Two-level type theory and applications

Mathematical Structures in Computer Science (2023), **33**, pp. 688–743
doi:10.1017/S0960129523000130

CAMBRIDGE UNIVERSITY PRESS

PAPER

Two-level type theory and applications

Danil Annenkov[1], Paolo Capriotti[2], Nicolai Kraus[3] and Christian Sattler[4]

[1]Concordium Blockchain Research Center, Aarhus University, Aarhus, Denmark, [2]Technische Universität Darmstadt, Darmstadt, Germany, [3]University of Birmingham, Birmingham, UK and [4]University of Nottingham, Nottingham NG7 2RD, UK

Corresponding author: Christian Sattler; Email: sattler.christian@gmail.com

(Received 17 December 2019; revised 28 February 2023; accepted 26 March 2023; first published online 30 May 2023)

Abstract

We define and develop *two-level type theory* (2LTT), a version of Martin-Löf type theory which combines two different type theories. We refer to them as the 'inner' and the 'outer' type theory. In our case of interest, the inner theory is *homotopy type theory* (HoTT) which may include univalent universes and higher inductive types. The outer theory is a traditional form of type theory validating *uniqueness of identity proofs* (UIP). One point of view on it is as internalised meta-theory of the inner type theory. There are two motivations for 2LTT. Firstly, there are certain results about HoTT which are of meta-theoretic nature, such as the statement that semisimplicial types up to level n can be constructed in HoTT for any externally fixed natural number n. Such results cannot be expressed in HoTT itself, but they can be formalised and proved in 2LTT, where n will be a variable in the outer theory. This point of view is inspired by observations about conservativity of presheaf models. Secondly, 2LTT is a framework which is suitable for formulating additional axioms that one might want to add to HoTT. This idea is heavily inspired by Voevodsky's *Homotopy Type System* (HTS), which constitutes one specific instance of a 2LTT. HTS has an axiom ensuring that the type of natural numbers behaves like the external natural numbers, which allows the construction of a universe of semisimplicial types. In 2LTT, this axiom can be assumed by postulating that the inner and outer natural numbers types are isomorphic. After defining 2LTT, we set up a collection of tools with the goal of making 2LTT a convenient language for future developments. As a first such application, we develop the theory of Reedy fibrant diagrams in the style of Shulman. Continuing this line of thought, we suggest a definition of $(\infty, 1)$-*category* and give some examples.

Keywords: two-level type theory, 2LTT, homotopy type system, homotopy type theory, conservativity, higher category

1. Introduction

The literature on homotopy type theory (HoTT), and type theory in general, offers a great variety of results. Some developments are completely internal to a specific type theory; that is, they can be expressed in type-theoretic syntax and mechanised using a proof assistant which itself is an implementation of a sufficiently good approximation of the considered type theory. Examples include most of the material in the homotopy type theory book (The Univalent Foundations Program, 2013), many theorems of which have been formalised in the proof assistants Coq (Bertot and Castéran, 2010), Agda (Norell, 2007), and Lean (de Moura et al., 2015). A second kind of literature presents results of inherently meta-theoretic nature, for example the development of models of type theory, or proofs that a system is strongly normalising, and so on.

This article was updated 18 June 2023.

What we are particularly interested in is a third kind of result. Some developments are partially internal to homotopy type theory, and often one would *want* them to be completely internal and formalisable in a proof assistant, but unfortunately, it is either unknown how this is doable or it is known to be impossible. The most well-known and most frequently discussed example for this situation is the definition of *semisimplicial types* (Lumsdaine et al., 2013). An informal explanation of the problem is the following. A *semisimplicial type of level 1* is the same as a type A_0 in a fixed universe $\mathcal{U}$. A *semisimplicial type of level 2* is a pair (A_0, A_1) of a type $A_0 : \mathcal{U}$ and a family $A_1 : A_0 \to A_0 \to \mathcal{U}$. A *semisimplicial type of level 3* is a triple (A_0, A_1, A_2) with A_0 and A_1 as before, and A_2 of the type

$$A_2 : \Pi(x, y, z : A_0).A_1\, x\, y \to A_1\, y\, z \to A_1\, x\, z \to \mathcal{U}.$$

We think of A_0 as a type of points, $A_1\, x\, y$ as a type of lines from x to y, and $A_2\, x\, y\, z\, f\, g\, h$ as a type of 'triangle fillers' for the triangle spanned by f, g and h. It is tedious but intuitively clear how to extend this definition to levels 4 or 5 (or even 100) in this style. An open problem of HoTT asks: Is it possible to construct a function $S : \mathbb{N} \to \mathcal{U}_1$ such that, for every n, the type $S(n)$ encodes the type of semisimplicial types of level n? It is known that, for any externally fixed number k (e.g. 4,5,100), we can construct a type S_k that encodes semisimplicial types of level k. However, this is not enough to construct an internal function S. Two questions arise naturally:

(1) How can we formalise the construction of S_k for every external natural number k?
(2) How can we extend the type theory such that S itself can be constructed?

In order to answer question 2, Voevodsky suggested a type theory called *homotopy type system (HTS)* (Voevodsky, 2013). This theory introduces a type of what they call *exact equalities*. Exact equality is an internalised version of judgemental (a.k.a. definitional) equality and co-exists with, but is very different from, the usual internal equality type (a.k.a. identity type, identification type, path type). In HTS, exact equality comes with some additional built-in assumptions.

We can think of the type theory of HTS as divided into two levels, each of which is a version of type theory on its own. The first is the actual object of study and close to HoTT; HTS refers to it as the *fibrant fragment*, and its equality type is the usual path type. In the second type theory, it is possible to reason about, rather than in, HoTT; this is the 'full' type theory of HTS with all types including not necessarily fibrant ones, and its equality type is viewed as 'internalised judgemental equality'.

We call a system of this form (and the study of such systems) *two-level type theory* (2LTT). The two levels are respectively called *inner* and *outer*; for this paper and for HTS, the inner level is a version of HoTT.[1]

We strive to keep the assumptions on 2LTT to a minimum. For example, while HTS assumes that types of the inner level are (in some appropriate sense) a 'subset' of the outer level, we do not make this assumption. Instead, we only request that there is a conversion function from the inner level to the outer. This is not as drastic of a change as it may sound, and it allows for a larger class of models. Of course, the HTS setup is the special case where the conversion function is simply an inclusion, and a user of 2LTT is free to add this or other such assumptions to the theory they consider, as least as long as these assumptions are justified by a model. We discuss such variations in Subsection 2.4. However, most of the theory in the later part of this paper is developed without such assumptions.

One intuition for the two levels is as follows: from a type in HoTT, we can extract a statement that can be phrased in the meta-theory. From a meta-theoretical statement *about* HoTT, it is not always possible to construct a type. Thus, we can convert inner types into outer one, but not always vice versa.

The fact that HTS calls inner types *fibrant* suggests an interpretation in model categories or similar models of abstract homotopy theory. We will adopt a similar terminology, but choose to

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

reserve the term *fibrant* for a slightly different notion: an outer type is fibrant if it is isomorphic to an inner one. This makes mixing the two levels more convenient when working internally.

While HTS is an answer to question 2, it does not seem suitable as a tool to address question 1: the theory of HTS has not been designed to be conservative over HoTT. Thus, it is unclear what the relation is between statements provable in the inner level of HTS and statements provable in HoTT. The version of 2LTT that we study in this paper *does* have this conservativity property over HoTT (Capriotti, 2016) (see Subsection 2.6). That is, given a type in any context in HoTT, if the corresponding type in the inner level of 2LTT is inhabited, we obtain an inhabitant of the original type in HoTT. This can be seen as a canonicity property of the inner level with respect to the full system. It is unknown whether the same is the case for HTS, but we do not expect it (see the discussion in Subsection 2.4). This crucial difference between our version of 2LTT and HTS is however not due to the differences mentioned so far. Instead, the reason is that HTS assumes that many type formers (apart from equality), in particular empty, unit, and natural number type are shared between the two levels. In our setting, this would correspond to the assumption (A1) of Subsection 2.4 that the conversion function from the inner to the outer level preserves these type formers, and such a strong assumption is not covered by the mentioned conservativity result (Capriotti, 2016). In summary, the basic version of 2LTT that we work with in this paper is suitable to address question 1, and the framework makes it easy to add additional axioms, also weaker ones such as (A2), which allows it to address question 2.

The idea of semisimplicial types can be developed further as demonstrated by Shulman, who considers diagrams over a larger class of categories (Shulman, 2015*b*). For clarity of what happens, we switch back to the setting of HoTT rather than 2LTT. Shulman shows that, given externally a category $\mathcal{C}$ with a certain property (being inverse), we have a well-behaved notion of type-valued diagrams over $\mathcal{C}$ (called Reedy fibrant). For finite $\mathcal{C}$, there will even be a *type* of such diagrams. Note that $\mathcal{C}$ is not a variable that we can quantify over inside the type theory: it is assumed to be given externally. In other words, if we choose a concrete instance for $\mathcal{C}$, we can take a proof assistant, implement the type of these diagrams and work with them internally. However, if $\mathcal{C}$ is an internal variable, this is not possible. 2LTT offers a setting in which this situation can be developed and formalised: $\mathcal{C}$ becomes a variable in the outer level, and we will demonstrate in this paper how this can be done (see Section 4). Semisimplicial types restricted to level n are the special case where $\mathcal{C}$ is taken to be the initial segment of length n of the semisimplex category.

Without 2LTT, a standard approach to the development of such a theory of diagrams over $\mathcal{C}$ is to fix a (possibly arbitrary) model of type theory and work in the corresponding category, using categorical tools. This requires carefully mixing *internal* notions with *external* ones, and there may not always be a clean way to achieve that. In the mentioned work by Shulman, the role of the model of HoTT is played by a *type-theoretic fibration category* (TTFC). Most results of their paper are formulated at that level, that is, as categorical constructions within a particular TTFC. This requires changing style of presentation compared to a more 'traditional' type-theoretic exposition, like for example that of the book on HoTT (The Univalent Foundations Program, 2013). For instance, one has to work with morphisms rather than terms, fibrations rather than families of types, talk about pullbacks rather than just performing substitutions and use "diagrammatic" instead of "equational" reasoning techniques. Although these stylistic variations are not necessarily bad in themselves, the fact that one is essentially *forced* to apply them can lead to difficulties. A testament to that is the fact that, on occasions, the discussed work (Shulman, 2015*b*) falls back to the internal language to formulate certain definitions and properties, as this is much easier than expressing them in a category-theoretic form. 2LTT offers an alternative strategy: Type theory is the only language that is needed, meaning that internal and external reasoning go hand in hand, and the mixing feels very natural.

There are three equally valid ways to think about 2LTT: We can start with the type theory that we want to study (e.g. HoTT), take it as the inner level and build some of its meta-theory as an additional layer on top of it. Vice versa, we can start with a theory that is suitable as the outer level

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

(e.g. MLTT with function extensionality and unique identity proofs), expose a type family declared as the universe of inner types and develop the inner level from there. As the middle ground, we can think of the inner and outer theories as coexisting side-by-side, with a shared notion of context, related only by a conversion function from the inner to the outer level. We take this middle ground as our setup, but our point of view, reflected in our choice of terminology, is the second: we consider the outer level as the "default" type theory, and in particular, *equality* and *equality type* will always mean the equality type of the outer level. When talking about constructions that happen at the inner level, we will make this clear explicitly, and the equalities of the inner level are referred to as *inner equalities* or *path-equalities*. We make this choice for a number of reasons. First, it is reasonable when considering models, since the equality of the outer level is much closer to actual "external equality" than path-equality is (see Subsection 2.5). Second, our choice is also pragmatic since, in practice, one works in the outer theory most of the time as the outer theory is more expressive. For example, in the outer theory, a (small) diagram of types always has a limit that can be calculated in the same way as in the category of sets. In general, of course this will not be fibrant, but the ability to talk about it will be useful nevertheless. Finally, our choice also matches the way that 2LTT can be implemented in existing proof assistants such as Coq, Agda or Lean. We have formalised some of the results in this paper in this style.[2]

1.1 Context of this paper and related work

The current paper significantly reworks and extends the idea of 2LTT that Altenkirch and two of the current authors have presented at the CSL'16 conference (Altenkirch et al., 2016). As discussed, a main inspiration for the development presented in the current paper is Voevodsky's HTS (Voevodsky, 2013), which itself was suggested as an answer to question 2. The other aspect (question 1) is perhaps a bit closer to the motivation for Maietti's *minimalist two-level foundation for constructive mathematics* (Maietti, 2009) (also cf. the work with Sambin Maietti and Sambin 2005). There, the reason for the split of the theory into two levels is that it allows to have *minimal type theory* as (what we call) the inner level, a type theory that is free of extensionality principles and implements a specific formulation of the proofs-as-programs paradigm, while still having an (in our terminology) outer level with powerful principles. Somewhat similarly, 2LTT has an inner level that can be taken to be free from principles that are not part of HoTT, while such principles can then be added via the outer level. Angiuli et al. (2018) present cartesian cubical type theory as a two-level system.

2LTT as presented in the current paper (or, rather, a previous draft of it that had been available for a while) has been used by and connected to several other lines of work. One is the book *The Univalence Principle* by Ahrens et al. (2021), using the setting to formulate and prove a very general result stating that equivalent mathematical structures are indistinguishable.[3] Going in a different direction, Kovács uses 2LTT for staging with dependent types (Kovács, 2022) and, in particular, shows that staging with stability and soundness corresponds to conservativity over the inner level. Yet another application was given by Barras and Maestracci (2020), using a two-level type theory in Dedukti (Assaf et al., 2016) to encode cubical type theory. Finally, 2LTT provides a framework which may be expressive enough to "eat" (model in a partially synthetic sense) HoTT (Kraus, 2021).

Other suggestions to address question 2, i.e. systems that make it possible to develop a theory of semisimplicial types and higher categories, have been made. One is the type theory for synthetic ∞-categories by Riehl and Shulman (2017), a setting that uses additional context layers to express structure that, in 2LTT, would be expressed via the outer equality type. A setting closer to standard HoTT, but less expressive, was suggested by Finster, Allioux and Sozeau: By equipping HoTT with a universe of judgementally associative and unital polynomial monads, they can encode higher coherent algebraic structures including ∞-groupoids.

Our formalisation approach of 2LTT mentioned above uses the type theory of Lean as the outer level, and uses type classes to keep track of and automatically propagate fibrancy constraints. We

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

discuss this further in the conclusions (Section 5). This strategy is similar to the one used by Boulier and Tabareau (2017) in Coq, although their development proceeds in a direction different from the one pursued in the present paper. They define the fibrant equality type as a *private* inductive type (Bertot, 2013). Exposing a custom induction principle for such a private inductive type allows one to retain computational behaviour while restricting the user to explicitly provided eliminators. However, private inductive types are not available in all proof assistants. Agda supports a version of 2LTT more directly via a universe of "strict sets" SSet.[4] Agda's approach is somewhat different: instead of starting in the outer level and encoding inner types from there, it treats Agda's types as fibrant by default and adds a new universe to simulate the outer level instead. 2LTT in this setting has been explored by Uskuplu (2022).

1.2 Outline

The outline of the paper is as follows. In Section 2, we specify the version of 2LTT that we consider in this paper. We intentionally include as few assumptions on the theory as possible, but we also discuss a number of reasonable additional assumption that one would like to make. The section also discusses the semantics of 2LTT, with some minor differences to the development of Capriotti (2016). Indeed, we think of 2LTT as being defined via its category of models. We show basic results and introduce the useful notions of *fibrancy* and *cofibrancy* in Section 3. This section, we hope, turns 2LTT into a language which can be useful for the study of concepts that are not completely internal to HoTT. A first such application can be found in Section 4, where we develop the theory of Reedy fibrant diagrams over inverse categories. We conclude in Section 5 with a short discussion on formalisations.

2. Two-level type theory

The basic idea of 2LTT is that it contains two separate levels of types:

- the *outer* level, which is a form of traditional Martin-Löf type theory with intensional equality types and the principle of uniqueness of identity proofs (UIP);
- the *inner* level, which is essentially homotopy type theory, and contains univalent universes and potentially higher inductive types (The Univalent Foundations Program, 2013).

In this section, we start by suggesting a syntax for 2LTT. We strive to be close to the standard syntax of MLTT and HoTT as used in the book (The Univalent Foundations Program, 2013). In a nutshell, we have *two* copies of each basic type or type former, one outer and one inner. Inner types can be converted to outer types via a separate operation, and contexts are shared between the two levels.

Our suggested syntax should not be understood as a complete specification of 2LTT: such syntactical specifications require many more rules than we give, most of which are obvious and standard but nevertheless important. Instead, we give a precise specification of 2LTT with a semantic approach. We define what a model of 2LTT is (essentially a combination of two *categories with families* Dybjer 1995 which share a common category of contexts). From this definition, it is clear that the suggested syntax can be used to perform constructions in any model of 2LTT.

Remark 2.1. (Initiality of the syntax)**.** We can view the syntax as notation which works in any model, and this is how we understand the developments in later sections of the paper. Our category of two-level models will be the category of models of a generalised algebraic theory and thus be locally finitely presentable. As such, there is in particular an *initial* model for two-level type theory, and of course, all constructions will work in this initial model. If one were to make the syntax precise (cf. Bonacina and Ahrens 2021; Bonacina et al. 2021), then one would expect this

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

initial model to coincide with the *term model*. However, it is known in the community that a complete proof for this sort of statement requires a lot of work. For the calculus of constructions, this was carefully worked out by Streicher (1993), and formalisation projects for intensional Martin-Löf type theory were described by de Boer, Brunerie, Lumsdaine, and M (Brunerie et al., 2019; Brunerie and Lumsdaine, 2018, 2020; Lumsdaine and Mörtberg, 2018). An Agda formalisation is available as part of the licentiate thesis by de Boer (2020). It may be possible to adapt these proofs to two-level type theory, but this is beyond the scope of the paper. While the question is of course important for type theory in general, it is orthogonal to the specific idea of having two levels.

After specifying 2LTT via models, we observe some immediate consequences from the definitions: for example, Π- and Σ-types are preserved up to isomorphism when converting from outer to inner types. Other properties do not follow from the definitions but could be added as assumptions, leading to systems such as HTS, and we discuss these assumptions separately. We also discuss several specific example models (or classes of example models) and prove a conservativity property for 2LTT without further assumptions. Further, we examine the possibility of an *inner replacement* (or *fibrant replacement*).

2.1 Syntax

We stay close to the presentation of type theory given in the appendix in the homotopy type theory book (The Univalent Foundations Program, 2013, Appendix A.2). There is however one technical difference that we want to make. The semantics of Russell-style universes (where terms of the universe are types) is less elegant than the one of Tarski-style universes (if A is a term of a universe, then $\mathsf{El}\ A$ is a type), and the former can be seen as a special case of the latter. This is a general observation in type theory which has little to do with the idea of having two levels; see also point (M2) in Subsection 2.4.

We consider the judgements Γ ctx, $\Gamma \vdash a : A$, and $\Gamma \vdash a \equiv a' : A$. In addition, we consider the two judgements

$$\Gamma \vdash A\ \mathrm{type}_j \qquad\qquad \Gamma \vdash A\ \mathrm{type}^{\mathsf{i}}_j$$

Here, j is a natural number, the *size* of A. The first means that A is an outer type, the second that A is an inner type. Similar to the judgement $\Gamma \vdash a \equiv a' : A$, we consider equality judgements for types.

For the outer level of the theory that we consider, we have the following basic types and type formers:

- Π, the type former of dependent functions;
- Σ, the type former of dependent pairs;
- $+$, the coproduct type former;
- $\mathbf{1}$, the unit type;
- $\mathbf{0}$, the empty type;
- $\mathbb{N}$, the type of natural numbers;
- $=$, the equality type;
- a cumulative hierarchy $\mathcal{U}_0$, $\mathcal{U}_1$, . . . of universes;
- inductive and quotient types (not used in this paper).

The inner level of our 2LTT has the same basic types and type formers. We annotate them to avoid confusion:[5]

- Π^{i}, the type former of inner dependent functions;
- Σ^{i}, the type former of inner dependent pairs;
- $+^{\mathsf{i}}$, the inner coproduct type former;

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

- $\mathbf{1}^{\mathsf{i}}$, the inner unit type;
- $\mathbf{0}^{\mathsf{i}}$, the inner empty type;
- $\mathbb{N}^{\mathsf{i}}$, the inner type of natural numbers;
- $=^{\mathsf{i}}$, the inner equality (or *path-equality*) type (in the sense of HoTT);
- a cumulative hierarchy $\mathcal{U}_0^{\mathsf{i}}, \mathcal{U}_1^{\mathsf{i}}, \ldots$ of inner universes;
- possibly inductive and higher inductive types.

The basic rules for Π, $+$, $\mathbf{1}$, $\mathbf{0}$, $\mathbb{N}$, as well as Π^{i}, $+^{\mathsf{i}}$, $\mathbf{1}^{\mathsf{i}}$, $\mathbf{0}^{\mathsf{i}}$, $\mathbb{N}^{\mathsf{i}}$ are the standard ones and match those given in the HoTT book (The Univalent Foundations Program, 2013, Appendix A.2), modulo the difference between Russell and Tarski universes. For Σ and Σ^{i}, we assume in addition the judgemental η-law $x \equiv (\pi_1(x), \pi_2(x))$.[6] Note that all inference rules in the cited appendix are stated in terms of universes, and in our situation, all occurrences of $A : \mathcal{U}_j$ are replaced by $A\ \mathrm{type}_j$ and $A : \mathcal{U}_j^{\mathsf{i}}$ by $A\ \mathrm{type}_j^{\mathsf{i}}$; this keeps the two levels separate. For example, for the formation of coproducts, we have the rules

$$\frac{\Gamma \vdash A\ \mathrm{type}_j \qquad \Gamma \vdash B\ \mathrm{type}_j}{\Gamma \vdash A + B\ \mathrm{type}_j}\ \textsc{form-+}$$

$$\frac{\Gamma \vdash A\ \mathrm{type}_j^{\mathsf{i}} \qquad \Gamma \vdash B\ \mathrm{type}_j^{\mathsf{i}}}{\Gamma \vdash A +^{\mathsf{i}} B\ \mathrm{type}_j^{\mathsf{i}}}\ \textsc{form-+}^{\mathsf{i}}$$

To emphasise, these rules do *not* allow us to form a coproduct of an inner and an outer type! The outer equality type $=$ and inner equality (path-equality) type $=^{\mathsf{i}}$ have the usual rules as well. Since equality is the central aspect of 2LTT, we state the rules for the outer equality type explicitly, although they are completely standard:

$$\frac{\Gamma \vdash A\ \mathrm{type}_j \qquad \Gamma \vdash a, b : A}{\Gamma \vdash a = b\ \mathrm{type}_j}\ \textsc{form-=} \qquad \frac{\Gamma \vdash a : A}{\Gamma \vdash \mathsf{refl}_a : a = a}\ \textsc{intro-=}$$

$$\frac{\Gamma \vdash a : A \qquad \Gamma.(b : A).(p : a = b) \vdash P\ \mathrm{type}_j \qquad \Gamma \vdash d : P[a, \mathsf{refl}_a]}{\Gamma.(b : A).(p : a = b) \vdash J_P(d) : P}\ \textsc{elim-=},$$

together with the usual computation rule:

$$J_P(d)[a, \mathsf{refl}_a] \equiv d.$$

The rules of the inner type are the same, with type_j replaced by $\mathrm{type}_j^{\mathsf{i}}$, $=$ by $=^{\mathsf{i}}$, and refl_a by $\mathsf{refl}_a^{\mathsf{i}}$. The El operator is assumed to be an isomorphism between terms of $\mathcal{U}_j$ (or $\mathcal{U}_j^{\mathsf{i}}$) and (inner/outer) types at level i. For the outer equality type, we furthermore assume the principles of UIP and function extensionality:

$$\frac{\Gamma \vdash a_1, a_2 : A \qquad \Gamma \vdash p, q : a_1 = a_2}{\Gamma \vdash K(p, q) : p = q}\ \textsc{uip}$$

$$\frac{\Gamma \vdash f, g : \Pi_{a:A} B(a) \qquad \Gamma.(a : A) \vdash p(a) : f(a) = g(a)}{\Gamma \vdash \mathsf{funext}(p) : f = g}\ \textsc{funext}$$

Instead of UIP, we could add the slightly stronger principle called *Axiom K* (Streicher, 1993). This is a version of (2.1) (with its computation rule) for inducting on loops with a fixed base point. It does not make a difference in our treatment.

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

All inner universes $\mathcal{U}_j^{\mathsf{i}}$ are assumed to be univalent in the sense of homotopy type theory.

Context extension also follows the rules of The Univalent Foundations Program (2013, Appendix A.2), but note that there is only *one* judgement of the form $\Gamma \vdash \mathrm{ctx}$ and there are *two* hierarchies of types. Since context extension works for every type, we have:

$$\frac{\Gamma\ \mathrm{ctx} \qquad \Gamma \vdash A\ \mathrm{type}_j}{\Gamma.A\ \mathrm{ctx}}\ \textsc{ctx-ext} \qquad \frac{\Gamma\ \mathrm{ctx} \qquad \Gamma \vdash A\ \mathrm{type}_j^{\mathsf{i}}}{\Gamma.A\ \mathrm{ctx}}\ \textsc{ctx}^{\mathsf{i}}\textsc{-ext}$$

This means that contexts are shared between the two levels.

Finally, we have a conversion operation c which turns inner types into outer types: Whenever we have $\Gamma \vdash A\ \mathrm{type}_j^{\mathsf{i}}$, we have a type $\mathrm{c}(A)$ such that $\Gamma \vdash \mathrm{c}(A)\ \mathrm{type}_j$ and this operation preserves context extension, in the sense that $\Gamma.A$ and $\Gamma.\mathrm{c}(A)$ are the same context. This operation is natural in Γ, and the detailed specification is given in the next subsection. Preservation of context extension in particular means that the set of terms of A and $\mathrm{c}(A)$ are isomorphic. For the 'forwards-direction', we again write c, that is, for $\Gamma \vdash a : A$, we have $\Gamma \vdash \mathrm{c}(a) : \mathrm{c}(A)$.

2.2 Semantics

We define what it means to be a model of two-level type theory. For this, we use the language of *categories with families* (cwfs) (Dybjer, 1995). We have a choice of how to handle universe hierarchies. In this work, we aim for concreteness and fix a cumulative hierarchy indexed by natural numbers, with no notion of top-level type.

Given a presheaf F over a category $\mathcal{E}$, we denote by $\mathcal{E}/F$ its category of elements. Recall the following notion.

Definition 2.2 (Dybjer 1995)**.** A *category with families* (cwf) is a category $\mathcal{E}$ with:

- a presheaf Ty over $\mathcal{E}$ (*types*),
- a presheaf Tm over $\mathcal{E}/\mathsf{Ty}$ (*terms*),
- a terminal object $1 \in \mathcal{E}$ (*global* or *empty context*),
- for all $\Gamma \in \mathcal{E}$ and $A \in \mathsf{Ty}(\Gamma)$, a terminal object $(\Gamma.A, p_A, q_A)$ in the category of triples (Δ, σ, t) where $\Delta \in \mathcal{E}$, $\sigma : \Delta \to \Gamma$, and $t \in \mathsf{Tm}(\Delta, A[\sigma])$ (*context extension*).

