrightarrow{g} \overline{y}$ *and a displayed 2-cell* $\overline{\theta} : \overline{f} \overset{\theta}{\Rightarrow} \overline{g}$*, we say that* $\overline{\theta}$ *is 2-opcartesian (or just opcartesian) if for all 1-cells* $h : x \to y$*, displayed 1-cells* $\overline{h} : \overline{x} \xrightarrow{h} \overline{y}$*, 2-cells* $\tau : g \Rightarrow h$*, and displayed 2-cells* $\overline{\gamma} : \overline{f} \overset{\tau \bullet \theta}{\Longrightarrow} \overline{h}$*, there is a unique displayed 2-cell* $\overline{\tau} : \overline{g} \overset{\tau}{\Rightarrow} \overline{h}$ *such that* $\overline{\theta} \bullet \overline{\tau} = \overline{\gamma}$*.*

Being an opcartesian 2-cell is always a property. The notion of *cartesian 2-cells* is analogous.

Definition 30 (`local_opcleaving`). *A local opcleaving on* D *is given by, for every* $\theta : f \Rightarrow g$ *and* $\overline{f} : \overline{x} \xrightarrow{f} \overline{y}$*, a displayed 1-cell* $\overline{g} : \overline{x} \xrightarrow{g} \overline{y}$ *and an opcartesian 2-cell* $\overline{\theta} : \overline{f} \overset{\theta}{\Rightarrow} \overline{g}$*.*

The notions of local cleaving and local isocleaving are defined analogously.

Remark 31. Every displayed bicategory on a univalent bicategory has a local isocleaving. The construction is the same as for categories (Ahrens and Lumsdaine 2019, Construction 5.12).

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

Remark 32. The notions of opcartesian 2-cell and of local opcleaving as in Definition 30 give rise to notions of opcartesian 2-cell and of cloven local opfibration on the total bicategory $\int \mathsf{D}$.

Note that one can construct a local isocleaving from either a local cleaving or a local opcleaving.

Proposition 33 (`CleavingOfBicatIsAProp.v`). *Suppose that* B *is a univalent bicategory and* D *is a univalent displayed bicategory over* D. *Then the types of local (resp. global) (op)cleavings on* D *are propositions.*

Now let us look at some examples of these notions.

Example 34 (`TrivialCleaving.v`). The trivial displayed bicategory ${\mathsf{B}_1}^{+\mathsf{B}_2}$ over B_1 comes equipped with a cloven global fibration. Cartesian 1-cells in ${\mathsf{B}_1}^{+\mathsf{B}_2}$ correspond to adjoint equivalences in B_2. As such, we can take the identity 1-cell as the global lift. In addition, ${\mathsf{B}_1}^{+\mathsf{B}_2}$ also has both a cloven local fibration and a cloven local opfibration.

Example 35 (`CodomainCleaving.v`). Suppose that B is a locally groupoidal bicategory with pullbacks. Since cartesian 1-cells in $\mathsf{B}^{\downarrow}$ correspond to pullback squares, we can construct a global cleaving for $\mathsf{B}^{\downarrow}$ by taking pullbacks. Note that all 2-cells in $\mathsf{B}^{\downarrow}$ are cartesian, because B is locally groupoidal, and thus $\mathsf{B}^{\downarrow}$ also has a local cleaving and a local opcleaving.

Example 36 (`FunctorsIntoCatCleaving.v`). The displayed bicategory $\mathsf{IndexedCat}$ has a global cleaving and a local opcleaving. To construct these, we first observe that a displayed 1-cell $\gamma : G_1 \Rightarrow F_1 \cdot G_2$ from $G_1 : \mathsf{C}_1 \to \underline{\mathsf{StrictCat}}$ to $G_2 : \mathsf{C}_2 \to \underline{\mathsf{StrictCat}}$ is cartesian if and only if it is a natural isomorphism. Now we construct a global cleaving for $\mathsf{IndexedCat}$ as follows: whenever we have functors $F : \mathsf{C}_1 \to \mathsf{C}_2$ and $G : \mathsf{C}_2 \to \underline{\mathsf{StrictCat}}$, then we also get a functor $F \cdot G : \mathsf{C}_1 \to \underline{\mathsf{StrictCat}}$.

Every displayed 2-cell in $\mathsf{IndexedCat}$ is opcartesian. To find local opcartesian lifts, suppose that we have functors $F_1, F_2 : \mathsf{C}_1 \to \mathsf{C}_2$, $G_1 : \mathsf{C}_1 \to \underline{\mathsf{StrictCat}}$ and $G_2 : \mathsf{C}_2 \to \underline{\mathsf{StrictCat}}$, and natural transformations $n : F_1 \Rightarrow F_2$ and $\gamma : G_1 \Rightarrow F_1 \cdot G_2$. Our goal is to construct a natural transformation $G_1 \Rightarrow F_2 \cdot G_2$, and for the desired transformation we take $\gamma \bullet (n \triangleright G_2)$. Hence, $\mathsf{IndexedCat}$ has a local opcleaving.

Example 37 (`OpFibrationCleaving.v`). The displayed bicategory $\mathsf{OpCleav}$ has a global cleaving and a local opcleaving. Given a functor $F : \mathsf{C}_1 \to \mathsf{C}_2$ and a displayed category D_2 over C_2, we construct a displayed category $F^*(\mathsf{D}_2)$ over C_1:

- The displayed objects over $x : \mathsf{C}_1$ are displayed objects in D_2 over $F(x)$.
- The displayed morphisms over $f : x \to y$ from $\overline{x}$ to $\overline{y}$ are displayed morphisms over $\overline{x} \xrightarrow{F(f)} \overline{y}$.

Note that $F^*(\mathsf{D}_2)$ inherits any opcleaving from D_2. In addition, we have a displayed functor over F from $F^*(\mathsf{D}_2)$ to D_2 that preserves cartesian morphisms.

Opcartesian 2-cells in $\mathsf{OpCleav}$ correspond to displayed natural transformations of which all components are opcartesian. We form local lifts pointwise. Since displayed 1-cells in $\mathsf{OpCleav}$ preserve cartesian morphisms, cartesian 2-cells are preserved under both left and right whiskering.

Similarly, we can show that the displayed bicategory Cleav has a global and a local cleaving (`FibrationCleaving.v`). We finish this section by defining comprehension bicategories. Examples of those are given in Section 5.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

Definition 38 (`comprehension_bicat`). *A weak comprehension bicategory is given by a bicategory* B*, a displayed bicategory* D *over* B*, and a displayed pseudofunctor* χ *over the identity on* B *as pictured below,*

$$\begin{array}{ccc} \mathsf{D} & \xrightarrow{\chi} & \mathsf{B}^{\downarrow} \\ & \searrow \quad \swarrow & \\ & \mathsf{B} & \end{array}$$

satisfying the following properties (see also Proposition 33):

(1) D *is equipped with a cloven global fibration;*
(2) D *has a cloven local opfibration;*
(3) opcartesian 2-cells of D *are preserved under both left and right whiskering.*

A comprehension bicategory is a weak comprehension bicategory such that

(4) χ *preserves cartesian 1-cells and opcartesian 2-cells.*

Remark 39. Recall from Remarks 14, 28, and 32 that the displayed pseudofunctor $\chi : \mathsf{D} \to \mathsf{B}^{\downarrow}$ of Definition 38 gives rise to a *strictly* commuting diagram of pseudofunctors

$$\begin{array}{ccc} \int \mathsf{D} & \xrightarrow{\int \chi} & \int \mathsf{B}^{\downarrow} = \mathsf{B}^{\rightarrow} \\ & \searrow \quad \swarrow_{\mathrm{cod}} & \\ & \mathsf{B} & \end{array}$$

with the structure of a global fibration and local opfibration in the sense of Buckley (2014).

Remark 40. In the definition of comprehension categories, one usually requires that the functor χ preserves cartesian morphisms. The analogous requirement for comprehension bicategories is that the pseudofunctor χ preserves both cartesian 1-cells and opcartesian 2-cells – this is condition 4 in Definition 38.

We have factored out this requirement, since for our interpretation of the language BTT (introduced in Section 8), we do not need this condition; that is, we can interpret BTT in any *weak* comprehension bicategory. The reason has to do with how we interpret terms of the language BTT in a (weak) comprehension bicategory. Traditionally, in a comprehension category, terms of MLTT are interpreted as sections of projections. Specifically, every type A in context Γ gives rise, in the interpretation, to a morphism $\pi_A : \Gamma.A \to \Gamma$ in the arrow category. A term of type A in context Γ in MLTT is then interpreted as a section of π_A.

In our setting, terms of BTT will be interpreted differently. Instead of interpreting BTT terms as sections of projections, we interpret them as morphisms over the identity. More concretely, in a bicategory D displayed over B, a term in context Γ of type B with source A of BTT is interpreted as a morphism from A to B over the identity of Γ.

In the 1-categorical case, these two ways of interpreting terms are equivalent in many examples. The reason is that, in practice, most comprehension categories are full, which means that χ is fully faithful, and that every fiber has a terminal object. From these two assumptions, one can conclude that these two ways of interpreting terms are actually equivalent. However, in the bicategorical setting, these two ways are often *not* equivalent: for example, in the comprehension bicategory of displayed categories with opcleavings, i.e. opfibrations (Examples 12, 37, and 45). There, morphisms over the identity are required to preserve opcartesian morphisms, but such a requirement is not present for sections of the projection. If one were to include MLTT style terms (which are

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

interpreted as sections of projections) in BTT, then to give an interpretation one would also need that χ preserves cartesian 1-cells and opcartesian 2-cells in order to interpret substitution. This is why we present here both definitions.

In short, all the examples presented in Section 5 are comprehension bicategories; the interpretation of BTT given in Section 8 only requires a *weak* comprehension bicategory.

Remark 41. Contravariant and isovariant comprehension bicategories are defined analogously with the notions of (iso)cleaving and (iso)cartesian taking the place of opcleaving and opcartesian in Items 2 and 3 of Definition 38. Some of our examples of comprehension bicategories can similarly be equipped with a contravariant and isovariant comprehension structure.

5. Examples of Comprehension Bicategories

In this section, we give several (classes of) instances of comprehension bicategories. In some of these instances, we recognize structures studied previously in the context of higher-dimensional and directed type theory.

Example 42 (`trivial_comprehension_bicat`). Suppose we have a bicategory B with products. Then we have the following comprehension bicategory

$$\mathsf{B}^{+\mathsf{B}} \xrightarrow{\ \chi\ } \mathsf{B}^{\downarrow}$$

$$\mathsf{B}$$

The displayed pseudofunctor $\chi : \mathsf{B}^{+\mathsf{B}} \to \mathsf{B}^{\downarrow}$ sends the object y_2 over y_1 to the product projection $y_1 \times y_2 \to y_1$, that is, the corresponding total pseudofunctor $\mathsf{B} \times \mathsf{B} \to \mathsf{B}^{\downarrow}$ is defined by $(y_1, y_2) \mapsto \pi_1 : y_1 \times y_2 \to y_1$.

Example 42 corresponds roughly to the semantics studied by Fiore and Saville (2019), but without the type formers.

Another example of a comprehension bicategory comes from locally groupoidal bicategories, such as Grpd.

Example 43 (`locally_grpd_comprehension_bicat`). Let B be a locally groupoidal bicategory with pullbacks. Then we get the following comprehension bicategory

$$\mathsf{B}^{\downarrow} \xrightarrow{\ \mathrm{id}\ } \mathsf{B}^{\downarrow}$$

$$\mathsf{B}$$

Since every 2-cell is opcartesian in $\mathsf{B}^{\downarrow}$ if B is locally groupoidal, the displayed bicategory $\mathsf{B}^{\downarrow}$ has a local opcleaving.

Example 43 models *undirected* reductions between terms. It is thus related to the groupoid model of type theory by Hofmann and Streicher (1994) and to the definition of comprehension 2-category by Garner (2009). A more detailed comparison to Garner's comprehension 2-categories is given in Remark 65.

We can also consider *directed* versions of this example by using categories instead of groupoids.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

Example 44 (`functors_into_cat_comprehension_bicat`). We construct the following comprehension bicategory using cloven split opfibrations.

$$\mathsf{IndexedCat} \xrightarrow{\quad\chi\quad} \mathsf{StrictCat}^{\downarrow}$$

$$\mathsf{StrictCat}$$

To define χ, we use the Grothendieck construction. Specifically, from a functor $G : \mathsf{C} \to \underline{\mathsf{StrictCat}}$ we construct a category $\int G$ as follows

(1) The objects of $\int G$ are pairs (x, y) where $x : \mathsf{C}$ and $y : G(x)$.
(2) The morphisms from (x_1, y_1) to (x_2, y_2) are pairs (f, g) where $f : x_1 \to x_2$ and $g : G(f)(y_1) \to y_2$.

Note that we also have a projection $\pi_1 : \int G \to \mathsf{C}$. In addition, from a displayed 1-cell $\gamma : G_1 \Rightarrow F \cdot G_2$, we get a functor $\int \gamma : \int G_1 \to \int G_2$ and a natural isomorphism $\gamma_c : \pi_1 \cdot F \Rightarrow \int \gamma \cdot \pi_1$. If we have $p : \gamma_1(x) \cdot G_2(n(x)) = \gamma_2(x)$ for every $x : \mathsf{C}_1$, then we get a natural transformation $\int p$ from $\int \gamma_1$ to $\int \gamma_2$.

Example 44 is related to the interpretations given by Licata and Harper (2011) and North (2019) in their works on directed type theories, albeit without considering type formers. However, their notion of term, in both the syntax and the interpretation, is different from ours; see Section 10 for details.

In Example 44, contexts are categories and types over a context C are cloven split opfibrations on C. We also consider a version of this interpretation where there are more types: instead of looking at only those opfibrations that are split, *all* opfibrations are considered.

Example 45 (`opcleaving_comprehension_bicat`). From opcleavings we build the following comprehension bicategory:

$$\mathsf{OpCleav} \xrightarrow{\quad\chi\quad} \mathsf{Cat}^{\downarrow}$$

$$\mathsf{Cat}$$

The pseudofunctor χ sends a displayed category D over C to the functor $\pi_1 : \int \mathsf{D} \to \mathsf{C}$.

Similarly, we can define a *contravariant* comprehension bicategory (`cleaving_contravariant_comprehension_bicat`) using cleavings instead of opcleavings.

6. Internal Street (Op)Fibrations

In this section, we discuss Street (op)fibrations internal to a fixed bicategory B. They will yield, in Section 7, many examples of comprehension bicategories, see Example 57 and Remark 59. The examples of Street opfibrations internal to bicategories of stacks are particularly interesting (see Remark 59).

