Math. Struct. in Comp. Science (2009), *vol.* 19, *pp.* 687–736.
doi:10.1017/S0960129509007646 First published online 2 July 2009 Printed in the United Kingdom

Two-dimensional models of type theory

RICHARD GARNER[†]

Department of Pure Mathematics and Mathematical Statistics,
Wilberforce Road, Cambridge CB3 0WB, United Kingdom
Email: rhgg2@cam.ac.uk

Received 11 September 2008; revised 21 January 2009

We describe a non-extensional variant of Martin-Löf type theory, which we call *two-dimensional type theory*, and equip it with a sound and complete semantics valued in 2-categories.

1. Introduction

This is the second in a series of papers detailing the author's investigations into the intensional type theory of Martin-Löf, as described in Nordström *et al.* (1990). The first of these papers, Garner (2009), investigated syntactic issues relating to its dependent product types. The present paper is a contribution to its categorical semantics.

Seely (1984) proposed that the correct categorical models for extensional Martin-Löf type theory should be locally cartesian closed categories: these being categories $\mathscr{C}$ with finite limits in which each of the functors $f^* : \mathscr{C}/X \to \mathscr{C}/Y$ induced by pulling back along a morphism $f : Y \to X$ has a right adjoint. The idea is to think of each object X of a locally cartesian closed category $\mathscr{C}$ as a closed type, each morphism as a term and each object of the slice category $\mathscr{C}/X$ as a type dependent upon X. Now substitution of terms in types may be interpreted by pullback between the slices of $\mathscr{C}$, dependent sum and product types by left and right adjoints to pullback and the equality type on X by the diagonal morphism $\Delta : X \to X \times X$ in $\mathscr{C}/X \times X$. It was later pointed out in Hofmann (1995b) that this picture, whilst very appealing, is not wholly accurate, since in the syntax, the operation that assigns to each morphism of types $f : Y \to X$ the corresponding substitution operation $\mathbf{Type}(X) \to \mathbf{Type}(Y)$ is strictly functorial in f; whilst in the semantics, the corresponding assignation $(f : Y \to X) \mapsto (f^* : \mathscr{C}/X \to \mathscr{C}/Y)$ is rarely so. Thus, this notion of model is not *sound* for the syntax, and we are forced to refine it slightly: essentially by equipping our locally cartesian closed category with a split fibration $\mathscr{T} \to \mathscr{C}$ equivalent to its codomain fibration $\mathscr{C}^{\mathbf{2}} \to \mathscr{C}$. Types over X are now interpreted as objects of the fibre category $\mathscr{T}(X)$; and since $\mathscr{T} \to \mathscr{C}$ is a split fibration, the interpretation is sound for substitution.

The question of how the above should generalise from extensional to intensional Martin-Löf type theory is a delicate one. It is possible to paraphrase the syntax of intensional type theory in categorical language and so arrive at a notion of model – as

[†] Supported by a Research Fellowship of St John's College, Cambridge and a Marie Curie Intra-European Fellowship, Project No. 040802.

done, for example, in Dybjer (1996) or Hofmann (1995a) – but then we lose sight of a key aspect of the extensional semantics, namely that dependent sums and products may be characterised universally, as left and right adjoints to substitution. To obtain a similar result for the intensional theory requires a more refined sort of semantics. More specifically, we are thinking of a semantics valued in higher-dimensional categories, motivated by works such as Awodey and Warren (2009), Gambino and Garner (2008) and Hofmann and Streicher (1998), which identify in intensional type theory certain higher-dimensional features. The idea is that, in such a semantics, we should be able to characterise dependent sums and products universally in terms of weak, higher-dimensional adjoints to substitution.

Eventually, we expect to be able to construct a sound and complete semantics for intensional type theory valued in weak ω-categories. At the moment the theory of weak ω-categories is insufficiently well developed for us to describe this semantics, though we can at least take steps towards it by describing semantics valued in simpler kinds of higher-dimensional category. In this paper, we describe such a semantics valued in 2-categories, which, as well as objects representing types and morphisms $f : X \to Y$ representing terms, has 2-cells $\alpha : f \Rightarrow g$ representing witnesses for the propositional equality of terms f and g. Intuitively, the 2-categorical models we consider provide a notion of two-dimensional locally cartesian closed category, though, bearing in mind the above concerns regarding the functoriality of substitution, it is in fact a 'split' notion of two-dimensional model that we will describe here. Relating this to a notion of two-dimensional local cartesian closed category will require a 2-categorical coherence result along the lines of Hofmann (1995b), but we defer this to a subsequent paper.