In the above, $[\sigma]$ denotes the action of Ty on the morphism σ. The action of Tm on morphisms is written similarly. These operations are referred to as *substitutions* of types and terms, respectively.

We treat Definition 2.2 as having algebraic character, immediately giving rise to a category of cwfs. This applies to all the definitions in this subsection. They are to be read as not just introducing a certain concept (algebraic in character), but also the corresponding notion of morphism for it, part of a category structure.

Lemma 2.3. *Naturally in the cwf $\mathcal{E}$, $\Gamma \in \mathcal{E}$ and $A \in \mathsf{Ty}(\Gamma)$, the set $\mathsf{Tm}(\Gamma, A)$ is isomorphic to the set of sections of p_A, with the isomorphism given by terminality of $(\Gamma.A, p_A, q_A)$ and substitution of q_A.*

In the usual fashion, one has notions of cwfs with type formers and axioms such as Π-types, identity types and function extensionality. In the following, we abbreviate a generic selection of such type formers and axioms by T and speak of cwfs having type formers T.

Given a cwf $\mathcal{E}$, we also speak of a *cwf structure* on the category $\mathcal{E}$. We abbreviate such a cwf structure just by its presheaf of types. By restriction, we obtain a category of cwf structures with type formers T on a fixed category $\mathcal{E}$. Note that the action of its morphisms on terms is an isomorphism (this follows from preservation of extension).

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

Definition 2.4. A *cwf hierarchy* Ty *with type formers* T on a category $\mathcal{E}$ is a sequential diagram

$$\mathsf{Ty}_0 \longrightarrow \mathsf{Ty}_1 \longrightarrow \dots$$

of cwf structures with type formers T on $\mathcal{E}$.

We can regard a cwf hierarchy as a multi-sorted cwf indexed over the poset ω. This makes it a cumulative hierarchy. Note that the natural transformation $\mathsf{Ty}_j \to \mathsf{Ty}_{j+1}$ preserve the type formers T. We will omit the subscript index into the hierarchy when it is inferable or we are only interested at a fixed index.

Definition 2.5. A *model of Martin-Løf type theory with type formers* T on a category $\mathcal{E}$ is a cwf hierarchy Ty with type formers T on $\mathcal{E}$ together with, for each j, a global section $\mathcal{U}_j$ of Ty_{j+1} with an isomorphism $\mathsf{El}_j : \mathsf{Tm}_{j+1}(\Gamma, \mathcal{U}_j) \simeq \mathsf{Ty}_j(\Gamma)$ natural in $\Gamma \in \mathcal{E}$.

We refer to such a model by just its cwf hierarchy Ty. We call $\mathcal{U}_j$ the *j-th universe*. If unambiguous, we will omit the universe index j. Note that the above definition models a cumulative hierarchy of universes (closed under type formers), with no top-level notion of type.

We consider two important kinds of models of Martin-Löf type theory.

Definition 2.6. A *model of set type theory* is a model of Martin-Löf type theory with the following formers:

(i) $\mathbf{1}/\Sigma/\Pi$-types, all with η-laws (i.e. satisfying universal properties);
(ii) identity types, empty types, (binary) coproduct types and natural number types;
(iii) uniqueness of identity proofs and function extensionality.

Definition 2.7. A *model of homotopy type theory* is a model of Martin-Löf type theory with the type formers (i) and (ii) of Definition 2.6 that is *univalent*, i.e. $(\mathcal{U}_j, \mathsf{El}_j)$ is univalent in Ty_{j+1} for each j.

In applications, one can add more type formers to these notions as desired, for example higher inductive types to Definition 2.7 or quotient types to Definition 2.6. All our example models of set type theory are in fact models of extensional type theory, i.e. have equality reflection. For our developments here, the given type formers will suffice.

Definition 2.8. A *two-level model* (of Martin-Löf type theory) with *inner type formers* T^{i} and *outer type formers* T consists of:

- a category $\mathcal{E}$ (*contexts*),
- the *inner level*, a model Ty^{i} with type formers T^{i} on $\mathcal{E}$,
- the *outer level*, a model Ty with type formers T on $\mathcal{E}$,
- a *conversion* morphism $\mathsf{c} : \mathsf{Ty}^{\mathsf{i}} \to \mathsf{Ty}$ of cwf hierarchies (with no type formers) on $\mathcal{E}$, converting inner types to outer types.

Ignoring type formers, a two-level model can be seen as a multi-sorted cwf indexed over the poset $\{0 \to 1\} \times \omega$. With type formers, we have two separate multi-sorted cwfs (with inner and outer type formers, respectively) indexed over ω. The conversion morphism c can be described as a morphism of multi-sorted cwfs that is the identity on the category of contexts. Consequently, context extension is automatically preserved: if we have $\Gamma \vdash A\ \mathsf{type}^{\mathsf{i}}_j$, then $\Gamma.A$ and $\Gamma.\mathsf{c}(A)$ are the

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

same object of $\mathcal{E}$.[7] Note that we assume no interaction between inner and outer type formers under the conversion morphism.

Finally, we can make precise two-level type theory.

Definition 2.9. A *model of two-level type theory* is a two-level model of Martin-Löf type theory where:

- the inner level is a model of homotopy type theory,
- the outer level is a model of set type theory.

2.3 Preservation of type formers by conversion

For this subsection, we fix a model $\mathcal{E}$ of two-level type theory. We have assumed very little about the conversion morphism $\mathsf{c} : \mathsf{Ty}^{\mathsf{i}} \to \mathsf{Ty}$, but in this subsection, we see that it being a morphism of cwf hierarchies allows us to derive important properties.

Lemma 2.10. *For $\Gamma \vdash A\ \mathrm{type}^{\mathsf{i}}_j$, the conversion operator c from the set of terms of A to the set of terms of $c(A)$ is an isomorphism.*

Proof. This follows from 2.3 since c preserves context extesion. □

Further, we can use that many types are characterised by universal properties (the syntactical equivalent of which are elimination rules). The consequences are summarised in the following statement.

Lemma 2.11. *Let $\Gamma \in \mathcal{E}$, $A \in \mathsf{Ty}^{\mathsf{i}}(\Gamma)$ and $B \in \mathsf{Ty}^{\mathsf{i}}(\Gamma.A)$. We have the following morphisms natural in Γ. All morphisms live in the slice over Γ (and (7) lives in the slice over $\Gamma.A.A$):*

$$\Gamma.c(\mathbf{1}^{\mathsf{i}}) \xrightarrow{\sim} \Gamma.\mathbf{1} \tag{1}$$

$$\Gamma.c(\Sigma^{\mathsf{i}}_A B) \xrightarrow{\sim} \Gamma.\Sigma_{c(A)} c(B) \tag{2}$$

$$\Gamma.c(\Pi^{\mathsf{i}}_A B) \xrightarrow{\sim} \Gamma.\Pi_{c(A)} c(B) \tag{3}$$

$$\Gamma.(c(A) + c(B)) \to \Gamma.c(A +^{\mathsf{i}} B) \tag{4}$$

$$\Gamma.\mathbf{0} \to \Gamma.c(\mathbf{0}^{\mathsf{i}}) \tag{5}$$

$$\Gamma.\mathbb{N} \to \Gamma.c(\mathbb{N}^{\mathsf{i}}) \tag{6}$$

$$\Gamma.(u, v : A).\left(c(u) =_{c(A)} c(v)\right) \to \Gamma.(u, v : A).c(u =^{\mathsf{i}}_A v) \tag{7}$$

$$\Gamma.c(\mathcal{U}^{\mathsf{i}}_j) \to \Gamma.\mathcal{U}_j \tag{8}$$

Moreover, the three annotated morphisms are natural isomorphisms.

Before constructing the morphisms, let us make some remarks. Via the adjunction with Π, the above morphisms give rise to internal functions at the outer level. This way, isomorphisms become judgemental internal isomorphisms, i.e. the functions in both directions compose judgementally to the identity.

It is also worth emphasising the asymmetry that the conversion function c introduces: In general, the rules of the system mean that it is usually easier to eliminate from outer types into inner types than vice versa. While Π, Σ and $\mathbf{1}$ at the outer level are 'the same' as at the inner level, in the sense made precise in the lemma above, the same is not automatically the case for the remaining types and type formers. For the case of equality types however, this assumption would destroy our motivation for two-level type theory altogether. Although invertibility of (4)–(6), discussed in Subsection 2.4 under (A1), does hold in some of the intended models, it is not valid with the point of view of the outer level as internalised meta-theory of the object theory given by the inner level,

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

made precise by the presheaf model of two-level type theory in Subsection 2.5.3. From that point of view, some of the maps and the absence of their invertibility can be understood as follows:

- Coproducts $A + B$: given a term of A or a term of B in the meta-theory, we get an element of $A +^{\mathsf{i}} B$, but not vice versa. For example, the context might not allow us to normalise an element of a coproduct type to a coprojection.
- Empty type $\mathbf{0}$: from a contradiction in the meta-theory, one can get a contradiction in the object theory, but not vice versa.
- Natural numbers $\mathbb{N}$: we think of an external natural number as a numeral. From a numeral, one can get an internal natural number, but from an element of the inner natural numbers type, we do not always get a numeral.
- Equality $x = y$: two meta-theoretically (e.g. syntactically) equal expressions are provably equal via reflexivity, but provably equal expressions might not be equal meta-theoretically (syntactically).

The last morphism (8) says that inner types are (up to c) outer types, but we would not expect the reverse since not every meta-theoretic statement can be internalised.

Proof of 2.11. We start with the three isomorphisms. Recall that inner and outer $\mathbf{1}/\Sigma$-types come with η-laws. Since c preserves context extensions, it easily follows that c preserves these type formers up to canonical isomorphism. In fact, this is true for cwf morphisms in general, without the requirement that the underlying functor is an identity. In detail, the isomorphism (1) is given by

$$\begin{aligned}\Gamma.\mathsf{c}(\mathbf{1}^{\mathsf{i}}) &= \Gamma.\mathbf{1}^{\mathsf{i}}\\ &\simeq \Gamma\\ &\simeq \Gamma.\mathbf{1}\end{aligned}$$

and the isomorphism (2) is given by

$$\begin{aligned}\Gamma.\mathsf{c}(\Sigma^{\mathsf{i}}_A B) &= \Gamma.\Sigma^{\mathsf{i}}_A B\\ &\simeq \Gamma.A.B\\ &= \Gamma.c(A).c(B)\\ &\simeq \Gamma.\Sigma_{\mathsf{c}(A)}\mathsf{c}(B).\end{aligned}$$

Since Π-types in the inner and outer level come with the η-law, their terms are uniquely characterised as terms of the codomain type. We have, naturally in $\sigma : \Delta \to \Gamma$:

$$\begin{aligned}\mathcal{E}/\Gamma\left(\Delta, \Gamma.\mathsf{c}(\Pi^{\mathsf{i}}_A B)\right) &= \mathcal{E}/\Gamma\left(\Delta, \Gamma.\Pi^{\mathsf{i}}_A B\right)\\ &\simeq \mathcal{E}/\Gamma.A\ (\Delta.A[\sigma], \Gamma.A.B)\\ &= \mathcal{E}/\Gamma.\mathsf{c}(A)\ (\Delta.\mathsf{c}(A[\sigma]), \Gamma.\mathsf{c}(A).\mathsf{c}(B))\\ &= \mathcal{E}/\Gamma.\mathsf{c}(A)\ (\Delta.\mathsf{c}(A)[\sigma], \Gamma.\mathsf{c}(A).\mathsf{c}(B))\\ &\simeq \mathcal{E}/\Gamma\left(\Delta, \Gamma.\Pi_{\mathsf{c}(A)}\mathsf{c}(B)\right),\end{aligned}$$

from which (3) follows by Yoneda. The morphism (4) is given by the usual properties of outer and inner coproduct. We have $\Gamma.A \to \Gamma.\mathsf{c}(A +^{\mathsf{i}} B)$ and $\Gamma.B \to \Gamma.\mathsf{c}(A +^{\mathsf{i}} B)$ due to the inner coproduct, and the outer coproduct lets us construct (4). The morphism (5) comes from the outer empty type $\mathbf{0}$; no property of the inner empty type is used.

The usual morphism $\Gamma.\mathsf{c}(\mathbf{1}^{\mathsf{i}} +^{\mathsf{i}} \mathbb{N}^{\mathsf{i}}) \to \Gamma.\mathsf{c}(\mathbb{N}^{\mathsf{i}})$, by composition with (1) and (4), gives rise to a morphism $\Gamma.(\mathbf{1} + \mathsf{c}(\mathbb{N}^{\mathsf{i}})) \to \Gamma.\mathsf{c}(\mathbb{N}^{\mathsf{i}})$. By the universal property of $\mathbb{N}$, we get the morphism (6).

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

Outer equality implies inner equality in the sense of (7) by the J-eliminator of the outer equality and 2.10. Finally, (8) uses the isomorphisms El_j and $\mathsf{El}^{\mathsf{i}}_j$. □

Remark 2.12. For "positive" type formers such as empty types, coproduct types, natural number types or identity types, we cannot hope for preservation up to isomorphism by c, even if we add η-laws to both the inner and outer type former. This is because their universal property talks about maps out of the type in question into another type of the same level, rather than an arbitrary object of the slice. For example, for empty types with η-law in both inner and outer level, we have natural isomorphisms $\mathcal{E}/\Gamma(\Gamma.0^{\mathsf{i}}, \Gamma.C) \simeq 1$ for $C \in \mathsf{Ty}^{\mathsf{i}}(\Gamma)$ and $\mathcal{E}/\Gamma(\Gamma.0, \Gamma.C) \simeq 1$ for $C \in \mathsf{Ty}(\Gamma)$. To be able to conclude that $\Gamma.0^{\mathsf{i}} \simeq \Gamma.0$ over Γ, we would have to apply the universal property of 0^{i} to an outer type C.

Our proof of the isomorphism (2) uses the judgemental η-law for Σ-types, a rule that is not assumed by all authors. If we only assume the induction principle with which the HoTT book (The Univalent Foundations Program, 2013, Chapter 1.6) characterises Σ-types, without a judgemental η-law, then Σ becomes a positive type former, not generally preserved by c.

In some of the models, we discuss in Subsection 2.5, the comparison maps (4)–(8) are indeed not isomorphisms.

2.4 Strengthenings and extensions

In Subsection 2.2, we have defined two-level type theory in a minimalistic fashion, eschewing properties that one might argue are reasonable to ask for. Indeed, other two-level type theories such as HTS are much more rigid.

We separate possible strengthenings into three groups. The first group has nothing to do with two-level type theory proper, concerning only the basic structure of models of Martin-Löf type theory as per Definition 2.5. In a model of two-level type theory, this applies to both inner and outer levels separately.

(M1) We can ask that the step maps $\mathsf{Ty}_j \to \mathsf{Ty}_{j+1}$ in the cwf hierarchy are mono. In a set-theoretic meta-theory, we could go further and demand the step maps are subpresheaf inclusions.

(M2) We can ask that the natural isomorphism $\mathsf{El} : \mathsf{Tm}(\Gamma, \mathcal{U}_j) \simeq \mathsf{Ty}_j(\Gamma)$ is an equality on the nose, i.e. that $\mathsf{Tm}(\Gamma, \mathcal{U}_j) = \mathsf{Ty}_j(\Gamma)$ and $\mathsf{El} = \mathsf{id}$. This has the effect of modelling Russell-style universes.

Implementing both of these points makes it possible to forgo the distinction between types and terms (with types just being terms of universes), treating the typing relation as going between terms. Together with univalence, this yields a type theory as in the homotopy type theory book (The Univalent Foundations Program, 2013, Appendix A.2) (but note that the η-law for Σ-types is not included there).

The second group concerns strictness properties of the conversion morphism c relating the inner to the outer level in a model of two-level type theory as per Definition 2.9.

(T1) We can ask that the conversion morphism $\mathsf{c} : \mathsf{Ty}^{\mathsf{i}} \to \mathsf{Ty}$ is mono on types, i.e. that $\mathsf{c} : \mathsf{Ty}^{\mathsf{i}}(\Gamma) \to \mathsf{Ty}(\Gamma)$ is injective for $\Gamma \in \mathcal{E}$. In a set-theoretic meta-theory, we could demand this is on the nose, i.e. that c is a subpresheaf inclusion on types (and that the action on terms is not just an isomorphism, but an equality).

(T2) We can ask that the isomorphisms of 2.11 witnessing preservation of $\mathbf{1}/\Sigma/\Pi$-types under the conversion morphism c are equalities on the nose. For Π-types, this means the following: given $A \in \mathsf{Ty}^{\mathsf{i}}(\Gamma)$ and $B \in \mathsf{Ty}^{\mathsf{i}}(\Gamma.A)$, we have $\mathsf{c}(\Pi^{\mathsf{i}}_A B) = \Pi_{\mathsf{c}(A)} \mathsf{c}(B)$; given further

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

$f \in \mathsf{Tm}^{\mathsf{i}}(\Gamma, \Pi^{\mathsf{i}}(A,B))$ and $a \in \mathsf{Tm}^{\mathsf{i}}(\Gamma, A)$, we have $\mathsf{app}^{\mathsf{i}}(f,a) = \mathsf{app}(f,a)$ (preservation of abstraction is implied by this).

If we ask for any other type formers to be preserved by c up to canonical isomorphism (such as in (A1) below), we can similarly ask that these isomorphisms are equalities on the nose.

(T3) We can ask that inner types are *replete* within outer types, i.e. that any outer type with extension isomorphic to that of an inner type is itself the image of an inner type under conversion. In detail, given $A \in \mathsf{Ty}(\Gamma)$ and $B \in \mathsf{Ty}^{\mathsf{i}}(\Gamma)$ with $\Gamma.A \simeq \Gamma.B$ over Γ, we have $A' \in \mathsf{Ty}^{\mathsf{i}}(\Gamma)$ with $A = \mathsf{c}(A')$, naturally in Γ.

Implementing (T1) and (T2) essentially yields an (outer) type theory with a predicate of "being inner" on types that some type formers are closed under. With inner types named "fibrant", this is the perspective taken in HTS.

The third group concerns more semantical extensions or axioms that will differ depending on what kinds of models one is interested in.

(A1) We can ask that conversion preserves certain "positive" type formers (minus identity types) up to canonical isomorphism, making for example the following canonical comparison maps over $\Gamma \in \mathcal{E}$ invertible:

$$\Gamma.0 \to \Gamma.0^{\mathsf{i}} \qquad \text{cf. (5)}$$

$$\Gamma.(\mathsf{c}(A) + \mathsf{c}(B)) \to \Gamma.(A +^{\mathsf{i}} B) \qquad \text{for } A, B \in \mathsf{Ty}^{\mathsf{i}}(\Gamma) \qquad \text{cf. (4)}$$

$$\Gamma.\mathbb{N} \to \Gamma.\mathbb{N}^{\mathsf{i}} \qquad \text{cf. (6)}$$

We can weaken this by asking for an inverse only up to the outer identity type. Then, these properties become axioms internal to two-level type theory.

(A2) The following is a weakening of the assertion of (A1) for natural numbers that still allows for the construction of inner types of Reedy fibrant semisimplicial types (see 4.38 and afterwards). We can ask that countably infinite towers of (trivial) fibrations have (trivially) fibrant limits. The notion of fibration used here will be defined in Subsection 3.2 in terms of inner types. This is an internalisation (to the outer level) of the corresponding axiom considered for (co)fibration categories (Radulescu-Banu, 2006, Definition 1.6.1) (see also Shulman 2015*b*, Lemma 11.8).

(A3) Weakening (A2) further, we can ask that the outer natural number type is *cofibrant*. The notion of cofibrant type will be defined in Subsection 3.4, essentially meaning that exponentiation with it preserves inner types up to isomorphism. This axiom has been suggested by Shulman.

(A4) We can ask that the outer universes are "fibrant", i.e. that there is $u \in \mathsf{Ty}^{\mathsf{i}}(\Gamma)$ such that $\Gamma.\mathcal{U} \simeq \Gamma.u$ over Γ (or even $\mathcal{U} = \mathsf{c}(u)$), naturally in $\Gamma \in \mathcal{E}$. Again, we can weaken this to an isomorphism up to the outer identity type, making it an axiom internal to two-level type theory concerning universes.

(A5) We can ask that the outer level validates the equality reflection rule, i.e. forms a model of extensional type theory. This is the case in all the example models we are interested in.

We phrase this as an extension so that the base systems retains good meta-theoretical properties such as decidability of type checking. This is relevant for faithful implementation by current proof assistants (note though that some systems such as Andromeda (Bauer et al.) model equality reflection).

As a compromise, one may add features to the outer level that are partially extensional while retaining decidability of type checking. An example is the recent addition of universes

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

of strict propositions (with judgemental uniqueness of elements) to Agda and Coq (Gilbert et al., 2019).

(A6) We may add more type formers to the inner or outer level as desired. For example, since the inner level is simply a version of homotopy type theory, it is natural to add inner higher inductive types. We can even add higher inductive-inductive types (see e.g. The Univalent Foundations Program 2013 and Kaposi and Kovács 2020 for a specification). Similarly, we can add quotient types or quotient inductive-inductive types (Altenkirch et al., 2018, 2017; Altenkirch and Kaposi, 2016) to the outer level (note that the presence of uniqueness of identity proofs makes higher equalities moot).

Note that one may identify HTS as two-level type theory in our sense extended with the axioms (A1), (A4), (A5), (T1) and (T2), in their strongest form. In the following subsection, we will discuss which of the above strengthenings and extensions hold in each of several example models. This will provide justification for not including most of the above conditions as blanket assumptions. It will also serve as a guide to the reader on which assumption to include in their two-level type theory when they have a certain class of models in mind.

2.5 Example models

We discuss some key models of two-level type theory. All have in common that the underlying category is presheaves $\widehat{\mathcal{C}}$ over a category $\mathcal{C}$ and that the outer level is given by the standard presheaf model of extensional type theory (in particular, (A5) is validated) where types are (small) presheaves over the category of elements of their context. The only exception to this is in Proposition 2.16, where the outer level is different.

We briefly recall key details of this presheaf model in a set-theoretic meta-theory. Fix a sequence of Grothendieck universes $M_0 \in M_1 \in \ldots$ such that $\mathcal{C}$ lives in M_0. Given $\Gamma \in \widehat{\mathcal{C}}$, then $\mathsf{Ty}_j(\Gamma)$ consists of presheaves over $\mathcal{C}/\Gamma$ valued in M_j and $\mathsf{Tm}_j(\Gamma, A)$ is the set of global sections of such a presheaf A. Then, Ty_j is represented by $\mathcal{U}_j \in \widehat{\mathcal{C}}$ where $\mathcal{U}_j(X)$ is the set of presheaves over $\mathcal{C}/X$ valued in M_j, which itself lives in M_{j+1}. This defines the universe $\mathcal{U}_j \in \mathsf{Ty}_{j+1}(1)$. With types presented in this displayed form, the standard definition of type formers is substitution-stable and preserved under size change.

The above definition of types and universes is essentially that of Hofmann and Streicher (1997). Other constructions are possible, for example following Voevodsky (Kapulkin and Lumsdaine, 2018, Subsection 2.1) or Shulman (2015a) (the latter construction works equally in the non-univalent setting of classifying all maps, not fibrations), but necessitate further work to split type formers.

2.5.1 Simplicial sets

The first model of homotopy type theory was in simplicial sets (Kapulkin and Lumsdaine, 2018). As already remarked in that paper, simplicial sets, being a presheaf category, exhibit two separate, but related, structures of models of type theory: the one constructed in the paper itself and the one that every presheaf category has, modelling extensional type theory. This idea can be expanded by making simplicial sets a model of two-level type theory. We suspect that an observation along these lines motivated Voevodsky's HTS.[8]

Letting $\mathcal{C} = \Delta$ be the simplex category, the outer level is the presheaf model of simplicial sets as explained above. The inner types over $\Gamma \in \widehat{\Delta}$ are interpreted as the subset of those outer types whose corresponding "display map" with target Γ is a Kan fibration, making (T1) hold. Kan fibrations are closed under isomorphism, hence (T3) holds (in fact, this can be strengthened to closure under retracts).

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

Outer $\mathbf{1}/\Sigma/\Pi$-types preserve fibrancy, giving their inner interpretation and enforcing (T2). This applies also to empty types, coproduct types and natural number types, giving (A1)–(A3). For a monomorphism i, pullback and pushforward form a reflection $i^* \dashv \Pi_i$, giving trivial fibrancy of outer universes (A4).

The inner identity type is modelled by the cotensor with Δ^1. Recall that its elimination operation has a splitting issue. We follow the splitting strategy introduced by Kapulkin and Lumsdaine (2018), interpreting the offending operation in the universal context that captures its inputs. For this, one might try to use the representing object $\mathcal{U}_j$ for i-small types. However, the size change map $\mathsf{Ty}^{\mathsf{i}}_j \to \mathsf{Ty}^{\mathsf{i}}_{j+1}$ would then not preserve the operation.[9] Instead, as in Kapulkin and Lumsdaine (2018), we introduce yet another Grothendieck universe M_ω containing $M_0, M_1, \ldots$, define a presheaf $\mathcal{U}_\omega$ as above and use it to build the universal context.

Since generating trivial cofibrations in the form of horn inclusions have representable codomain, the presheaves of inner types are representable, yielding the inner universes. They are fibrant and univalent as in Kapulkin and Lumsdaine (2018).

Following the setup of Pitts and Orton (2018), a more internal development of the simplicial set model in line of the above choices is described in Coquand et al. (2019, Appendix D). It also describes (following a suggestion by Andrew Swan) how the higher inductive types constructed in the cubical setting in Coquand et al. (2018) interpret in the simplicial model (A6).

2.5.2 Cubical sets

A similar class of models of two-level type theory is given by cubical sets for various choices of a cubical site and notion of fibrations (Bezem et al., 2014; Cohen et al., 2017). In contrast to Kapulkin and Lumsdaine (2018), these models of homotopy type theory have been developed from the start with Hofmann-Streicher universes in mind and all type formers split by construction. Thus, they immediately fit our setup and we can simply declare them to form the inner level.

The development of cubical models of Pitts and Orton (2018), Licata et al. (2018) can be interpreted as *defining* the inner level internally to the outer level. Note that this requires extending the outer level to crisp type theory (Licata et al., 2018; Shulman, 2018).

A major difference to the simplicial model is that cubical Kan lifts are part of the structure of inner types. That is, an inner type is not just an outer type satisfying a lifting property, but has an additional datum in its Kan composition operation. This invalidates (T1). Other strengthenings (T2) and (T3) and (A1)–(A6) hold in the same manner as discussed above for simplicial sets.

The reason that simplicial and cubical sets model validate (A1) is, in both cases, that fibrant objects (in slices) are closed under small coproducts.

2.5.3 Presheaves over models of homotopy type theory

The material in the first half of this subsubsection follows Capriotti (2016, Chapter 3.2). The resulting models have guided the design choices of our two-level type theory.

We will first establish some preliminaries. A *weak morphism* $F : \mathcal{C} \to \mathcal{D}$ of cwfs is a functor between underlying categories with natural transformations on types and terms that preserves the global context and extension only up to canonical isomorphism. Given interpretations of type formers T in $\mathcal{C}$ and $\mathcal{D}$, it still makes sense to ask that F preserves the operations of T, transporting along these preservation isomorphisms when required. Indeed, by expressing the type formers T in a cwf $\mathcal{E}$ as operations internal to its presheaf category $\widehat{\mathcal{E}}$, one may completely avoid the dependency of T on extension as an algebraic operation (Capriotti, 2016; Uemura, 2019). Thus, we obtain a notion of weak morphism of cwfs with type formers T.