Note that $\mathsf{B}^{\downarrow}$ comes equipped with a cloven *global* fibration if B has pullbacks. However, to obtain a *local* (op)cleaving, we used that B is locally groupoidal in Example 35. This assumption is avoided in Example 37 where $\mathsf{B} = \mathsf{Cat}$: instead of looking at arbitrary functors, one only considers the opfibrations. We can generalize this idea to arbitrary bicategories by using *internal Street (op)fibrations* (Buckley 2014).

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

The displayed bicategory OpCleav has a local opcleaving where the desired lifts are constructed pointwise. To generalize this to arbitrary bicategories, we need to adjust this definition so that we can lift arbitrary 2-cells. Furthermore, the notion of cloven Grothendieck opfibration of categories is stricter than appropriate for bicategories. If we have $x \to y$ and an object over y, then a cloven Grothendieck fibration gives an object *strictly* living over x, while a Street fibration only gives an object living *weakly* over x (i.e., up to an isomorphism). More information can be found in work by Loregian and Riehl (Loregian and Riehl 2020, Example 4.1.2).

Definition 46 (`internal_sfib`). *Let $p : e \to b$ be a 1-cell in a bicategory* B. *Then p is an internal Street fibration if*

- *For every x : B, the functor $p_* : \underline{\mathsf{B}(x, e)} \to \underline{\mathsf{B}(x, b)}$ of hom-categories is a Street fibration.*
- *For every $f : x \to y$, the following square is a morphism of Street fibrations:*

$$\begin{array}{ccc} \underline{\mathsf{B}(y, e)} & \xrightarrow{f^*} & \underline{\mathsf{B}(x, e)} \\ {\scriptstyle p_*}\downarrow & & \downarrow{\scriptstyle p_*} \\ \underline{\mathsf{B}(y, b)} & \xrightarrow{f^*} & \underline{\mathsf{B}(x, b)} \end{array}$$

A 1-cell is called an internal Street opfibration if it is an internal Street fibration in B^{co} (`internal_sopfib`).

In Cat, internal Street fibrations are the same as Street fibrations of categories. However, the notion of internal Street fibrations can be applied in a wider variety of settings: for example, one could also look at internal Street fibrations in the bicategories from Example 8 or in presheaves or stacks valued in Cat. A classical result on internal Street (op)fibrations is that they are closed under taking pullbacks, see Gray (1966) and Street (1980).

Proposition 47 (`pb_of_sfib_cleaving`). *Street (op-)fibrations are closed under pullback. Concretely, given a pullback square*

$$\begin{array}{ccc} e_1 & \longrightarrow & e_2 \\ {\scriptstyle p_1}\downarrow & & \downarrow{\scriptstyle p_2} \\ b_1 & \longrightarrow & b_2 \end{array}$$

where p_2 is a Street (op)fibration, then p_1 is so, too.

7. Display Map Bicategories

In this section, we introduce "display map bicategories" as a convenient way to build comprehension bicategories.

Let B be a bicategory with pullbacks. We aim to construct a displayed bicategory SOpFib(B) of Street opfibrations over B with both a global cleaving and a local opcleaving. The construction should be done in a modular and general way, since using the same techniques we aim to construct a displayed bicategory SFib(B) of Street fibrations over B with a global and local cleaving. One can imagine more examples, such as the intersection of Street opfibrations with discrete or fully faithful 1-cells.

The common pattern among all these examples is that the displayed bicategory actually represents a subbicategory of the arrow bicategory of B. In the 1-categorical case, such examples

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

are captured by display map categories (Taylor 1999). We adapt that notion to the bicategorical setting.

However, there is a difference between the 1-categorical and the bicategorical case. A display map category on a 1-category C is a full subcategory of the arrow category on C satisfying some requirements. Such a notion would not be useful in the bicategorical case, since neither SOpFib(B) nor SFib(B) are full subbicategories of $\mathsf{B}^{\downarrow}$. Indeed, the 1-cells in SOpFib(B) and SFib(B) are required to preserve (op)cartesian cells, which is not required in $\mathsf{B}^{\downarrow}$. As such, to obtain a notion of display map bicategory that captures these two examples, some care is needed in the definition of display map bicategories.

For that reason, our notion of display map bicategory comes in three different flavors.[3]

Definition 48 (`disp_map_bicat`). *Let* B *be a bicategory. A display map bicategory on* B *consists of*

(1) a predicate P on the 1-cells of B*, and*
(2) a choice of pullbacks along 1-cells satisfying P

such that

(3) if a morphism satisfies P, then it is an internal Street opfibration, and
(4) P is closed under pullback.

Given a display map bicategory B*, a display map of* B *is a 1-cell of* B *together with a proof of the predicate P.*

We also define notions of contravariant display map bicategory and isovariant display map bicategory: for the former, condition (3) above is replaced by the condition that the display maps are required to be internal Street fibrations, while for the latter, condition (3) is omitted. The reason for this terminology is that every display map bicategory of a given variance gives rise to a comprehension bicategory with the same variance.

Example 49 (`sopfib_disp_map_bicat_is_covariant`). Let B be a locally univalent bicategory with pullbacks. Since B has pullbacks, pullbacks exist along all 1-cells, and in particular, along Street opfibrations. In addition, Street opfibrations are closed under pullbacks by Proposition 47. Hence, we have a display map bicategory in which the predicate P says that a 1-cell is an internal Street opfibration.

Remark 50. Note that in Example 49, we assume B to be locally univalent. This is to guarantee that the notion of being a Street opfibration is actually a proposition as required in Definition 48. However, we expect that this assumption can be dropped by suitably truncating the notion of Street opfibration.

Analogously, one can define a contravariant display map bicategory of Street fibrations. There are other examples of display map bicategories as well, for instance discrete Street opfibrations. To define those, we first need the notion of discrete morphisms.[4]

Definition 51 *A 1-cell $f : a \to b$ in a bicategory* B *is called*

- *(`faithful_1cell`) faithful if for every object x, the functor $f_* : \underline{\mathsf{B}(x, a)} \to \underline{\mathsf{B}(x, b)}$ given by postcomposition with f is faithful;*

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

- *(`conservative_1cell`) conservative if for every object x, the functor $f_* : \underline{\mathsf{B}(x,a)} \to \underline{\mathsf{B}(x,b)}$ is conservative;*
- *(`discrete_1cell`) discrete if for every object x, the functor $f_* : \underline{\mathsf{B}(x,a)} \to \underline{\mathsf{B}(x,b)}$ is faithful and conservative.*

Example 52 (`discrete_sopfib_disp_map_bicat_is_covariant`). Let B be a bicategory with pullbacks. Since both faithful and conservative 1-cells are closed under pullbacks, discrete 1-cells are as well. Hence, we get a display map bicategory of discrete Street opfibrations.

Every display map bicategory gives rise to a displayed bicategory as follows.

Definition 53 (`disp_map_bicat_to_disp_bicat`). *Let B be a bicategory and let D be a display map bicategory on B. We define a displayed bicategory $\overline{\mathsf{D}}$ over B as follows:*

- *The objects over $b : B$ are pairs $e : \mathsf{B}$ and $p : e \to b$ such that p is a display map;*
- *The 1-cells over $f : b_1 \to b_2$ from $p_1 : e_1 \to b_1$ to $p_2 : e_2 \to b_2$ are pairs of a 1-cell $g : e_1 \to e_2$ and an invertible 2-cell $g \cdot p_2 \Rightarrow p_1 \cdot f$ such that g preserves opcartesian 2-cells;*
- *The 2-cells are as in Example 10.*

Similarly, every contravariant display map bicategory gives rise to a displayed bicategory. However, the 1-cells are required to preserve cartesian 2-cells. For isovariant display map bicategories, we do not make any restriction on the 1-cells. This way, we obtain displayed bicategories $\mathsf{SOpFib}(\mathsf{B})$ and $\mathsf{SFib}(\mathsf{B})$ over a bicategory B with pullbacks. In $\mathsf{SOpFib}(\mathsf{B})$, 2-cells are the same as 2-cells in $\mathsf{B}^{\to}$. However, 1-cells are a bit different: while 1-cells in $\mathsf{B}^{\to}$ are squares

$$\begin{array}{ccc} e_1 & \xrightarrow{f_e} & e_2 \\ {\scriptstyle p_1}\downarrow & & \downarrow{\scriptstyle p_2} \\ b_1 & \xrightarrow[f_b]{} & b_2 \end{array}$$

that commute up to invertible 2-cell, 1-cells in $\mathsf{SOpFib}(\mathsf{B})$ have the additional requirement that whiskering with f_e preserves opcartesian 2-cells. Similarly, we can define the bicategory $\mathsf{SFib}(\mathsf{B})$ where the objects are internal Street fibrations and where the 1-cells preserve cartesian 2-cells.

Now we can construct the desired cleavings.

Example 54 (`DisplayMapBicatCleaving.v`). Let B be a bicategory and let D be a display map bicategory. As in Example 35, cartesian 1-cells are the same as pullback squares. Hence, we can construct a global cleaving for D using pullbacks and Proposition 47. To construct a local opcleaving for D, we use that it is contained in $\mathsf{SOpFib}(\mathsf{B})$, and then we can use the same construction as for $\mathsf{SFib}(\mathsf{B})$ (Buckley 2014, Example 3.4.6).

Similarly, given a contravariant display map bicategory, one obtains both a global and a local cleaving. However, from an isovariant display map bicategory, one only gets a global cleaving and a local isocleaving. One can instantiate Example 54 to internal Street opfibrations to obtain a global cleaving and a local opcleaving for $\mathsf{SOpFib}(\mathsf{B})$ if B has pullbacks.

Using Example 54, we can generalize the example of (op)fibrations to arbitrary display map bicategories. We discuss the interest in this generalization in Remark 59.

Problem 55. Given a display map bicategory D on a bicategory B, to define a comprehension bicategory.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

Construction 56 for Problem 55; `disp_map_bicat_comprehension_bicat`. From bicategory B and a display map bicategory D on B, we construct the comprehension bicategory

$$\begin{array}{ccc} \mathsf{D} & \xrightarrow{\chi} & \mathsf{B}^{\downarrow} \\ & \searrow & \downarrow \\ & & \mathsf{B} \end{array}$$

The (displayed) pseudofunctor χ is the inclusion of D into $\mathsf{B}^{\downarrow}$. □

Similarly, we get a contravariant comprehension bicategory from a contravariant display map bicategory, and an isovariant comprehension bicategory from an isovariant display map bicategory. We can instantiate Example 56 to internal Street opfibrations to get the following comprehension bicategory.

Example 57 (`internal_sopfib_comprehension_bicat`). For a bicategory B with pullbacks (see, e.g., Example 19), we construct the comprehension bicategory

$$\begin{array}{ccc} \mathsf{SOpFib(B)} & \xrightarrow{\chi} & \mathsf{B}^{\downarrow} \\ & \searrow & \downarrow \\ & & \mathsf{B} \end{array}$$

The (displayed) pseudofunctor χ forgets that the morphisms in SOpFib(B) are internal Street opfibrations.

Example 58 By Example 57 any bicategory with pullbacks, thus in particular any bicategory of stacks (Street 1982), gives rise to a comprehension bicategory.

Remark 59. Recall from Section 2.3.1 that Coquand et al. (2017) constructed models of MLTT in stacks valued in groupoids. Using those models, they proved the independence of countable choice in univalent foundations.

With our comprehension bicategories of stacks (not just ones valued in groupoids), one could follow Coquand et al. (2017) and study the validity and independence of logical principles in *directed* type theory.

Note that we can specialize Example 57 to each of the examples given in Example 8. This is because we showed in Example 19 that those bicategories have pullbacks. In the case of $\mathsf{Cat}_{\mathsf{Terminal}}$, we get a comprehension bicategory in which

- the "contexts" are categories with a terminal object; and
- the "types" are cloven opfibrations of categories which preserve terminal objects.

Similarly, we can instantiate this to the other categories mentioned in Example 8.

Remark 60. Note that, similarly, we can construct a contravariant comprehension bicategory from SFib(B).

8. The Type Theory BTT

In this section, we extract a core syntax for two-dimensional type theory from our semantic model. We call the resulting type theory *Bicategorical Type Theory* (BTT). In Section 9, we prove

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

soundness of our syntax by giving an interpretation of the syntax in any weak comprehension bicategory.

The syntax extracted here is *maximally general*, in the sense that it reflects the structure of a general comprehension bicategory. In Section 10, we propose several orthogonal simplifications to the syntax, along with the corresponding semantic structure and properties. As such, our syntax and semantics are to be viewed as a framework to study different semantic structures and their corresponding internal languages, rather than as one particular pair of syntax and semantics.

In Section 8.1 we present the judgment forms of BTT, as well as the rules extracted from the bicategories of contexts and of types, respectively. In Sections 8.2 and 8.3 we present the comprehension and substitution rules, respectively.

For reasons of space we omit here several rules, namely the naturality rules in Figs. 8 and 9, and the coherencies of trifunctors. Such rules, written out in the "linear" syntax used here, would take up several lines and would be difficult to understand. Consequently, the syntax presented here is not *complete*. The presentation of the complete syntax and a suitable completeness theorem is left for future work.

8.1 Judgments and basic rules

BTT features contexts, substitutions, types, *generalized* terms, and reductions between terms. As befits the bicategorical semantics, judgmental equality is only postulated between parallel reductions.

There are eight kinds of *judgments* in BTT:

(1) Γ ctx, which is read as "Γ is a context";
(2) $\Delta \vdash s : \Gamma$ (given Δ, Γ ctx), which is read as "s is a substitution from Δ to Γ";
(3) $\Delta \vdash r : s \rightsquigarrow t : \Gamma$ (where $\Delta \vdash s, t : \Gamma$), which is read as "$r$ is a reduction from s to t";
(4) $\Delta \vdash r \equiv r' : s \rightsquigarrow t : \Gamma$ (where $\Delta \vdash r, r' : s \rightsquigarrow t : \Gamma$), which is read as "$r$ is equal to r'";
(5) $\Gamma \vdash T$ type (where Γ ctx), which is read as "T is a type in context Γ";
(6) $\Gamma \mid S \vdash t : T$ (where $\Gamma \vdash S, T$ type), which is read as "t is a term in T depending on S in context Γ";
(7) $\Gamma \mid S \vdash \rho : t \rightsquigarrow t' : T$ (where $\Gamma \mid S \vdash t, t' : T$), which is read as "$\rho$ is a reduction from t to t'";
(8) $\Gamma \mid S \vdash \rho \equiv \rho' : t \rightsquigarrow t' : T$ (where $\Gamma \mid S \vdash \rho, \rho' : t \rightsquigarrow t' : T$), which is read as "$\rho$ is equal to ρ'".