Our 2-categorical semantics is sound and complete neither for intensional nor extensional type theory, but rather for an intermediate theory, which we call *two-dimensional type theory*. Recall that extensional type theory distinguishes itself from intensional type theory by its admission of an *equality reflection rule*, which states that any two terms of type A that are propositionally equal are also definitionally equal. The two-dimensional type theory that we will consider admits instances of the equality reflection rule at just those types that are themselves identity types. The effect of this is to collapse the higher-dimensional aspects of the intensional theory, but only above dimension two, and it is this that allows a complete semantics in 2-categories. The leading example of a model for our semantics is the *groupoid model* of Hofmann and Streicher (1998); indeed, it plays the same fundamental role for two-dimensional type theory as the **Set**-based model does for extensional type theory. However, we expect there to be many more examples: on the categorical side, *prestack* and *stack* models, which will provide two-dimensional analogues of the *presheaf* and *sheaf* models of extensional type theory; and on the type-theoretic side, an *E-groupoid* model (Aczel 1994), which extends to two dimensions the *setoid* model of extensional type theory (Hofmann 1994). Once again, the task of describing these models will be deferred to a subsequent paper.

Our hope is that the semantics we describe in this paper will provide a useful guide in setting up more elaborate semantics for intensional type theory: both of the ω-categorical kind outlined above, and of the homotopy-theoretic kind espoused in Awodey and Warren (2009). Indeed, most of the problematic features of these higher-dimensional semantics are fully alive in the two-dimensional case – in particular, the rather subtle

issues regarding stability of structure under substitution – and the analysis we give of them here should prove useful in understanding these more general situations.

The paper is set out as follows. In Section 2, we review the syntax of intensional and extensional Martin-Löf type theory and describe our intermediate two-dimensional theory, ML_2. In Sections 3 and 4, we describe a 2-categorical structure built from the syntax of ML_2. Section 3 makes use of the non-logical rules of ML_2 together with the rules for identity types in order to construct: a 2-category $\mathfrak{C}$ of contexts; a two-dimensional fibration $\mathfrak{T} \to \mathfrak{C}$ of types over contexts; and a comprehension 2-functor $E : \mathfrak{T} \to \mathfrak{C}^{\mathbf{2}}$, sending each type-in-context $\Gamma \vdash A$ type to the corresponding *dependent projection* map $(\Gamma, x : A) \to \Gamma$. So far we have given nothing more than a simple-minded extension of the one-dimensional semantics; the twist is that each dependent projection in our 2-categorical model carries the structure of a *normal isofibration*. This can be seen as the semantic correlate of the *Leibniz rule* in dependent type theory. Section 4 considers the extra structure imposed on this basic framework by the logical rules of ML_2. The identity types are characterised as *arrow objects* in the slices of the 2-category of contexts; whilst the unit type, dependent sums and dependent products admit description in terms of a notion of weak 2-categorical adjointness, which we call *retract biadjunction*. Where a plain adjunction concerns itself with isomorphisms of hom-sets $\mathbb{C}(FX, Y) \cong \mathbb{D}(X, GY)$, a retract biadjunction instead requires retract equivalences of hom-categories $\mathbf{C}(FX, Y) \simeq \mathbf{D}(X, GY)$. In particular, dependent sums and products are characterised as left and right retract biadjoints to weakening. These syntactic investigations lead us to define a notion of *model* for two-dimensional type theory, this being an arbitrary 2-fibration $\mathfrak{T} \to \mathfrak{C}$ equipped with the structure outlined above, and the results of Sections 3 and 4 can be summarised as saying that to each type theory S extending ML_2, we may assign a *classifying* two-dimensional model $\mathbb{C}(\mathrm{S})$. In Section 5, we provide a converse to this result by showing that we can assign to each two-dimensional model $\mathbb{C}$ a two-dimensional type theory $\mathrm{S}(\mathbb{C})$ that represents the model faithfully. We call this type theory the *internal language* of $\mathbb{C}$. Finally, we show that these two constructions – classifying model and internal language – give rise to a *functorial semantics* in the sense of Lawvere (1968), which is to say that they induce an equivalence between suitably defined categories of two-dimensional type theories, and of two-dimensional models.