There is an evident notion of 2-morphism between weak cwf morphisms $F, G : \mathcal{C} \to \mathcal{D}$, a natural transformation $u : F \to G$ such that $FA = (GA)[u_\Gamma]$ for $A \in \mathsf{Ty}_{\mathcal{C}}(\Gamma)$ and $Ft = (Gt)[u_\Gamma]$ for additionally $t \in \mathsf{Tm}_{\mathcal{C}}(\Gamma, A)$. Note that this implies commutativity of

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

$$\begin{array}{ccc} F(\Gamma.A) & \xrightarrow{\simeq} & F\Gamma.FA \\ \downarrow{\scriptstyle u_{\Gamma.A}} & & \downarrow{\scriptstyle u_{\Gamma}.GA} \\ G(\Gamma.A) & \xrightarrow{\simeq} & G\Gamma.GA \end{array}$$

for $\Gamma \in \mathcal{C}$ and $A \in \mathsf{Ty}_{\mathcal{C}}$. This extends to a notion of 2-morphism for cwfs with type formers T by requiring that substitution along the components of u preserves the operations of T. With this, we obtain a (strict) 2-category of cwfs (with type formers T) and weak morphisms. It has the 1-category of cwfs (with type formers T) as a wide sub-2-category.

Importantly, the initial object $\mathcal{C}$ of the 1-category of cwfs (with type formers T) becomes biinitial in the 2-category of cwfs (with type formers T) and weak morphisms. That is, given an object $\mathcal{D}$ with a weak morphism $H : \mathcal{C} \to \mathcal{D}$, there is an isomorphism $H \cong F$ between weak morphisms where $F : \mathcal{C} \to \mathcal{D}$ is the unique morphism. This may be derived from making the strict pseudolimit of H (seen as a diagram indexed by the walking arrow) into a cwf $\mathcal{E}$ (with type formers T):

- objects are triples (X, X', f) where $X \in \mathcal{C}$, $X' \in \mathcal{D}$, and $f : H(X) \simeq X'$,
- types over such an object are pairs (A, A') with $A \in \mathsf{Ty}_{\mathcal{C}}(X)$ and $A' \in \mathsf{Ty}_{\mathcal{D}}(X')$ such that $H(A)$ and A' correspond over f,
- terms of such a type are pairs (t, t') with $t \in \mathsf{Tm}_{\mathcal{C}}(X, A)$ and $t' \in \mathsf{Tm}_{\mathcal{D}}(X', A')$ such that $H(t)$ and t' correspond over f.

Note that A' and t' in the above description are redundant. We have projection morphisms $p_{\mathcal{C}} : \mathcal{E} \to \mathcal{C}$ and $p_{\mathcal{D}} : \mathcal{E} \to \mathcal{D}$. By initiality of $\mathcal{C}$, we have $G : \mathcal{C} \to \mathcal{E}$ such that $p_{\mathcal{C}} \circ G = \mathsf{id}_{\mathcal{C}}$ and $p_{\mathcal{D}} \circ G = F$. The isomorphism $H \cong F$ is read off from it.

Everything we have said above extends analogously to cwf hierarchies (with type formers T), models of Martin-Löf type theory (with type formers T), models of homotopy type theory and models of two-level type theory.

Recall from Hofmann (1997) that for any category $\mathcal{C}$, the category $\widehat{\mathcal{C}}$ of presheaves over $\mathcal{C}$ forms a model of extensional type theory. Here, the types and terms are defined as follows. Given a presheaf P, which we think of as a context, we define $\mathsf{Ty}(P)$ to consists of (small) presheaves over the category $\mathcal{C}/P$ of elements of P (specifically, for $\mathsf{Ty}_j(P)$, we require these presheaves to be valued in the Grothendieck universe M_j). Given such a type $A \in \mathsf{Ty}^{\mathsf{i}}(P)$ over P, the set $\mathsf{Tm}(A)$ of terms is the set of global sections of the corresponding presheaf over $\mathcal{C}/P$.

In the special case where $\mathcal{C}$ is itself a cwf, we can define an additional, *inner* cwf structure Ty^{i} on presheaves $\widehat{\mathcal{C}}$ with a morphism $\mathsf{c} : \mathsf{Ty}^{\mathsf{i}} \to \mathsf{Ty}$ to the presheaf cwf structure Ty. This works as follows. We can single out a special context (in other words, a type in the empty context), namely $\mathsf{Ty}_{\mathcal{C}}$ itself. This context can play the role of a universe in the presheaf cwf $\widehat{\mathcal{C}}$. In fact, we have a type $\mathsf{Tm}_{\mathcal{C}} \in \mathsf{Ty}(\mathsf{Ty}_{\mathcal{C}})$, acting as the universal family of this universe. The cwf structure induced by this universe forms the inner cwf structure Ty^{i} of $\widehat{\mathcal{C}}$. In detail, we define $\mathsf{Ty}^{\mathsf{i}} = y(\mathsf{Ty}_{\mathcal{C}})$, i.e. $\mathsf{Ty}^{\mathsf{i}}(P) = \widehat{\mathcal{C}}(P, \mathsf{Ty}_{\mathcal{C}})$, and let c send $A \in \mathsf{Ty}^{\mathsf{i}}(P)$ to the restriction of $\mathsf{Tm}_{\mathcal{C}}$ along the functor $\mathcal{C}/P \to \mathcal{C}/\mathsf{Ty}_{\mathcal{C}}$ induced by A, i.e. to $c(A) \in \widehat{\mathcal{C}/P}$ sending (Γ, x) to $\mathsf{Tm}_{\mathcal{C}}(\Gamma, A(x))$. We are then forced to define $\mathsf{Tm}^{\mathsf{i}}(P, A)$ as the set of global sections of $\mathsf{c}(A)$.

The Yoneda embedding $y_{\mathcal{C}} : \mathcal{C} \to \widehat{\mathcal{C}}$ becomes a weak morphism of cwfs

$$y_{\mathcal{C}} : \mathcal{C} \to (\widehat{\mathcal{C}}, \mathsf{Ty}^{\mathsf{i}}), \tag{9}$$

whose actions on types and terms are bijective by construction of Ty^{i}.

Let $\mathcal{C}$ now support a selection of type formers T. We can lift the rules in T to corresponding operations and laws on the "universe" $\mathsf{Ty}_{\mathcal{C}}$ in $(\widehat{\mathcal{C}}, \mathsf{Ty})$ and thereby to interpretations of the type

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

formers T in the inner cwf $(\widehat{\mathcal{C}}, \mathsf{Ty}^{\mathsf{i}})$. Furthermore, the weak morphism (9) preserves these, i.e. $y_{\mathcal{C}}$ becomes a weak morphism of cwfs with type formers T. This process and its properties are explained in Hofmann (1997) for a specific set of type formers, and in Capriotti (2016) for a generic notion of type former.

We illustrate the above process for the formation operation for dependent products. We desire the following judgement in the presheaf cwf:

$$A : \mathsf{Ty}_{\mathcal{C}}, B : \mathsf{Tm}_{\mathcal{C}}(A) \to \mathsf{Ty}_{\mathcal{C}} \vdash \Pi(A, B) : \mathsf{Ty}_{\mathcal{C}}. \tag{10}$$

Naturally in $\Gamma \in \mathcal{C}$, we are given:

(1) $A \in \mathsf{Ty}_{\mathcal{C}}(\Gamma)$,
(2) naturally in $(\Delta, \sigma : \Delta \to \Gamma) \in \mathcal{C}/\Gamma$, a map $B_\sigma : \mathsf{Tm}_{\mathcal{C}}(\Delta, A[\sigma]) \to \mathsf{Ty}_{\mathcal{C}}(\Delta)$,

and have to produce an element $\Pi(A, B) \in \mathsf{Ty}_{\mathcal{C}}(\Gamma)$. By the universal property of context extension in $\mathcal{C}$, data in (2) are uniquely induced by just the element $B_{p_A} \in \mathsf{Ty}(\Gamma.A)$. Thus, our obligation precisely corresponds to the formation rule for Π in $\mathcal{C}$. Furthermore, after lifting to the cwf $(\widehat{\mathcal{C}}, \mathsf{Ty}^{\mathsf{i}})$, we can check that the weak morphism (9) preserves the formation rule.

The above example makes it reasonable to expect that every type former can be lifted, rule by rule, from $\mathcal{C}$, producing judgements that replicate each rule internally in the theory of $\widehat{\mathcal{C}}$ when expressed using the "universe" Ty_C, and that hence one can interpret each rule in the inner presheaf cwf.

The above construction is functorial in the cwf structure $\mathsf{Ty}_{\mathcal{C}}$ (with type formers T) on $\mathcal{C}$ and 2-functorial in $\mathcal{C}$ as an object of the 2-category of cwfs (with type formers T) and weak morphisms. From this, we obtain the following.

Proposition 2.13. *Let $\mathcal{C}$ be a model of Martin-Löf type theory with type formers T. Assume that $(\mathsf{Tm}_{\mathcal{C}})_j$ is valued in the Grothendieck universe M_j for every i. Then, $\widehat{\mathcal{C}}$ forms a two-level model of Martin-Löf type theory with inner type formers T and outer types formers from extensional type theory. The Yoneda embedding extends to a weak morphism $y : \mathcal{C} \to (\widehat{\mathcal{C}}, \mathsf{Ty}^{\mathsf{i}})$ of models of Martin-Löf type theory that acts bijectively on types and terms.*

Furthermore, this operation is 2-functorial in $\mathcal{C}$ as an object of the 2-category of models of Martin-Löf type theory with type formers T and weak morphisms. The action on a weak morphism $F : \mathcal{C} \to \mathcal{D}$ is as follows. The left Kan extension $F_! : \widehat{\mathcal{C}} \to \widehat{\mathcal{D}}$ extends to a weak morphism of two-level models as above. The natural isomorphism $F_! \circ y_{\mathcal{C}} \simeq y_{\mathcal{D}} \circ F$ of functors lifts to the 2-category of models of Martin-Löf type theory with type formers T and weak morphisms.

Proof. Applying the above discussion to the sequence of cwf structures $(\mathsf{Ty}_{\mathcal{C}})_j$ with type formers T on $\mathcal{C}$, we obtain a corresponding sequence of cwf structures

$$\mathsf{Ty}^{\mathsf{i}}_0 \longrightarrow \mathsf{Ty}^{\mathsf{i}}_1 \longrightarrow \dots$$

with type formers T on $\widehat{\mathcal{C}}$. Yoneda preserves terminal objects, so sends the global section $\mathcal{U}_j$ of $(\mathsf{Ty}_{\mathcal{C}})_{j+1}$ to a global section $\mathcal{U}^{\mathsf{i}}_j$ of $\mathsf{Ty}^{\mathsf{i}}_{j+1}$. Naturally in $\Gamma \in \mathcal{C}$, we have

$$\mathsf{Tm}^{\mathsf{i}}(y(\Gamma), \mathcal{U}^{\mathsf{i}}_j) = \mathsf{Tm}^{\mathsf{i}}(y(\Gamma), y(\mathcal{U}_j)) \simeq \mathsf{Tm}_{\mathcal{C}}(\Gamma, \mathcal{U}_j) \simeq (\mathsf{Ty}_{\mathcal{C}})_j(\Gamma) \simeq \mathsf{Ty}^{\mathsf{i}}_j(y(\Gamma)),$$

using that the action of (9) on types and terms is bijective. By cocontinuous extension, we thus have $\mathsf{Tm}^{\mathsf{i}}(X, \mathcal{U}^{\mathsf{i}}_j) \simeq \mathsf{Ty}^{\mathsf{i}}_j(X)$ naturally in $X \in \widehat{\mathcal{C}}$. By the smallness assumption, the map $\mathsf{Ty}^{\mathsf{i}}_j \to \mathsf{Ty}$ restricts to $\mathsf{Ty}^{\mathsf{i}}_j \to \mathsf{Ty}_j$. □

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

Seeing univalence as just another type former, the universes $\mathcal{U}_j^{\mathsf{i}}$ in the above model are univalent if the original universes $\mathcal{U}_j$ in $\mathcal{C}$ are.

Corollary 2.14. *Let $\mathcal{C}$ be a model of homotopy type theory. Assume that $(\mathsf{Tm}_{\mathcal{C}})_j$ is valued in the Grothendieck universe M_j for every i. Then, $\widehat{\mathcal{C}}$ forms a model of two-level type theory. The Yoneda embedding extends to a weak morphism $y : \mathcal{C} \to (\widehat{\mathcal{C}}, \mathsf{Ty}^{\mathsf{i}})$ of models of homotopy type theory that acts bijectively on types and terms. Furthermore, this operation is 2-functorial in $\mathcal{C}$ as in Proposition 2.13.*

We call this the *presheaf model* $\widehat{\mathcal{C}}$ of two-level type theory over the given model $\mathcal{C}$ of homotopy type theory. It will be key for proving conservativity of two-level type theory over homotopy type theory in Subsection 2.6.

Let us discuss strictness properties of conversion satisfies by the presheaf model. Depending on the specifics of the implementation of the outer types, c may or may not have a chance to be mono. With our choice of presheaves over categories of elements, (T1) holds as long as the action of the presheaf $\mathsf{Tm}_{\mathcal{C}}$ on objects is injective. Note that this can always be achieved by passing through the Grothendieck construction. None of the other strictness properties (T2) and (T3) are satisfied.

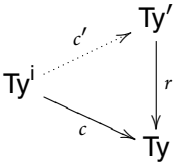
Remark 2.15. Concerning (T2), some effort is expended in Capriotti (2016, Chapter 3.2) to achieve strict preservation of $\mathbf{1}/\Sigma/\Pi$-types under what we here call conversion from the inner to outer level. This is achieved by defining the inner types more cleverly as certain free expressions involving the types of $\mathcal{C}$ and formal $\mathbf{1}/\Sigma/\Pi$-type forming operations. Unfortunately, this only works for "top-level" $\mathbf{1}/\Sigma/\Pi$-types and breaks whenever the kind of type in question has a classifier that is itself a type (of higher size). Thus, this technique is not applicable here.

The presheaf model does not preserve (up to isomorphism) "positive" type formers as in (A1). Note that the interpretation of empty types, coproducts, and natural numbers in the outer level is levelwise. Were (A1) to hold, then in an arbitrary context Γ in $\mathcal{C}$, there would be no terms of empty type, every term of coproduct type would be a constructor application, and every natural number term would be a canonical numeral. Even the weaker versions (A2) and (A3) do not hold in general. Note that (A2) holds if $\mathcal{C}$ supports dependent sums of "arity" ω (also known as record types with countably infinitely many fields). Axiom (A4) is also generally not satisfied.

As for the other example models, the outer level models extensional type theory, i.e. (A5) holds. Extension (A6) holds as far as permitted by the given model $\mathcal{C}$ of homotopy type theory.

We end this subsection by giving a modified version of the presheaf model where (T1) and (T2) hold. This is achieved by modifying the interpretation of the outer level. The technique is inspired by Shulman's modification (Shulman, 2019, Appendix A) of the local universe splitting technique in the presence of universes (recall though that we do not make use of the local universe splitting technique).

Proposition 2.16. *Denote by $(\mathsf{Ty}^{\mathsf{i}}, \mathsf{Ty}, c)$ the presheaf model of two-level type theory on $\widehat{\mathcal{C}}$ as established by Corollary 2.14. There is a factorisation*



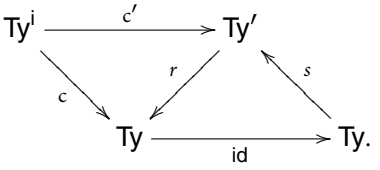
in the category of cwf hierarchies such that $(\mathsf{Ty}^{\mathsf{i}}, \mathsf{Ty}', c')$ forms a model of two-level type theory satisfying (T1) and (T2).

Furthermore, this operation is 2-functorial in $\mathcal{C}$ in the same sense as Corollary 2.14.

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

Proof. The following is to be understood as happening for every size index i, which we omit. By a small set, we mean an element of the Grothendieck universe M_j.

Let $\mathcal{U} \in \widehat{\mathcal{C}}$ denote the representing object of Ty, with universal element $\mathsf{El} \in \mathsf{Ty}(\mathcal{U})$. Define Ty' as the presheaf represented by $\mathsf{Ty}_{\mathcal{C}} + \mathcal{U}$. We have a map $[\mathsf{c}, \mathsf{id}_{\mathcal{U}}] : \mathsf{Ty}_{\mathcal{C}} + \mathcal{U} \to \mathcal{U}$. Applying Yoneda, this induces the map $r : \mathsf{Ty}' \to \mathsf{Ty}$. The cwf structure of Ty' is inherited from Ty via r. Applying Yoneda to the coprojections $\mathsf{inl} : \mathsf{Ty}_{\mathcal{C}} \to \mathsf{Ty}_{\mathcal{C}} + \mathcal{U}$ and $\mathsf{inr} : \mathcal{U} \to \mathsf{Ty}_{\mathcal{C}} + \mathcal{U}$, we obtain respective cwf structure morphisms $\mathsf{c}' : \mathsf{Ty}^{\mathsf{i}} \to \mathsf{Ty}'$ and $s : \mathsf{Ty} \to \mathsf{Ty}'$. These fit into a commuting diagram as follows:



Unfolding the definition, we find that, given $X \in \widehat{\mathcal{C}}$, an element of $\mathsf{Ty}'(X)$ consists of a partition $X = X_0 \sqcup X_1$ of X into subpresheaves X_0 and X_1 together with $A_0 \in \mathsf{Ty}^{\mathsf{i}}(X_0)$, i.e. $A_0 : X_0 \to \mathsf{Ty}_{\mathcal{C}}$, and $A_1 \in \mathsf{Ty}(X_1)$, i.e. a small presheaf A_1 over $\mathcal{C}/X_1$.

The interpretation of type formers in Ty' other than $\mathbf{1}/\Sigma/\Pi$-types is as for Ty and is defined such that it is preserved by r. For the type forming operations, we first transport the given types in Ty' to Ty via r, use the corresponding type forming operation there, and apply s to the result. Since $r \circ s = \mathsf{id}$, this makes r preserve the type forming operation, meaning the remainder of the operations dealing with terms can be copied from Ty to Ty'.

It remains to interpret $\mathbf{1}/\Sigma/\Pi$-types in Ty'. Since the terms of these type formers are characterised by universal properties, it will suffice to define their type forming operations such that r preserves $\mathbf{1}/\Sigma/\Pi$-types up to isomorphism. The remainder of their operations dealing with terms is then uniquely induced. In order to ensure that c' preserves $\mathbf{1}/\Sigma/\Pi$-types, we only have to check that c' preserves the type forming operations and that the isomorphisms used in the penultimate sentence are the ones of Lemma 2.11 whenever the input types come from

The case of $\mathbf{1}$-types is trivial: given $X \in \widehat{\mathcal{C}}$, we imply take $\mathbf{1}' = \mathsf{inl}(\mathbf{1}^{\mathsf{i}}) \in \mathsf{Ty}'(X)$ using $\mathbf{1}^{\mathsf{i}} \in \mathsf{Ty}^{\mathsf{i}}$. Since it has no inputs, there is nothing to show. The type forming operations for Σ-types and Π-types are of the same form. To save space, we only show the case of Π-types.

Recall how the Π-type forming operation of Ty^{i} was inherited from the one of $\mathsf{Ty}_{\mathcal{C}}$ via the operation (10). The cwf structure of Ty' is induced by $\mathsf{El}[\mathsf{c}, \mathsf{id}_{\mathcal{U}}] \in \mathsf{Ty}(\mathsf{Ty}_{\mathcal{C}} + \mathcal{U})$ rather than $\mathsf{Tm}_{\mathcal{C}} \in \mathsf{Ty}(\mathsf{Ty}_{\mathcal{C}})$ as for (10). Here, we have to interpret the analogous operation

$$A : \mathsf{Ty}_{\mathcal{C}} + \mathcal{U}, B : \mathsf{El}([\mathsf{c}, \mathsf{id}_{\mathcal{U}}](A)) \to (\mathsf{Ty}_{\mathcal{C}} + \mathcal{U}) \vdash \Pi'(A, B) : \mathsf{Ty}_{\mathcal{C}} + \mathcal{U}$$

together with

$$\mathsf{El}([\mathsf{c}, \mathsf{id}_{\mathcal{U}}](\Pi'(A, B))) \simeq \mathsf{El}(\Pi([\mathsf{c}, \mathsf{id}_{\mathcal{U}}](A), [\mathsf{c}, \mathsf{id}_{\mathcal{U}}] \circ B))$$

in the same context.

To define the action at level $\Gamma \in \mathcal{C}$, we take

$$\begin{aligned} &A \in \mathsf{Ty}_{\mathcal{C}}(\Gamma) + \mathcal{U}(\Gamma), \\ &B \in \widehat{\mathcal{C}/\Gamma}(\mathsf{El}([\mathsf{c}, \mathsf{id}_{\mathcal{U}(\Gamma)}](A)), \mathsf{Ty}_{\mathcal{C}} + \mathcal{U}) \end{aligned} \tag{11}$$

(omitting restriction in the target of B) and must define $\Pi'(A, B) \in \mathsf{Ty}_{\mathcal{C}}(\Gamma) + \mathcal{U}(\Gamma)$ with an isomorphism

$$\mathsf{El}([\mathsf{c}, \mathsf{id}_{\mathcal{U}}](\Pi'(A, B))) \simeq \mathsf{El}(\Pi([\mathsf{c}, \mathsf{id}_{\mathcal{U}}](A), [\mathsf{c}, \mathsf{id}_{\mathcal{U}}] \circ B)) \tag{12}$$

of presheaves over $\mathcal{C}/\Gamma$. We perform a case distinction on A.

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

(i) Suppose $A = \mathsf{inr}(A_1)$ with $A_1 \in \mathcal{U}(\Gamma)$. Then, we take

$$\Pi'(A,B)(\Gamma) = \mathsf{inr}(\Pi(A_1, [\mathrm{c}, \mathsf{id}_{\mathcal{U}}] \circ B)),$$

using the type forming operation of Ty, and let (12) be the identity.

(ii) Suppose $A = \mathsf{inl}(A_0)$ with $A_0 \in \mathsf{Ty}_{\mathcal{C}}(\Gamma)$. Then, $\mathsf{El}([\mathrm{c}, \mathsf{id}_{\mathcal{U}(\Gamma)}](A) = \mathsf{El}(\mathrm{c}(A))$ is the presheaf over $\mathcal{C}/\Gamma$ sending $\sigma : \Delta \to \Gamma$ to $\mathsf{Tm}(\Delta, A[\sigma])$. Since $\mathcal{C}$ has extension, this presheaf is representable (represented by $\Gamma.A$). In particular, mapping out of it as in (11) preserves coproducts. We can thus make a further case distinction on B.

(a) If $B = \mathsf{inr} \circ B_1$ with $B_1 \in \widehat{\mathcal{C}/\Gamma}(\mathsf{El}(\mathrm{c}(A)), \mathcal{U})$, we take

$$\Pi'(A,B)(\Gamma) = \mathsf{inr}(\Pi(\mathrm{c}(A_0), B_1)),$$

again using the type forming operation of Ty, and let (12) be the identity.

(b) If $B = \mathsf{inl} \circ B_0$ with $B_0 \in \widehat{\mathcal{C}/\Gamma}(\mathsf{El}(\mathrm{c}(A)), \mathsf{Ty}_{\mathcal{C}})$, we take

$$\Pi'(A,B)(\Gamma) = \mathsf{inl}(\Pi^{\mathsf{i}}(A_0, B_0)),$$

using the type forming operation of Ty^{i}, and let (12) be the isomorphism given by Lemma 2.11.

One checks that this definition is natural in Γ. Recalling that c' is given by the action of Yoneda on $\mathsf{inl} : \mathsf{Ty}_{\mathcal{C}} \to \mathsf{Ty}_{\mathcal{C}} + \mathcal{U}$, we find that c' preserves Π-type formation by construction (case (ii.b)) with the required coherence isomorphism.

This finishes the verification that Ty' forms a cwf hierarchy with the type formers of the outer level of a model of two-level type theory. Note that uniqueness of identity proofs and function extensionality are inherited from Ty (this is immediate for the former; for the latter, use that the outer identity type respects the isomorphism relating Π' and Π).

To obtain the universes in Ty', we must encode the representing object $(\mathsf{Ty}_{\mathcal{C}})_j + (\mathcal{U})_j$ as $\mathsf{El}(r(\mathcal{U}'_j))$ (under the isomorphism $\widehat{\mathcal{C}} \simeq \widehat{\mathcal{C}/1}$) for some $\mathcal{U}'_j \in \mathsf{Ty}'(1)$. We have $V_j \in \mathsf{Ty}_{j+1}(1)$ such that $(\mathsf{Ty}_{\mathcal{C}})_j + (\mathcal{U})_j$ is $\mathsf{El}(V_j)$ (under the isomorphism $\widehat{\mathcal{C}} \simeq \widehat{\mathcal{C}/1}$). So we simply take $\mathcal{U}'_j = s(V_j)$.

2-Functoriality in $\mathcal{C}$ is a straightforward calculation. □

Note that the cwf hierarchy morphism s in the above proof preserves almost all type formers: the only one not preserved is the unit type. Note also that the technique of Proposition 2.16 is constructive only for finitary type formers: were we to add product types of infinite arity or dependent sums of "arity ω (ω^{op}-Reedy limits) to homotopy type theory, then to make c' preserve these using the above approach, we would have to perform an infinite number of case distinctions before deciding on the result of the corresponding type forming operation in Ty' on given inputs, which requires classical logic.

In Subsection 2.6, we will use this modified presheaf model to strengthen conservativity of two-level type theory over homotopy type theory to additionally include conservativity of (T1) and (T2).

None of the other properties (A1) and (T3) to (A6) are generally impacted by the model construction of proposition 2.16.

2.6 Conservativity

Two-level type theory is an extension of homotopy type theory, which forms its inner level. As such, it makes sense to ask if two-level type theory is *conservative* over homotopy type theory.

Here, we take the perspective regarding homotopy type theory and two-level type theory simply as the initial models in their respective categories of models, which are the primary notion. Syntax is treated as notation, that is, merely as a device for working within such models. Expressions of

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

our syntax denoting types and terms are just stand-ins denoting certain derivations. We do not analyse them as raw syntactic objects independently from the associated derivation, although such considerations are of course important for the implementation of proof assistants.

What does conservativity mean under this perspective? We have a forgetful functor $(-)^{\mathsf{i}}$ from the category of models of two-level type theory to the category of models of homotopy type theory. Letting 0_{HoTT} and 0_{2LTT} denote their respective initial objects, we have a unique morphism $0_{\mathsf{HoTT}} \to 0_{\mathsf{2LTT}}{}^{\mathsf{i}}$. This expresses that any derivation in homotopy type theory can also be performed in two-level type theory. For conservativity, we wish to know reversely that any construction of a type or term, or equality of such, in $0_{\mathsf{2LTT}}{}^{\mathsf{i}}$, with given context (and type, in the case of constructions for terms) coming from 0_{HoTT}, can be lifted to 0_{HoTT}.