We often abbreviate the above judgments and write, e.g., just $\rho : t \rightsquigarrow t'$ instead of $\Gamma \mid S \vdash \rho : t \rightsquigarrow t' : T$. For these judgments, we have rules that express the bicategorical structure of contexts and types. Rules are given in Fig. 1 for the bicategory of contexts, and in Fig. 2 for the bicategory of types.

We also introduce symbols which read like judgments but stand for several judgments, using the composition and identities introduced in Figs. 1 and 2.

(1) $\Delta \vdash \rho : s \tilde{\rightsquigarrow} t : \Gamma$ stands for the following four judgments.
   - $\Delta \vdash \rho : s \rightsquigarrow t : \Gamma$ $\qquad$ $\Delta \vdash \rho^{-1} : t \rightsquigarrow s : \Gamma$
   - $\rho \bullet \rho^{-1} \equiv 1_s$ $\qquad$ $\rho^{-1} \bullet \rho \equiv 1_t$
(2) $\Gamma \mid S \vdash \rho : t \tilde{\rightsquigarrow} t' : T$ stands for the following four judgments.
   - $\Gamma \mid S \vdash \rho : t \rightsquigarrow t' : T$ $\qquad$ $\Gamma \mid S \vdash \rho^{-1} : t' \rightsquigarrow t : T$
   - $\rho \bullet \rho^{-1} \equiv 1_t$ $\qquad$ $\rho^{-1} \bullet \rho \equiv 1_{t'}$

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

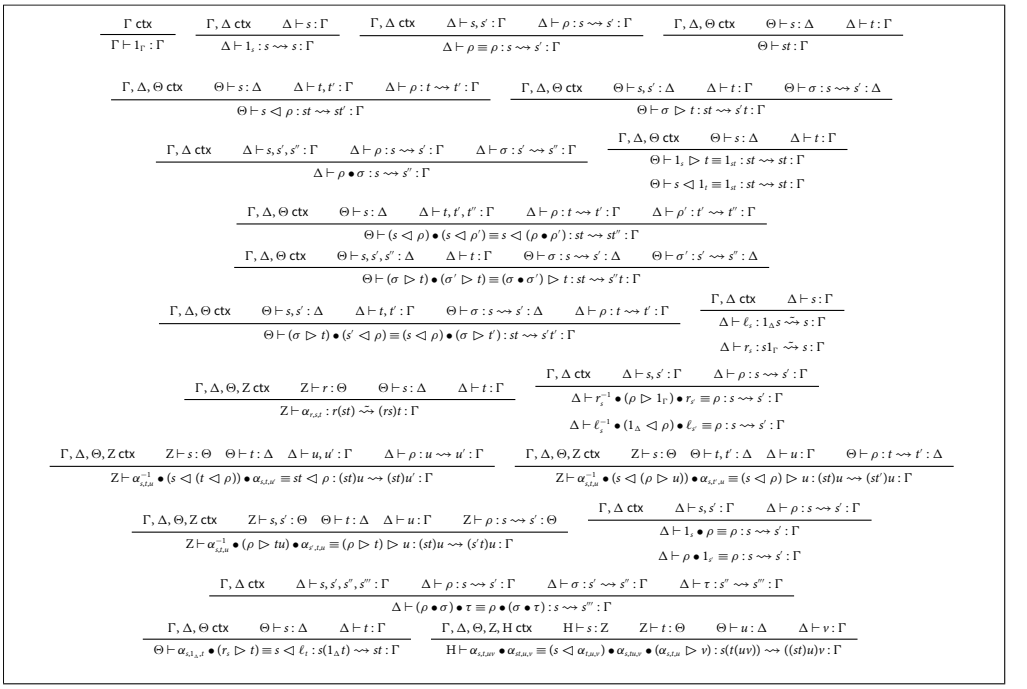


Figure 1. Rules for the bicategory of contexts.

(3) $\Delta \mathrel{\tilde{\vdash}} s : \Gamma$ stands for the following four judgments.
- $\Delta \vdash s : \Gamma \qquad \Gamma \vdash s^{-1} : \Delta$
- $s^{\ell} : ss^{-1} \mathrel{\tilde{\rightsquigarrow}} 1_{\Delta} \qquad s^{\rho} : s^{-1}s \mathrel{\tilde{\rightsquigarrow}} 1_{\Gamma}$

We also require that s^{ℓ} and s^{ρ} form an adjoint equivalence. This can be specified by formulating the usual triangle equalities as equality judgments.

(4) $\Gamma \mid S \mathrel{\tilde{\vdash}} t : T$ stands for the following four judgments.
- $\Gamma \mid S \vdash t : T \qquad \Gamma \mid T \vdash t^{-1} : S$
- $t^{\ell} : tt^{-1} \mathrel{\tilde{\rightsquigarrow}} 1_S \qquad t^{\rho} : t^{-1}t \mathrel{\tilde{\rightsquigarrow}} 1_T$

Remark 61. By abuse of notation, we write several topically related rules that share all the same hypotheses as one rule with several conclusions. These rules then also share the same name, e.g., extend-con-Ty. When referring to a rule by name, it will be clear from the context which of the possible rules we refer to. The names of inference rules in the text are hyperlinks to the corresponding rules (e.g., map). The equality $\equiv$ is assumed to be a congruence for every other constructor and judgment. For brevity, we have not recorded here the resulting rules. When parentheses are omitted, everything is associated to the left: that is, rst stands for $((rs)t)$. Note also that in several rules in which it is necessary to re-associate several four or more terms or substitutions, we have written α instead of a long composition of whiskered associators $\alpha_{\bullet,\bullet,\bullet}$ in the interest of readability.

8.2 Comprehension structure

Comprehension, that is, context extension, is extracted from the pseudofunctor χ. The rules for comprehension are given in Fig. 3. There are some notable differences to comprehension in MLTT. First, the rule extend-con-Tm, which forms a substitution, comes together with a reduction

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

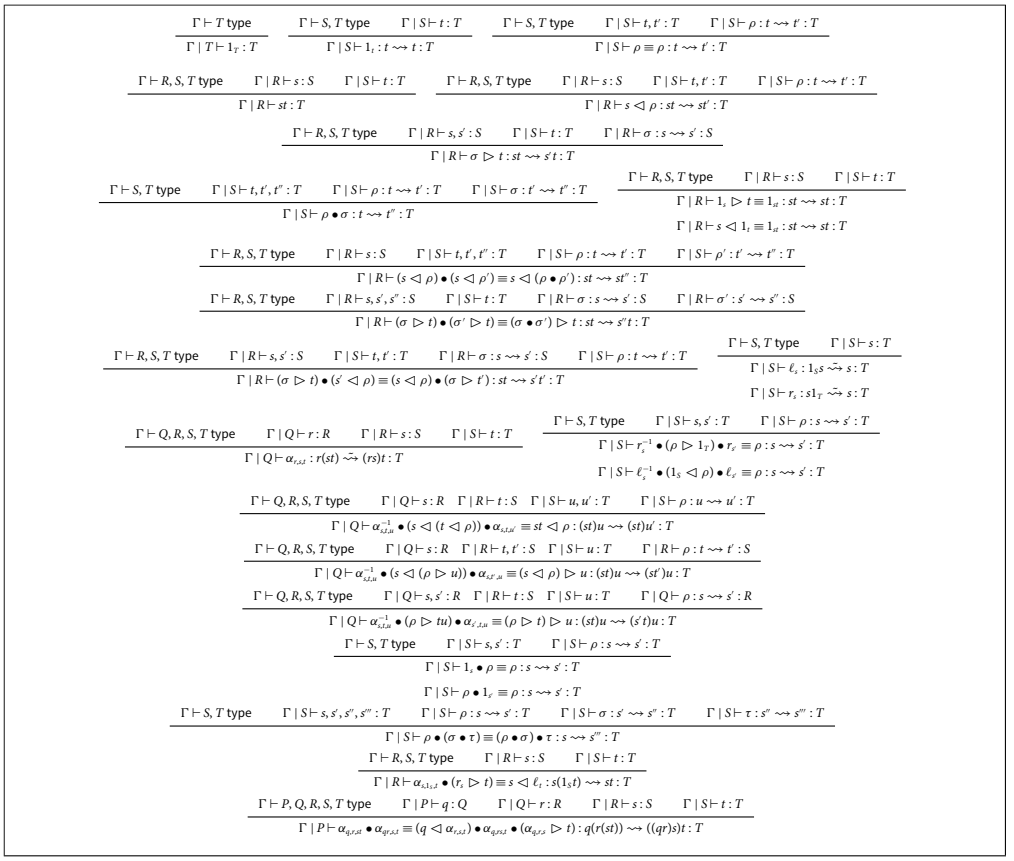


Figure 2. Rules for the the bicategory of types.

that expresses the commutativity of a triangle. Second, we also have a rule extend-con-Red that extends a substitution with a reduction. Since reductions are proof-relevant, this rule comes with a coherency on the commutativity.

8.3 Substitution structure

Substitution is given, in the semantics, by the global and local (op)cleaving structure. We reflect this into the syntax as *explicit substitution*, as was also used, e.g., by Fiore and Saville (2019), Licata and Harper (2012) in their respective settings.

The rules for substitution are given in Figs. 4, 5, 6, 7, 8, and 9. We distinguish them based on whether we need the global cleaving or the local opcleaving to interpret them. There are several important observations to be made about these rules. First, in line with our truly bicategorical approach, we do not assume the comprehension bicategory is split. In particular, no equality between $T[\mathrm{id}\,]$ and T is postulated. Instead, there is an equivalence between them (see sub-id and sub-comp), and terms of these types are transported along the equivalence.

The rule map expresses that each type T behaves "functorially": for each $\Delta \vdash s : \Gamma$ (i.e., object in "hom (Δ, Γ)") we get a type $T[s]$ (i.e., object in "the category of types in context Δ") by sub-ty and for each $r : s \rightsquigarrow s'$ (i.e., morphism in 'hom (Δ, Γ)') we get a term $\Delta \mid T[s] \vdash \mathsf{map}\ T\ \theta : T[s']$ (i.e., morphism in "the category of types in context Δ") by map. The rules map-id and map-comp

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

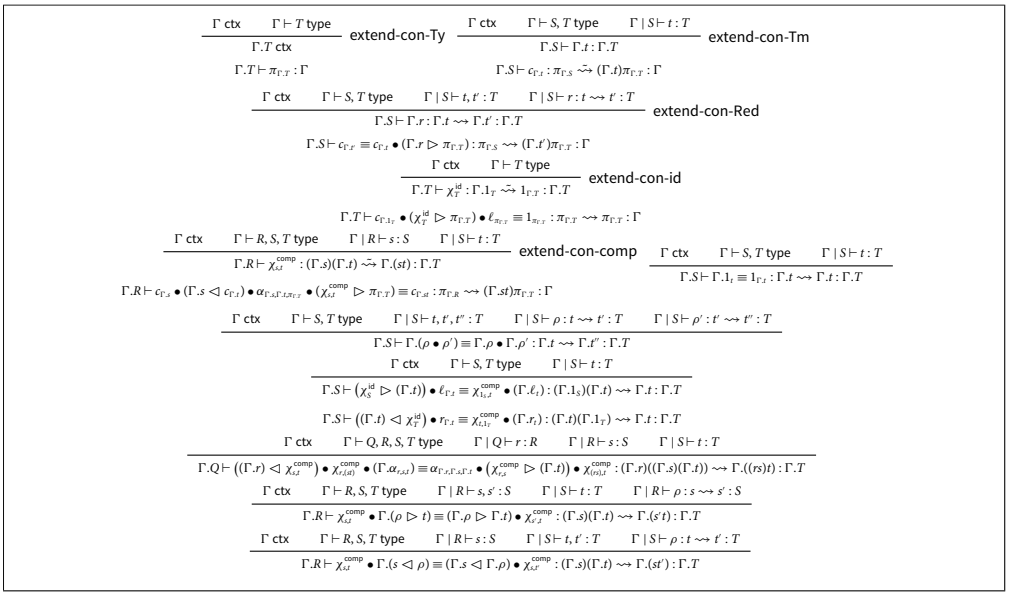


Figure 3. Rules for comprehension.

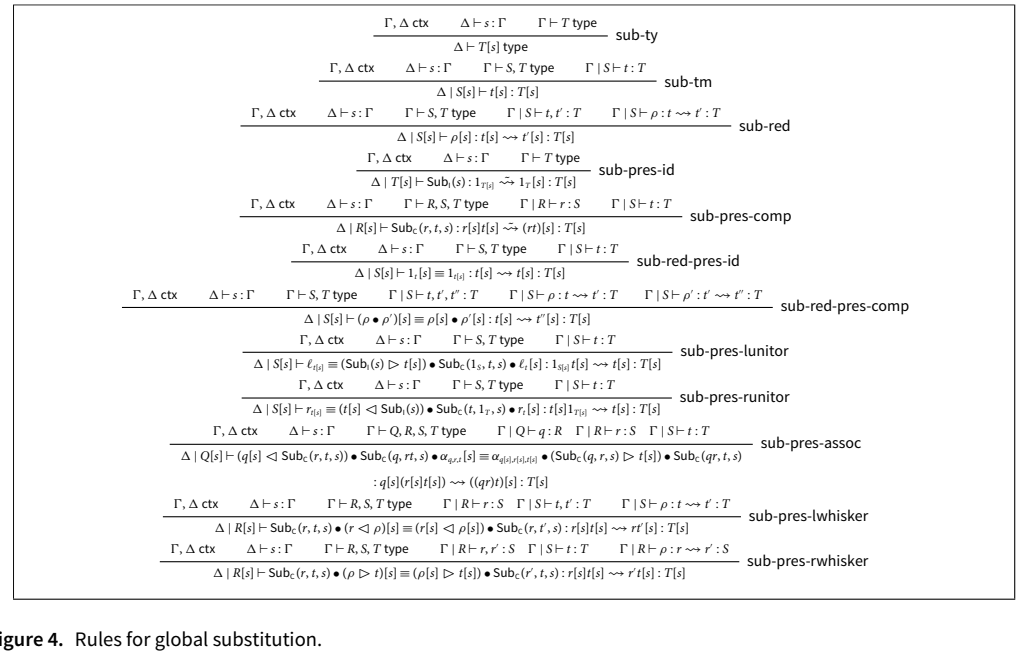


Figure 4. Rules for global substitution.

ensure that $T[-]$ preserves identity and composition. With rew-tm we can then understand terms to be "natural transformations."