2. Intensional, extensional and two-dimensional type theory

2.1. *Intensional type theory*

When we refer to *intensional Martin-Löf type theory*, we mean the logical calculus set out in Part II of Nordström *et al.* (1990). In this paper we consider only the core calculus ML_I, with type-formers for dependent sums, dependent products, identity types and the unit type. We will now summarise this calculus, partly to fix notation and partly because there are a few peculiarities worth commenting on. The calculus has four basic forms of judgement: A type ('A is a type'); $a : A$ ('a is an element of the type A'); $A = B$ type ('A and B are definitionally equal types'); and $a = b : A$ ('a and b are definitionally equal elements of the type A'). These judgements may be made either absolutely or relative to

a context Γ of assumptions, and in the latter case we write them as

$$\Gamma \vdash A \text{ type}, \qquad \Gamma \vdash a : A, \qquad \Gamma \vdash A = B \text{ type} \qquad \text{and} \qquad \Gamma \vdash a = b : A,$$

respectively. Here, a *context* is a list $\Gamma = (x_1 : A_1,\ x_2 : A_2,\ \ldots,\ x_n : A_n)$, where each A_i is a type relative to the context $(x_1 : A_1,\ \ldots,\ x_{i-1} : A_{i-1})$. There are now some rather natural requirements for well-formed judgements: in order to assert that $a : A$, we must first know that A type; to assert that $A = B$ type, we must first know that A type and B type; and so on. We specify intensional Martin-Löf type theory as a collection of inference rules over these forms of judgement. First, we have the *equality rules*, which assert that the two judgement forms $A = B$ type and $a = b : A$ are congruences with respect to all the other operations of the theory. Then we have the *structural rules*, which deal with weakening, contraction, exchange and substitution[†]. Finally, we have the *logical rules*, which we list in Table 1. Note that we commit the usual abuse of notation in leaving implicit an ambient context Γ common to the premisses and conclusions of each rule. We also omit the rules expressing stability under substitution in this ambient context. We will find it convenient to use the following extended forms of the identity elimination and computation rules:

$$\frac{\begin{array}{c} x, y : A,\ p : \mathsf{Id}_A(x, y),\ \Delta \vdash C(x, y, p) \text{ type} \\ x : A,\ \Delta[x, x, \mathrm{r}(x)/x, y, p] \vdash d(x) : C(x, x, \mathrm{r}(x)) \end{array}}{x, y : A,\ p : \mathsf{Id}_A(x, y),\ \Delta \vdash \mathrm{J}_d(x, y, p) : C(x, y, p)}$$
$$\frac{\begin{array}{c} x, y : A,\ p : \mathsf{Id}_A(x, y),\ \Delta \vdash C(x, y, p) \text{ type} \\ x : A,\ \Delta[x, x, \mathrm{r}(x)/x, y, p] \vdash d(x) : C(x, x, \mathrm{r}(x)) \end{array}}{x : A,\ \Delta[x, x, \mathrm{r}(x)/x, y, p] \vdash \mathrm{J}_d(x, x, \mathrm{r}(x)) = d(x) : C(x, x, \mathrm{r}(x))} \tag{1}$$

These rules may be derived from the elimination and computation rules in Table 1 by using the Π-types to shift the additional contextual parameter Δ onto the right-hand side of the turnstile.

Notation 2.1.1. We may omit from the premisses of a rule or deduction any hypothesis that may be inferred from later hypotheses of that rule. Where it improves clarity, we may omit brackets in function applications, writing $hgfx$ in place of $h(g(f(x)))$, for example. We may drop the subscript A in an identity type $\mathsf{Id}_A(a, b)$ where no confusion seems likely to occur. We may write a sum type $\Sigma x : A.\, B(x)$ as $\Sigma(A, B)$, a product type $\Pi x : A.\, B(x)$ as $\Pi(A, B)$, and a λ-abstraction $\lambda x.\, f(x)$ as $\lambda(f)$ (or using our applicative convention, simply λf). It will occasionally be useful to perform lambda-abstraction at the meta-theoretic level, for instance writing $[x]\, f(x)$ to denote a term f of the form $x : A \vdash f(x) : B(x)$. We may write $\Gamma \vdash a \approx b : A$ to indicate that the type $\Gamma \vdash \mathsf{Id}_A(a, b)$ is inhabited, and say that a and b are *propositionally equal*. We will also make use of *vector notation* in the style of de Bruijn (1991). Given a context $\Gamma = (x_1 : A_1, \ldots, x_n : A_n)$, we may abbreviate a series of judgements

$$\vdash a_1 : A_1, \qquad \vdash a_2 : A_2(a_1), \qquad \ldots \qquad \vdash a_n : A_n(a_1, \ldots, a_{n-1})$$

as $\vdash a : \Gamma$, where $a := (a_1, \ldots, a_n)$, and say that a is a *global element* of Γ. We may also use this notation to abbreviate sequences of hypothetical elements on the left-hand

[†] Note, in particular, that we take substitution to be a *primitive* rather than a *derived* operation as done in Jacobs (1999), for instance.