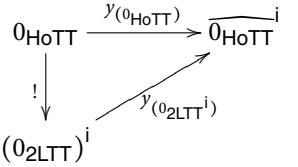
We will employ the following definition of conservativity for a cwf morphism $F : \mathcal{C} \to D$, which is, in some sense, the weakest possible. It essentially states that F reflects inhabitation of terms. This formalises the idea that we can use the language of $\mathcal{D}$ to prove statements in $\mathcal{C}$.

Definition 2.17. A cwf morphism $F : \mathcal{C} \to \mathcal{D}$ is called *conservative* if for all contexts $\Gamma \in \mathcal{C}$ and types $A \in \mathsf{Ty}_{\mathcal{C}}(\Gamma)$ with an element of $\mathsf{Tm}_{\mathcal{D}}(F\Gamma, FA)$, we have an element of $\mathsf{Tm}_{\mathcal{C}}(\Gamma, A)$.

Note that is definition is unrelated to the underlying functor F being conservative, i.e. reflecting isomorphisms. Stronger definitions are of course possible, for example requiring that F acts (split) surjectively or bijectively on terms and types, perhaps up to internal notions of equality in $\mathcal{D}$.

Proposition 2.18. *Two-level type theory is conservative over homotopy type theory. That is, the morphism* $! : 0_{\mathsf{HoTT}} \to (0_{\mathsf{2LTT}})^{\mathsf{i}}$ *is conservative. This stays true when the outer level is extended with any type former validated by the standard presheaf model, such as equality reflection (A5).*

Proof. We apply the construction of Corollary 2.14 to 0_{HoTT} and $0_{\mathsf{2LTT}}{}^{\mathsf{i}}$, obtaining a diagram



commuting up to isomorphism in the 2-category of models of homotopy type theory and weak morphisms. Conservativity of the vertical map now follows immediately from the fact that the Yoneda embedding acts bijectively on terms. □

Using proposition 2.16, we may strengthen the above statement to two-level type theory with injective conversion morphisms that strictly preserve $1/\Sigma/\Pi$-types.

Proposition 2.19. *Two-level type theory with (T1) and (T2) is conservative over homotopy type theory. This stays true when the outer level is extended with any type former validated by the outer level of the modified presheaf model, such as equality reflection (A5).*

Proof. This is a copy of the proof of Proposition 2.18, with the presheaf model of Corollary 2.14 replaced by the modified presheaf model of Proposition 2.16. □

We conjecture a stronger conservativity result: if equality reflection (A5) holds in the outer level, then the actions of the cwf morphism $! : 0_{\mathsf{HoTT}} \to (0_{\mathsf{2LTT}})^{\mathsf{i}}$ on types and terms are bijective (and hence the underlying functor is fully faithful). We believe this result can be obtained using the technique of categorical glueing. A concrete argument has been given by Kovács (2022, Corollary 5.5), seen there as "soundness and stability of staging.

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

2.7 On the possibility of a fibrant replacement

In homotopical models of two-level type theory, outer types in context Γ correspond to arbitrary maps into Γ, whereas inner types correspond to fibrations with base Γ. From this viewpoint, it is natural to ask whether we could extend our theory with a *fibrant replacement* operation, allowing us to replace any outer type by its "closest inner approximation. A syntactic presentation of rules for such a fibrant replacement type former might look as follows:

$$\frac{\Gamma \vdash A\ \mathrm{type}_j}{\Gamma \vdash RA\ \mathrm{type}^{\mathsf{i}}_j}\ \textsc{form-r} \qquad \frac{\Gamma \vdash a : A}{\Gamma \vdash r(a) : RA}\ \textsc{intro-r}$$

$$\frac{\Gamma.RA \vdash P\ \mathrm{type}^{\mathsf{i}}_j \qquad \Gamma.(a:A) \vdash d : P[r(a)]}{\Gamma.RA \vdash \mathsf{elim}^P_R(d) : \mathrm{c}(P)}\ \textsc{elim-r}$$

$$\frac{\Gamma.RA \vdash P\ \mathrm{type}^{\mathsf{i}}_j \qquad \Gamma.(a:A) \vdash d : \mathrm{c}(P[r(a)])}{\Gamma.(a:A) \vdash \mathsf{elim}^P_R(r(a)) \equiv d}\ \textsc{comp-r}$$

Phrased internally, given an outer type A, we get an inner type RA together with a function $r : A \to c(RA)$ with the universal property that, for any inner type X, to define a function $RA \to X$ is to give a function $A \to c(X)$. Note the similarity of the above rules to those of the propositional truncation modality; the only difference is, of course, that R makes types *fibrant* rather than propositional.

A type former along these lines is considered in Boulier and Tabareau (2017), where the authors construct a model structure on a universe of outer types using fibrant replacement.

Unfortunately, the fibrant replacement operation cannot actually be internalised in the above form while still retaining interesting homotopical models. This is shown by the following theorem.

Theorem 2.20. *Assume a fibrant replacement type former R as defined by the rules* FORM-R *to* COMP-R. *Then, the inner level satisfies uniqueness of identity proofs.*

Proof. For an inner type A with $u, v : A$, the internalisation of (7) gives us a canonical map

$$i : \big(\mathrm{c}(u) =_{\mathrm{c}(A)} \mathrm{c}(v)\big) \to \mathrm{c}(u =^{\mathsf{i}}_A v).$$

We claim the following:

$$\Pi_{u,v:A,p:u=^{\mathsf{i}}_A v} R(\Pi_{h:\mathrm{c}(u)=_{\mathrm{c}(A)}\mathrm{c}(v)} \mathrm{c}(\mathrm{c}^{-1}(i(h)) =^{\mathsf{i}} p)). \tag{13}$$

By inner path induction, we can assume $p \equiv \mathsf{refl}^{\mathsf{i}}_u$. Using INTRO-R it remains to show that, for $h : \mathrm{c}(u) = \mathrm{c}(u)$, we have

$$\mathrm{c}(\mathrm{c}^{-1}(i(h)) =^{\mathsf{i}} \mathsf{refl}^{\mathsf{i}}_u). \tag{14}$$

Because of UIP, we can replace h by $\mathsf{refl}_{\mathrm{c}(u)}$, and by observing that i maps the trivial outer equality to the trivial inner equality we get (14).

Our goal is to show that A satisfies UIP. Assume now $u : A$ and $p : u =^{\mathsf{i}} u$. It suffices to show $p =^{\mathsf{i}} \mathsf{refl}^{\mathsf{i}}_u$. This follows from (13), choosing h to be $\mathsf{refl}_{\mathrm{c}(u)}$. □

While homotopical models do have fibrant replacement operations coming from weak factorisation systems, they are usually not stable under base change. This prevents internalisation of this operation in the form of the above rules. That is, we may replace an inner type by an outer type, but this operation is not natural in the context. If one still wishes to expose this operation, one option is to make two-level type theory into a modal type theory extended with a notion of crisp types as in Licata et al. (2018). Then, one can state the above replacement operation *crisply*.

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

2.8 Notational conventions

The rest of the paper does not concern the meta-properties of 2LTT. Instead, we develop some theory internally to 2LTT. As described above, we use the syntax suggested in Subsection 2.1. For notational convenience, we omit applications of El and pretend that we work with Russell-style universes. As it is fairly standard, we also omit universe indices in the style of *typical ambiguity*. Similarly, we will keep the conversion operation c between inner and outer types implicit.

Note that we did not assume a built-in universe of propositions in either level (but cf. (A5)). Instead, we define

$$\mathsf{Prop}^{\mathsf{i}} :\equiv \Sigma\left(X : \mathcal{U}^{\mathsf{i}}\right).\ \Pi_{x,y:X}\left(x =^{\mathsf{i}} y\right) \tag{15}$$

$$\mathsf{Prop} :\equiv \Sigma\ (X : \mathcal{U})\ .\ \Pi_{x,y:X}\left(x = y\right). \tag{16}$$

Most of the time, we work with the outer level, which is why we treat that level as the default; note how the inner type formers are annotated with the symbol $^{\mathsf{i}}$, while the outer do not carry annotations. This also means that, when we say that a diagram commutes, it commutes up to the outer equality type.

For outer types A and B, we can form the type of isomorphisms, written $A \simeq B$. Note that, because of UIP, asking for maps in both directions such that both compositions are pointwise equal to the identity is well-behaved. For inner types A and B, the inner type $A \simeq^{\mathsf{i}} B$ is the usual type of equivalences.

3. Basic Tools: Categories, Fibrations, and Cofibrations

Before we can start working inside two-level type theory, it is helpful to develop some basic theory. As the outer level of the theory is simply a version of MLTT with UIP, we have access to a vast pool of results that are already known. In particular, finite types and the basics of category theory work in the expected way. We will summarise some of that here.

Later on, we will need several notions more specific to two-level type theory. Namely, we are going to define what it means for a function to be a fibration or a cofibration, and for types to be fibrant or cofibrant. These notions will allow us to use the outer level to obtain results that are really about the inner one, without having to explicitly coerce from inner types to outer.

3.1 Preliminaries

Although somewhat trivial, the importance of finite types for our development justifies that we introduce them explicitly. Recall that, in usual type-theoretic terminology, Fin_n is the finite type with n elements. In our development, we will use this notation exclusively to refer to finite types in the outer level. One explicit definition is as the type of natural numbers smaller than n, where the order on natural numbers is defined as usual.

We will say that a type X is *finite* if it is isomorphic to Fin_n, for some n, i.e. if we have $\Sigma\ (n : \mathbb{N})\ .\ X \cong \mathsf{Fin}_n$. The type Fin_n is not to be confused with its inner counterpart $\mathsf{Fin}^{\mathsf{i}}_n$, which exists for $n : \mathbb{N}^{\mathsf{i}}$. Of course, we have a canonical function $\mathsf{Fin}_n \to \mathsf{Fin}^{\mathsf{i}}_n$, where the application of the function $\mathbb{N} \to \mathbb{N}^{\mathsf{i}}$ is kept implicit. In a two-level theory satisfying (A1), this would be an isomorphism, but in general, we do not even assume a function in the other direction.

In the following, will make heavy use of category-theoretic notions. Categories are defined in the usual way, within the outer level of the theory.

Definition 3.1 (category)**.** A *category* $\mathcal{C}$ is given by

- a type $|\mathcal{C}| : \mathcal{U}$ of *objects*;
- for all pairs $x, y : |\mathcal{C}|$, a type $\mathcal{C}(x, y) : \mathcal{U}$ of *arrows* or *morphisms*;

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

- an *identity* arrow $\mathsf{id} : \mathcal{C}(x, x)$ for every object x;
- and a *composition* function $\circ : \mathcal{C}(y, z) \to \mathcal{C}(x, y) \to \mathcal{C}(x, z)$ for all objects x, y, z;
- such that the usual categorical laws holds, that is, we have $f \circ \mathsf{id} = f$ and $\mathsf{id} \circ f = f$, as well as $h \circ (g \circ f) = (h \circ g) \circ f$.

Given objects x and y of a category $\mathcal{C}$, we also write $f : x \to y$ for a morphism from x to y, that is, an element of the type $\mathcal{C}(x, y)$. It will always be clear from the context if x and y are types or objects of a category, so that there is no confusion with the function type former. (In the case of a category of types, the two notions agree.)

Readers familiar with the chapter on category theory in the HoTT book (The Univalent Foundations Program, 2013) (and Ahrens et al. 2015) will note that our definition is exactly the same as that of *precategories* there. Of course, since our outer theory validates UIP, and therefore every type is a set, we do not need to explicitly add a truncation condition on homsets.

A canonical example of a category is the category of types, whose objects are the types in a given universe $\mathcal{U}$, and whose morphisms are functions. By a slight abuse of notation, we will simply write $\mathcal{U}$ to denote this category. Analogously, if $\mathcal{C}$ is a category, we allow ourselves to denote the type of objects by $\mathcal{C}$ itself.

The usual theory of categories can be reproduced in the context of our categories (as long as we stay constructive). We write $[\mathcal{C}, \mathcal{D}]$ for the *functor category* of categories $\mathcal{C}$ and $\mathcal{D}$, with the type of *natural transformations* from a functor F to a functor G also written $\mathbf{Nat}(\mathcal{C}, \mathcal{D})$. Functors and natural transformations form the objects and morphisms of a (larger) category of categories. We have the usual concepts such as *limits* and *adjunctions* and can prove all their usual properties, for example that limits (if they exist) are unique up to isomorphism.

Remark 3.2. We will not indulge in the exercise of replicating the whole of category theory in our outer level, and simply assume, on the empirical evidence provided by several existing developments in the major implementations of type theory, like the aforementioned Agda, Coq and Lean, that doing so is simply a matter of diligence and patience, and it ultimately should present no mathematical difficulties.

Remark 3.3. Despite the above remark, it is perhaps appropriate to add a small explanation of how one might reasonably deal with "size issues in a formal development of category theory within the outer theory.

When translating category-theoretical statements originally formulated in the metalanguage of set theory, one is posed with the question of what precise type-theoretic meaning to give to the term "small".

As most incarnations of type theory, including our outer level, provide the user with an infinite tower of universes, it feels unnecessarily restrictive to constrain a general term like "small" to a predetermined choice of a universe level.

For this reason, we will not make such a choice, and simply continue the tradition of writing "small" for a type that resides in a universe which is one step below a "default" unspecified universe level. This makes it clear that the absolute level that certain constructions happen to end in is not particularly important, rather what we have to pay attention to are the differences in *relative* size.

Note that the universe $\mathcal{U}^{\mathsf{i}}$ of inner types also forms a category, although it is not as well behaved as $\mathcal{U}$. For example, it does not have pullbacks (but see part (i) of Lemma 3.10).

3.2 Fibrant types

Definition 3.4 ((trivially) fibrant type)**.** A type $A : \mathcal{U}$ is *fibrant* if it is isomorphic to an inner type $A' : \mathcal{U}^{\mathsf{i}}$. It is *trivially fibrant* if the inner type A' is furthermore contractible.

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

Note that fibrancy (and similarly trivial fibrancy) is a proof-relevant notion, in that being fibrant is not a proposition (in the sense of having at most one element). A fibrant type A carries with it a choice of an inner type A' and an isomorphism $f : A \cong A'$ relating its coercion to a type to the original type A (and a trivially fibrant type furthermore carries an element witnessing inner contractibility of A'). This should be kept in mind in our use of language when we use being fibrant as an adjective. For example, when we say that A is fibrant exactly if B is fibrant, what we mean is functions back and forth between the types witnessing fibrancy of A and B.

Generally speaking, our use of informal language in the outer level follows the mantra of "propositions as types". Thus, similar conventions as established in the previous paragraph apply to notions such as fibrations and cofibrations defined below (and their Reedy variants considered later).

We write $\mathcal{U}_{\mathsf{fib}}$ for the type of fibrant types in $\mathcal{U}$. Note that it is itself not generally fibrant (although it is in some of the intended models such as simplicial sets). We let $\mathcal{U}_{\mathsf{fib}}$ inherit the category structure of $\mathcal{U}$. Note that the functor $\mathcal{U}_{\mathsf{fib}} \to \mathcal{U}$ is the replacement of the coercion functor $\mathcal{U}^{\mathsf{i}} \to \mathcal{U}$ by an isofibration. In particular, fibrant types are closed under isomorphism. We allow ourselves to implicitly coerce from $\mathcal{U}_{\mathsf{fib}}$ to $\mathcal{U}$.

Lemma 3.5. *Fibrant types enjoy the following closure properties.*

(i) The unit type is trivially fibrant.
(ii) Given $A : \mathcal{U}_{\mathsf{fib}}$ and $B : A \to \mathcal{U}_{\mathsf{fib}}$, then $\Sigma_A B$ is fibrant. It is trivially fibrant if A and B are valued in trivially fibrant types.
(iii) Given $A : \mathcal{U}_{\mathsf{fib}}$ and $B : A \to \mathcal{U}_{\mathsf{fib}}$, then $\Pi_A B$ is fibrant. It is trivially fibrant if B is valued in trivially fibrant types.

Proof. Recall from Lemma 2.11 that the coercion map $\mathcal{U}^{\mathsf{i}} \to \mathcal{U}$ preserves unit type, dependent sums and dependent products up to canonical isomorphism.

We do the case of dependent products in detail. Note that we can internalise the inner dependent product as an operation $\Pi^{\mathsf{i}} : (\Sigma\,(A : \mathcal{U}^{\mathsf{i}})\,.\,(A \to \mathcal{U}^{\mathsf{i}})) \to \mathcal{U}^{\mathsf{i}}$ and that given $A : \mathcal{U}^{\mathsf{i}}$ and $B : A \to \mathcal{U}^{\mathsf{i}}$, we have a comparison isomorphism $\Pi^{\mathsf{i}}(A, B) \cong \Pi_{a:A} B(a)$. This shows that the (outer) dependent product of $A : \mathcal{U}^{\mathsf{i}}$ and $B : A \to \mathcal{U}^{\mathsf{i}}$ is fibrant. Isomorphic families have isomorphic dependent product, generalising the statement to $B : A \to \mathcal{U}_{\mathsf{fib}}$. Finally, reindexing a family along an isomorphism gives isomorphic dependent product, generalising the statement to $A : \mathcal{U}_{\mathsf{fib}}$. □

Definition 3.6 (equivalence)**.** A function $f : A \to B$ between fibrant types with underlying inner types A' and B' is an *equivalence* if the corresponding inner function $A' \to^{\mathsf{i}} B'$ is an equivalence (in the sense of homotopy type theory and using the inner identity type).

Note that the notion of f being an equivalence in the above definition formally depends on the witnesses of fibrancy of A and B. Different witnesses yield non-isomorphic types of f being an equivalence. However, they will still be logically equivalent (in the sense of maps back and forth) and are in fact propositionally fibrant (meaning that their underlying inner type is a proposition in the sense of homotopy type theory). As per our convention, we can thus say that the notion of equivalence is invariant under isomorphism.

Properties of equivalences are directly lifted from the inner level. For example, equivalences satisfy 2-out-of-6.

3.3 Fibrations

Recall that the *fibre* $p^{-1}(x)$ of a function $p : Y \to X$ over $x : X$ is given by the type $\Sigma\,(y : Y)\,.\,p(e) = b$. This is not to be confused with the notion of homotopy fibre, which is only available for inner types

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

(using the inner identity type). More generally, we say that a type A is the fibre of p over x if A arises as a pullback of p along $1 \to X$, in which case it is isomorphic to $p^{-1}(x)$.

Definition 3.7 ((trivial) fibration)**.** A function $p : Y \to X$ is a *(trivial) fibration* if its fibres are (trivially) fibrant.

In the running text and diagrams, a fibration $Y \to X$ is denoted $Y \twoheadrightarrow X$.

Remark 3.8.

(i) Note that $A \to 1$ is a (trivial) fibration exactly if A is (trivially) fibrant. This matches the terminology in abstract homotopy theory, where the notion of fibration is taken as primitive and fibrant objects are the special case of maps into the terminal object. Note also that every trivial fibration is a fibration.

(ii) In the models of simplicial sets and cubical sets, our fibration coincide with the fibrations in the sense of the model. The reader should be aware that our internal pointwise definition of fibration, talking just about the fibres of a map, does not correspond to an external pointwise or fibrewise property. For example, a map $Y \to X$ in simplicial sets is not necessarily a Kan fibration if the fibre Y_x of every point $x \in X_0$ is a Kan complex. Rather, the internal quantification over elements of the base type X externally becomes a quantification over $[n] : \Delta$ with $x \in X_n$. The witness of fibrancy is given by a family of elements $\sigma([n], x) \in \mathcal{U}_n$, natural in $[n]$, where $\mathcal{U}$ is the universe of Kan fibrations.

Since isomorphic maps have isomorphic fibres and the notion of (trivial) fibrancy is invariant under isomorphism, the notion of (trivial) fibration is invariant under isomorphism as well. From this, the following lemma is an immediate consequence of the definitions.

Lemma 3.9. *The following are equivalent for a function $p : Y \to X$:*

(i) p is a fibration,
(ii) p is isomorphic over X to $\Sigma_X Y' \to X$ for some $Y' : X \to \mathcal{U}_{\mathsf{fib}}$,
(iii) p is isomorphic over X to $\Sigma_X Y' \to X$ for some $Y' : X \to \mathcal{U}^{\mathsf{i}}$,

and if X is fibrant with underlying inner type $X' : \mathcal{U}^{\mathsf{i}}$:

(iv) p is isomorphic to the map $\Sigma^{\mathsf{i}}_{X'} Y' \to X'$ corresponding to the inner dependent projection $\Sigma^{\mathsf{i}}_{X'} Y' \to^{\mathsf{i}} X'$ for some $Y' : X' \to \mathcal{U}^{\mathsf{i}}$.

We have analogous equivalences for trivial fibrations with $\mathcal{U}^{\mathsf{i}}$ replaced by the type of inner contractible types and $\mathcal{U}_{\mathsf{fib}}$ replaced by the type of trivially fibrant types.

Fibrations and trivial fibrations enjoy a number of closure properties. We start with the following easy collection.

Lemma 3.10. *(Trivial) fibrations are closed under:*

(i) pullbacks,
(ii) finite compositions.
(iii) finite products,

Proof. For part (i), note that the fibres of a pullback of a map f are also fibres of f.

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

For part (ii), the nullary and binary case reduce to parts (i) and (ii) of Lemma 3.5, respectively. The general case follows by induction.

Part (iii) is a consequence of (i) and (ii). □

Lemma 3.11. *Every trivial fibration has a section.*

Proof. By Lemma 3.9, we can assume that the given trivial fibration is of the form $\Sigma\,(b:B)\,.\,X(b) \to B$ for a type B and a family $X : B \to \mathcal{U}^{\mathsf{i}}$ of contractible fibrant types over B. We obtain a section $\Pi_{b:B} X(b)$ by extracting the centre of contraction from the contractibility proof of $X(b)$. □

Lemma 3.12. *Let $p : E \to B$ be a map between fibrant types. Then, p is a trivial fibration if and only if it is both a fibration and an equivalence.*

Proof. Let p be a fibration. By invariance under isomorphism and Lemma 3.9, we can assume that p is of the form $\Sigma\,(b:B)\,.\,X(b) \to B$ for an inner type $B : \mathcal{U}^{\mathsf{i}}$ and a family $X : B \to \mathcal{U}^{\mathsf{i}}$ of inner types over B, with identity isomorphisms witnessing fibrancy of B and E. Then, p is a trivial fibration exactly if $X(b)$ is contractible for all $b : B$. Note that dependent projection $\Sigma\,(b:B)\,.\,X(b) \to B$ corresponds to the inner dependent projection $\Sigma^{\mathsf{i}}\,(b:B)\,.\,X(b) \to^{\mathsf{i}} B$ under the isomorphisms relating inner and outer dependent sums and function types. Thus, p is an equivalence exactly if this inner dependent projection is an equivalence in the inner level, where from homotopy type theory we know to be equivalent to its fibres $X(b)$ being contractible for all $b : B$. □

3.4 Cofibrations

Cofibrations and cofibrant types are further technical concepts which help us to study the actual objects of interest, i.e. fibrant types. We will see their usefulness later in this article, but let us in addition try to give some motivation here. If B is a fibrant type, then so are $B \times B$ and $B \times B \times B$. More generally, if we work in HoTT and fix any natural number n in the meta-theory, we can consider the n-fold product of B. In our setting, this corresponds to the function type $\mathsf{Fin}_n \to B$. It is important to note that, while $(\mathsf{Fin}_2 \to B)$ is isomorphic to $B \times B$, this and analogous isomorphisms do in general not hold if we use the inner version $\mathsf{Fin}^{\mathsf{i}}_2$ instead; we will only get the weaker notion of an inner equivalence. While $\mathsf{Fin}_n \to B$ is fibrant, this is not directly given by the rules of 2LTT and instead requires a proof (see Lemma 3.25). This, we hope, makes it plausible that it is useful to have a notion of cofibrant types, which we want to be those that exponentiation with preserves fibrancy, and it is not too far-fetched that we also want an analogous notion for functions. In fact, one might expect that *any* setting which allows to reason about HoTT externally in such a way benefits from a notion of cofibration. For example, it is worth comparing the characterisation in Remark 3.14(i) with the *extension types* by Riehl and Shulman (2017).

To define and reason about cofibrations (as well as Reedy cofibrations later on), we make use of the theory of *Leibniz constructions* established in Riehl and Verity (2014). Given a bifunctor $F : \mathcal{C} \times \mathcal{D} \to \mathcal{E}$, the *Leibniz action* $\widehat{F}(f,g)$ of F on maps $f : A \to B$ in $\mathcal{C}$ and $g : C \to D$ in $\mathcal{D}$ is the induced map from the pushout corner in the square

$$\begin{array}{ccc} F(A,C) & \longrightarrow & F(A,D) \\ \downarrow & & \downarrow \\ F(B,C) & \longrightarrow & F(B,D), \end{array}$$

i.e. the map

$$F(A,D) +_{F(A,C)} F(B,C) \xrightarrow{\widehat{F}(f,g)} F(B,D),$$

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

assuming that this pushout exists. If $\mathcal{E}$ has all pushouts, this gives rise to a bifunctor $\widehat{F} : \mathcal{C}^{\to} \times \mathcal{D}^{\to} \to \mathcal{E}^{\to}$, the Leibniz construction of F.

In a category $\mathcal{C}$ with finite products, the Leibniz action of the product functor $(-) \times (-) : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ in a category $\mathcal{C}$ is called the *pushout product*. For $\mathcal{C}$ Cartesian closed, the Leibniz action of the exponential functor $\exp^{\mathrm{op}} : \mathcal{C} \times \mathcal{C}^{\mathrm{op}} \to \mathcal{C}^{\mathrm{op}}$ is called the *pullback exponential*. Note the dualised functor signature we have given exp here (as opposed to $\exp : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{C}$). This causes the pushout (in $\mathcal{C}^{\mathrm{op}}$) of the Leibniz construction to become a pullback in $\mathcal{C}$, explaining the naming.

The category $\mathcal{U}$ of types is Cartesian closed and has pullbacks; thus, we have the pullback exponential bifunctor $\widehat{\exp} : (\mathcal{U}^{\mathrm{op}})^{\to} \times \mathcal{U}^{\to} \to \mathcal{U}^{\to}$, sending functions $f : A \to B$ and $p : Y \to X$ to the function

$$(B \to Y) \xrightarrow{\widehat{\exp}(f,p)} (B \to X) \times_{A \to X} (A \to Y).$$

Definition 3.13. A function $f : A \to B$ between types is:

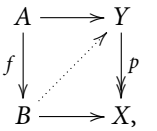
- a *cofibration* if $\widehat{\exp}(f, -)$ preserves fibrations and trivial fibrations,
- a *trivial cofibration* if $\widehat{\exp}(f, -)$ sends fibrations to trivial fibrations.

A type B is *(trivially) cofibrant* if the function $0 \to B$ is a (trivial) cofibration.

We thank Mike Shulman for pointing out that we were missing the condition on trivial fibrations for cofibration in an earlier version of this definition.

Remark 3.14.

(i) Unfolding the above definition, we obtain the following phrasing. A function $f : A \to B$ is a cofibration exactly if for all fibrations $p : Y \twoheadrightarrow X$ and commuting squares



the type of diagonal fillers (indicated by the dotted arrow) is fibrant and trivially fibrant whenever p is a trivial fibration.

(ii) Analogously to (i), f is a trivial cofibration exactly if, for all fibrations p as above, the type of diagonal fillers is trivially fibrant.

(iii) The notions of (trivial) cofibration and (trivially) cofibrant type are invariant under isomorphism.