Remark 62. For contravariant comprehension bicategories (Remark 41), the rule map would be in the opposite direction, while for isovariant comprehension bicategories, this rule would be restricted to isomorphisms in the base.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

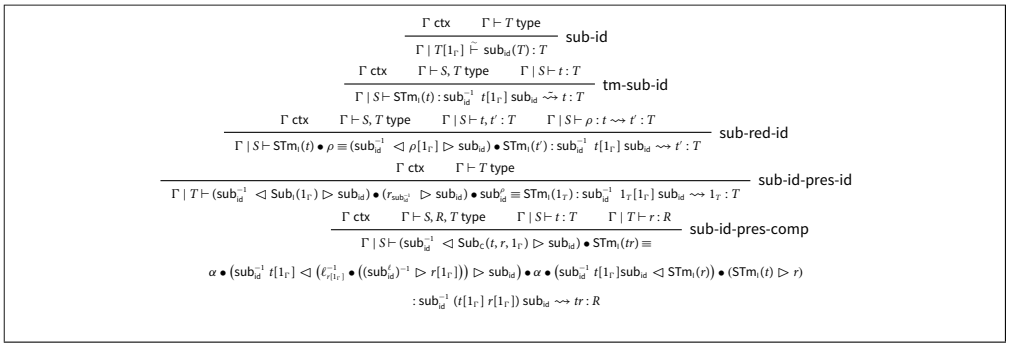


Figure 5. Rules for global substitution (preservation of identity).

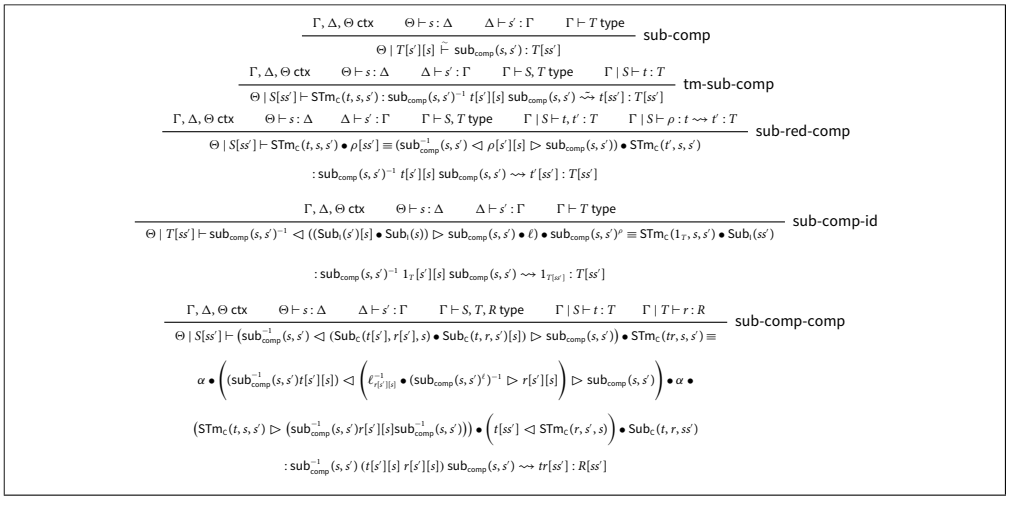


Figure 6. Rules for global substitution (preservation of composition).

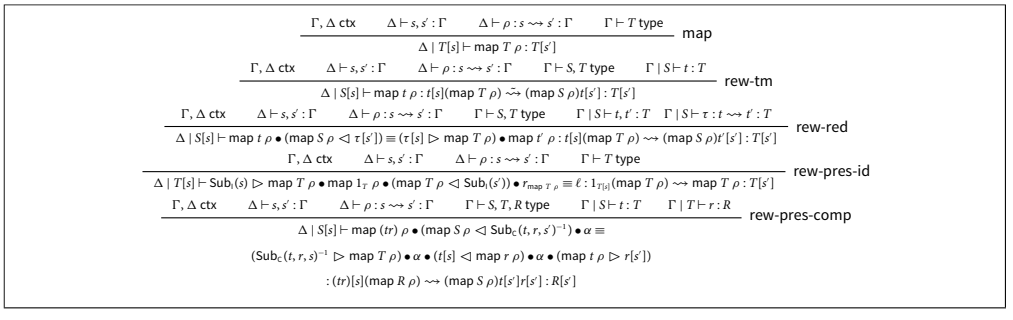


Figure 7. Rules for local substitution.

Remark 63. Using the Grothendieck construction, we can view the rules from the perspective of fiber bicategories. If $P : \mathsf{E} \to \mathsf{B}$ has a global cleaving and a local opcleaving, and furthermore opcartesian 2-cells are preserved under whiskering, then we obtain a trifunctor $\mathsf{B}^{\mathsf{op}} \to \mathsf{Bicat}$, which sends every $x : \mathsf{B}$ to the fiber of x along P. The rules given in Fig. 4 say that any 1-cell

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

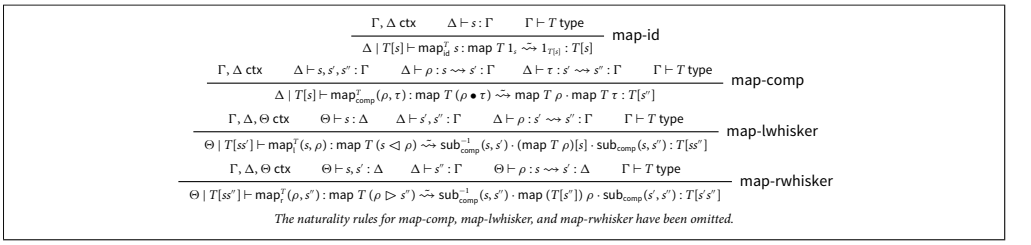


Figure 8. Some rules for local substitution (preservation).

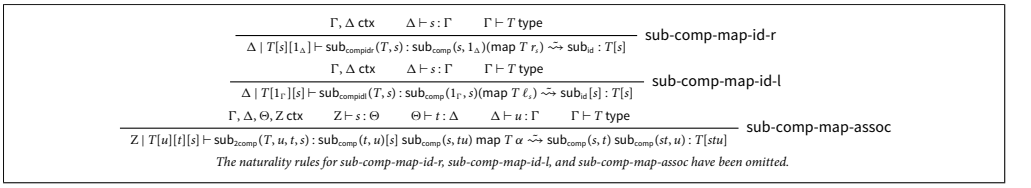


Figure 9. Some of the rules for coherence of substitution.

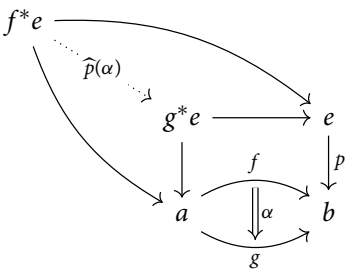
in the base gives rise to a pseudofunctor. For example, the rules sub-ty, sub-tm, and sub-red represent the actions on objects, 1-cells, and 2-cells, respectively, while sub-pres-id and sub-pres–comp represent the invertible 2-cells that witness the preservation of the identity and composition of 1-cells. The other rules in that table are the coherence laws of pseudofunctors: for example, sub-red-pres-id is the preservation of identity 2-cells.

Figs. 5 and 6 give pseudonatural equivalences expressing that the assignment of the previously mentioned pseudofunctor is actually pseudofunctorial: the identity and composition are preserved up to pseudonatural equivalence. For example, the rules sub-id and tm-sub--id in Fig. 5 represent the action on objects and the naturality squares. On the other hand, sub-red-id, sub-id-pres-id, and sub-id-pres-comp are the usual coherencies of pseudonatural transformations.

The rules of Fig. 7 state that we obtain a pseudonatural transformation from a 2-cell in the base. The action on objects and 1-cells is given by map and rew-tm, respectively, while rew-red, rew-pres-id, and rew-pres-comp describe the usual coherencies of pseudonatural transformations. The remaining rules can be found in Fig. 8. In this figure, four invertible modifications are described: map-id and map-comp describe the preservation of identity and composition, respectively, while map-lwhisker and map-rwhisker describe the preservation of both left and right whiskering. Note that we left out the naturality conditions of these modifications. In addition, two additional coherencies are required which express that all this data together forms a trifunctor. We do not write them down, but instead, we refer the reader to Definition 3.3.1 of Gurski (2006).

Remark 64. A similar rule to map of Fig. 7 appears in work of Johnstone (1993). There, it is shown that a 1-cell p in a 2-category is an internal fibration in a sense similar to our Definition 46 if and only if a specific functor given by precomposition with p has a semi-oplax right adjoint satisfying some properties, assuming that pullbacks along p exist. In Lemma 2.5 of Johnstone (1993), another equivalent characterization of p being a fibration is given assuming that pullbacks of p exist. This characterization uses several operations and compatibility requirements. One of the operations is described by the following diagram:

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press



We assume that pullbacks along p are given; in particular, we have f^*e and g^*e. The characterization of p being a fibration entails, in particular, that for every 2-cell $\alpha : f \Rightarrow g$ there is a 1-cell $\widehat{p}(\alpha) : f^*e \to g^*e$. This takes place in a 2-category, though Johnstone claims that such a characterization would also hold in a bicategory. Modulo this difference, this operation is expressed by rule map of Fig.7.

9. Soundness: Interpretation in Comprehension Bicategories

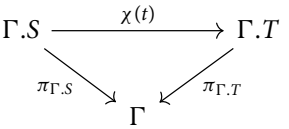
In this section, we give an interpretation of BTT in any weak comprehension bicategory. To this end, we fix a comprehension bicategory $\mathsf{D} \xrightarrow{\chi} \mathsf{B}^{\downarrow}$ over B. We interpret the judgments as follows.

- Γ ctx is interpreted as an object $[\![\Gamma]\!]$ of B.
- $\Delta \vdash s : \Gamma$ is interpreted as a 1-cell $[\![s]\!] : [\![\Delta]\!] \to [\![\Gamma]\!]$ in B.
- $\Delta \vdash r : s \rightsquigarrow t : \Gamma$ is interpreted as a 2-cell $[\![r]\!] : [\![s]\!] \Rightarrow [\![t]\!]$ in B.
- $\Delta \vdash r \equiv r' : s \rightsquigarrow t : \Gamma$ is interpreted as an equality $[\![r]\!] = [\![r']\!]$.
- $\Gamma \vdash T$ type is interpreted as an object $[\![T]\!]$ in D over $[\![\Gamma]\!]$.
- $\Gamma \mid S \vdash t : T$ is interpreted as a 1-cell $[\![t]\!] : [\![S]\!] \to [\![T]\!]$ over the identity on $[\![\Gamma]\!]$.
- $\Gamma \mid S \vdash r : t \rightsquigarrow t' : T$ is interpreted as a 2-cell $[\![r]\!] : [\![t]\!] \Rightarrow [\![t']\!]$ over the identity 2-cell.
- $\Gamma \mid S \vdash r \equiv r' : t \rightsquigarrow t' : T$ is interpreted as an equality $[\![r]\!] = [\![r']\!]$.

Regarding the "bicategorical" rules of Figs. 1 and 2, each rule is analogous to one of the operations or laws of a bicategory – see, for instance, Definition 3.1 of Ahrens et al. (2021). This also indicates how it is interpreted.

9.1 Comprehension

In this section, we interpret the rules related to comprehension of Fig. 3. Suppose that we have a context $\Gamma : \mathsf{B}$ and a type T over Γ. Its image $\chi(T)$ in $\mathsf{B}^{\downarrow}$ gives rise to an object $\Gamma.T : \mathsf{B}$ and a 1-cell $\pi_{\Gamma.T} : \Gamma.T \to \Gamma$ which interprets extend-con-Ty. For a 1-cell t from S to T over the identity, the resulting 1-cell $\chi(t)$ is part of a triangle



which commutes up to invertible 2-cell. This yields the interpretation of the rules in extend–con-Tm. Furthermore, a reduction $r : t \rightsquigarrow t'$ is mapped by χ to a 2-cell from $\chi(t)$ to $\chi(t')$ in $\mathsf{B}^{\downarrow}$, and this is how we interpret extend-con-Red. The rules extend-con-id and extend-con-comp are interpreted by the identitor and compositor of χ, respectively, while the remaining rules in Fig. 3 are satisfied because they translate to the laws that express that this data forms a pseudofunctor.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

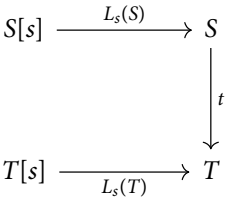
9.2 Global substitution

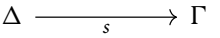
Next, we interpret the rules for global substitution of Fig. 4 (in Sections 9.2.1, 9.2.2, 9.2.3, and 9.2.4) and the rules regarding preservation of identity under global substitution of Fig. 5 (in Sections 9.2.6, 9.2.5, and 9.2.7). The interpretation of the rules regarding preservation of composition of Fig. 6 is analogous to that of Fig. 5 and is not spelled out.

The interpretation of the rule sub-ty is given directly by the global cleaving. The interpretation of the other rules require more explanation.

9.2.1 Interpretation of sub-tm

To interpret sub-tm, we assume that we have a substitution $s : \Delta \to \Gamma$ between contexts Δ and Γ. We also assume that we have types S, T over Γ and a morphism $t : S \to T$ over 1_Γ. This situation is encapsulated in the following diagram, using the notation introduced in Definition 27.

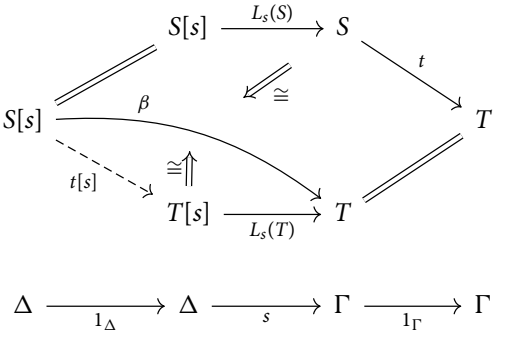




Our goal is to construct a 1-cell $t[s] : S[s] \to T[s]$ that lies over the identity on Δ. Since the morphism $L_s(T) : T[s] \to T$ is cartesian, it suffices to construct a 1-cell, say, $\beta : S[s] \to T$ that lies over $1_\Delta \cdot s$. We then obtain the desired 1-cell as the factorization $\mathsf{F}_1(L_s(T), \beta)$ of that 1-cell via $L_s(T)$. (Recall that $\mathsf{F}_1(\_, \_)$ was defined in Definition 21.) We have an invertible 2-cell $r_s \bullet {\ell_s}^{-1} : s \cdot 1_\Gamma \Rightarrow 1_\Delta \cdot s$, and we set $\beta := (r_s \bullet {\ell_s}^{-1})_!(L_s(S) \cdot t)$. Define

$$t[s] := \mathsf{F}_1(L_s(T), \beta).$$

Overall, we can summarize the construction with the following diagram.