Dependent sum types

$$\frac{A\ \mathsf{type} \qquad x : A \vdash B(x)\ \mathsf{type}}{\Sigma x : A.\, B(x)\ \mathsf{type}}\ \Sigma\text{-FORM} \qquad \frac{a : A \qquad b : B(a)}{\langle a, b\rangle : \Sigma x : A.\, B(x)}\ \Sigma\text{-INTRO}$$

$$\frac{z : \Sigma x : A.\, B(x) \vdash C(z)\ \mathsf{type} \qquad x : A,\ y : B(x) \vdash d(x, y) : C(\langle x, y\rangle)}{z : \Sigma x : A.\, B(x) \vdash \mathrm{E}_d(z) : C(z)}\ \Sigma\text{-ELIM}$$

$$\frac{z : \Sigma x : A.\, B(x) \vdash C(z)\ \mathsf{type} \qquad x : A,\ y : B(x) \vdash d(x, y) : C(\langle x, y\rangle)}{x : A,\ y : B(x) \vdash \mathrm{E}_d(\langle x, y\rangle) = d(x, y) : C(\langle x, y\rangle)}\ \Sigma\text{-COMP}$$

Unit type

$$\frac{}{\mathbf{1}\ \mathsf{type}}\ \mathbf{1}\text{-FORM} \qquad \frac{}{\star : \mathbf{1}}\ \mathbf{1}\text{-INTRO}$$

$$\frac{z : \mathbf{1} \vdash C(z)\ \mathsf{type} \qquad d : C(\star)}{z : \mathbf{1} \vdash \mathrm{U}_d(z) : C(z)}\ \mathbf{1}\text{-ELIM}$$

$$\frac{z : \mathbf{1} \vdash C(z)\ \mathsf{type} \qquad d : C(\star)}{\mathrm{U}_d(\star) = d : C(\star)}\ \mathbf{1}\text{-COMP}$$

Identity types

$$\frac{A\ \mathsf{type} \qquad a, b : A}{\mathsf{Id}_A(a, b)\ \mathsf{type}}\ \mathsf{Id}\text{-FORM} \qquad \frac{A\ \mathsf{type} \qquad a : A}{\mathrm{r}(a) : \mathsf{Id}_A(a, a)}\ \mathsf{Id}\text{-INTRO}$$

$$\frac{x, y : A,\ p : \mathsf{Id}_A(x, y) \vdash C(x, y, p)\ \mathsf{type} \qquad x : A \vdash d(x) : C(x, x, \mathrm{r}(x))}{x, y : A,\ p : \mathsf{Id}_A(x, y) \vdash \mathrm{J}_d(x, y, p) : C(x, y, p)}\ \mathsf{Id}\text{-ELIM}$$

$$\frac{x, y : A,\ p : \mathsf{Id}_A(x, y) \vdash C(x, y, p)\ \mathsf{type} \qquad x : A \vdash d(x) : C(x, x, \mathrm{r}(x))}{x : A \vdash \mathrm{J}_d(x, x, \mathrm{r}(x)) = d(x) : C(x, x, \mathrm{r}(x))}\ \mathsf{Id}\text{-COMP}$$

Dependent product types

$$\frac{A\ \mathsf{type} \qquad x : A \vdash B(x)\ \mathsf{type}}{\Pi x : A.\, B(x)\ \mathsf{type}}\ \Pi\text{-FORM} \qquad \frac{x : A \vdash f(x) : B(x)}{\lambda x.\, f(x) : \Pi x : A.\, B(x)}\ \Pi\text{-ABS}$$

$$\frac{m : \Pi x : A.\, B(x)}{y : A \vdash m \cdot y : B(y)}\ \Pi\text{-APP} \qquad \frac{x : A \vdash f(x) : B(x)}{y : A \vdash \big(\lambda x.\, f(x)\big) \cdot y = f(y) : B(y)}\ \Pi\text{-}\beta$$