(iv) Since trivial fibrations are fibrations, trivial cofibrations are cofibrations.

(v) A type B is cofibrant exactly if the representable functor $\mathcal{U}(B, -)$ preserves fibrations and trivial fibrations.

Remark 3.15. The conditions for (trivial) cofibrations given by Definition 3.13 are reminiscent of the pushout product axiom of Cartesian closed model categories. In fact, they correspond exactly to the dual phrasing of this axiom in terms of pullback exponentials. This is the intuition behind our naming scheme.

In the simplicial set model, our notion of (trivial) cofibration (interpreted in the empty context) coincides with the (trivial) cofibrations of the Kan model structure. This can be seen as follows. The Kan model structure is Cartesian closed; hence, a (trivial) cofibration of the model structure is a (trivial) cofibration in our sense. Reversely, let $f : A \to B$ be a cofibration in our sense. In particular, Leibniz exponential with f preserves trivial fibrations. Then, f lifts against trivial fibrations

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

by Lemma 3.11, i.e. is a cofibration of the model structure. A similar argument applies to trivial cofibrations, and our reasoning is not specific to simplicial sets but applies to any cartesian closed model structure.

One can ask for a local version of this correspondence. Given a simplicial set Γ, the above argument in the slice over Γ shows that the (trivial) cofibrations of the model structure over Γ include all of our (trivial) cofibrations in context Γ. However, the reverse inclusions fails as the slices of the Kan model structure are not Cartesian closed. For example, taking $\Gamma = \Delta^1$, the map $\{0\} \to \Delta^1$ (sitting over Δ^1 via the identity) is an example of a trivial cofibration of the model structure that is not even a cofibration in context Γ in our sense. For if it were, pullback exponentials over Δ^1 with $\{0\} \to \Delta^1$ would preserve fibrations, which, by the adjunction between product and exponential, would imply that pushout product over Δ^1 with $\{0\} \to \Delta^1$ preserves trivial cofibrations. However, the pushout product over Δ^1 of $\{0\} \to \Delta^1$ and $\{1\} \to \Delta^1$ is simply their union $\partial\Delta^1 \to \Delta^1$, which is not a trivial cofibration.

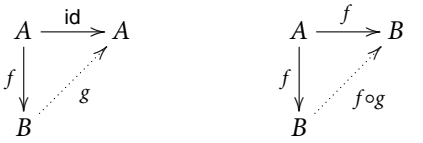
Remark 3.16. Our notions of (trivial) cofibrations and (trivial) fibrations do not form weak factorisation systems. One may be tempted to consider the collection of maps with the left lifting property with respect to (trivial) fibrations. This notion does not capture the intuition described at the beginning of Subsection 3.4, and we instead call such maps *anodyne* (cf. Definition 4.11).

Within fibrant types, trivial cofibrations can be characterised as cofibrations that are equivalences.

Lemma 3.17. *Let $f : A \to B$ be a function between fibrant types. Then, f is a trivial cofibration if and only if it is both a cofibration and an equivalence.*

Proof. By part Remark (iv) of Remark 3.14, we can assume that f is a cofibration and prove that it is a trivial cofibration if and only if it is an equivalence.

If f is a trivial cofibration, then, for all fibrant types X, the restriction map $(B \to X) \to (A \to X)$ is an equivalence. A Yoneda-like argument implies that f is an equivalence in this case. In detail, the characterisation (ii) of Remark 3.14 implies that the type of dotted diagonal fillers is trivially fibrant in each of the following two diagrams:



As indicated, we call g the unique morphism $B \to A$ that makes the left triangle commute. In the right triangle, observe that $f \circ g$ and id both make the triangle commute, and the desired result follows by uniqueness.

Conversely, let $p : Y \to X$ be a fibration. Then, the corresponding pullback-exponential fibration

$$(B \to Y) \to (B \to X) \times_{(A \to X)} (A \to Y)$$

is an equivalence by preservation of equivalences under exponentiation with a fixed base and 2-out-of-3, hence a trivial fibration by Lemma 3.12. Thus, f is a trivial cofibration. □

The following key lemma helps us characterise cofibrations.

Lemma 3.18. *For any function $f : A \to B$:*

(i) f is a cofibration exactly if for any family $Y' : B \to \mathcal{U}_{\mathsf{fib}}$ of (trivially) fibrant types, the induced map

$$\Pi_B Y' \to \Pi_A (Y' \circ f)$$

is a (trivial) fibration.

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

(ii) f is a trivial cofibration exactly if for any family $Y' : B \to \mathcal{U}_{\mathsf{fib}}$ of fibrant types, the induced map

$$\Pi_B Y' \to \Pi_A (Y' \circ f)$$

is a trivial fibration.

Before giving its proof, let us recall the following characterisation of dependent functions in terms of non-dependent functions into a type of pairs where the first component is fixed.

Lemma 3.19. *For a function $f : A \to B$ and a family $X : B \to \mathcal{U}$, the following diagram is a pullback:*

$$\begin{array}{ccc} \Pi_A (X \circ f) & \xrightarrow{g \mapsto (\lambda a.\, (f(a), g(a)))} & (A \to \Sigma_B X) \\ \downarrow & & \downarrow{\scriptstyle \pi_1 \circ -} \\ \mathbf{1} & \xrightarrow{\quad f \quad} & (A \to B) \end{array}$$

Proof. By direct calculation, the pullback is $\Sigma\,(g : A \to \Sigma_B X)\,.\,(\pi_1 \circ g = f)$, which is indeed isomorphic to $\Pi_A (X \circ f)$. □

Proof of Lemma 3.18. Let $p : Y \to X$ be a fibration. It is isomorphic to $\Sigma_X Y' \to X$ for some family $Y' : X \to \mathcal{U}_{\mathsf{fib}}$ of fibrant types. Given $v : B \to X$, we have the following diagram:

$$\begin{array}{ccccc} \Pi_B (Y' \circ v) & \longrightarrow & (B \to Y) & & \\ \downarrow & & \downarrow{\scriptstyle \widehat{\exp}(f,p)} & & \\ \Pi_A (Y' \circ v \circ f) & \longrightarrow & (B \to X) \times_{(A \to X)} (A \to Y) & \longrightarrow & (A \to Y) \\ \downarrow & & \downarrow & & \downarrow{\scriptstyle p \circ -} \\ 1 & \xrightarrow{\quad v \quad} & (B \to X) & \xrightarrow{\quad - \circ f \quad} & (A \to X). \end{array}$$

The right bottom square is a pullback by construction. The composite bottom square and the composite left square are pullbacks by Lemma 3.19. By pullback pasting, the top left square is a pullback.

Note that a map over a type Z is a (trivial) fibration exactly if all its pullbacks along maps $1 \to Z$ are (trivial) fibrations. It follows that $\widehat{\exp}(f, p)$ is a (trivial) fibration exactly if $\Pi_B (Y' \circ v) \to \Pi_A (Y' \circ v \circ f)$ is a (trivial) fibration for all v. This shows the desired equivalences: going forward, we put $X = B$ and $v = \mathsf{id}_B$; going backward, we use the given condition for the family $Y' \circ v$ for all v. □

By setting $A = 0$ in Lemma 3.18, we obtain the following important special case.

Corollary 3.20. *For any type B:*

(i) B is cofibrant exactly if for any family $Y' : B \to \mathcal{U}_{\mathsf{fib}}$ of (trivially) fibrant types, the type $\Pi_B Y'$ is (trivially) fibrant,
(ii) B is trivially cofibrant exactly if for any family $Y' : B \to \mathcal{U}_{\mathsf{fib}}$ of fibrant types, the type $\Pi_B Y'$ is trivially fibrant.

In the following, we make use of standard properties of the Leibniz calculus (most of which are established at a general level in Riehl and Verity 2014). They allow us to infer closure properties of (trivial) cofibrations from corresponding closure properties of (trivial) fibrations established in Lemma 3.10.

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

Lemma 3.21. *(Trivial) cofibrations are closed under:*

(i) pushouts (whenever they exist),
(ii) finite compositions,
(iii) finite coproducts.

Recall that $\mathcal{U}$ does not possess pushouts in general. Part (i) does not establish the existence of pushouts, but merely asserts that any pushout of a (trivial) cofibration is also one.

Proof of Lemma 3.21. For part (i), recall from Riehl and Verity (2014) that the functorial action of the pullback exponential in its first argument sends morphisms of arrows that are pushouts to morphisms of arrows that are pullbacks. Thus, the claim follows from part (i) of Lemma 3.10.

In detail, consider a pushout

$$\begin{array}{ccc} A & \longrightarrow & C \\ {\scriptstyle f}\downarrow & \ulcorner & \downarrow{\scriptstyle g} \\ B & \longrightarrow & D. \end{array}$$

Assuming that f is a cofibration, we wish to show that g is a cofibration (the case of trivial cofibrations is analogous). Let $Y \twoheadrightarrow X$ be a (trivial) fibration. From the universal property of the pushout and pullback pasting, we get that the square

$$\begin{array}{ccc} (D \to Y) & \longrightarrow & (B \to Y) \\ \downarrow & \lrcorner & \downarrow \\ (C \to Y) \times_{(C \to X)} (D \to X) & \longrightarrow & (A \to Y) \times_{(A \to X)} (B \to X) \end{array}$$

is a pullback. Since $f : A \to B$ is a cofibration, the right vertical map is a (trivial) fibration, hence so is the left vertical map by part (i) of Lemma 3.10. This makes g a cofibration. An alternative argument uses Lemma 3.18 and the dependent version of the universal property of pushouts.

For part (ii), recall from Riehl and Verity (2014) that the pullback exponential of a map p with a finite composition is a finite composition of pullbacks of pullback exponentials of p with the individual factors. Thus, the claim follows from parts (i) and (ii) of Lemma 3.10.

For an alternative proof, we can make use of the characterisation of (trivial) cofibrations given by Lemma 3.18. Let us only deal with the case of a binary composition

$$A \xrightarrow{f} B \xrightarrow{g} C$$

of cofibrations. Given a family $X : C \to \mathcal{U}_{\mathsf{fib}}$ of (trivially) fibrant types, we have to show that $\Pi_C X \to \Pi_A (X \circ g \circ f)$ is a (trivial) fibration. This writes as the composite

$$\Pi_C X \xrightarrow{-\circ g} \Pi_B (X \circ g) \xrightarrow{-\circ f} \Pi_A (X \circ g \circ f),$$

whose factors are (trivial) fibrations by assumption. The statements then follows from closure of fibrations under composition, i.e. part (ii) of Lemma 3.10.

For part (iii), recall that the exponential bifunctor sends colimits in its exponent to limits. By Riehl and Verity (2014), this property transfers to the pullback exponential bifunctor. In particular, for a fixed function p, the operation $\widehat{\exp}(-, p)$ sends coproducts (in $\mathcal{U}^{\to}$) to products. The claim then reduces to part (iii) of Lemma 3.10. Alternatively, it is a consequence of (i) and (ii) in the same fashion as for Lemma 3.10. □

Corollary 3.22. *For any type A and cofibrant type B, the inclusion $A \to A + B$ is a cofibration.*

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

Proof. Since $\mathcal{U}$ has binary coproducts, it has pushouts along $0 \to A$. Instantiating part (i) of Lemma 3.21 to the pushout of the cofibration $0 \to B$ along $0 \to A$, we obtain the claim.

Alternatively, we could instantiate part (iii) of Lemma 3.21 to the cofibrations id_A (using the nullary case of part (ii) of Lemma 3.21) and $0 \to B$. □

Note that the map $0 \to 1$ is the unit for the pushout product. Thus, the pullback exponential with $0 \to 1$ is equivalent to the identity functor. Thus, 1 is the simplest example a cofibrant type. This can be seen as the nullary version of the following statement.

Lemma 3.23. *Any pushout product $f \mathbin{\widehat{\times}} g$ (if it exists) of a cofibration f with a cofibration g is again a cofibration. Furthermore, $f \mathbin{\widehat{\times}} g$ is a trivial cofibration if one of f and g is a trivial cofibration.*

Proof. Recall that binary product and exponential form a two-variable adjunction. As explained in Riehl and Verity (2014), this lifts to Leibniz constructions. In particular, given a function p, we have isomorphisms

$$\widehat{\exp}(f \mathbin{\widehat{\times}} g, p) \cong \widehat{\exp}(g, \widehat{\exp}(f, p)) \cong \widehat{\exp}(f, \widehat{\exp}(g, p)).$$

With this, the claim reduces to the definition of (trivial) cofibrations. □

Corollary 3.24. *Cofibrations are closed under products with cofibrant types.*

We are now able to characterise a large class of cofibrant types.

Lemma 3.25.

(i) Fibrant types are cofibrant.
(ii) Cofibrant types are closed under finite coproducts.
(iii) Finite types are cofibrant.
(iv) Given a cofibrant type A and a family $B : A \to \mathcal{U}$ of cofibrant types, then $\Sigma_A B$ is cofibrant.
(v) Cofibrant types are closed under finite products.

Proof. Part (i) is immediate from Corollary 3.20 since (trivially) fibrant types are closed under exponentiating with fibrant types by part Lemma (iii) of Lemma 3.5.

Part (ii) is an instance of part (iii) of Lemma 3.21.

Part (iii) is a corollary of (ii) and cofibrancy of 1.

For part (iv), we work with the characterisation of cofibrancy given by Corollary 3.20. Let $X : (\Sigma_A B) \to \mathcal{U}_{\mathsf{fib}}$ be a (trivially) fibrant family. We have to show that $\Pi_{\Sigma_{AB} X}$ is (trivially) fibrant. This type is strictly isomorphic to $\Pi_{a:A} \Pi_{b:B(a)} X(a, b)$ and (trivially) fibrant by two applications of Corollary 3.20.

For part (v), the nullary case is given by cofibrancy of 1. With this, the general situation reduces to the case of binary products. This is the non-dependent instance of (iv). Alternatively, it is the instance of Lemma 3.23 for maps $0 \to A$ and $0 \to B$ (whose pushout product is $0 \to A \times B$). □

We note that part (iv) of Lemma 3.25 has a relative generalisation: given cofibrant A and a family $f_a : C_a \to D_a$ of cofibrations indexed over $a : A$, the functorial action $\Sigma_A C \to \Sigma_A D$ of Σ_A on f is a cofibration. This is proved using Lemma 3.19 instead of Lemma 3.18. In principle, one could generalise further to a 'Leibniz dependent sum' of a cofibration $A \to B$ with a family of cofibrations indexed over B, but we have no need for that here.

4. Reedy fibrant diagrams

4.1 Motivation

The connection between dependency structures of types and type families and what is known as *Reedy fibrancy* (Reedy, 1974) is well-known in the type theory community, although people have

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

used different names for this concept; for example, Makkai's *FOLDS* (Makkai, 1995) builds on the same insights. Let us explain the idea with the help of a very concrete example. The category $\Delta_+^{<3}$ is the category with three objects $[0]$, $[1]$, $[2]$ ('vertices', 'edges', 'triangles'), and morphisms generated by $s, t : [0] \to [1]$ ('source', 'target') and $u, v, w : [1] \to [2]$, subject to the following three equations:

$$[0] \underset{t}{\overset{s}{\rightrightarrows}} [1] \underset{w}{\overset{u}{\overset{v}{\rightrightarrows}}} [2] \qquad \begin{aligned} u \circ s &= v \circ s \\ u \circ t &= w \circ s \\ v \circ t &= w \circ t \end{aligned}$$

The type of functors $F : (\Delta_+^{<3})^{\mathrm{op}} \to \mathcal{U}$ is defined in the usual way and is isomorphic to the type of 9-tuples $(X_0, X_1, X_2, f_s, f_t, f_u, f_v, f_w, e_0, e_1, e_2)$ where:

$$\begin{array}{lll} X_0 : \mathcal{U} & f_s : X_1 \to X_0 & e_0 : f_s \circ f_u = f_s \circ f_v \\ X_1 : \mathcal{U} & f_t : X_1 \to X_0 & e_1 : f_t \circ f_u = f_s \circ f_w \\ X_2 : \mathcal{U} & f_u : X_2 \to X_1 & e_2 : f_t \circ f_v = f_t \circ f_w \\ & f_v : X_2 \to X_1 & \\ & f_w : X_2 \to X_1 & \end{array}$$

While this type of 9-tuples is unfortunately not fibrant, we can identify a class of diagrams – the *Reedy fibrant* ones (Definition 4.6) – whose type will turn out to be fibrant, and such that every functor is equivalent a Reedy fibrant one for a suitably weak notion of equivalence.

We will illustrate the idea using $(\Delta_+^{<3})^{\mathrm{op}}$. We will regard this category as a presentation of a dependency structure of types and type families. That is, instead of asking for source and target map from the type of edges to the type of vertices, we let the type of edges be indexed twice over the type of vertices, and similarly for triangles over edges. This leads to a type of triples (A_0, A_1, A_2) with the following components:

$$\begin{aligned} &A_0 : \mathcal{U} \\ &A_1 : A_0 \to A_0 \to \mathcal{U} \\ &A_2 : \Pi(a, b, c : A_0).A_1\ a\ b \to A_1\ b\ c \to A_1\ a\ c \to \mathcal{U} \end{aligned}$$

The type of triples (A_0, A_1, A_2) encodes the so-called Reedy fibrant functors from $(\Delta_+^{<3})^{\mathrm{op}}$ to $\mathcal{U}$ and can be formulated in homotopy type theory without the notion of strict equality; or in other words, it can be formulated in our inner theory without using equality from the outer theory.

Every triple (A_0, A_1, A_2) determines a functor, but, unfortunately, it is not the case that the latter type of triples is isomorphic (as a type) to the former type of tuples that used strict equality. Instead, we will show later that for every functor there exists a Reedy fibrant one that is equivalent to it in a weaker sense (Corollary 4.27) – its *Reedy fibrant replacement*.

Before this, we introduce the basic required categorical infrastructure within the language of 2LTT. This is a first demonstration of results which can be expressed in full in 2LTT although they would require meta-theoretic reasoning in more traditional approaches. We define the notion of Reedy fibration and show that Reedy fibrant diagrams $\mathcal{C} \to \mathcal{U}$ have limits in $\mathcal{U}$ for a finite inverse category $\mathcal{C}$. This is an internalised version of Shulman's results which can be found in Shulman (2015*b*). In the second half of this section, we describe how to construct Reedy fibrant replacements, discuss classifiers for Reedy fibrations (a special case of which would be the type of semisimplicial types restricted to level n), and finally, we develop the theory of exponentials of diagrams.

4.2 Inverse categories

In the following, when considering categories of diagrams, we will mostly focus on diagrams over *inverse* categories. In contrast with the setup leading, for instance, to the Reedy model structure

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

on simplicial presheaves over a Reedy category, in our setting it is necessary to impose the further restriction on the index category to be 'one way'. We can either express this in terms of *direct* categories and contravariant functors, or *inverse* categories and covariant functors. We have arbitrarily picked the second representation. For readers familiar with Reedy categories, inverse categories are simply the special case of those obtained by requiring that there be no non-trivial positive arrows. For the sake of illustrating how to encode the details of the definition in two-level type theory, however, we choose to give an explicit definition.

Consider the category ω. Its objects are the natural numbers $\mathbb{N}$ and its morphisms are given by

$$\omega(n, m) :\equiv m \leq n,$$

with $\leq : \mathbb{N} \to \mathbb{N} \to \mathsf{Prop}$ defined in the usual way. This category is in fact a poset. In general, Reedy categories can be defined in terms of arbitrary ordinals, but ω is enough for our purposes. This lets us evade the question of how to encode more general ordinals in the theory.

Definition 4.1 (inverse category)**.** We say that a category $\mathcal{C}$ is *inverse* if there is a functor $\varphi : \mathcal{C}^{\mathrm{op}} \to \omega$ reflecting identities; i.e. given $f : x \to y$ with $\varphi(x) = \varphi(y)$, then f is an identity, i.e. $(x, y, f) = (x, x, \mathsf{id}_x)$ as elements of the type of arrows of $\mathcal{C}$. We call φ the *rank functor* and say $x : |\mathcal{C}|$ has rank $\varphi(x)$. We write $\mathcal{C}^{<n}$ for the full subcategory of $\mathcal{C}$ consisting of objects of rank less than n.

Given category $\mathcal{C}$ and a functor $F : \mathcal{C} \to \mathcal{U}$, the *category of elements* F, denoted $F/\mathcal{C}$ in analogy with the notation for coslices, is defined as usual via the Grothendieck construction. In particular, objects are pairs (a, x) where $a : |\mathcal{C}|$ and $x : F(a)$ and morphisms from (a, x) to (b, y) are maps $f : a \to b$ such that $Ff(x) = y$. We have a forgetful functor $F/\mathcal{C} \to \mathcal{C}$ that is discrete Grothendieck opfibration. This is a cosieve exactly if F is valued in propositions.

Let $\mathcal{C}$ now be an inverse category. Then, the forgetful functor $F/\mathcal{C} \to \mathcal{C}$ induces a rank functor also on $F/\mathcal{C}$. This makes $F/\mathcal{C}$ into an inverse category.

Given a category $\mathcal{C}$, recall that the *coslice* $x/\mathcal{C}$ over an object $x : \mathcal{C}$ has as objects pairs (y, f) of $y : \mathcal{C}$ and a map $f : x \to y$. Morphisms between (y, f) and (y', f') are given maps $h : y \to y'$ with $h \circ f = f'$ in $\mathcal{C}$. The evident forgetful functor $x/\mathcal{C}$ is a discrete Grothendieck opfibration.

The formulation of Reedy fibrancy for diagrams over inverse categories makes use of the notion of *matching object*. The following definition can be used (see Definition 4.4) to give a formulation of matching objects in terms of limits.

Definition 4.2 (reduced coslice)**.** Given a category $\mathcal{C}$ and an object $x : \mathcal{C}$, the *reduced coslice* $x /\!\!/ \mathcal{C}$ is the full subcategory of the coslice $x/\mathcal{C}$ on objects (y, f) that are not equal to (x, id_x), i.e. come with a proof $p : (y, f) \neq (x, \mathsf{id}_x)$ in the type of objects of $x/\mathcal{C}$ (equivalently, $(x, y, f) \neq (x, x, \mathsf{id}_x)$ in the type of morphisms of $\mathcal{C}$). We write such an object as a triple (y, f, p).

By definition, we have a fully faithful forgetful functor $x /\!\!/ \mathcal{C} \to x/\mathcal{C}$. Composing it with the forgetful functor to $\mathcal{C}$, we obtain $\mathsf{forget} : x /\!\!/ \mathcal{C} \to \mathcal{C}$, sending an object (y, f, p) to y.

Definition 4.2 may seem slightly unnatural from a category theory point of view. There is, however, a different perspective that can perhaps help to shed some light on the above definition, motivated by thinking of inverse categories as generalisations of Δ_+^{op}.

For a given inverse $\mathcal{C}$ with an object $x : \mathcal{C}$, we have the *representable diagram* $\mathcal{C}[x]$ defined by $\mathcal{C}[x]_y :\equiv \mathcal{C}(y, x)$ (with evident functorial action). Note that the coslice $x/\mathcal{C}$ is the category of elements of $\mathcal{C}[x]$. Now, the representable diagram $\mathcal{C}[x]$ has a maximal non-trivial subfunctor consisting of all the elements of $\mathcal{C}[x]$ except the identity arrow $\mathsf{id}_x : x \to x$. We detail its construction below.

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

Recall that subfunctors of representable functors correspond to *cosieves*, i.e. families of propositions:

$$\varphi : \Pi_{i:\mathcal{C}} \left(\mathcal{C}(x, i) \to \mathsf{Prop}\right),$$

such that, for maps $f : x \to y$ and $g : y \to z$, we have

$$\varphi_y(f) \to \varphi_z(g \circ f). \tag{17}$$

The functor $\Phi : \mathcal{C} \to \mathcal{U}$ corresponding to such a cosieve φ is then given by

$$\Phi(y) :\equiv \Sigma\, (f : \mathcal{C}(x, y))\,.\, \varphi_y(f)$$

on objects, and rule (17) ensures that the functor can act on morphisms by function composition, thereby giving a subfunctor of $\mathcal{C}[x]$.

Now, if δ is defined by

$$\delta(y, f) :\equiv (y \neq x),$$

the condition on $\mathcal{C}$ being inverse guarantees that δ is a well-defined cosieve. The corresponding subfunctor of $\mathcal{C}[x]$ is denoted by $\partial C[x]$ and referred to as the *abstract boundary* of x. In the following, we will denote by $i^x : \partial\mathcal{C}[x] \to \mathcal{C}[x]$ the obvious inclusion map.

The following result is an immediate consequence of the definitions.

Lemma 4.3. *For any inverse category $\mathcal{C}$ and object $x : \mathcal{C}$, the reduced coslice $x /\!\!/ \mathcal{C}$ is (up to isomorphism) the category of elements of the abstract boundary $\partial\mathcal{C}[x]$. Extending this, the category of elements functor sends the natural transformation $\partial\mathcal{C}[x] \to \mathcal{C}[x]$ (up to isomorphism) to the inclusion $x /\!\!/ \mathcal{C} \to x/\mathcal{C}$.*

4.3 Reedy fibrations

Part (i) of Lemma 3.10 makes it possible to construct fibrant limits of certain "well-behaved" functors from inverse categories. We follow Shulman (2015b), but our setup allows a slightly more general development. We will give a short analysis after the proof of Theorem 4.8.

In the following, we always assume that $\mathcal{C}$ is an inverse category.

Definition 4.4 (matching object; see Shulman 2015*b*, Chapter 11)**.** Let $X : \mathcal{C} \to \mathcal{U}$ be a functor. For any $z : \mathcal{C}$, we define the *matching object* M_z^X to be the limit of the composition $z /\!\!/ \mathcal{C} \xrightarrow{\text{forget}} \mathcal{C} \xrightarrow{X} \mathcal{U}$.

Using the universal property of the limit defining the matching object, we obtain a map $X_z \to M_z^X$. Abstracting over X, this gives a natural transformation $(-)_z \to M_z$ between functors from $[\mathcal{C}, \mathcal{U}]$ to $\mathcal{U}$.

Alternatively, we can formulate matching objects in terms of abstract boundaries:

Lemma 4.5. *Given a diagram X over $\mathcal{C}$, and an object $z : \mathcal{C}$, the matching object M_z^X is isomorphic to $\mathbf{Nat}(\partial\mathcal{C}[z], X)$, naturally in X. Under this and the Yoneda isomorphism $X_z \cong \mathbf{Nat}(\mathcal{C}[z], X)$, the map $X_z \to M_z^X$ corresponds to $\mathbf{Nat}(i^z, X)$.*

Proof. This is straightforward to verify directly. Alternatively, one can observe that weighted limits of X with weight W can be implemented by taking the limit over the restriction of X to category of elements of W, and this is natural in W. □

Definition 4.6 (Reedy (trivial) fibration; see Shulman 2015*b*, Definition 11.3)**.** Let $X, Y : \mathcal{C} \to \mathcal{U}$ be two diagrams. Further, let $p : Y \to X$ be a natural transformation. We say that p is a *Reedy (trivial) fibration* if, for all $z : \mathcal{C}$, the canonical map

$$Y_z \to M_z^Y \times_{M_z^X} X_z,$$

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

induced by the universal property of the pullback, is a (trivial) fibration.

A diagram X is said to be *Reedy (trivially) fibrant* if the canonical map $X \to 1$ is a Reedy (trivial) fibration, where here 1 denotes the diagram that is constantly the unit type.