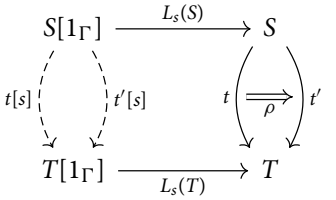


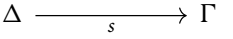
9.2.2 Interpretation of sub-red

To interpret sub-red, we assume that we have a substitution $s : \Delta \to \Gamma$ and types S and T in context Γ. We also assume that we have 1-cells $t, t' : S \to T$ over 1_Γ and a 2-cell $\rho : t \Rightarrow t'$. Consider

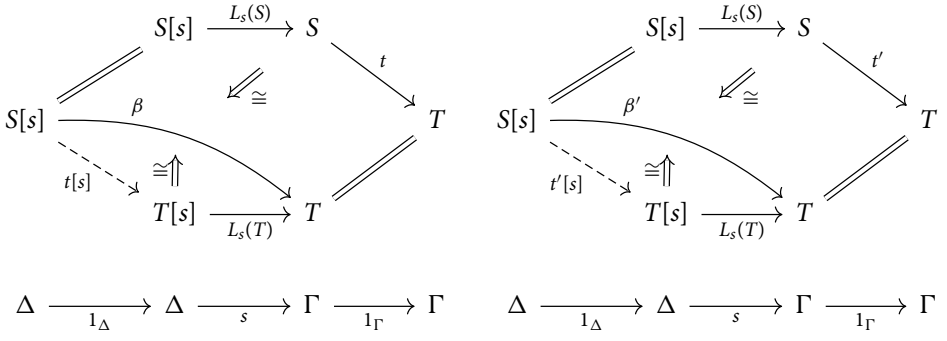
https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

the following diagram.





Our goal is to construct a 2-cell from $t[s]$ to $t[s']$ over the identity on Δ. The source and target, respectively, were constructed by factoring a 1-cell via a cartesian 1-cell, as follows.

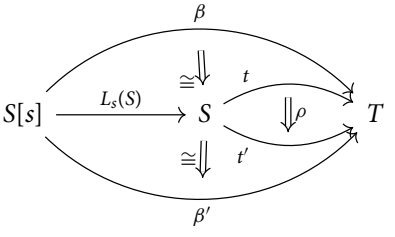


To construct the desired 2-cell $\rho[s] : t[s] \Rightarrow t[s']$, we use Item 2 of Definition 21. More specifically, it suffices to construct

- a 2-cell δ from 1_Δ to 1_Δ, and
- a 2-cell $\overline{\sigma}$ from β to β'.

Once we have defined those, we define $\rho[s] := \mathsf{E}(\delta, \overline{\sigma})$.

For the first of the two, we take the identity 2-cell 1_{1_Δ}. For the second, we take the composition of 2-cells shown in the following diagram.



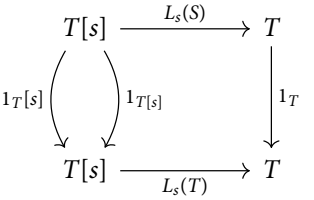
The interpretation of the rules sub-red-id and sub-red-comp follows from the uniqueness of factorization 2-cells.

9.2.3 Interpretation of sub-pres-id

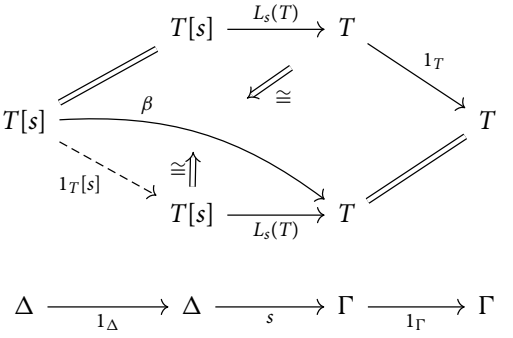
The rules sub-pres-id and sub-pres-comp are interpreted similarly, and we show the interpretation of sub-pres-id. Suppose that we have a substitution $s : \Delta \to \Gamma$ and a type T in context Γ.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

We consider two morphisms: the identity on $T[s]$ and $1_T[s]$. These are illustrated in the following diagram.



Our goal is to construct an invertible 2-cell from $1_{T[s]}$ to $1_T[s]$. To do so, we first recall the construction of $1_T[s]$.



To construct the desired invertible 2-cell, we again use Item 2 of Definition 21. As such, we need to construct the following:

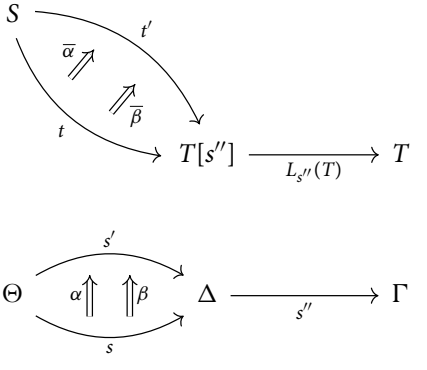
- An invertible 2-cell δ from 1_Δ to 1_Δ; and
- An invertible 2-cell $\overline{\sigma}$ from $1_T[s] \cdot L_s(T)$ to $1_{T[s]} \cdot L_s(T)$.

With those in place, we define $\mathsf{Sub_I}(s)^{-1} := \mathsf{E}(\delta, \overline{\sigma})$. For δ, we take the identity, while for $\overline{\sigma}$, we take the following composition.

$$1_{T[s]} \cdot L_s(T) \cong L_s(T) \cdot 1_T \cong \beta \cong 1_T[s] \cdot L_s(T)$$

9.2.4 Interpretation of sub-pres-lunitor

The remaining laws stating equalities of 2-cells, such as sub-pres-comp and sub-id-pres-id, are all proven in a similar way. Our goal is to prove an equality $\overline{\alpha} = \overline{\beta}$ with 1-cells and 2-cells as in the diagram below:



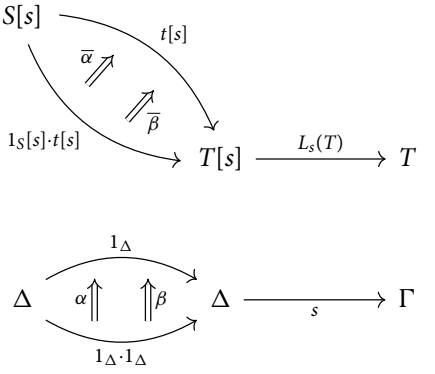
https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

Here t and t' live over s and s', respectively, while $\overline{\alpha}$ and $\overline{\beta}$ live over α and β, respectively. To prove the equality, we take the following steps.

(1) We construct an invertible 2-cell θ from $t \cdot L_{s''}(T)$ to $t' \cdot L_{s''}(T)$.
(2) We prove that $\alpha = \beta$.
(3) We prove that $\overline{\alpha} \rhd L_{s''}(T) = \theta = \overline{\beta} \rhd L_{s''}(T)$.

From the first item, we get that both t and t' are cartesian factorizations of the same $S \to T$. From the second and third item, we get that both $\overline{\alpha}$ and $\overline{\beta}$ are factorization 2-cells from t to t', and since such 2-cells are unique, we get $\overline{\alpha} = \overline{\beta}$.

We demonstrate this for sub-pres-lunitor. Suppose we have a substitution $s : \Delta \to \Gamma$, types S and T in context Γ, and a morphism $t : S \to T$ over 1_Γ. We are in the following situation.

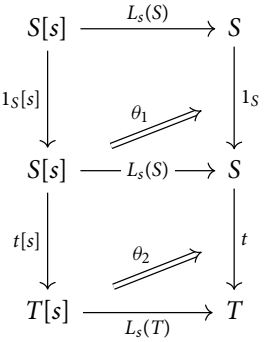


Here, we define $\alpha := (1_{1_\Delta} \rhd 1_\Delta) \bullet \ell_{1_\Delta}$ and $\beta := 1_{1_\Delta \cdot 1_\Delta} \bullet \ell_{1_\Delta}$. We also define $\overline{\alpha}$ and $\overline{\beta}$ as follows.

$$\overline{\alpha} := (\mathsf{Sub_I}(s)^{-1} \rhd t[s]) \bullet \ell_{t[s]} \qquad \overline{\beta} := \mathsf{Sub_C}(1_S, t, s) \bullet \ell_t[s]$$

Then by construction, both α and β are equal to ℓ_{1_Δ}, and hence, $\alpha = \beta$.

Next we construct an invertible 2-cell from $1_S[s] \cdot t[s] \cdot L_s(T)$ to $t[s] \cdot L_s(T)$. By construction of $t[s]$, we have an invertible 2-cell $\theta_1 : t[s] \cdot L_s(T) \cong \cdot L_s(S) \cdot t$. In addition, we have an invertible 2-cell $\theta_2 : 1_S[s] \cdot L_s(S) \cong L_s(S) \cdot 1_S$. This is illustrated in the following diagram.



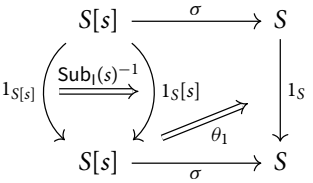
Now we define an invertible 2-cell $\theta_3 : 1_S[s] \cdot t[s] \cdot L_s(T) \cong L_s(S) \cdot t$ as follows.

$$\theta_3 := \alpha^{-1} \bullet (1_S[s] \lhd \theta_2) \bullet \alpha \bullet (\theta_1 \rhd t) \bullet (r_\sigma \rhd t)$$

The desired invertible 2-cell $\theta : 1_S[s] \cdot t[s] \cdot L_s(T) \cong t[s] \cdot L_s(T)$ is $\theta_3 \bullet \theta_2^{-1}$.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

For the last step, we only prove that $\overline{\alpha} \rhd L_s(T) = \theta$. We first recall how we constructed $\mathsf{Sub}_{\mathsf{I}}(s)$. For brevity, we write $\sigma := L_s(S)$ and $\tau := L_s(T)$.



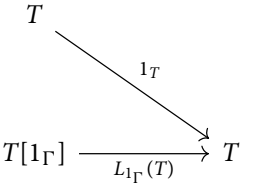
By construction, we have that $(\mathsf{Sub}_{\mathsf{I}}(s)^{-1} \rhd \sigma) \bullet \ell_\sigma \bullet r_\sigma^{-1} = \theta_1$. In particular, we have $\mathsf{Sub}_{\mathsf{I}}(s)^{-1} \rhd \sigma = \theta_1 \bullet r_\sigma \bullet \ell_\sigma^{-1}$.

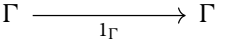
To show that $((\mathsf{Sub}_{\mathsf{I}}(s)^{-1} \rhd t[s]) \bullet \ell_{t[s]}) \rhd \tau = \theta_3 \bullet \theta_2^{-1}$, it suffices to show that $((\mathsf{Sub}_{\mathsf{I}}(s)^{-1} \rhd t[s]) \bullet \ell_{t[s]}) \rhd \tau \bullet \theta_2 = \theta_3$. This follows from the following chain of equalities.

$$
\begin{aligned}
&\alpha^{-1} \bullet (1_S[s] \lhd \theta_2^{-1}) \bullet \alpha \bullet ((\mathsf{Sub}_{\mathsf{I}}(s)^{-1} \rhd t[s]) \bullet \ell_{t[s]}) \rhd \tau \bullet \theta_2 \bullet (r_\sigma \rhd t)^{-1} \\
&= \alpha^{-1} \bullet (1_S[s] \lhd \theta_2^{-1}) \bullet \alpha \bullet (\mathsf{Sub}_{\mathsf{I}}(s)^{-1} \rhd t[s]) \rhd \tau \bullet (\ell_{t[s]} \rhd \tau) \bullet \theta_2 \bullet (r_\sigma \rhd t)^{-1} \\
&= \alpha^{-1} \bullet (1_S[s] \lhd \theta_2^{-1}) \bullet \mathsf{Sub}_{\mathsf{I}}(s)^{-1} \rhd (t[s] \cdot \tau) \bullet \alpha \bullet (\ell_{t[s]} \rhd \tau) \bullet \theta_2 \bullet (r_\sigma \rhd t)^{-1} \\
&= \alpha^{-1} \bullet \mathsf{Sub}_{\mathsf{I}}(s)^{-1} \rhd (\sigma \cdot t) \bullet (1_{S[s]} \lhd \theta_2^{-1}) \bullet \alpha \bullet (\ell_{t[s]} \rhd \tau) \bullet \theta_2 \bullet (r_\sigma \rhd t)^{-1} \\
&= \mathsf{Sub}_{\mathsf{I}}(s)^{-1} \rhd \sigma \rhd t \bullet \alpha^{-1} \bullet (1_{S[s]} \lhd \theta_2^{-1}) \bullet \alpha \bullet (\ell_{t[s]} \rhd \tau) \bullet \theta_2 \bullet (r_\sigma \rhd t)^{-1} \\
&= (\theta_1 \bullet r_\sigma \bullet \ell_\sigma^{-1}) \rhd t \bullet \alpha^{-1} \bullet (1_{S[s]} \lhd \theta_2^{-1}) \bullet \alpha \bullet (\ell_{t[s]} \rhd \tau) \bullet \theta_2 \bullet (r_\sigma \rhd t)^{-1} \\
&= \theta_1 \rhd t \bullet (r_\sigma \bullet \ell_\sigma^{-1}) \rhd t \bullet \alpha^{-1} \bullet (1_{S[s]} \lhd \theta_2^{-1}) \bullet \alpha \bullet (\ell_{t[s]} \rhd \tau) \bullet \theta_2 \bullet (r_\sigma \rhd t)^{-1} \\
&= \theta_1 \rhd t \bullet (r_\sigma \bullet \ell_\sigma^{-1}) \rhd t \bullet \alpha^{-1} \bullet (1_{S[s]} \lhd \theta_2^{-1}) \bullet \ell_{t[s]\tau} \bullet \theta_2 \bullet (r_\sigma \rhd t)^{-1} \\
&= \theta_1 \rhd t \bullet (r_\sigma \bullet \ell_\sigma^{-1}) \rhd t \bullet \alpha^{-1} \bullet \ell_{\sigma \cdot t} \bullet \theta_2^{-1} \bullet \theta_2 \bullet (r_\sigma \rhd t)^{-1} \\
&= \theta_1 \rhd t \bullet (r_\sigma \bullet \ell_\sigma^{-1}) \rhd t \bullet \alpha^{-1} \bullet \ell_{\sigma \cdot t} \bullet (r_\sigma \rhd t)^{-1} \\
&= \theta_1 \rhd t \bullet (r_\sigma \bullet \ell_\sigma^{-1}) \rhd t \bullet \ell_\sigma \rhd t \bullet (r_\sigma \rhd t)^{-1} \\
&= \theta_1 \rhd t
\end{aligned}
$$

9.2.5 Interpretation of sub-id

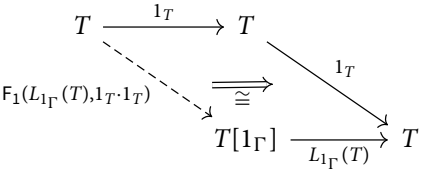
Next we interpret sub-id. Suppose we have a context Γ and a type T in Γ. Note that we have a diagram as follows.