Table 1. *Logical rules of intensional Martin-Löf type theory* (ML_I)

side of the turnstile: so, for example, we may specify a dependent type in context Γ as $x : \Gamma \vdash A(x)\ \mathsf{type}$. We will also make use of the *telescope* notion of de Bruijn (1991). Given a context Γ as before, this allows us to abbreviate the series of judgements

$$\begin{gathered} x : \Gamma \vdash B_1(x)\ \mathsf{type} \\ x : \Gamma,\ y_1 : B_1 \vdash B_2(x, y_1)\ \mathsf{type} \\ \vdots \\ x : \Gamma,\ y_1 : B_1,\ \ldots,\ y_{m-1} : B_{m-1} \vdash B_m(x, y_1, \ldots y_{m-1})\ \mathsf{type} \end{gathered}$$

as $x : \Gamma \vdash \Delta(x)\ \mathsf{ctxt}$, where $\Delta(x) :\equiv (y_1 : B_1(x),\ y_2 : B_2(x, y_1),\ \ldots)$. We say that Δ is a *context in context* Γ, or a *context dependent upon* Γ, and refer to contexts like Δ as

dependent contexts, and to those like Γ as *closed contexts*. Given a dependent context $x : \Gamma \vdash \Delta(x)\ \mathsf{ctxt}$, we may abbreviate the series of judgements

$$\begin{gathered} x : \Gamma \vdash f_1(x) : B_1(x) \\ \vdots \\ x : \Gamma \vdash f_m(x) : B_m(x, f_1(x), \ldots, f_{m-1}(x)) \end{gathered}$$

as $x : \Gamma \vdash f(x) : \Delta(x)$, and say that f is a *dependent element* of Δ. We can similarly assign a meaning to the judgements $x : \Gamma \vdash \Delta(x) = \Theta(x)\ \mathsf{ctxt}$ and $x : \Gamma \vdash f(x) = g(x) : \Delta(x)$ expressing the definitional equality of two dependent contexts and the definitional equality of two dependent elements of a dependent context, respectively.

2.2. *Extensional type theory*

We obtain extensional Martin-Löf type theory ML_E by augmenting the intensional theory with the two *equality reflection rules*

$$\frac{a, b : A \qquad \alpha : \mathsf{Id}(a, b)}{a = b : A} \qquad\qquad \frac{a, b : A \qquad \alpha : \mathsf{Id}(a, b)}{\alpha = \mathrm{r}(a) : \mathsf{Id}(a, b)}$$

together with the rule of *function extensionality*

$$\frac{m, n : \Pi(A, B) \qquad x : A \vdash m \cdot x = n \cdot x}{m = n : \Pi(A, B)}\ .$$

The addition of these three rules yields a type theory that is intuitively simpler, and more natural from the perspective of categorical models, but is proof-theoretically unpleasant: we lose the decidability of definitional equality and the decidability of type-checking. Note that if one develops Martin-Löf type theory in a framework admitting higher-order inference rules (such as the Logical Framework of Nordström *et al.* (1990)), then the above three rules are equipotent with the definitional η-rule.

2.3. *Two-dimensional type theory*

The type theory we investigate in this paper lies between the intensional theory of Section 2.1 and the extensional theory of Section 2.2. We denote it by ML_2, and call it *two-dimensional type theory*, because as we will see, it has a natural semantics in two-dimensional categories. It is obtained by augmenting intensional type theory with the rules of Tables 2 and 3. These provide restricted versions of the equality reflection rules (Table 2) and the function extensionality rules (Table 3). To motivate the rules in Table 2, we introduce the notion of a discrete type. We say that $\Gamma \vdash A\ \mathsf{type}$ is *discrete* if the judgements

$$\frac{\Gamma \vdash a, b : A \qquad \Gamma \vdash p : \mathsf{Id}(a, b)}{\Gamma \vdash a = b : A}\ \mathsf{Id}\text{-}\mathrm{REFL}_1\text{-}A$$

$$\frac{\Gamma \vdash a, b : A \qquad \Gamma \vdash p : \mathsf{Id}(a, b)}{\Gamma \vdash p = \mathrm{r}(a) : \mathsf{Id}(a, b)}\ \mathsf{Id}\text{-}\mathrm{REFL}_2\text{-}A$$

$$\frac{a, b : A \qquad p, q : \mathsf{Id}(a, b) \qquad \alpha : \mathsf{Id}(p, q)}{p = q : \mathsf{Id}(a, b)}\ \mathsf{Id}\text{-}\mathrm{DISC}_1$$

$$\frac{a, b : A \qquad p, q : \mathsf{Id}(a, b) \qquad \alpha : \mathsf{Id}(p, q)}{\alpha = \mathrm{r}(p) : \mathsf{Id}(p, q)}\ \mathsf{Id}\text{-}\mathrm{DISC}_2$$