In terms of the natural transformation $(-)_z \to M_z$, we can express that p is a Reedy fibration by saying that the *pullback application* of $(-)_z \to M_z$ to p is a fibration for all $z : \mathcal{C}$. Here, we have applied the Leibniz calculus to the bifunctor $[[\mathcal{C},\mathcal{U}],\mathcal{U}] \times [\mathcal{C},\mathcal{U}] \to \mathcal{U}$.

We can use abstract boundaries to improve on this. The *pullback hom* $\widehat{\mathbf{Nat}}$ in the category of diagrams on $\mathcal{C}$ is the Leibniz construction of the natural transformation bifunctor **Nat**. Concretely, given natural transformations $f : A \to B$ and $p : Y \to X$ between diagram on $\mathcal{C}$, their pullback hom is the induced map

$$\mathbf{Nat}(B, Y) \xrightarrow{\widehat{\mathbf{Nat}}(f,p)} \mathbf{Nat}(B, X) \times_{\mathbf{Nat}(A,X)} \mathbf{Nat}(A, Y).$$

With this, we can give an alternative characterisation of Definition 4.6 that is occasionally useful.

Lemma 4.7. *A natural transformation $p : Y \to X$ between diagrams on $\mathcal{C}$ is a Reedy (trivial) fibration if and only if, for all objects $z : \mathcal{C}$, the map $\widehat{\mathbf{Nat}}(i^z, p)$ is a (trivial) fibration (where $i^z : \partial\mathcal{C}[z] \to \mathcal{C}[z]$).*

Proof. Immediate consequence of Lemma 4.5. □

Using Definition 4.6, we can make precise the claim that we can construct fibrant limits of certain well-behaved diagrams.

Theorem 4.8 (see Shulman 2015*b*, Lemma 11.8)**.** *Assume that $\mathcal{C}$ is an inverse category with a finite type of objects $|\mathcal{C}|$ and that $X : \mathcal{C} \to \mathcal{U}$ is a Reedy (trivially) fibrant diagram. Then, X has a (trivially) fibrant limit.*

Proof. By induction on the cardinality of $|\mathcal{C}|$. In the case $|\mathcal{C}| \cong \mathsf{Fin}_0$, the limit is the unit type.

Otherwise, we have $|\mathcal{C}| \cong \mathsf{Fin}_{n+1}$. Let us consider the rank functor

$$\varphi : \mathcal{C}^{\mathrm{op}} \to \omega.$$

Choose an object $z : \mathcal{C}$ such that $\varphi(z)$ is maximal; this is possible (constructively) due to the finiteness of $|\mathcal{C}|$. Let us call $\mathcal{C}'$ the category that we get if we remove z from $\mathcal{C}$; that is, we set $|\mathcal{C}'| :\equiv \Sigma\,(x : |\mathcal{C}|)\,.\,x \neq z$. Clearly, $\mathcal{C}'$ is still inverse, and we have $|\mathcal{C}'| \cong \mathsf{Fin}_n$.

Denote by 1 the terminal diagram on $\mathcal{C}$, and by $1'$ the left Kan extension of the terminal diagram on $\mathcal{C}'$ to $\mathcal{C}$. So $1'$ has value 1 on all objects except z, where it has value 0. Similarly, let X' be the restriction of X on $\mathcal{C}'$.

The following square of diagrams on $\mathcal{C}$

$$\begin{array}{ccc} \partial\mathcal{C}[z] & \longrightarrow & \mathcal{C}[z] \\ \downarrow & & \downarrow \\ 1' & \longrightarrow & 1 \end{array}$$

is a pushout, as one can easily check levelwise, using the fact that equality on objects in $\mathcal{C}$ is decidable, since $|\mathcal{C}|$ is finite.

By applying the bifunctor **Nat** with X on the right, we get that the square

$$\begin{array}{ccc} \lim X & \longrightarrow & X_z \\ \downarrow & & \downarrow \\ \lim X' & \longrightarrow & M_z^X \end{array}$$

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

is a pullback. The right vertical map is a fibration by Reedy fibrancy of X; hence, the left vertical map is a fibration by part (i) of Lemma 3.10. Now $\mathsf{lim}\, X'$ is fibrant by induction hypothesis; hence ,$\mathsf{lim}\, X$ is fibrant. □

In Shulman's work (Shulman, 2015*b*), the notion of an outer type does not exist, and every type that occurs is inner. This means that every diagram is valued in inner types, and consequently, matching objects do not necessarily exist, so the definition of a Reedy fibration has to include the condition that all the matching objects involved are available.

If we wanted to precisely reproduce Shulman's definition of a Reedy fibration, we would have to modify Definition 4.6: first, that all occurring diagrams are valued in fibrant types, and second, that all occurring matching objects, obtained by coercing to outer types then taking a limit, happen to be fibrant.

In our setting, it is more natural to work with outer types from the beginning. We recover Shulman's notion in the special case where the outer types that occur are coercions of inner types. Although diagrams of fibrant types ($\mathcal{C} \to \mathcal{U}^{\mathsf{i}}$) are what we are ultimately interested in, we can at no cost enlarge the class of diagrams that we talk about to those that are built out of outer types, possibly not even isomorphic to inner ones. The ability to make this choice is an important feature of two-level type theory as a meta-theoretic reasoning framework, and it can help to obtain some slightly more general results compared to a traditional approach.

Similarly to the notation $\mathcal{C}^{<n}$ (see Definition 4.1), we will denote by $X|n$ the restriction of a diagram $X : \mathcal{C} \to \mathcal{U}$ to $\mathcal{C}^{<n}$.

The following lemma generalises Lemma 3.10 to Reedy fibrations.

Lemma 4.9. *Reedy (trivial) fibrations are closed under:*

(i) pullbacks (see Shulman 2015b, Theorem 11.11),
(ii) finite compositions.
(iii) finite products,

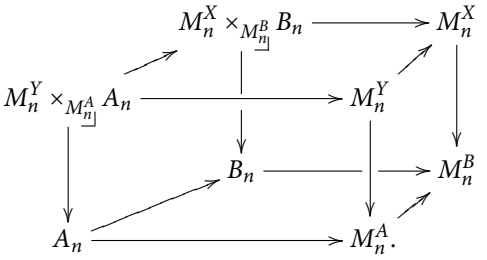
Proof. As before, part (iii) is a consequence of (i) and (ii).

For part (i), we first give an abstract argument. Recall from Riehl and Verity (2014) that the functorial action of the pullback hom in its second argument preserves morphisms between arrows that are pullbacks. It follows that $\widehat{\mathbf{Nat}}(i^z, p^* f)$ is a pullback of $\widehat{\mathbf{Nat}}(i^z, f)$ for any $z : \mathcal{C}$. This reduces the claim to part (i) to Lemma 3.10.

For the benefit of the reader, we also include a more direct proof, which is obtained by unfolding the abstract argument. Suppose we have a pullback square:

$$\begin{array}{ccc} Y & \xrightarrow{g} & X \\ {\scriptstyle q}\downarrow & \lrcorner & \downarrow{\scriptstyle p} \\ A & \xrightarrow[f]{} & B, \end{array}$$

where p is a Reedy fibration. We want to show that q is a Reedy fibration. Now fix an object $n : |\mathcal{C}|$, and consider the cube:



https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

The front and back faces are pullbacks by construction, and the right face is a pullback because it is the limit of a pullback square. By a pullback pasting argument, the square determined by the front left and the back right vertical arrow is a pullback. By a second pullback pasting argument, the left face is a pullback.

Now consider the diagram:

$$\begin{array}{ccc} Y_n & \longrightarrow & X_n \\ \downarrow & & \downarrow \\ M_n^Y \times_{M_n^A} A_n & \longrightarrow & M_n^X \times_{M_n^B} B_n \\ \downarrow & & \downarrow \\ A_n & \longrightarrow & B_n. \end{array}$$

We have proved that the lower square is a pullback, and the outermost square is a pullback because limits in categories of diagrams are pointwise. It follows that the upper square is a pullback for all $n : \mathcal{C}$, which shows that the map $Y \to A$ is a Reedy fibration.

For part (ii), recall from Riehl and Verity (2014) that the pullback hom with a fixed map $i^z : \partial\mathcal{C}[Z] \to \mathcal{C}[Z]$ of a finite composition is a finite composition of pullbacks of pullback homs of i^z with the individual factors. Thus, the claim reduces to parts (i) and (ii) of Lemma 3.10. □

Reedy fibrations admit "change of base" along discrete Grothendieck fibrations.

Lemma 4.10. *Let $H : \mathcal{C} \to \mathcal{D}$ be a discrete Grothendieck fibration of inverse categories. Then, the restriction functor $H^* : [\mathcal{D}, \mathcal{U}] \to [\mathcal{C}, \mathcal{U}]$ preserves Reedy (trivial) fibrations.*

Proof. There are various way to check this. Because H is a discrete fibration, for $z : \mathcal{C}$, the induced functor $H : z/\mathcal{C} \to Hz/\mathcal{D}$ on slices is an isomorphism, and it restricts to an isomorphism $H : z /\!\!/ \mathcal{C} \to Hz /\!\!/ \mathcal{D}$ of reduced slices. The Reedy fibrancy conditions for a map p in $[\mathcal{D}, \mathcal{U}]$ thus restrict to particular cases of the Reedy fibrancy condition for H^*p in $[\mathcal{C}, \mathcal{U}]$. The same holds for Reedy trivial fibrations.

Alternatively, one checks for $z : \mathcal{C}$ that left Kan extension $H_!$ along H sends the map $i^z : \partial\mathcal{C}[z] \to \mathcal{C}[z]$ in $[\mathcal{C}, \mathcal{U}]$ to the map $i^{Hz} : \partial\mathcal{C}[Hz] \to \mathcal{C}[hZ]$ (this always holds for the codomain, independently of requiring that H is a discrete Grothendieck fibration). Indeed, this follows from the following special case. The map i^z is itself the left Kan extension of the maximal non-trivial subobject of the terminal object in diagrams over $z/\mathcal{C}$, and similarly for the abstract boundary i^{Hz} and $\mathcal{D}$. The claim then follows from commutativity of the diagram

$$\begin{array}{ccc} z/\mathcal{C} & \xrightarrow[\simeq]{H} & Hz/\mathcal{D} \\ \downarrow & & \downarrow \\ \mathcal{C} & \xrightarrow{H} & \mathcal{D} \end{array}$$

of categories. □

4.4 Reedy fibrant factorisations

The goal of the current section is to show that any functor X from an *admissible* inverse category $\mathcal{C}$ to $\mathcal{U}_{\mathsf{fib}}$ has a Reedy fibrant replacement; that is, we can construct a Reedy fibrant diagram which is equivalent to X in a suitable sense. More generally, given any map of pointwise fibrant diagrams, we can always factor it into a (pointwise) equivalence followed by a fibration.

We emphasise that we are talking about diagrams that are valued in fibrant types. An obvious reason why this is necessary is that the only notion of equivalence for general diagrams that we

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

could use would be (strict) isomorphism, which clearly would be too strong. But even if we came up with a weak notion of equivalence of general diagrams, we could not expect it to be possible to start with *any* diagram $\mathcal{C} \to \mathcal{U}$ and derive a Reedy fibrant one from it which is in some sense equivalent. Already in the special case that $\mathcal{C}$ is the discrete category with exactly one object, this would correspond to finding a *fibrant replacement* of an outer type. By Theorem 2.20, such a fibrant replacement cannot be defined internally.

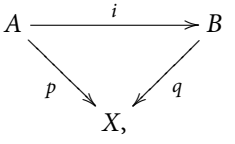
Our construction is an internalisation of the known analogous construction in traditional mathematics (see e.g. Shulman 2015*b*, Lemma 11.10 or Riehl and Verity 2014).

Definition 4.11. We say that a map $i : A \to B$ is *anodyne* if it has the left lifting property with respect to fibrations. More precisely, for all fibrations $p : Y \to X$, the pullback exponential $(B \to Y) \to (B \to X) \times_{A \to X} (A \to Y)$ has a section.

The following characterisations of anodyne maps are straightforward to establish.

Lemma 4.12. *A function $i : A \to B$ is anodyne if and only if for all families $X : B \to \mathcal{U}_{\mathsf{fib}}$ of fibrant types over B, and all terms $t : \prod_{a:A} X(i(a))$, we can find a term $t' : \prod_{b:B} X(b)$ such that $t'(i(a)) = t(a)$ for all $a : A$.*

Lemma 4.13. *Assume we are given a commutative triangle*



where the map i is fibrewise anodyne, i.e. for all $x : X$, the induced map $p^{-1}(x) \to q^{-1}(x)$ between the fibres is anodyne. Then, i is anodyne.

Since anodyne maps are defined by a left lifting property, they exhibit familiar closure properties, of which we recall the following:

Lemma 4.14. *Anodyne maps are closed under retracts.*

When focusing on fibrant types, we can say more about anodyne maps.

Lemma 4.15. *If $i : A \to B$ is an anodyne map between fibrant types, then i is an equivalence.*

Proof. Since A is fibrant, we get a map $r : B \to A$ such that $r \circ i = \mathsf{id}$, in particular $r \circ i =^{\mathsf{i}} \mathsf{id}$. To show that $i \circ r =^{\mathsf{i}} \mathsf{id}$, it is enough, by Lemma 4.12, to show that $i \circ r \circ i =^{\mathsf{i}} i$, which follows from the first part. □

Trivial cofibrations are in particular anodyne maps, as the following lemma shows.

Lemma 4.16. *If $i : A \to B$ is a trivial cofibration, then it is anodyne.*

Proof. For any fibration p, the pullback exponential $\widehat{\exp}(i, p)$ is a trivial fibration, so it has a section by Lemma 3.11. □

The following lemma provides an important example of anodyne map which is not necessarily a trivial cofibration.

Lemma 4.17. *Let A be a fibrant type, and $a_0 : A$ a term. Then the map $i : 1 \to \Sigma\,(a : A)\,.\,a =^{\mathsf{i}} a_0$ which selects the pair $(a_0, \mathsf{refl}_{a_0})$ is anodyne.*

Proof. Immediate consequence of the elimination rule for the fibrant identity type and its computation rule. □

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

Corollary 4.18. *Let $f : A \to B$ be a function between fibrant types. Then, there exists a fibrant type N, an anodyne map $i : A \to N$, and a fibration $p : N \to B$, such that $f = p \circ i$.*

Proof. Let $N :\equiv \Sigma\,(a:A)\,.\,\Sigma\,(b:B)\,.\,(f(a) =^{\mathsf{i}} b)$. The function i is given by $i(a) :\equiv (a, f(a), \mathsf{refl}_{f(a)})$, while p is simply the projection into the component of type B. Then p is a fibration by construction.

Now consider the projection $q : N \to A$ on the first component. If we use q to regard i as a map over A, then it is clear that the fibres of i have the form of Lemma 4.17 up to isomorphism, hence they are anodyne. It then follows from Lemma 4.13 that i itself is anodyne, as required. □

We will refer to the type N constructed in the proof of Corollary 4.18 as the *mapping cocylinder* of f.

Definition 4.19. Let $\mathcal{C}$ be an inverse category. We say that $\mathcal{C}$ is *admissible* if, for all $a : \mathcal{C}$, Reedy (trivially) fibrant diagrams over the reduced coslice $a /\!\!/ \mathcal{C}$ have a (trivially) fibrant limit.

Let $\mathcal{C}$ be an inverse category with a functor $F : \mathcal{C} \to \mathcal{U}$. Since the category of elements $F/\mathcal{C} \to \mathcal{C}$ is a discrete Grothendieck opfibration, the induced functor $(a, x) /\!\!/ F/\mathcal{C} \to a /\!\!/ \mathcal{C}$ is an isomorphism for any $(a, x) : F/\mathcal{C}$, and similarly for the (non-reduced) coslices. It follows that admissibility of $\mathcal{C}$ implies admissibility of $F/\mathcal{C}$.

Lemma 4.20. *Let $\mathcal{C}$ an admissible inverse category, and let $X : \mathcal{C} \to \mathcal{U}$ be a Reedy fibrant diagram. Then, X is pointwise fibrant, and all the matching objects of X are fibrant.*

Proof. First observe that if X is a Reedy fibrant diagram on $\mathcal{C}$, and $z : \mathcal{C}$ is any object, then $j^* X$ is a Reedy fibrant diagram on $z /\!\!/ \mathcal{C}$, where $j : z /\!\!/ \mathcal{C} \to \mathcal{C}$ is the forgetful functor. This follows from the above observation that j induces an isomorphism between (reduced) coslices of $z /\!\!/ \mathcal{C}$ and (reduced) coslices of $\mathcal{C}$.

Therefore, if $\mathcal{C}$ is admissible, the matching object M_z^X of X is fibrant for all $z : \mathcal{C}$. Since furthermore the map $X_z \to M_z^X$ is a fibration by the assumption that X is Reedy fibrant, we get that X_z is fibrant as well. □

The main example of an admissible inverse category is Δ_+^{op}, the opposite of the simplex category restricted to strictly monotone maps. We can define it concretely as follows:

Definition 4.21. (category Δ_+^{op})**.** The category Δ_+^{op} has natural numbers as objects, written $[0]$, $[1]$, $[2]$, …, and morphisms $\Delta_+^{\mathrm{op}}([m], [k])$ are strictly increasing functions $\mathsf{Fin}_{k+1} \to \mathsf{Fin}_{m+1}$. Composition of morphisms is given by function composition.

That Δ_+^{op} is admissible follows from Theorem 4.8 and the fact that all the reduced coslices of Δ_+^{op} are finite.

The notion of anodyne can be extended to diagrams in a pointwise fashion.

Definition 4.22. Let $\mathcal{C}$ be a category, and X, Y be diagrams on $\mathcal{C}$. A natural transformation $f : X \to Y$ is said to be *anodyne* if it is so pointwise, i.e. for all $n : \mathcal{C}$, the function $f_n : X_n \to Y_n$ is anodyne.

Similarly, we have a notion of equivalence of diagrams, defined pointwise.

Definition 4.23. For a category $\mathcal{C}$, let $X, Y : \mathcal{C} \to \mathcal{U}$ be pointwise fibrant diagrams. A natural transformation $f : X \to Y$ is said to be an *equivalence* if, for all $n : \mathcal{C}$, the function $f_n : X_n \to Y_n$ is a ("homotopy") equivalence.

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

Lemma 4.24. *An anodyne natural transformation between pointwise fibrant diagrams is an equivalence.*

Proof. Immediate consequence of Lemma 4.15. □

Lemma 4.25. *Let $p : X \to Y$ be a (trivial) Reedy fibration of diagrams over an inverse category $\mathcal{C}$. Suppose that Reedy (trivially) fibrant diagrams over $\mathcal{C}$ have (trivially) fibrant limits. Then, the limit $\lim p : \lim X \to \lim Y$ is a (trivial) fibration.*

Proof. We only do the case of fibrations.

Let $y : \lim Y$ be an arbitrary element of the limit. We can think of y as a natural transformation $y : 1 \to Y$. Consider the following pullback of diagrams:

$$\begin{array}{ccc} X[y] & \longrightarrow & X \\ \downarrow & \lrcorner & \downarrow{\scriptstyle p} \\ 1 & \xrightarrow{y} & Y. \end{array}$$

By part (i) of Lemma 4.9, $X[y]$ is a Reedy fibrant diagram, hence its limit is fibrant by the assumption on $\mathcal{C}$. Since limits commute with pullbacks, we get a pullback diagram:

$$\begin{array}{ccc} \lim X[y] & \longrightarrow & \lim X \\ \downarrow & \lrcorner & \downarrow{\scriptstyle \lim p} \\ 1 & \xrightarrow{y} & \lim Y, \end{array}$$

showing that the fibre of $\lim p$ over y is fibrant. □

Lemma 4.26. *Let $f : X \to Z$ be a natural transformation of pointwise fibrant diagrams over an admissible inverse category $\mathcal{C}$. Then, f can be factored as:*

$$f : X \xrightarrow{i} Y \xrightarrow{p} Z,$$

where i is anodyne, and p is a Reedy fibration.

Proof. We will construct, by induction on the natural number n, a diagram $Y^{(n)}$ over $\mathcal{C}^{<n}$, and a factorisation of f:

$$X|n \xrightarrow{i^{(n)}} Y^{(n)} \xrightarrow{p^{(n)}} Z|n,$$

where $i^{(n)}$ is anodyne and $p^{(n)}$ is a Reedy fibration.

For $n = 0$ there is nothing to construct, so assume the existence of $Y^{(n)}$, and fix any object $x : \mathcal{C}$ of rank $n+1$. The forgetful functor $j_x : x /\!\!/ \mathcal{C} \to \mathcal{C}$ factors through $\mathcal{C}^{<n}$; hence, we can consider the composition $Y^{(n)} \circ j_x$ and take its limit L. Note that L is not necessarily fibrant, but the map $L \to M_x^Z$ induced by $p^{(n)}$ is a fibration by Lemma 4.25 and the admissibility of $\mathcal{C}$. By part (i) of Lemma 3.10, we get a fibration $L \times_{M_x^Z} Z_x \to Z_x$, hence $L \times_{M_x^Z} Z_x$ is a fibrant type.

Now f, together with $i^{(n)}$, determine a map $X_x \to L \times_{M_x^Z} Z_x$. Define $Y_x^{(n+1)}$ to be the mapping cocylinder of this map. For any object y of rank n or less, define $Y_y^{(n+1)}$ as $Y_y^{(n)}$, and for any morphism $f : x \to y$ in $\mathcal{C}$, the corresponding function $Y_x^{(n+1)} \to Y_y^{(n+1)}$ is given by the projection from the mapping cocylinder, followed by a map of the universal cone of the limit L. The action of $Y^{(n)}$ on morphisms between objects of ranks n or less is defined to be the same as that of $Y^{(n)}$.

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

It is easy to see that those definitions make $Y^{(n+1)}$ into a diagram that extends $Y^{(n)}$ to objects of rank $n+1$. We can also extend $i^{(n)}$ by defining $i_x^{(n+1)}$ to be the embedding of X_x into the mapping cocylinder $Y_x^{(n+1)}$, which is anodyne by Corollary 4.18.

Similarly, we define $p_x^{(n+1)}$ to be the composition of the projection from the mapping cocylinder with the map $L \times_{M_x^Z} Z_x \to Z_x$ defined above. The fact that $p^{(n+1)}$ is a Reedy fibration follows immediately from the construction, since L is exactly the matching object of $Y^{(n+1)}$ at x.

To conclude the proof, we glue together all the $Y^{(n)}$, $i^{(n)}$ and $p^{(n)}$ into a single diagram Y and natural transformations i, p. Clearly, p is a Reedy fibration, and i is an equivalence. □

Corollary 4.27. *Let X be a pointwise fibrant diagram. Then, there exists a Reedy fibrant diagram Y and an anodyne natural transformation $\eta : X \to Y$.*

Proof. Apply Lemma 4.26 with Z equal to the constant diagram on the unit type. □

Corollary 4.28. *Let $i : A \to B$ be a natural transformation of pointwise fibrant diagrams. Then, i is anodyne if and only if it has the left lifting property with respect to Reedy fibrations, i.e. for all Reedy fibrations $p : Y \to X$, the induced map $\mathbf{Nat}(B, Y) \to \mathbf{Nat}(B, X) \times_{\mathbf{Nat}(A,X)} \mathbf{Nat}(A, Y)$ has a section.*

Proof. First suppose that $i : A \to B$ is anodyne, and let $p : Y \to X$ be any fibration. Consider a commutative square

$$\begin{array}{ccc} A & \xrightarrow{u} & Y \\ {\scriptstyle i}\downarrow & & \downarrow{\scriptstyle p} \\ B & \xrightarrow[v]{} & X. \end{array}$$

For all natural numbers n, we will construct a lift w for the square of the restrictions of all the diagrams involved to $\mathcal{C}^{<n}$. The base of the induction is trivial, so suppose we have constructed such a lift for n, and let $x \in C$ be an object of degree $n+1$. Consider the square:

$$\begin{array}{ccc} A_x & \xrightarrow{u} & Y_x \\ {\scriptstyle i}\downarrow & & \downarrow{\scriptstyle p} \\ B_x & \xrightarrow[(w,v)]{} & M_x^Y \times_{M_x^X} X_x, \end{array}$$

where the bottom map is obtained from the inductively constructed lift w, together with the bottom natural transformation v of the original square.

The right map is a fibration by the assumption that p is a Reedy fibration, which lets us construct a diagonal lift w_x for the square. This gives an extension of the lift w to all objects x of degree n, and naturality of w follows immediately from the commutativity of the bottom right triangle. Note that we have not used the assumption that A and B are pointwise fibrant for this direction.

Conversely, suppose that $i : A \to B$ has the stated lifting property. Since A and B are pointwise fibrant, we can factor i as $A \xrightarrow{j} N \xrightarrow{p} B$, where j is anodyne and p is a Reedy fibration, thanks to Lemma 4.26. At this point, a standard retract argument shows that i must be anodyne. More explicitly, first consider the square

$$\begin{array}{ccc} A & \xrightarrow{i} & M \\ {\scriptstyle j}\downarrow & & \downarrow{\scriptstyle p} \\ B & \xrightarrow[\mathsf{id}]{} & B, \end{array}$$

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

and use the lifting property of i to get a natural transformation $r : B \to M$. It then follows that i is a retract of j, so in particular i_x is a retract of j_x for all objects $x : \mathcal{C}$. The conclusion now follows from Lemma 4.14. □

4.5 Classifiers for Reedy fibrations

Let $\mathcal{C}$ be an admissible category. The goal of this section is to construct a type that classifies, in the appropriate sense, Reedy fibrant diagrams over $\mathcal{C}$. In certain cases, this type will itself be fibrant, giving a construction of a classifier for diagrams which is completely internal to the inner level.

The construction of this classifier makes use of a stricter notion of fibration than the one we have used so far.

Definition 4.29. Let X be a type. A *strict fibration* on X is simply a family of inner types indexed over X.

Any strict fibration A over X determines a map $Y \to X$, where $Y :\equiv \Sigma_X A$. We will sometimes abuse language and refer to a map $p : Y \to X$ itself as a strict fibration, with the convention that strict fibrations are always assumed to be equipped with a corresponding choice of a family A, which we will refer to as a *strict fibration structure* on p. Fibrations can then be characterised as those maps $Y \to X$ that are isomorphic over X to some strict fibration. In particular, strict fibrations are fibrations.

Lemma 4.30. *If X is a cofibrant type, then the type of strict fibrations over X is fibrant.*

Proof. Immediate consequence of the fact that this type is, by definition, $X \to \mathcal{U}^{\mathsf{i}}$. □

Correspondingly, we get a notion of strict Reedy fibration, simply by replacing fibrations with strict fibrations in Definition 4.6.

Definition 4.31. Let $\mathcal{C}$ be an inverse category. A *strict Reedy fibration* is a natural transformation $p : Y \to X$, where X and Y are diagrams on $\mathcal{C}$, together with, for all $z : \mathcal{C}$, a choice of a strict fibration structure $\overline{Y}_z$ for the canonical map

$$Y_z \to M_z^Y \times_{M_z^X} X_z. \tag{18}$$

A strictly Reedy fibrant diagram is a strict Reedy fibration $X \to 1$.

It is crucial to observe that the notion of strict fibration is not invariant under isomorphism, since it uses strict equality of types. Consequently, (18) has to be intended with a specific choice of pullbacks and limits in the target type. For concreteness, given a functor $F : \mathcal{A} \to \mathcal{U}$, we will always choose the limit of F to be the type given by $\mathbf{Nat}(1, F)$.