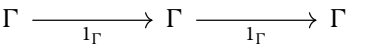




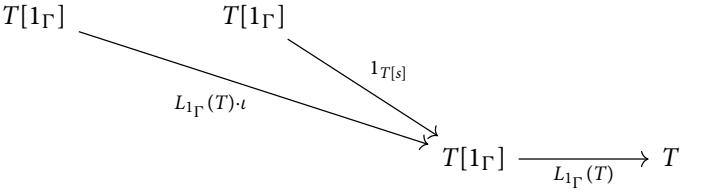
https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

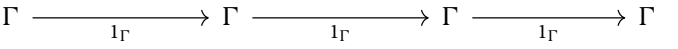
We interpret $\mathsf{sub}_{\mathsf{id}}(T)$[5] by $L_{1_\Gamma}(T)$, so it suffices to show that $L_{1_\Gamma}(T)$ is an equivalence. To construct the inverse, we use the following diagram.





We set $\iota := \mathsf{F}_1(L_{1_\Gamma}(T), 1_T \cdot 1_T)$, and it remains to show that we have an invertible 2-cell from $L_{1_\Gamma}(T) \cdot \iota$ to 1_T. To do so, we use Item 2 of Definition 21, and we consider the following diagram.





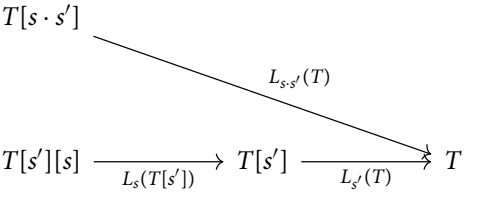
We need to construct an invertible 2-cell δ from $1_\Gamma \cdot 1_\Gamma$ to 1_Γ and an invertible 2-cell $\overline{\sigma}$ from $L_{1_\Gamma}(T) \cdot \iota \cdot L_{1_\Gamma}(T)$ to $1_{T[s]} \cdot L_{1_\Gamma}(T)$. For δ, we take λ. For $\overline{\sigma}$, we use that $\iota \cdot L_{1_\Gamma}(T) \cong 1_T \cdot 1_T$, and thus we have the following composition of invertible 2-cells.

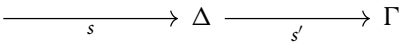
$$\begin{aligned} L_{1_\Gamma}(T) \cdot \iota \cdot L_{1_\Gamma}(T) &\cong L_{1_\Gamma}(T) \cdot 1_T \cdot 1_T \\ &\cong L_{1_\Gamma}(T) \\ &\cong 1_{T[s]} \cdot L_{1_\Gamma}(T) \end{aligned}$$

The desired invertible 2-cell is thus defined by $\mathsf{E}(\delta, \overline{\sigma})$.

9.2.6 *Interpretation of sub-comp*

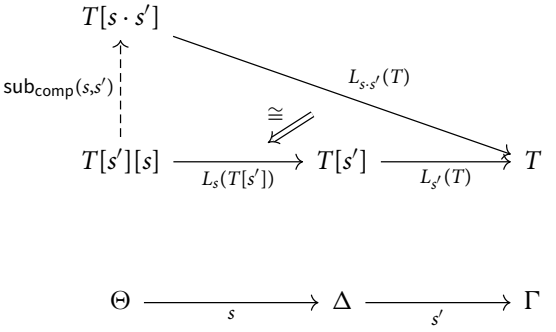
To interpret sub-comp, we suppose that we have substitutions $s' : \Theta \to \Delta$ and $s : \Delta \to \Gamma$, and a type T in Γ. Hence, we have the following diagram.





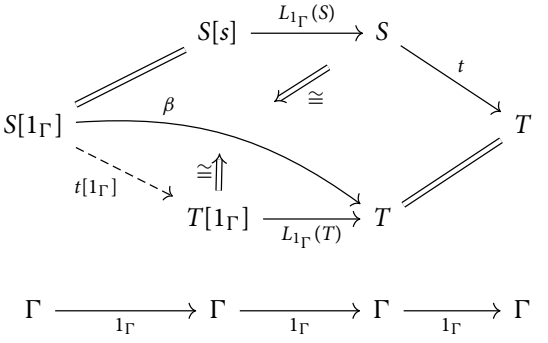
https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

Note that both $L_{s\cdot s'}(T)$ and $L_s(T[s']) \cdot L_{s'}(T)$ are cartesian 1-cells over $s \cdot s'$, since cartesian 1-cells are closed under composition by Proposition 22. For that reason, we get an adjoint equivalence $\mathsf{sub}_{\mathsf{comp}}(s, s') : T[s'][s] \to T[s \cdot s']$ making the diagram below commute up to invertible 2-cell by Construction 24.



9.2.7 Interpretation of tm-sub-id

Next we interpret tm-sub-id, and for that, we suppose that we have a context Γ, types S and T over Γ, and a 1-cell $t : S \to T$ over 1_Γ. We need to construct an invertible 2-cell from $\mathsf{sub}_{\mathsf{id}}^{-1} \cdot t[1_\Gamma] \cdot \mathsf{sub}_{\mathsf{id}}$ to t, and for that, it suffices to construct an invertible 2-cell from $t[1_\Gamma] \cdot \mathsf{sub}_{\mathsf{id}}$ to $\mathsf{sub}_{\mathsf{id}} \cdot t$. Recall that we defined $\mathsf{sub}_{\mathsf{id}}$ to be $L_{1_\Gamma}(T)$ and recall that we defined $t[1_\Gamma]$ as follows.



By composing the two invertible 2-cells depicted in this diagram, we get the desired invertible 2-cell. The rule tm-sub-comp is interpreted analogously.

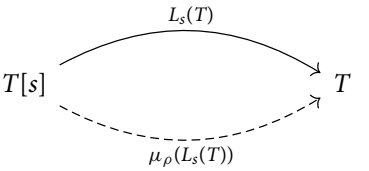
9.3 Local substitution

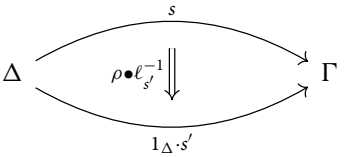
Next we interpret the rules in Fig. 7 (in Sections 9.3.1 and 9.3.2), Fig. 8 (in Sections 9.3.3 and 9.3.4), and Fig. 9 (in Section 9.3.5). For the interpretation, we use the local opcleaving.

9.3.1 Interpretation of map

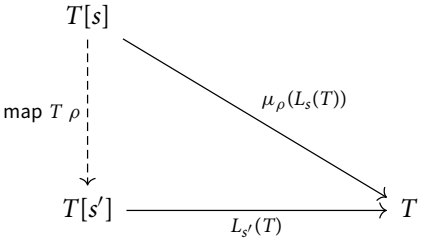
We start with map. Suppose we have two substitutions $s, s' : \Delta \to \Gamma$ and a 2-cell $\rho : s \Rightarrow s'$. In addition, we assume that we have a type T in context Γ. Our goal is to construct a 1-cell $\mathsf{map}\ T\ \rho : T[s] \to T[s']$. From the local opcleaving, we obtain $\mu_\rho(L_s(T))$ over $1_\Delta \cdot s'$ as the pushforward of $L_s(T)$ along $\rho \bullet \ell_{s'}^{-1}$.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press





Since $L_{s'}(T)$ is cartesian, we can factor $\mu_\rho(L_s(T))$ through it. From that, we obtain $\mathsf{map}\ T\ \rho$.



9.3.2 Interpretation of rew-tm

Next, we give the interpretation for the rule rew-tm. Suppose we have two 1-cells $s, s' : \Delta \to \Gamma$ and a 2-cell $\rho : s \to s'$. In addition, we assume that we have types S, T in context Γ and a 1-cell $\Gamma \mid S \vdash t : T$. Our goal is to construct an invertible 2-cell between $t[s] \bullet (\mathsf{map}\ T\ \rho)$ and $(\mathsf{map}\ S\ \rho) \bullet t[s']$. For that it suffices to construct:

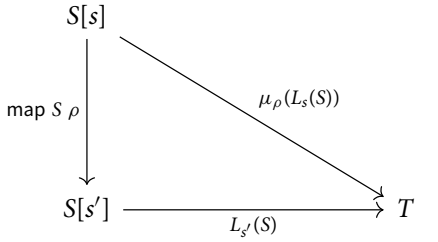
(1) An invertible 2-cell δ from $1_\Delta \cdot 1_\Delta$ to $1_\Delta \cdot 1_\Delta$; and
(2) An invertible 2-cell $\overline{\sigma}$ from $t[s] \bullet (\mathsf{map}\ T\ \rho) \bullet L_{s'}(T)$ to $(\mathsf{map}\ S\ \rho) \bullet t[s'] \bullet L_{s'}(T)$.

Then we define $t[\rho]$ to be $\mathsf{E}(\delta, \overline{\sigma})$. For δ we take $1_{1_\Delta \cdot 1_\Delta}$.

To construct $\overline{\sigma}$, we first note that by construction of $t[s']$, we have the following invertible 2-cell.

$$(\mathsf{map}\ S\ \rho) \bullet t[s'] \bullet L_{s'}(T) \cong (\mathsf{map}\ S\ \rho) \bullet L_{s'}(S) \bullet t$$

Note that by construction of $\mathsf{map}\ S\ \rho$, the following triangle commutes up to invertible 2-cell.



Hence, we get

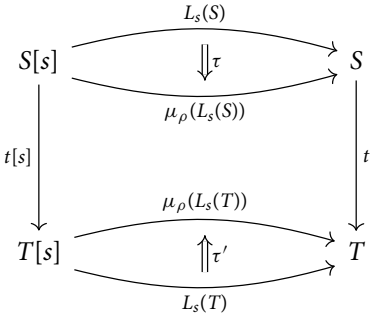
$$\gamma_1 : (\mathsf{map}\ S\ \rho) \bullet L_{s'}(S) \bullet t \cong \mu_\rho(L_s(S)) \bullet t.$$

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

Similarly, we have an invertible 2-cell $\mathsf{map}\ T\ \rho \cdot L_{s'}(S) \cong \mu_\rho(L_s(T))$, and thus we obtain

$$\gamma_2 : t[s] \bullet (\mathsf{map}\ T\ \rho) \bullet L_{s'}(T) \cong t[s] \bullet \mu_\rho(L_s(T)).$$

It thus suffices to construct an invertible 2-cell from $t[s] \bullet \mu_\rho(L_s(T))$ to $\mu_\rho(L_s(S)) \bullet t$. We can depict the situation with the following square.



In this diagram, τ and τ' are the opcartesian 2-cells coming from the opcartesian lift. Note that by construction of $t[s]$, the outer square of this diagram commutes. More precisely, we have an invertible 2-cell $\theta : t[s] \cdot L_s(T) \cong L_s(S) \cdot t$.

To construct an invertible 2-cell from $\mu_\rho(L_s(S)) \cdot t$ to $t[s] \cdot \mu_\rho(L_s(T))$, we are going to construct two opcartesian 2-cells. First, note that we have a 2-cell

$$\tau \rhd t : L_s(S) \cdot t \Rightarrow \mu_\rho(L_s(S)) \cdot t'$$

that lies over $\beta_1 := (\rho \bullet l^{-1}) \rhd 1_\Gamma : s \cdot 1_\Gamma \Rightarrow (1_\Delta \cdot s') \cdot 1_\Gamma$. This 2-cell is opcartesian, because opcartesian 2-cells are closed under right whiskering.

Note that we also have the 2-cell

$$\theta^{-1} \bullet t[s] \lhd \tau : L_s(S) \cdot t \Rightarrow t[s] \cdot \mu_\rho(L_s(T))$$

which lies over $\beta_2 := r \bullet \ell^{-1} \bullet 1_\Gamma \lhd (\rho \bullet \ell^{-1}) : s \cdot 1_\Gamma \Rightarrow 1_\Delta \cdot (1_\Delta \cdot s')$. It is opcartesian, because left whiskering also preserves opcartesian 2-cells.

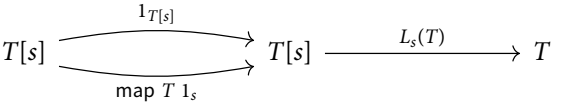
We also have the following invertible 2-cell

$$\beta_3 := r \bullet \ell^{-1} : (1_\Delta \cdot s') \cdot 1_\Gamma \cong 1_\Delta \cdot (1_\Delta \cdot s'),$$

and note that $\beta_1 \bullet \beta_3 = \beta_2$ by naturality of the unitors. From all of this we get the desired invertible 2-cell $\mu_\rho(L_s(S)) \cdot t \cong t[s] \cdot \mu_\rho(L_s(T))$.

9.3.3 Interpretation of map-id

To interpret map-id, we assume that we have a 1-cell $s : \Delta \to \Gamma$ and a type T in context Γ. Our goal is to construct an invertible 2-cell between the following compositions.



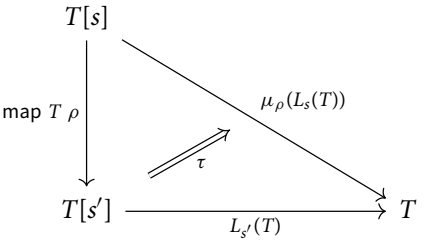
To do so, we need to construct:

(1) An invertible 2-cell δ from 1_Δ to 1_Δ; and
(2) An invertible 2-cell $\overline{\sigma}$ from $\mathsf{map}\ T\ 1_s \cdot L_s(T)$ to $L_s(T)$.