Nevertheless, given a type X, the type of strict fibrations over X is isomorphic to the type of functions $X \to \mathcal{U}^{\mathsf{i}}$, so it is a representable functor of X, hence in particular if $X \cong Y$, then also the types of strict fibrations over X and Y are isomorphic.

Again, since strict fibrations are in particular fibrations, it follows that strict Reedy fibrations are Reedy fibrations. Recall that both notions are equipped with structure, namely a choice of a family of inner types for all objects of the base category. For our usage of Reedy fibration, this choice usually does not matter; however, for strict Reedy fibrations, it is important.

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

Definition 4.32. Given strict Reedy fibrations $X \to A$ and $Y \to B$, a *morphism* between them is a pullback square

$$\begin{array}{ccc} X & \longrightarrow & Y \\ \downarrow & \lrcorner & \downarrow \\ A & \longrightarrow & B, \end{array}$$

such that for all $z : \mathcal{C}$, the induced triangle

$$\begin{array}{ccc} M_z^X \times_{M_z^A} A_z & \longrightarrow & M_z^Y \times_{M_z^B} B_z \\ & \searrow \quad \swarrow & \\ & \mathcal{U}^{\mathsf{i}} & \end{array} \tag{19}$$

commutes.

With this definition of morphisms, the collection of strict Reedy fibrations on an inverse category $\mathcal{C}$ forms a category, which we will denote by $\mathcal{R}_{\mathcal{C}}$.

Lemma 4.33. *The functor $\mathcal{R}_{\mathcal{C}} \to [\mathcal{C}, \mathcal{U}]$ which maps a fibration $X \to A$ to its base A is a discrete Grothendieck fibration.*

Proof. First, let us prove that every morphism of $\mathcal{R}_{\mathcal{C}}$ is Cartesian over $[\mathcal{C}, \mathcal{U}]$. Let $X \to A$, $Y \to B$ and $Z \to C$ be strict Reedy fibrations, $f : A \to B$, $g : B \to C$ natural transformations, and suppose we are given morphisms

$$\begin{array}{ccc} Y & \longrightarrow & Z \\ \downarrow & \lrcorner & \downarrow \\ B & \xrightarrow{g} & C \end{array} \qquad \begin{array}{ccc} X & \longrightarrow & Z \\ \downarrow & \lrcorner & \downarrow \\ A & \xrightarrow{g \circ f} & C, \end{array}$$

then it is clear we can uniquely construct a map $X \to Y$ that fits into a pullback square

$$\begin{array}{ccc} X & \longrightarrow & Y \\ \downarrow & \lrcorner & \downarrow \\ A & \xrightarrow{f} & B. \end{array}$$

To show that this is a morphism of strict Reedy fibrations, observe that in the induced diagram

$$\begin{array}{ccccc} M_z^X \times_{M_z^A} A_z & \longrightarrow & M_z^Y \times_{M_z^B} B_z & \longrightarrow & M_z^Z \times_{M_z^C} C_z \\ & \searrow & \downarrow & \swarrow & \\ & & \mathcal{U}^{\mathsf{i}} & & \end{array}$$

the outermost triangle and the right triangle commute, and hence the left triangle commutes as well.

Now, let $f : A \to B$ be a natural transformation, and $Y \to B$ a strict Reedy fibration. We want to show that there exists a unique strict Reedy fibration $X \to A$ equipped with a morphism $Y \to B$. For all natural numbers n, we construct a strict Reedy fibration $X^{(n)} \to A|n$ on $\mathcal{C}^{<n}$, together with a morphism to $Y|n \to B|n$, with the property that $X^{(n+1)}|n = X^{(n)}$.

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

The base case is trivial as usual; hence, we can assume that we have constructed $X^{(n)}$. If $z : \mathcal{C}$ has rank n, let $\overline{X}_z^{(n+1)}$ be the composition

$$M_z^{X^{(n)}} \times_{M_z^A} A_z \to M_z^Y \times_{M_z^B} B_z \to \mathcal{U}^i,$$

and define $X_z :\equiv \Sigma_{M_z^{X^{(n)}} \times_{M_z^A} A_z} \overline{X}_z$. It is easy to verify that this defines an extension $X^{(n+1)}$ of $X^{(n)}$ to objects of degree n. The map $X^{(n+1)} \to A$ is a strict Reedy fibration, and the choice of $\overline{X}_z^{(n+1)}$ induces a morphism of strict Reedy fibrations $X^{(n+1)} \to Y$, essentially by construction.

As for uniqueness, note that the choice of $\overline{X}_z^{(n+1)}$ is forced by the requirement that the triangle (19) commute; hence, the fibration $X^{(n+1)} \to A$ and the morphism to $Y|n \to B|n$ are uniquely determined by the corresponding data for n. It follows that the whole fibration $X \to A$ and morphism to $Y \to B$, obtained by gluing all the $X^{(n)}$, are uniquely determined. □

The discrete Grothendieck fibration $\mathcal{R}_\mathcal{C} \to [\mathcal{C}, \mathcal{U}]$ determines a presheaf Reedy on $[\mathcal{C}, \mathcal{U}]$. For the next lemma, recall that, for a category $\mathcal{A}$ with pullbacks, a diagram $X : I \to \mathcal{A}$ is said to have a *van Kampen colimit* if the colimit of X exists, and the reindexing functor induces an equivalence of categories

$$\mathcal{A}/\operatorname{colim} X \xrightarrow{\cong} \lim_i \mathcal{A}/X_i.$$

Lemma 4.34. *The functor* $\mathsf{Reedy}_\mathcal{C} : [\mathcal{C}, \mathcal{U}]^{op} \to \mathcal{U}$ *maps van Kampen colimits in* $[\mathcal{C}, \mathcal{U}]$ *to limits in* $\mathcal{U}$.

Proof. Let I be any small category, and $X : I \to \mathcal{R}_\mathcal{C}$ a diagram of strict Reedy fibrations. Let $X^i \to A^i$ denote the component of the diagram X at an object $i : I$ and assume that the A^i have a van Kampen colimit B. It is enough to show that there exists a unique cocone for X in $\mathcal{R}_\mathcal{C}$ over the colimit cocone of the A^i.

We will construct a strict Reedy fibration $Y \to B$, by induction on the rank of an object $z : \mathcal{C}$, and prove that Y is a colimit of the X^i over the corresponding colimit cocone of the A^i. Suppose that we have constructed Y on objects of $\mathcal{C}^{<n}$, and let z have rank n. From the fact that the colimit of the A^i is van Kampen, it follows that

$$\begin{array}{ccc} X^i|n & \longrightarrow & Y|n \\ \downarrow & \lrcorner & \downarrow \\ A^i|n & \longrightarrow & B|n, \end{array}$$

is Cartesian, and therefore in the diagram

$$\begin{array}{ccccc} M_z^{X^i} \times_{M_z^{A^i}} A^i & \longrightarrow & M_z^{X^i} & \longrightarrow & M_z^Y \\ \downarrow & & \downarrow & \lrcorner & \downarrow \\ A_z^i & \longrightarrow & M_z^{A^i} & \longrightarrow & M_z^B \end{array}$$

both squares are Cartesian, hence so is the outer rectangle. This, and universality of colimits in $[\mathcal{C}, \mathcal{U}]$, imply that

$$M_z^Y \times_{M_z^B} B_z \cong \operatorname*{colim}_i M_z^{X^i} \times_{M_z^{A^i}} A^i,$$

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

hence the collection of inner families $\overline{X}^i_z : M_z^{X^i} \times_{M_z^{A^i}} A^i \to \mathcal{U}^{\mathsf{i}}$ uniquely determines an inner family $\overline{Y}_z : M_z^Y \times_{M_z^B} B_z \to \mathcal{U}^{\mathsf{i}}$, and if we define

$$Y_z :\equiv \Sigma_{M_z^Y \times_{M_z^B} B_z} \overline{Y}_z,$$

it follows again from the van Kampen property that Y_z is a colimit of the X^i_z, and that the corresponding squares

$$\begin{array}{ccc} X^i_z & \longrightarrow & Y_z \\ \downarrow & \lrcorner & \downarrow \\ A^i & \longrightarrow & B, \end{array}$$

are Cartesian. It is then easy to verify that our choice of strict Reedy fibration structure on the extension of Y to objects of rank n makes the colimit injections $X^i \to Y$ into morphisms of strict Reedy fibrations.

As for uniqueness, let $Y' \to B$ be a strict Reedy fibration, together with morphisms

$$\begin{array}{ccc} X^i & \longrightarrow & Y' \\ \downarrow & & \downarrow \\ A^i & \longrightarrow & B, \end{array}$$

forming a cocone for the diagram X.

Universality of colimits, applied to the identity map $\mathsf{colim}_i A^i \to B$, yields that $Y' \to B$ is a colimit of the $X^i \to A^i$, and therefore, Y and Y' are isomorphic over B. From the fact that the colimit injections into Y' are morphisms of strict Reedy fibrations, and the choice of strict Reedy fibration structure on Y, it then follows easily that the isomorphism $Y \cong Y'$ can be chosen to be a morphism of strict Reedy fibrations over the identity of B. Since strict Reedy fibrations form a discrete Grothendieck fibration over $[\mathcal{C}, \mathcal{U}]$ by Lemma 4.33, we get that $Y = Y'$ on the nose, as required. □

The next step is defining a *universe* of strictly Reedy fibrant types in the category of diagrams on $\mathcal{C}$. We use the basic idea for the construction of a universe in presheaves (Hofmann and Streicher, 1997), but we specialise it to strict Reedy fibrations rather than arbitrary natural transformations.

Definition 4.35. Let $\mathcal{V}$ be the diagram on $\mathcal{C}$ defined by:

$$\mathcal{V}_z :\equiv \mathsf{Reedy}_{\mathcal{C}}(\mathcal{C}[z]),$$

with the obvious action on morphisms.

Lemma 4.36. *For any diagram B on $\mathcal{C}$, there is a natural isomorphism*

$$\boldsymbol{Nat}(B, \mathcal{V}) \cong \mathsf{Reedy}_{\mathcal{C}}(B).$$

Proof. We can write B as a van Kampen colimit of representables

$$B \cong \operatorname*{colim}_{\substack{z:\mathcal{C} \\ b:B_z}} \mathcal{C}[z].$$

Since $\mathbf{Nat}(-, \mathcal{V})$ clearly maps colimits to limits, and $\mathsf{Reedy}_{\mathcal{C}}$ maps van Kampen colimits to limits by Lemma 4.34, we get that $\mathbf{Nat}(B, \mathcal{V}) \cong \mathsf{Reedy}_{\mathcal{C}}(B)$, and naturality easily follows. □

Corollary 4.37. *If $\mathcal{C}$ is admissible, the diagram $\mathcal{V}$ is Reedy fibrant.*

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

Proof. Let $z : \mathcal{C}$ be any object. We have to show that the map

$$\mathbf{Nat}(\mathcal{C}[z], \mathcal{V}) \to \mathbf{Nat}(\partial\mathcal{C}[z], \mathcal{V}),$$

obtained by applying $\widehat{\mathbf{Nat}}$ to the inclusion $\partial\mathcal{C}[z] \to \mathcal{C}[z]$ and the map $\mathcal{V} \to 1$, is a fibration. By Lemma 4.36, this map is isomorphic to the restriction map

$$\mathsf{Reedy}_{\mathcal{C}}(\mathcal{C}[z]) \to \mathsf{Reedy}_{\mathcal{C}}(\partial\mathcal{C}[z]).$$

Given a strict Reedy fibration X over $\partial\mathcal{C}[z]$, an extension of X to a strict Reedy fibration over $\mathcal{C}[z]$ is uniquely determined by the choice of a strict fibration over M_z^X. Since $\mathcal{C}$ is admissible, M_z^X is fibrant, hence cofibrant, and therefore, the type of strict fibrations over M_z^X is itself fibrant by Lemma 4.30. □

The type of strictly Reedy fibrant diagrams over $\mathcal{C}$ can now be recovered as the limit of $\mathcal{V}$.

Lemma 4.38. *Assume that $\mathcal{C}$ is admissible. The type $\mathsf{lim}\,\mathcal{V}$ is isomorphic to the type of strictly Reedy fibrant diagrams on $\mathcal{C}$.*

Proof. The type $\mathsf{lim}\,\mathcal{V}$ is defined as $\mathbf{Nat}(1, \mathcal{V})$, which, by Lemma 4.36, is isomorphic to the type of strictly Reedy fibrant diagrams on $\mathcal{C}$. □

In particular, if $\mathcal{C}$ is admissible, and Reedy fibrant diagrams on $\mathcal{C}$ itself have fibrant limits, then we obtain a *fibrant* type of strictly Reedy fibrant diagrams. This applies in particular to the restrictions $(\Delta_+^{\mathrm{op}})^{<n}$ of the semisimplicial category to finite level.

The connection between general Reedy fibrant diagrams and strict ones is made explicit by the following result.

Lemma 4.39. *A diagram X is Reedy fibrant if and only if there exists a strictly Reedy fibrant diagram Y that is isomorphic to X.*

Proof. Let X be a Reedy fibrant diagram. We strictify X by induction on the rank of the objects of $\mathcal{C}$. Assume X already satisfies the strict Reedy condition for objects of rank lower than n. If $z : \mathcal{C}$ is an object of rank n, we know that the map $X_z \to M_z^X$ is a fibration, since X is Reedy fibrant. It follows that there exists a family of inner types $T_z : M_z^X \to \mathcal{U}^{\mathsf{i}}$ such that $X_z \cong \Sigma_{M_z^X} T_z$.

Define a new functor Y by setting $Y_z :\equiv \Sigma_{M_z^X} T_z$ for all z of degree n, and $Y_z :\equiv X_z$ otherwise. It is easy to see that we can extend Y to a functor so that the obvious pointwise isomorphism $X_z \cong Y_z$ is natural. Furthermore, Y is strictly Reedy fibrant up to rank n, by construction. □

4.6 Exponents of diagrams

In this section, we want to address Reedy fibrancy of exponentials, and fibrancy of types of natural transformations. As before, we fix an inverse category $\mathcal{C}$ and work with diagrams on $\mathcal{C}$. We begin by defining a notion of Reedy (trivial) cofibrations, analogous to that of Definition 3.13.

Definition 4.40. A natural transformation $f : A \to B$ between diagrams is:

- a *Reedy cofibration* if $\widehat{\exp}(f, -)$ preserves Reedy fibrations and Reedy trivial fibrations,
- a *Reedy trivial cofibration* if $\widehat{\exp}(f, -)$ sends Reedy fibrations to Reedy trivial fibrations.

A diagram B is *Reedy (trivially) cofibrant* if the natural transformation $0 \to B$ is a Reedy (trivial) cofibration.

Given $z : \mathcal{C}$, recall the boundary inclusion $i^z : \partial\mathcal{C}[z] \to \mathcal{C}[z]$.

Lemma 4.41. *For any $z : \mathcal{C}$ and map $f : A \to B$ in diagrams over $\mathcal{C}$, the pushout product $i^z \mathbin{\widehat{\times}} f$ exists.*

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

Proof. Pushouts are constructed levelwise. For $t : \mathcal{C}$, we have to construct a pushout

$$\begin{array}{ccc} \partial\mathcal{C}[z]_t \times A_t & \longrightarrow & \partial\mathcal{C}[z]_t \times B_t \\ \downarrow & & \downarrow \\ \mathcal{C}[z]_t \times A_t & \dashrightarrow & P_t. \end{array}$$

If z and t have different degrees, then $\partial\mathcal{C}[z]_t \to \mathcal{C}[z]_t$ is an isomorphism, and we take $P_t :\equiv \mathcal{C}[z]_t \times B_t$. If z and t have the same degree, then $\partial\mathcal{C}[z]_t$ is empty. In that case, the top map is an isomorphism, and we take $P_t :\equiv \mathcal{C}[z]_t \times A_t$. □

We say that a natural transformation $f : A \to B$ is a *pointwise (trivial) cofibration*, if for all objects $z : \mathcal{C}$, we have that $f_z : A_z \to B_z$ is a (trivial) cofibration. The following theorem implies that, under a mild assumption on the index category $\mathcal{C}$, pointwise cofibrations are in particular Reedy cofibrations.

Theorem 4.42. *Let $f : A \to B$ be a pointwise cofibration of diagrams over $\mathcal{C}$. Then f is a Reedy cofibration, i.e. given a Reedy (trivial) fibration p, the pullback exponential*

$$[B, Y] \xrightarrow{\widehat{\exp}(f,p)} [B, X] \times_{[A,X]} [A, Y]$$

of p with f is a Reedy (trivial) fibration.

Proof. Let $z : \mathcal{C}$ be any object. The forgetful functor $F : z/\mathcal{C} \to \mathcal{C}$ is a discrete Grothendieck fibration. The induced restriction functor $F^* : [\mathcal{C}, \mathcal{U}] \to [z/\mathcal{C}, \mathcal{U}]$ has a left adjoint, left Kan extension along F, sending a diagram X over $z/\mathcal{C}$ to the diagram $F_!X$ over $\mathcal{C}$ defined by $(F_!X)_t :\equiv \Sigma\,(f : \mathcal{C}(z,t))\,.\,X_{(t,f)}$. We have

$$F_!X \times Y \cong F_!(X \times F^*Y) \tag{20}$$

naturally in $X : [z/\mathcal{C}, \mathcal{U}]$ and $Y : [\mathcal{C}, \mathcal{U}]$. Abstractly, this is a consequence of F^* preserving exponentiation. Explicitly, it unfolds at level t to the natural isomorphism

$$(\Sigma\,(f : \mathcal{C}(z,t))\,.\,A_{(t,f)}) \times B_t \cong \Sigma\,(f : \mathcal{C}(z,t))\,.\,(A_{(t,f)} \times B_t).$$

Since $F_!$ preserves colimits, the isomorphism (20) lifts to an isomorphism

$$F_!u \mathbin{\widehat{\times}} v \cong F_!(u \mathbin{\widehat{\times}} F^*v) \tag{21}$$

naturally in u a map in $[z/\mathcal{C}, \mathcal{U}]$ and v a map in $[\mathcal{C}, \mathcal{U}]$ whenever the involved pushout products exist.

Let now $p : Y \to X$ be a Reedy fibration. We will argue that $\widehat{\exp}(f,p)$ is again a Reedy fibration. The case of Reedy trivial fibrations is analogous. Using Lemma 4.7, we have to show that $\widehat{\mathbf{Nat}}(i^z, \widehat{\exp}(f,p))$ is a fibration. Recall from the proof of Lemma 4.10 that $i^z : \partial\mathcal{C}[z] \to \mathcal{C}[z]$ is the image of

$$\partial(z/\mathcal{C})[(z, \mathsf{id}_z)] \xrightarrow{i^{(z,\mathsf{id}_z)}} \partial(z/\mathcal{C})[(z, \mathsf{id}_z)]$$

under $F_!$. We calculate

$$\begin{aligned} \widehat{\mathbf{Nat}}(i^z \mathbin{\widehat{\times}} f, p) &\cong \widehat{\mathbf{Nat}}(F_!i^{(z,\mathsf{id}_z)} \mathbin{\widehat{\times}} f, p) \\ &\cong \widehat{\mathbf{Nat}}(F_!(i^{(z,\mathsf{id}_z)} \mathbin{\widehat{\times}} F^*f), p) \\ &\cong \widehat{\mathbf{Nat}}(i^{(z,\mathsf{id}_z)} \mathbin{\widehat{\times}} F^*f, F^*p), \end{aligned}$$

using (21) in the second step.

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

We denote T the functor $1 \to z/\mathcal{C}$ selecting the initial object (z, id_z). Restriction $T^* : [z/\mathcal{C}, \mathcal{U}] \to \mathcal{U}$ has left adjoint $T_!$ the constant functor. Note that the unit $\mathsf{Id} \to T^* T_!$ of the adjunction $T_! \dashv T^*$ is invertible, i.e. the $T_!$ is a coreflective embedding. Its counit induces a map

$$i^{(z,\mathsf{id}_z)} \mathbin{\widehat{\times}} T_! T^* F^* f \longrightarrow i^{(z,\mathsf{id}_z)} \mathbin{\widehat{\times}} F^* f \tag{22}$$

of arrows. We claim that it is cocartesian, i.e. forms a pushout square. This we check at each level (t, f) of $z/\mathcal{C}$. If t has degree less than z, then $(i^{(z,\mathsf{id}_z)})_{(t,f)}$ is an isomorphism. Isomorphisms are absorbing for the pushout product, so both source and target of (22) at stage (t, f) are isomorphisms, making the square a pushout. Otherwise, we have $(t, f) = (z, \mathsf{id}_z)$. Evaluation at (z, id_z) is given by T^*, and $T_! T^* F^* f \to F^* f$ becomes invertible upon application of T^*. Thus, the map (22) at stage (t, f) is an isomorphism, in particular cocartesian.

The functorial action of the pullback hom in its first argument takes cocartesian maps in its first argument to cartesian maps, i.e. sends pushout squares to pullback squares. Thus, the map $\widehat{\mathbf{Nat}}(i^{(z,\mathsf{id}_z)} \mathbin{\widehat{\times}} F^* f, F^* p)$ is a pullback of $\widehat{\mathbf{Nat}}(i^{(z,\mathsf{id}_z)} \mathbin{\widehat{\times}} T_! T^* F^* f, F^* p)$. Using part (i) of Lemma 3.10, it thus suffices to show that this map is a fibration. We calculate

$$\begin{aligned}
\widehat{\mathbf{Nat}}(i^{(z,\mathsf{id}_z)} \mathbin{\widehat{\times}} T_! T^* F^* f, F^* p) &\cong \widehat{\mathbf{Nat}}(T_! T^* F^* f, \widehat{\exp}(i^{(z,\mathsf{id}_z)}, F^* p)) \\
&\cong \widehat{\mathbf{Nat}}(T^* F^* f, T^* \widehat{\exp}(i^{(z,\mathsf{id}_z)}, F^* p)) \\
&\cong \widehat{\exp}(f_z, \widehat{\mathbf{Nat}}(i^{(z,\mathsf{id}_z)}, F^* p)) \\
&\cong \widehat{\exp}(f_z, \widehat{\mathbf{Nat}}(i^z, p))
\end{aligned}$$

using exponential transposition, the adjunction $T_! \dashv T^*$, the adjunction $F_! \dashv F^*$. By assumption, f_z is a cofibration. It thus remains to show that $\widehat{\mathbf{Nat}}(i^z, p)$ is a fibration. This holds by Lemma 4.7. □

Lemma 4.43. *Let $f : A \to B$ be a Reedy cofibration and $p : Y \to X$ a Reedy fibration of diagrams over $\mathcal{C}$. Suppose further that all Reedy fibrant diagrams on $\mathcal{C}$ have fibrant limits. Then, the pullback hom*

$$\boldsymbol{Nat}(B, Y) \xrightarrow{\widehat{\boldsymbol{Nat}}(f,p)} \boldsymbol{Nat}(B, X) \times_{\boldsymbol{Nat}(A,X)} \boldsymbol{Nat}(A, Y)$$

of p with f is a fibration.

Proof. By the assumption on f, the pullback exponential $\widehat{\exp}(f, p)$ is a Reedy fibration. The pullback hom $\widehat{\mathbf{Nat}}(f, p)$ is just the limit functor applies to this map. Since Reedy fibrant diagrams on $\mathcal{C}$ have fibrant limits, it is a fibration by Lemma 4.25. □

Corollary 4.44. *Let X be a Reedy fibrant diagram and A a pointwise cofibrant diagram. Then $[A, X]$ is Reedy fibrant. If furthermore Reedy fibrant diagrams on $\mathcal{C}$ have fibrant limits, then $\boldsymbol{Nat}(A, X)$ is fibrant.*

4.7 Complete semi-Segal types

The basic facts about diagrams developed in the previous subsections allow us to use two-level type theory to formulate a variation of the classically well-established theory of *Segal spaces*.

Normally, Segal spaces are employed to reason about higher categorical structures such as $(\infty, 1)$-categories in a homotopy-invariant fashion. In other words, Segal spaces are a model of $(\infty, 1)$-categories that can be constructed purely in terms of existing models of spaces (∞-groupoids) and their homotopy theory.

By contrast, a model such as the one based on *quasicategories* (Boardman and Vogt, 1973; Joyal, 2002) works by encoding higher categories in terms of their simplicial nerves. The difference is that the homotopy-theoretic features of complete Segal spaces are *inherited* from the environment

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

in which they are defined (i.e. spaces), whereas for quasicategories, they have to be imposed by an ad hoc construction (the Joyal model structure on simplicial sets).

In the setting of type theory, only the first kind of approach is possible, if we are aiming for a formulation of higher category theory that can talk about higher categorical structures within the theory. In this subsection, we will see how to import the ideas of the framework of complete Segal spaces into two-level type theory.

Complete Segal spaces (Rezk, 2001) are first of all simplicial objects. Unfortunately, the fact that our development of Reedy fibrancy is restricted to inverse categories, rather than more general Reedy categories, implies that we have to limit ourselves to the semisimplicial case.

It has been shown by Harpaz (2015) that semisimplicial spaces satisfying a Segal condition and a version of the completeness condition form a model of $(\infty, 1)$-categories, by constructing a Quillen equivalence between the appropriate Bousfield localisations of marked semisimplicial spaces and simplicial spaces.

Inspired by Harpaz's construction, we give the following definitions in two-level type theory:

Definition 4.45. Let $\tau : F \to G$ be a Reedy cofibration between semisimplicial types (i.e. diagrams over the semisimplicial category Δ_+^{op}), and X a Reedy fibrant semisimplicial type. Assume that G is bounded, in the sense that it is the left Kan extension of a diagram over $(\Delta_+^{\mathrm{op}})^{<n}$ for some n. We say that X is *local* with respect to τ if the induced map

$$\mathbf{Nat}(G, X) \to \mathbf{Nat}(F, X) \tag{23}$$

is a trivial fibration.

Note that the assumptions on τ and G imply that the map (23) is already a fibration.

Definition 4.46. A *semi-Segal type* is a Reedy fibrant semisimplicial type X that is local with respect to all *inner horn inclusions* $\Lambda^k[n] \to \Delta[n]$, with $0 < k < n$.

Since Δ_+^{op} is inverse, $\Delta[n]$ is bounded. Furthermore, $\Lambda^k[n] \to \Delta[n]$ is a pointwise decidable monomorphism, hence a pointwise cofibration by Corollary 3.22, and therefore a Reedy cofibration by Theorem 4.42.

Equivalently (and perhaps more elegantly), we can define the semi-Segal condition as locality with respect to the pushout corner map in the image under Yoneda of

$$\begin{array}{ccc} [0] & \xrightarrow{\{0\}} & [b] \\ \downarrow{\scriptstyle \{a\}} & & \downarrow{\scriptstyle \{a,\ldots,a+b\}} \\ [a] & \xrightarrow{\{0,\ldots,a\}} & [a+b] \end{array} \tag{24}$$

for all $a, b \geq 0$, that is:

$$\Delta[a] +_{\Delta[0]} \Delta[b] \xrightarrow{c_{a,b}} \Delta[a+b] \tag{25}$$

Here, we can equivalently restrict to $a = 1$.