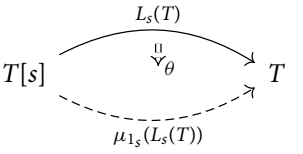
The desired 2-cell is then defined to be $\mathsf{E}(\delta, \overline{\sigma})$. For δ we take 1_{1_Δ}.

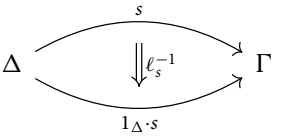
https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

To construct $\overline{\sigma}$, we first note that by construction of $\mathsf{map}\ T\ \rho$, the following triangle commutes up to an invertible 2-cell $\tau : \mathsf{map}\ T\ \rho \cdot L_{s'}(T) \cong \mu_\rho(L_s(T))$.



Next we recall that $\mu_\rho(L_s(T))$ was constructed as the following pushforward.





Since ℓ_s^{-1} is invertible, the opcartesian lift θ is invertible as well. Hence, we define $\overline{\sigma} := \tau \bullet \theta^{-1}$. We interpret map-comp similarly.

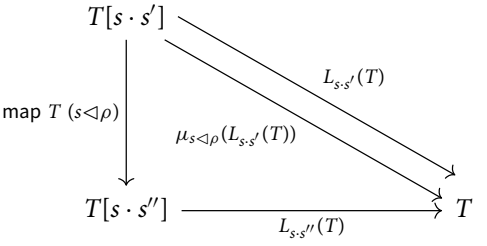
9.3.4 Interpretation of map-lwhisker

To interpret map-lwhisker and map-rwhisker, we use the same idea, and we need to use that we assumed that opcartesian 2-cells are closed under whiskering. We give the precise details for map-lwhisker. Suppose we have a 1-cell $s : \Theta \to \Delta$, a 2-cell $\rho : s' \Rightarrow s''$ where $s', s'' : \Delta \to \Gamma$, and a type T in context Γ. Our goal is to construct an invertible 2-cell from $\mathsf{map}\ T\ (s \lhd \rho)$ to $\mathsf{sub}_{\mathsf{comp}}^{-1}(s, s') \cdot (\mathsf{map}\ T\ \rho)[s] \cdot \mathsf{sub}_{\mathsf{comp}}(s, s'')$. Since $\mathsf{map}\ T\ \rho$ was constructed as a cartesian factorization, it suffices to construct:

(1) An invertible 2-cell δ from 1_Δ to $1_\Delta \cdot 1_\Delta \cdot 1_\Delta$; and
(2) An invertible 2-cell $\overline{\sigma}$ from $\mathsf{map}\ T\ (s \lhd \rho) \cdot L_{s \cdot s''}(T)$ to $\mathsf{sub}_{\mathsf{comp}}^{-1}(s, s') \cdot (\mathsf{map}\ T\ \rho)[s] \cdot \mathsf{sub}_{\mathsf{comp}}(s, s'') \cdot L_{s \cdot s''}(T)$.

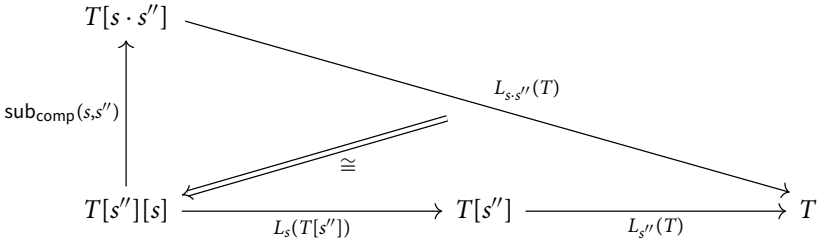
With those in place, the desired invertible 2-cell is defined to be $\mathsf{E}(\delta, \overline{\sigma})$. We define δ to be $\ell_{1_\Delta}^{-1} \cdot \ell_{1_\Delta \cdot 1_\Delta}^{-1}$.

For the construction of $\overline{\sigma}$, let us start by recalling the construction of $\mathsf{map}\ T\ (s \lhd \rho)$.



https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

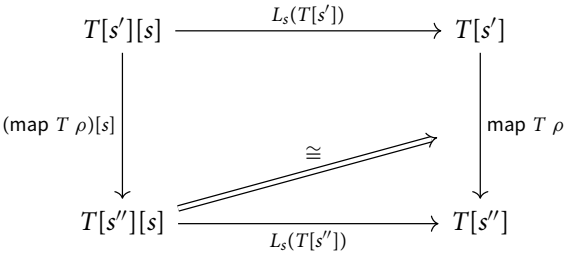
Next we inspect the definition of $\mathsf{sub}_{\mathsf{comp}}(s, s'')$.



From this, we already get the following invertible 2-cell.

$$\begin{aligned}\gamma_1 :& \mathsf{sub}^{-1}_{\mathsf{comp}}(s, s') \cdot (\mathsf{map}\ T\ \rho)[s] \cdot \mathsf{sub}_{\mathsf{comp}}(s, s'') \cdot L_{s \cdot s''}(T) \\ &\cong \\ & \mathsf{sub}^{-1}_{\mathsf{comp}}(s, s') \cdot (\mathsf{map}\ T\ \rho)[s] \cdot L_s(T[s'']) \cdot L_{s''}(T)\end{aligned}$$

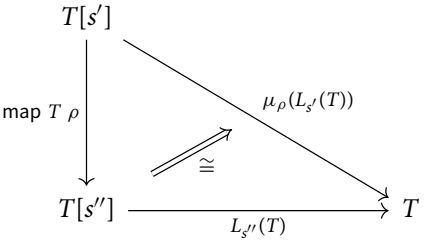
We also inspect the construction of $(\mathsf{map}\ T\ \rho)[s]$, from which we get the following invertible 2-cell.



As such, we get another invertible 2-cell as follows.

$$\begin{aligned}\gamma_2 :& \mathsf{sub}^{-1}_{\mathsf{comp}}(s, s') \cdot (\mathsf{map}\ T\ \rho)[s] \cdot L_s(T[s'']) \cdot L_{s''}(T) \\ &\cong \\ & \mathsf{sub}^{-1}_{\mathsf{comp}}(s, s') \cdot L_s(T[s']) \cdot \mathsf{map}\ T\ \rho \cdot L_{s''}(T).\end{aligned}$$

Let us recall the construction of $\mathsf{map}\ T\ \rho$ as well.



From this, we obtain yet another invertible 2-cell.

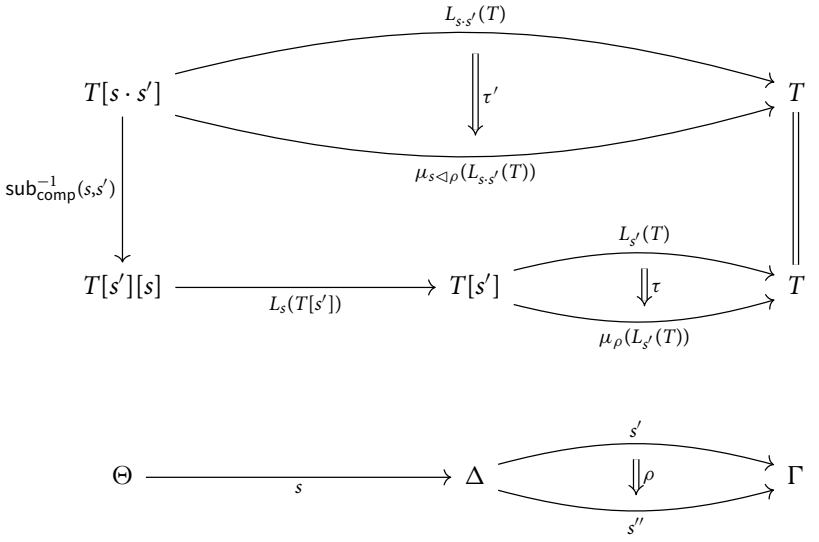
$$\begin{aligned}\gamma_3 :& \mathsf{sub}^{-1}_{\mathsf{comp}}(s, s') \cdot (\mathsf{map}\ T\ \rho)[s] \cdot L_s(T[s'']) \cdot L_{s''}(T) \\ &\cong \\ & \mathsf{sub}^{-1}_{\mathsf{comp}}(s, s') \cdot L_s(T[s']) \cdot \mu_\rho(L_{s''}(T))\end{aligned}$$

Hence, we achieve our goal if we construct an invertible 2-cell of the following form.

$$\gamma_4 : \mathsf{sub}^{-1}_{\mathsf{comp}}(s, s') \cdot L_s(T[s']) \cdot \mu_\rho(L_{s'}(T)) \cong \mu_{s \lhd \rho}(L_{s \cdot s'}(T))$$

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

We can depict this situation as follows.

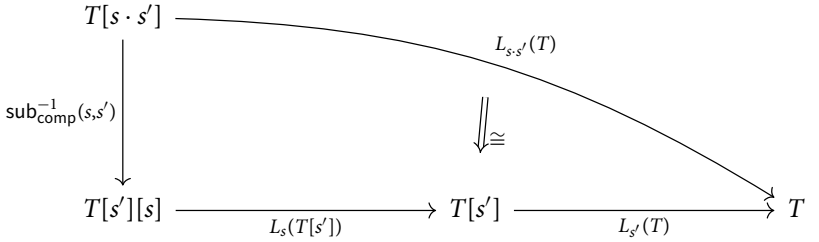


Here τ and τ' are the 2-cells coming from the opcartesian lifts.

Since whiskering preserves opcartesian 2-cells, the following 2-cell is opcartesian.

$$(\mathsf{sub}^{-1}_{\mathsf{comp}}(s,s') \cdot L_s(T[s'])) \lhd \tau \;\; : \;\; \mathsf{sub}^{-1}_{\mathsf{comp}}(s,s') \cdot L_s(T[s']) \cdot L_{s'}(T) \Rightarrow \mathsf{sub}^{-1}_{\mathsf{comp}}(s,s') \cdot L_s(T[s']) \cdot \mu_\rho(L_{s'}(T))$$

Now we unfold the definition of $\mathsf{sub}_{\mathsf{comp}}(s,s')$, and from that we get the following invertible 2-cell.



Since invertible 2-cells are opcartesian and opcartesian 2-cells are closed under composition, we get an opcartesian 2-cell

$$\theta : L_{s\cdot s'}(T) \Rightarrow \mathsf{sub}^{-1}_{\mathsf{comp}}(s,s') \cdot L_s(T[s']) \cdot \mu_\rho(L_{s'}(T)).$$

Since we already had an opcartesian 2-cell $\tau' : L_{s\cdot s'}(T) \Rightarrow \mu_{s \lhd \rho}(L_{s\cdot s'}(T))$, we get an invertible 2-cell between the codomains of θ and τ'. Hence, we obtain an invertible 2-cell

$$\gamma_5 : \mathsf{sub}^{-1}_{\mathsf{comp}}(s,s') \cdot L_s(T[s']) \cdot \mu_\rho(L_{s'}(T)) \cong \mu_{s \lhd \rho}(L_{s\cdot s'}(T)).$$

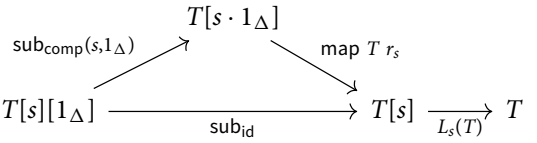
By chaining all the invertible 2-cells, we get the desired 2-cell $\overline{\sigma}$.

9.3.5 Interpretation of sub-comp-map-id-r

Finally, we show how to interpret sub-comp-map-id-r. Suppose that we have objects Γ and Δ, an object T over Γ, and a 1-cell $s : \Delta \to \Gamma$. Our goal is to construct an invertible 2-cell from

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

$\mathsf{sub}_{\mathsf{comp}}(s, 1_\Delta)(\mathsf{map}\ T\ r_s)$ to $\mathsf{sub}_{\mathsf{id}}$. To do so, it suffices to construct an invertible 2-cell from $\mathsf{sub}_{\mathsf{comp}}(s, 1_\Delta) \cdot (\mathsf{map}\ T\ r_s) \cdot L_s(T)$ to $\mathsf{sub}_{\mathsf{id}} \cdot L_s(T)$. Our situation is depicted in the following diagram.



Recall that $\mathsf{sub}_{\mathsf{id}} : T[s][1_\Delta] \to T[s]$ is defined to be $L_{1_\Delta}(T[s])$. By construction, we have an invertible 2-cell

$$\varphi_1 : \mathsf{map}\ T\ r_s \cdot L_s(T) \Rightarrow \mu_{r_s}(L_{s\cdot 1_\Delta}(T)).$$

In addition, since $\mathsf{map}\ T\ r_s$ was constructed as an opcartesian lift, we have the following invertible 2-cell.

$$\varphi_2 : L_{s\cdot 1_\Delta}(T) \Rightarrow \mu_{r_s}(L_{s\cdot 1_\Delta}(T))$$

Note that φ_2 is invertible: this is because it is the opcartesian lift of an invertible 2-cell. By construction of $\mathsf{sub}_{\mathsf{comp}}(s, 1_\Delta)$, we have the following invertible 2-cell.

$$\varphi_3 : \mathsf{sub}_{\mathsf{comp}}(s, 1_\Delta) \cdot L_{s\cdot 1_\Delta}(T) \Rightarrow L_{1_\Delta}(T[1_\Delta]) \cdot L_s(T).$$

The composition $\varphi_3 \bullet \varphi_2 \bullet \varphi_1^{-1}$ yields the desired 2-cell.

We omit the description of the verification of several equalities. All in all, we can state the following theorem.

Theorem (Soundness). *We can interpret* BTT *in any weak comprehension bicategory.*

10. Variations on Syntax

BTT is a very complicated type theory and might not be feasible to implement or use in practice. Its main purpose is to serve as a framework for studying specialized syntax and the corresponding semantics. Based on a user's goal, they might adopt some of the following simplifications.

- **V1:** **Strictness.** The rules in Figs. 1 and 2 are aimed at bicategories. When working with strict 2-categories instead, the unitors and associators would become equalities, and as a result, rules for inverse laws, naturality, and the pentagon and triangle equations are not needed. For this variant, an equality judgment on contexts, types, substitutions, and terms would be added to BTT.
- **V2:** **Splitness.** We could assume the comprehension bicategory to be split in addition to being strict; thus, in particular, the rules sub-id and sub-comp would collapse into ordinary equalities.
- **V3:** **Undirected TT.** We could add a rule postulating inverses of reductions. Semantically, this would amount to working in groupoid-enriched categories.
- **V4:** **Proof-irrelevant reductions.** Our syntax and the semantics allow us to distinguish parallel reductions (2-cells). We could instead "truncate" them, by moving to proof-irrelevant reductions, making the judgmental equality on them superfluous. This would yield a directed analog to the judgments of MLTT; semantically, it corresponds to working in poset-enriched categories instead of general bicategories.

We have developed comprehension bicategories with the explicit goal of encompassing previously defined interpretations of higher-dimensional and directed type theory. In the following remarks, we summarize the relationship with two previous works.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

Remark 65. (Comparison to Garner 2009). To summarize the differences between Garner's comprehension 2-categories (Garner 2009) and our comprehension bicategories: the former are full, strict, split, undirected (i.e., locally groupoidal), and incorporate type constructors.

Remark 66. (Licata and Harper's work). Licata and Harper (2011) interpret a type in context Γ as a strict 1-functor from the category interpreting Γ into a 1-category CAT of categories. Formally, they thus consider the slice bicategory $\mathsf{Cat}/\mathsf{CAT}$, where Cat is the bicategory of categories, in which CAT is assumed to be a 0-cell. The "domain" pseudofunctor $\mathsf{dom} : \mathsf{Cat}/\mathsf{CAT} \to \mathsf{Cat}$ carries the structure of a global cleaving and local opcleaving; this is more generally the case for any "domain" pseudofunctor $\mathsf{dom} : \mathsf{B}/a \to \mathsf{B}$ (`DomainCleaving.v`). To construct the comprehension pseudofunctor, one needs to use specifics of the category of strict categories, namely the Grothendieck construction of functors into CAT.

11. Terms in Directed Type Theory

The notion of term in BTT, and the interpretation of these terms, is different from the notion of term studied in the syntax and semantics by Licata and Harper (2011) or North (2019). Specifically, our terms correspond to 1-cells in the (total) category of types and are interpreted as such. Terms in the sense of Licata and Harper, and North, instead correspond to particular 1-cells in the category of contexts, specifically, to sections of the canonical projections $\pi_{\Gamma,A} : \Gamma.A \to \Gamma$.

One might attempt to reconcile these two, on the syntactic side, by

(1) adding a unit type to BTT; and
(2) asking for context morphisms to be built from terms, that is, from type morphisms. (Semantically, this corresponds to asking for χ to be full.)

However, note that most of our comprehension bicategories are *not* full. Furthermore, this syntactic modification would still leave a difference between the interpretations of terms given by Licata and Harper, and North, on the one hand, and our interpretation on the other hand. Specifically, in Example 45, terms are interpreted as functors that preserve opcartesian cells; by the Grothendieck construction, terms can equivalently be interpreted as *pseudonatural* transformations. In contrast, terms are interpreted as *lax* natural transformations by others (Licata and Harper 2011; North 2019). Hence, there is a mismatch between how terms are interpreted in comprehension bicategories compared to other interpretations.

We can analyze this mismatch more closely and suggest a potential way of reconciling it, by looking again at the intended interpretation of two-dimensional and directed type theory in categories via the opcleaving model. Taking this mismatch seriously leads us in the direction of directed type theory, where there can be different flavors of terms. For example, in work by Nuyts (2015) terms can be used isovariantly, covariantly, and contravariantly. In the opcleaving model (Example 45), these different notions of terms would correspond to different kinds of natural transformations: pseudonatural transformations, lax natural transformations, and oplax natural transformations.

To reconcile this difference, one would need to modify the syntax and the definition of comprehension bicategory. In the syntax, one would need to add a term judgment for every kind of term present in directed type theory. Accordingly, one would also add the corresponding operations like substitution. The notion of comprehension bicategory would also require slight modifications for this purpose; one necessary requirement would be that χ preserves both cartesian 1-cells and opcartesian 2-cells. The reason for this change is that this would be necessary to interpret substitution of a certain class of terms. More precisely, in the opfibration model, lax transformations

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

can be identified with sections of the projections $\int \mathsf{D} \to \mathsf{C}$. To interpret substitution for these, we need that certain squares in the arrow bicategory are pullbacks; to guarantee that, one requires that χ preserves cartesian 1-cells.

It is an open problem to develop categorical semantics for type theories that feature terms with different variances.

12. Conclusion

We have introduced the notion of comprehension bicategory, inspired by the notion of comprehension category. From this semantic notion, we have extracted a two-dimensional core syntax for dependent types, terms, and reductions and an interpretation of that syntax in comprehension bicategories. In future work, we hope to accompany this soundness result with a completeness result.

Our work is very general. We hope that it gives future investigations into two-dimensional and directed type theories a firm foundation. For instance, as outlined in Section 1.1, in future work, we will extend our structural rules with variances and a suitable hom-type former à la North (2019).

Acknowledgements. 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 Dan Licata and Bob Harper for their help in understanding their interpretation and how it relates to our framework. We thank the anonymous referees of the short version, and of the present long version, as well as editor Richard Garner, for their insightful comments and helpful advice. This work was partially funded by EPSRC under agreement number EP/T000252/1. 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).

Competing interests. Ahrens is employed at Delft University of Technology, North is employed at Utrecht University, and Van der Weide is employed at Radboud University.

Notes

1 Garner also relies on Hermida's (Hermida 1999) slightly incomplete definition of fibration of 2-categories; see Buckley's work (Buckley 2014, Remark 2.1.9) for details. We have not checked if Garner's work extends to Buckley's corrected definition of 2-fibration.

2 Note that $\to$ is used for both 1-cells and function types.

3 The definitions in the formalization differ slightly from the definitions in the paper. This is because the focus in the formalization is more general as it considers arbitrary display map bicategories that are not necessarily covariant.

4 In the formalization, we actually use unfolded versions of these definitions, and we prove these two versions are equivalent.

5 We omit the argument T in what follows.

References

Ahrens, B., Frumin, D., Maggesi, M., Veltri, N. and Van der Weide, N. (2021). Bicategories in univalent foundations. *Mathematical Structures in Computer Science* **31** (10) 1232–1269.

Ahrens, B., Kapulkin, K. and Shulman, M. (2015). Univalent categories and the Rezk completion. *Mathematical Structures in Computer Science* **25** (5) 1010–1039.

Ahrens, B. and Lumsdaine, P. L. (2019). Displayed categories. *Logical Methods in Computer Science* **15** (1) 20:1–20:18.

Ahrens, B., North, P. R. and Van der Weide, N. (2022). Semantics for two-dimensional type theory. In: Baier, C. and Fisman, D. (eds.) *LICS'22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2–5, 2022*, ACM, 12:1–12:14.

Bar, K., Kissinger, A. and Vicary, J. (2018). Globular: an online proof assistant for higher-dimensional rewriting. *Logical Methods in Computer Science* **14** (1) 1–16.

Bénabou, J. (1967). Introduction to bicategories. In: *Reports of the Midwest Category Seminar*, Lecture Notes in Mathematics, vol. 47, Berlin, Heidelberg, Springer Berlin Heidelberg, 1–77. https://doi.org/10.1007/BFb0074299.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

Benjamin, T., Finster, E. and Mimram, S. (2021). Globular weak ω-categories as models of a type theory. CoRR, abs/2106.04475. https://arxiv.org/abs/2106.04475.

Brunerie, G. (2016). *On the Homotopy Groups of Spheres in Homotopy Type Theory*. Phd thesis, Université Nice Sophia Antipolis.

Buchholtz, U. and Weinberger, J. (2021). Synthetic fibered $(\infty, 1)$-category theory. CoRR, abs/2105.01724. https://arxiv.org/abs/2105.01724.

Buckley, M. (2014). Fibred 2-categories and bicategories. *Journal of Pure and Applied Algebra* **218** (6) 1034–1074.

Coq Development Team, T. (2022). The Coq Proof Assistant. https://doi.org/10.5281/zenodo.5846982.

Coquand, T., Mannaa, B. and Ruch, F. (2017). Stack semantics of type theory. In: *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20–23, 2017*, IEEE Computer Society, 1–11.

Coraglia, G. and Di Liberti, I. (2021). Context, judgement, deduction. CoRR, abs/2111.09438. https://arxiv.org/abs/2111.09438.

Fajstrup, L., Goubault, E., Haucourt, E., Mimram, S. and Raussen, M. (2016). *Directed Algebraic Topology and Concurrency*, Springer. https://doi.org/10.1007/978-3-319-15398-8.

Finster, E. and Mimram, S. (2017). A type-theoretical definition of weak ω-categories. In: *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20–23, 2017*, IEEE Computer Society, 1–12.

Finster, E., Reutter, D., Vicary, J. and Rice, A. (2022). A type theory for strictly unital ∞-categories. In: Baier, C. and Fisman, D. (eds.) *LICS'22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2–5, 2022*, ACM, 48:1–48:12.

Fiore, M. and Saville, P. (2019). A type theory for cartesian closed bicategories (extended abstract). In: *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24–27, 2019*, IEEE, 1–13.

Garner, R. (2009). Two-dimensional models of type theory. *Mathematical Structures in Computer Science* **19** (4) 687–736.

Gray, J. W. (1966). Fibred and cofibred categories. In: Eilenberg, S., Harrison, D. K., MacLane, S. and Röhrl, H. (eds.) *Proceedings of the Conference on Categorical Algebra*, Berlin, Heidelberg, Springer Berlin Heidelberg, 21–83.

Gurski, M. N. (2006). *An Algebraic Theory of Tricategories*. Phd thesis, University of Chicago.

Hermida, C. (1999). Some properties of fib as a fibred 2-category. *Journal of Pure and Applied Algebra* **134** (1) 83–109.

Hirschowitz, T. (2013). Cartesian closed 2-categories and permutation equivalence in higher-order rewriting. *Logical Methods in Computer Science* **9** (3) 1–22.

Hofmann, M. and Streicher, T. (1994). The groupoid model refutes uniqueness of identity proofs. In: *Proceedings of the Ninth Annual Symposium on Logic in Computer Science (LICS'94), Paris, France, July 4–7, 1994*, IEEE Computer Society, 208–212.

Johnstone, P. T. (1993). Fibrations and partial products in a 2-category. *Applied Categorical Structures* **1** 141–179.

Kapulkin, K. and Lumsdaine, P. L. (2021). The simplicial model of Univalent Foundations (after Voevodsky). *Journal of the European Mathematical Society* **23** (6) 2071–2126.

Licata, D. R. (2011). *Dependently Typed Programming with Domain-Specific Logics*. Phd thesis, USA. AAI3476124.

Licata, D. R. and Harper, R. (2011). 2-dimensional directed type theory. In: Mislove, M. W. and Ouaknine, J. (eds.) *Twenty-seventh Conference on the Mathematical Foundations of Programming Semantics, MFPS 2011, Pittsburgh, PA, USA, May 25–28, 2011*, Electronic Notes in Theoretical Computer Science, vol. 276, Elsevier, 263–289.

Licata, D. R. and Harper, R. (2012). Canonicity for 2-dimensional type theory. In: Field, J. and Hicks, M. (eds.) *Proceedings of the 39th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2012, Philadelphia, Pennsylvania, USA, January 22–28, 2012*, ACM, 337–348.

Loregian, F. and Riehl, E. (2020). Categorical notions of fibration. *Expositiones Mathematicae* **38** (4) 496–514.

Lumsdaine, P. L. (2010). Weak omega-categories from intensional type theory. *Logical Methods in Computer Science* **6** (3) 1–19.

North, P. R. (2019). Towards a directed homotopy type theory. In: König, B. (ed.) *Proceedings of the Thirty-Fifth Conference on the Mathematical Foundations of Programming Semantics, MFPS 2019, London, UK, June 4–7, 2019*, Electronic Notes in Theoretical Computer Science, vol. 347, Elsevier, 223–239.

Nuyts, A. (2015). *Towards a Directed Homotopy Type Theory Based on 4 Kinds of Variance*. Master's thesis, KU Leuven. https://anuyts.github.io/files/mathesis.pdf.

Reutter, D. and Vicary, J. (2019). High-level methods for homotopy construction in associative n-categories. In: *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24–27, 2019*, IEEE, 1–13.

Riehl, E. and Shulman, M. (2017). A type theory for synthetic ∞-categories. *Higher Structures* **1** (1) 147–224.

Seely, R. A. G. (1987). Modelling computations: a 2-categorical framework. In: *Proceedings of the Symposium on Logic in Computer Science (LICS'87), Ithaca, New York, USA, June 22–25, 1987*, IEEE Computer Society, 65–71.

Shulman, M. (2010). Functorially dependent types. https://ncatlab.org/michaelshulman/show/functorially+dependent+types.

Shulman, M. (2011). Internal logic of a 2-category. https://ncatlab.org/michaelshulman/show/internal+logic+of+a+2-category.

Shulman, M. (2012). 2-categorical logic. https://ncatlab.org/michaelshulman/show/2-categorical+logic.

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press

Shulman, M. (2019). Fibrational slice. https://ncatlab.org/michaelshulman/show/fibrational+slice.

Street, R. (1980). Fibrations in bicategories. *Cahiers de Topologie et Géométrie Différentielle Catégoriques* **21** (2) 111–160.

Street, R. (1982). Characterization of bicategories of stacks. In: Kamps, K., Pumplün, D. and Tholen, W. (eds.) *Category Theory*, Lecture Notes in Mathematics, vol. 962, Springer.

Tabareau, N. (2011). Aspect oriented programming: a language for 2-categories. In: Rajan, H. (ed.) *Proceedings of the 10th International Workshop on Foundations of Aspect-Oriented Languages, FOAL 2011, Porto de Galinhas, Brazil, March 21–25, 2011*, ACM, 13–17.

Taylor, P. (1999). *Practical Foundations of Mathematics*, Cambridge Studies in Advanced Mathematics, vol. 59, Cambridge, United Kingdom: Cambridge University Press.

Van den Berg, B. and Garner, R. (2011). Types are weak ω-groupoids. *Proceedings of the London Mathematical Society* **102** (2) 370–394.

Voevodsky, V., Ahrens, B., Grayson, D., et al. (2022). UniMath — a computer-checked library of univalent mathematics. Available at https://unimath.org.

Weaver, M. Z. and Licata, D. R. (2020). A constructive model of directed univalence in bicubical sets. In: Hermanns, H., Zhang, L., Kobayashi, N. and Miller, D. (eds.) *LICS'20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8–11, 2020*, ACM, 915–928.

Cite this article: Ahrens B, North PR and van der Weide N (2023). Bicategorical type theory: semantics and syntax. *Mathematical Structures in Computer Science*. https://doi.org/10.1017/S0960129523000312

https://doi.org/10.1017/S0960129523000312 Published online by Cambridge University Press