However, semi-Segal types do not constitute the correct notion of $(\infty, 1)$-category that we are after, for essentially two reasons. Firstly, they are not *complete*, meaning that their type of vertices X_0 carries extra (non-categorical) information, i.e., they are not univalent, and secondly, they do not necessarily have identity arrows. They model $(\infty, 1)$-pre-semicategories.

It turns out, however, that one can define a notion of *equivalence* in a semi-Segal type, and using equivalences one can solve both problems at the same time.

For a semi-Segal type X, let $\overline{X}(x, y)$ be the type of edges that have vertices $x, y : X_0$ as endpoints. This is fibrant by the Reedy fibrancy condition. Thanks to locality with respect to the horn

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

inclusion $\Lambda^1[2] \to \Delta[2]$, we get a *composition map*:

$$- \circ - : \overline{X}(y,z) \times \overline{X}(x,y) \to \overline{X}(x,z).$$

Definition 4.47. Let X be a semi-Segal type. An *equivalence* in X is an edge $f : \overline{X}(x,y)$ such that for all vertices z, the functions $f \circ - : \overline{X}(z,x) \to \overline{X}(z,y)$ and $- \circ f : \overline{X}(y,z) \to \overline{X}(x,z)$ are equivalences (of fibrant types).

Since being an equivalence is a fibrant proposition, we get that equivalences in a semisimplicial type X form a fibrant type E, which we can think of as a subtype of X_1. We are now ready for the main definition.

Definition 4.48. A univalent $(\infty, 1)$-category is a semi-Segal type such that the source map $E \to X_0$ is an equivalence.

By source map in Definition 4.48, we mean the function mapping every equivalence $f : \overline{X}(x,y)$ to the source vertex x. The condition of Definition 4.48 is sometimes referred to as the *completeness condition*. One way to think of it is as a formulation of univalence internal to X. Since Definition 4.48 gives the only notion of $(\infty, 1)$-category that we consider here, we drop the attribute *univalent* for simplicity.

Note that Definitions 4.46–4.48 are all invariant under (levelwise) equivalence of Reedy fibrant semisimplicial types.

Two of the current authors have checked in detail that this definition is well-behaved, and equivalent to the manual definition that one might expect, for the truncated special cases of univalent ordinary categories and (2,1)-categories (Capriotti and Kraus, 2017). It is out of the scope of this paper to develop the theory of $(\infty, 1)$-category in two-level type theory. Therefore, we limit ourselves to sketching some basic examples of $(\infty, 1)$-categories, to give a taste of how our definition can be employed in practice.

For a given category $\mathcal{C}$, let $N_+(\mathcal{C})$ be the *semisimplicial nerve* of $\mathcal{C}$, i.e. the semisimplicial type whose n-simplices are given by functors $[n] \to \mathcal{C}$, where $[n]$ denotes the ordinal with $n+1$ elements, regarded as a category. Observe that the square (24) is a pushout in categories. It follows that $N_+(\mathcal{C})$ sends it to a pullback

$$\begin{array}{ccc} N_+(\mathcal{C})([a+b]) & \longrightarrow & N_+(\mathcal{C})([a]) \\ \downarrow & \lrcorner & \downarrow \\ N_+(\mathcal{C})([b]) & \longrightarrow & N_+(\mathcal{C})([0]). \end{array} \tag{26}$$

This is a strict version of the semi-Segal condition. We shall see below that under sufficient fibrancy conditions, it also forms a homotopy pullback and hence makes the Reedy fibrant replacement of $N_+(\mathcal{C})$ a semi-Segal type.

Lemma 4.49. *Let $\mathcal{C}$ be a category with slices that have fibrant types of objects. Then $N_+(\mathcal{C}) : \Delta_+^{op} \to \mathcal{U}$ sends the map $\{a, \ldots, a+b\} : [b] \to [a+b]$ to a fibration.*

Proof. By closure of fibrations under composition, it suffices to check the case $a = 1$. The map in question is the left map in (26). The right map is the *target map* $N_+(\mathcal{C})([1]) \to N_+(\mathcal{C})([0])$ induced by $\{1\} : [0] \to [1]$. This is a fibration by assumption: its fibre over $x : |\mathcal{C}|$ is isomorphic to $|\mathcal{C}/x|$. The claim follows since fibrations are closed under pullback. □

Corollary 4.50. *Let $\mathcal{C}$ be a category such that $\mathcal{C}$ and all its slices have fibrant types of objects. Then, $N_+(\mathcal{C}) : \Delta_+^{op} \to \mathcal{U}$ is valued in fibrant types.* □

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

From a pointwise fibrant semisimplicial type, we obtain a Reedy fibrant semisimplicial type (levelwise) equivalent to it using Lemma 4.26.

Lemma 4.51. *Let $\mathcal{C}$ be a category such that $\mathcal{C}$ and all its slices have fibrant types of objects. Let $j : N_+(\mathcal{C}) \to X$ be a Reedy fibrant replacement. Then X is a semi-Segal type.*

Proof. Using the map (25), we show that $\mathbf{Nat}(c_{a,b}, X)$ is an equivalence for all a, b. Consider the following square:

$$\begin{array}{ccc} \mathbf{Nat}(\Delta[a+b], N_+(\mathcal{C})) & \xrightarrow{\mathbf{Nat}(\Delta[a+b], j)} & \mathbf{Nat}(\Delta[a+b], X) \\ \downarrow{\scriptstyle \mathbf{Nat}(c_{a,b}, N_+(C))} & & \downarrow{\scriptstyle \mathbf{Nat}(c_{a,b}, X)} \\ \mathbf{Nat}(\Delta[a] +_{\Delta[0]} \Delta[b], N_+(\mathcal{C})) & \xrightarrow{\mathbf{Nat}(\Delta[a] +_{\Delta[0]} \Delta[b], j)} & \mathbf{Nat}(\Delta[a] +_{\Delta[0]} \Delta[b], X) \end{array}$$

By Yoneda, the left map is equivalently the pullback corner map in (26) and hence is invertible. The top left object is fibrant by Corollary 4.50, hence so is the bottom left object. Note that j evaluates to an equivalence at all of $[0], [a], [b], [a+b]$. Thus, the top map is an equivalence. The bottom map is the induced map between the pullbacks of two cospans of fibrant objects with one leg a fibration (by Lemma 4.49). The induced morphism between these cospans is (levelwise) an equivalence. By an argument analogous to the gluing lemma for fibration categories (Radulescu-Banu, 2006, Lemma 1.4.1, part (2)), the induced map between the two pullbacks is also an equivalence. Finally, since all other maps in the square are equivalences, so is $\mathbf{Nat}(c_{a,b}, X)$. □

Let $\mathcal{C}$ be a category with fibrant types of objects and morphisms (between any two given objects). As a consequence of Lemma 4.51, any Reedy fibrant replacement X of $N_+(\mathcal{C})$ is a semi-Segal type. We may start the construction of such a Reedy fibrant replacement with $X_0 :\equiv |\mathcal{C}|$ and $X_1(x, y) :\equiv \mathcal{C}(x, y)$. It is then easy to check that the composition map for X defined before Definition 4.47 agrees with the composition of $\mathcal{C}$. In particular, an edge in X is an equivalence exactly if the corresponding morphism in $\mathcal{C}$ is a *homotopy equivalence* (invertible up to inner equality). It follows that X is univalent exactly if $\mathcal{C}$ is *wildly univalent*, that is, the canonical map from $|\mathcal{C}|$ to the type of equivalences of $\mathcal{C}$ is an equivalence.

As an important special case of this construction, we can take for $\mathcal{C}$ the category of fibrant types. The resulting $(\infty, 1)$-category TYPE (large, but locally small) can be regarded as a classifier for families of types over $(\infty, 1)$-categories.

One way to construct a universal fibration $\mathsf{TYPE}^\bullet$ over TYPE is as follows. Let $\mathcal{U}^\bullet$ denote the category of fibrant types with an element (note that morphisms preserve the element strictly). We have a forgetful functor $\mathcal{U}^\bullet \to \mathcal{U}$ that is Reedy fibrant on underlying graphs. Taking a relative Reedy fibrant replacement, we obtain the following square:

$$\begin{array}{ccc} \mathsf{TYPE}^\bullet & \longrightarrow & N_+(\mathcal{U}^\bullet) \\ \downarrow & & \downarrow \\ \mathsf{TYPE} & \longrightarrow & N_+(\mathcal{U}). \end{array}$$

By a relative version of Lemma 4.51 for homotopy left fibrations, the fibration $\mathsf{TYPE}^\bullet \twoheadrightarrow \mathsf{TYPE}$ is a left fibration. One may go on to show that it is a classifier for left fibrations with small fibres. This gives one way to adapt the Grothendieck construction for presheaves of $(\infty, 1)$-categories to our settings. (An alternative is to directly define the universe of left fibrations and check the semi-Segal condition and univalence).

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

5. Conclusions

We believe that two-level type theory is a suitable framework for expressing and proving results which, in conventional homotopy type theory, require externally fixed data that one wishes to keep as variable as possible. We have demonstrated that this approach can be used effectively to express Shulman's results on diagrams over inverse categories (Shulman, 2015*b*). Starting from there, we have suggested the very beginning of an internal development of a theory of $(\infty, 1)$-categories. We expect that such a theory is helpful for other constructions which make use of the fact that types and universes are, naturally, higher categories; a short discussion is available in Kraus (2018).

Examples for existing results which can be expressed in and benefit from our framework of higher categories can be found in our previous work (Kraus, 2015*a*,b; Kraus and Sattler, 2017). These results use semisimplicial types to express large or even infinite towers of coherences, and it is unknown how to express such towers in standard settings of homotopy type theory. If we do these constructions in our suggested setting, it is important which precise version of two-level type theory we use. If we only use "basic" two-level type theory without any of the strengthenings discussed in Subsection 2.4, then the conservativity result means that we immediately get the corresponding result in homotopy type theory. For the results cited above, this is the case if the size of required coherence towers is bounded, with a bound given as an outer natural number; then, the construction works in homotopy type theory, with the bound fixed externally. An example for this situation is Kraus (2015*b*, Theorem 8.9.6). Other results however need one or more of the strengthenings of Subsection 2.4, and for those, it is in general unknown whether they can be expressed in usual settings of homotopy type theory. In case of the mentioned work, an assumption made for some results is that limits of Reedy fibrant towers are fibrant (A2), but we expect that this assumption can alternatively be substituted by the axiom (A3) that the outer natural numbers are cofibrant or even fibrant (A1). To give an example, Kraus (2015*b*, Theorem 8.8.5) depends on such an assumption. Possible future directions include developing a richer theory of $(\infty, 1)$-categories that includes standard concepts such as limits and colimits, and potentially based on that, a treatment of the internal semantics of higher inductive types as for example specified by Kaposi and Kovács (2020).

As a proof of concept, we have implemented some parts of our paper (with the main result being Theorem 4.8) in the proof assistant Lean.[10] Since Lean does not support two-level type theory directly, we have used type classes to keep track of and automatically propagate fibrancy constraints. An overall idea of the implementation is suitable for most existing proof assistants: we work in a type theory with universes of outer types (i.e. where UIP holds), outer types correspond to the ordinary types of the proof assistant, while fibrant types are represented as types "tagged" with the extra structure of being fibrant. The role of the outer equality is played by the ordinary propositional equality of the proof assistant (which, thanks to UIP, is indeed propositional in the sense of HoTT). We postulate the fibrant equality type, its elimination rule J and fibrancy preservation rules for Π and Σ resulting from the rules in Subsection 2.1. The usual computation (or β-) rule for J is defined using outer equality – the propositional equality of the proof assistant – and not judgemental equality. This means that this computation does a priori not happen automatically, and explicit rewrites along the propositional β-rules are needed in proof implementations when working in the inner level. In our development, besides the general two-level framework, we have implemented machinery required to define Reedy fibrant diagrams and have fully formalised a proof of Theorem 4.8. We did not find the lack of a definitional β-rule for J in the inner fragment to affect the internalisation of results on the theory of Reedy fibrant diagrams we have developed. For other formalisation approaches to 2LTT, we refer to Subsection 1.1 above.

Acknowledgements. We would like to thank Benedikt Ahrens, Thorsten Altenkirch, Simon Boulier and Michael Shulman for many interesting discussions and insightful comments. We also thank the anonymous referees for very helpful comments.

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

Funding notes. This work has been supported by

- the Royal Society, grant reference URF\R1\191055;
- the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations, Grounding Innovation in Informatics and Infocommunication)
- the Engineering and Physical Sciences Research Council (EPSRC), grant reference EP/M016994/1;
- USAF, Airforce office for scientific research, award FA9550-16-1-0029;
- and the HIPERFIT Research Centre, Danish Council for Strategic Research, contract number 10-092299.

Notes

1 Alternative terminology: Some authors refer to outer types as *pretypes* (Altenkirch et al., 2016), *strict types*, *non-fibrant types* or *exotypes* (Ahrens et al., 2021). Inner types are also simply referred to as *types* or *fibrant types*. In this paper, the term *fibrant* is reserved for outer types that are isomorphic to inner types.

2 The code is available at `https://github.com/annenkov/two-level`.

3 A shorter and earlier version of this work is also available (Ahrens et al., 2020).

4 A pull request to Agda's repository is available at `https://github.com/agda/agda/pull/4091`.

5 In a previous version of this article, inner components were annotated with o (e.g. Σ^{o}) instead of i (Σ^{i}). Other authors don't annotate the inner components at all and annotate *outer* type formers instead, e.g. Σ^{s} (*s* for *strict*) (Altenkirch et al., 2016) or Σ^{e} (*e* for *exo*) (Ahrens et al., 2021).

6 Not all authors assume the judgmental η-law for Σ-types, with the possibly most prominent example in the current context being the HoTT book (The Univalent Foundations Program, 2013). This law simplifies our theory somewhat: see (2) of 2.11 and the discussion in remark 2.12. In its absence, we would work with telescopes of inner types as a substitute for inner dependent sums.

7 Of course, the notion of a two-level model could be weakened to let c only preserve the universal property of context extension, not the context extension operation itself; i.e. one could require $\Gamma.A \simeq \Gamma.\mathrm{c}(A)$ instead of an equality. Similarly, one could consider two separated multi-sorted cwfs with a morphism that induces an equivalence on the categories of contexts. However, all our models in Subsection 2.5 satisfy the rather strict requirements of Definitions 2.8 and 2.9.

8 The introduction of Voevodsky's note (Voevodsky, 2013) states: *We call this system and its further extensions HTS for "homotopy type system". It is an extension of the Martin-Lof type system with some additional constructs which reflect the structures which exist in the target of the canonical univalent model of the Martin-Lof system.*

9 This can be rectified by reinterpreting the inner types as $\overline{\mathsf{Ty}}^{\mathsf{i}}_j = \coprod_{j \leq i} \mathsf{Ty}^{\mathsf{i}}_j$. Type forming operations in $\overline{\mathsf{Ty}}^{\mathsf{i}}_j$ are interpreted by lifting all input types to their maximum size index. Fibrancy of universes follows from closure of Kan complexes under finite coproducts. In this way, one can avoid the additional Grothendieck universe M_ω introduced below. It also enforces (M1). The same technique can be applied to the outer level.

10 Our development uses Lean 2: `https://github.com/leanprover/lean2/`.

References

Ahrens, B., Kapulkin, K. and Shulman, M. (2015). Univalent categories and the Rezk completion. *Mathematical Structures in Computer Science (MSCS)*, 1–30.

Ahrens, B., North, P. R., Shulman, M. and Tsementzis, D. (2020). A higher structure identity principle. In: *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS'20, New York, NY, USA, Association for Computing Machinery, 53–66. `https://doi.org/10.1145/3373718.3394755`.

Ahrens, B., North, P. R., Shulman, M. and Tsementzis, D. (2021). The univalence principle. ArXiv e-prints.

Altenkirch, T., Capriotti, P., Dijkstra, G., Kraus, N. and Forsberg, F. N. (2018). Quotient inductive-inductive types. In: *Foundations of Software Science and Computation Structures (FoSSaCS 2018)*, 293–310.

Altenkirch, T., Capriotti, P. and Kraus, N. (2016). Extending homotopy type theory with strict equality. In: Talbot, J.-M. and Regnier, L. (eds.) *25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*, vol. 62, 21:1–21:17.

Altenkirch, T., Danielsson, N. A. and Kraus, N. (2017). Partiality, revisited. In: Esparza, J. and Murawski, A. S. (eds.) *Foundations of Software Science and Computation Structures: 20th International Conference, FOSSACS 2017, Proceedings*, Springer Berlin Heidelberg, 534–549.

Altenkirch, T. and Kaposi, A. (2016). Type theory in type theory using quotient inductive types. In: *Principles of Programming Languages (POPL'16)*, vol. 51, ACM, 18–29.

Angiuli, C., Hou (Favonia), K.-B. and Harper, R. (2018). Cartesian cubical computational type theory: Constructive reasoning with paths and equalities. In: Ghica, D. and Jung, A. (eds.) *27th EACSL Annual Conference on Computer Science Logic (CSL 2018)*, Leibniz International Proceedings in Informatics (LIPIcs), vol. 119, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 6:1–6:17.

Assaf, A., Burel, G., Cauderlier, R., Delahaye, D., Dowek, G., Dubois, C., Gilbert, F., Halmagrand, P., Hermant, O. and Saillard, R. (2016). Dedukti: a logical framework based on the $\lambda\pi$-calculus modulo theory. Manuscript http://www.lsv.fr/~dowek/Publi/expressing.pdf.

Barras, B. and Maestracci, V. (2020). Implementation of two layers type theory in dedukti and application to cubical type theory. In: *LFMPT 2020 - Logical Frameworks and Meta-Languages: Theory and Practice 2020*, Paris, France.

Bauer, A., Gilbert, G., Haselwarter, P., Pretnar, M. and Stone, C. Andromeda. Implementation of a type theory with equality reflection. http://andromedans.github.io/andromeda/.

Bertot, Y. (2013). Private inductive types: Proposing a language extension. http://coq.inria.fr/files/coq5_submission_3.pdf.

Bertot, Y. and Castéran, P. (2010). *Interactive Theorem Proving and Program Development: Coq'Art: The Calculus of Inductive Constructions*, EATCS Texts in Theoretical Computer Science, Springer-Verlag, Springer Berlin, Heidelberg.

Bezem, M., Coquand, T. and Huber, S. (2014). A model of type theory in cubical sets. In: Matthes, R. and Schubert, A. (eds.) *Types for Proofs and Programs (TYPES)*, Leibniz International Proceedings in Informatics (LIPIcs), vol. 26, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 107–128.

Boardman, M. and Vogt, R. (1973). Homotopy Invariant Algebraic Structures on Topological Spaces, *Lecture Notes in Mathematics*, vol. 347, Springer Berlin, Heidelberg.

Bonacina, R. and Ahrens, B. (2021). Syntax for two-level type theory. Abstract, presented at TYPES'21.

Bonacina, R., Ahrens, B. and Kraus, N. (2021). Syntax for two-level type theory. Abstract, presented at HoTT/UF'21.

Boulier, S. and Tabareau, N. (2017). Model structure on the universe in a two level type theory. HAL e-prints. <hal-01579822>.

Brunerie, G., de Boer, M., Lumsdaine, P. L. and Mörtberg, A. (2019). A formalization of the initiality conjecture in agda. Talk given by Brunerie at the HoTT 2019 conference, slides available at https://guillaumebrunerie.github.io/pdf/initiality.pdf.

Brunerie, G. and Lumsdaine, P. L. (2018). Formalising the initiality conjecture in coq and agda. Talk at the Stockholm-Göteborg Type Theory Serminar.

Brunerie, G. and Lumsdaine, P. L. (2020). Initiality for martin-löf type theory. Talk at the Homotopy Type Theory Electronic Seminar Talks (HOTTEST).

Capriotti, P. (2016). *Models of Type Theory with Strict Equality*. Phd thesis, School of Computer Science, University of Nottingham.

Capriotti, P. and Kraus, N. (2017). Univalent higher categories via complete semi-segal types. *Proceedings of the ACM on Programming Languages* **2** (POPL'18) 44:1–44:29. Full version available at https://arxiv.org/abs/1707.03693.

Cohen, C., Coquand, T., Huber, S. and Mörtberg, A. (2017). Cubical type theory: A constructive interpretation of the univalence axiom. *IfCoLog Journal of Logics and their Applications* **4** (10) 3127–3169.

Coquand, T., Huber, S. and Mörtberg, A. (2018). On higher inductive types in cubical type theory. In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, ACM, 255–264.

Coquand, T., Huber, S. and Sattler, C. (2019). Homotopy canonicity for cubical type theory. In: *4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

de Boer, M. (2020). *A Proof and Formalization of the Initiality Conjecture of Dependent Type Theory*. Phd thesis, Stockholm University, Faculty of Science, Department of Mathematics, Stockholm, Sweden. Available online at https://su.diva-portal.org/smash/record.jsf?pid=diva2%3A1431287.

de Moura, L., Kong, S., Avigad, J., van Doorn, F. and von Raumer, J. (2015). The lean theorem prover. In: *Automated Deduction - CADE-25, 25th International Conference on Automated Deduction*.

Dybjer, P. (1995). Internal type theory. In: Berardi, S. and Coppo, M. (eds.) *Types for Proofs and Programs (TYPES)*, Lecture Notes in Computer Science, vol. 1158, Springer-Verlag, 120–134.

Gilbert, G., Cockx, J., Sozeau, M. and Tabareau, N. (2019). Definitional proof-irrelevance without K. *Proceedings of the ACM on Programming Languages* **3** (POPL) 3:1–3:28.

Harpaz, Y. (2015) Quasi-unital ∞–categories. *Algebraic & Geometric Topology* **15** (4) 2303–2381.

Hofmann, M. (1997). Syntax and semantics of dependent types. In: *Semantics and Logics of Computation*, Cambridge University Press, 79–130.

Hofmann, M. and Streicher, T. (1997). Lifting grothendieck universes. Unpublished note. Available at https://www2.mathematik.tu-darmstadt.de/~streicher/NOTES/lift.pdf.

Joyal, A. (2002). Quasi-categories and Kan complexes. *Journal of Pure and Applied Algebra* **175** 207–222.

Kaposi, A. and Kovács, A. (2020). Signatures and induction principles for higher inductive-inductive types. *Logical Methods in Computer Science* **16** (1). https://lmcs.episciences.org/6100.

Kapulkin, C. and Lumsdaine, P. L. (2018). The simplicial model of Univalent Foundations (after Voevodsky). ArXiv e-prints.

Kovács, A. (2022). Staged compilation with two-level type theory. *Proceedings of the ACM on Programming Languages* **6** (ICFP), 540–569.

Kraus, N. (2015a). The general universal property of the propositional truncation. In: Herbelin, H., Letouzey, P. and Sozeau, M. (eds.) *20th International Conference on Types for Proofs and Programs (TYPES 2014)*, Leibniz International Proceedings in Informatics (LIPIcs), vol. 39, Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 111–145.

Kraus, N. (2015b). *Truncation Levels in Homotopy Type Theory*. Phd thesis, School of Computer Science, University of Nottingham, Nottingham, UK.

https://doi.org/10.1017/S0960129523000130 Published online by Cambridge University Press

Kraus, N. (2018). On the role of semisimplicial types. Abstract, presented at TYPES'18.
Kraus, N. (2021). Internal ∞-categorical models of dependent type theory: Towards 2LTT eating HoTT. In: *Symposium on Logic in Computer Science (LICS 2021)*, 1–14.
Kraus, N. and Sattler, C. (2017). Space-valued diagrams, type-theoretically (extended abstract). ArXiv e-prints.
Licata, D. R., Orton, I., Pitts, A. M. and Spitters, B. (2018). Internal universes in models of homotopy type theory. In: Kirchner, H. (ed.) *3rd International Conference on Formal Structures for Computation and Deduction (FSCD 2018)*, Leibniz International Proceedings in Informatics (LIPIcs), vol. 108, Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 22:1–22:17.
Lumsdaine, P. L. and Mörtberg, A. (2018). Formalising the initiality conjecture in coq. Talk given by Lumsdaine at the Göteborg-Stockholm Joint Type Theory Seminar, slides available at http://peterlefanulumsdaine.com/research/Lumsdaine-2018-Goteborg-Initiality.pdf.
Lumsdaine, P. L. and The Univalent Foundations Program. (2013). Semi-simplicial types. Wiki page of the Univalent Foundations project at the Institute for Advanced Studies, https://uf-ias-2012.wikispaces.com/Semi-simplicial+types.
Maietti, M. E. (2009). A minimalist two-level foundation for constructive mathematics. *Annals of Pure and Applied Logic* **160** (3) 319–354. Computation and Logic in the Real World: CiE 2007.
Maietti, M. E. and Sambin, G. (2005). Toward a minimalist foundation for constructive mathematics. In: *From Sets and Types to Topology and Analysis: Towards Practicable Foundations for Constructive Mathematics*, Oxford University Press.
Makkai, M. (1995). First order logic with dependent sorts, with applications to category theory. preprint, available at http://www.math.mcgill.ca/makkai.
Norell, U. (2007). *Towards a Practical Programming Language Based on Dependent Type Theory*. Phd thesis, Department of Computer Science and Engineering, Chalmers University of Technology and Göteborg University.
Pitts, A. M. and Orton, I. (2018). Axioms for modelling cubical type theory in a topos. *Logical Methods in Computer Science* **14** (4). https://doi.org/10.23638/LMCS-14(4:23)2018.
Radulescu-Banu, A. (2006). Cofibrations in homotopy theory. arXiv preprint math/0610009.
Reedy, C. L. (1974). Homotopy theory of model categories. Unpublished manuscript. Available at http://www-math.mit.edu/~psh/reedy.pdf.
Rezk, C. (2001). A model for the homotopy theory of homotopy theory. *Transactions of the American Mathematical Society* **353** (3) 973–1007 (electronic).
Riehl, E. and Shulman, M. (2017). A type theory for synthetic ∞-categories. *Higher Structures* **1** (1) 147–224.
Riehl, E. and Verity, D. (2014). The theory and practice of Reedy categories. *Theory and Applications of Categories* **29** (9) 256–301.
Shulman, M. (2015a). The univalence axiom for elegant reedy presheaves. *Homology, Homotopy and Applications* **17** (2) 81–106.
Shulman, M. (2015b). Univalence for inverse diagrams and homotopy canonicity. *Mathematical Structures in Computer Science* **25** (5) 1203–1277.
Shulman, M. (2018). Brouwer's fixed-point theorem in real-cohesive homotopy type theory. *Mathematical Structures in Computer Science* **28** (6) 856–941.
Shulman, M. (2019). All $(\infty, 1)$-toposes have strict univalent universes. arXiv preprint arXiv:1904.07004.
Streicher, T. (1993). Investigations into intensional type theory. Habilitationsschrift, Ludwig-Maximilians-Universität München.
The Univalent Foundations Program. (2013). *Homotopy Type Theory: Univalent Foundations of Mathematics*. http://homotopytypetheory.org/book/.
Uemura, T. (2019). A general framework for the semantics of type theory. arXiv preprint arXiv:1904.04097.
Uskuplu, E. (2022). An implementation of 2LTT in Agda. Abstract, presented at HoTT/UF'22.
Voevodsky, V. (2013). A simple type system with two identity types. Unpublished note.

Cite this article: Annenkov D, Capriotti P, Kraus N and Sattler C (2023). Two-level type theory and applications. *Mathematical Structures in Computer Science* **33**, 688–743. https://doi.org/10.1017/S0960129523000130