Table 2. *Rules for discrete identity types*

$$\frac{m, n : \Pi(A, B) \qquad x : A \vdash p(x) : \mathsf{Id}(m \cdot x,\ n \cdot x)}{\mathsf{ext}(m, n, p) : \mathsf{Id}(m, n)}\ \Pi\text{-EXT}$$

$$\frac{m : \Pi(A, B)}{\mathsf{ext}(m, m, [x]\, \mathrm{r}(m \cdot x)) = \mathrm{r}(m) : \mathsf{Id}(m, m)}\ \Pi\text{-EXT-COMP}$$

$$\frac{m, n : \Pi(A, B) \qquad x : A \vdash p(x) : \mathsf{Id}(m \cdot x,\ n \cdot x)}{x : A \vdash \mathsf{ext}(m, n, p) * x = p(x) : \mathsf{Id}(m \cdot x,\ n \cdot x)}\ \Pi\text{-EXT-APP}$$

Table 3. *Rules for function extensionality*

are derivable. Thus the intensional theory says that no types need be discrete; the extensional theory says that all types are discrete; and the two-dimensional theory says that all identity types are discrete. Note that although two-dimensional type theory suffers from the same proof-theoretic deficiencies as the extensional theory, it does so in a less severe manner: indeed, only those types of ML_2 in whose construction the identity types have been used will have undecidable definitional equality. As we ascend to higher-dimensional variants of type theory, this undecidability will be pushed further and further up the hierarchy of iterated identity types, but it is only in the limit – which is intensional type theory – that we regain complete decidability.

The necessity for the rules in Table 3 will become clear when we reach Section 4.5. We require them in order to obtain a satisfactory notion of two-dimensional categorical model, in which dependent product formation is right adjoint to substitution (in a suitably weak 2-categorical sense). The first of the rules in Table 3 is a propositional version of the function extensionality principle of Section 2.2, whilst the second and third express coherence properties of the first. To understand the third rule, we must first explain the symbol $*$ appearing in it. It is a definable constant expressing the fact that two propositionally equal elements of a Π-type are pointwise propositionally equal. Explicitly, it satisfies the following introduction and computation rules

$$\frac{m, n : \Pi(A, B) \qquad p : \mathsf{Id}(m, n) \qquad a : A}{p * a : \mathsf{Id}(m \cdot a, n \cdot a)}\ *\text{-INTRO}$$

$$\frac{m : \Pi(A, B) \qquad a : A}{\mathrm{r}(m) * a = \mathrm{r}(m \cdot a) : \mathsf{Id}(m \cdot a, m \cdot a)}\ *\text{-COMP}$$

and we may define it by Id-elimination, taking $p * a := \mathrm{J}_{[x]\mathrm{r}(x \cdot a)}(m, n, p)$.

3. Categorical models for ML_2: structural aspects

The remainder of this paper will describe a notion of categorical semantics for ML_2. In this section and the next, we define a syntactic category and enumerate its structure, whilst in Section 5, we consider an arbitrary category endowed with this same structure, and derive from it a type theory incorporating the rules of ML_2. This yields a semantics that is both complete and sound. In this section, we define the basic syntactic category and look at the structure induced on it by the non-logical rules of ML_2; in the next section, we consider the logical rules. As mentioned in the Introduction, the syntactic category we define will in fact be a 2-*category* whose objects will be (vectors of) types, whose morphisms will be (vectors) of terms between those types and whose 2-cells will be (vectors of) identity proofs between these terms. The various forms of 2-cell composition will be obtained using the identity elimination rules; whilst the rules for discrete identity types given in Table 2 ensure that these compositions satisfy the 2-category axioms. For basic terminology and notation relating to 2-categories, see Kelly and Street (1974).

3.1. *One-dimensional semantics of type dependency*

We begin by recalling the construction of a one-dimensional categorical structure from the syntax of a dependent type theory. The presentation we will give follows Jacobs (1993) in its use of (full) *comprehension categories*. There are various other, essentially equivalent, presentations that we could have used: see, for example, Cartmell (1986), Dybjer (1996), Ehrhard (1988), Hyland and Pitts (1989) and Taylor (1999). We use comprehension categories because they afford a straightforward passage to a two-dimensional structure.

So we assume an arbitrary dependently typed calculus S admitting the same four basic judgement types and the same structural rules as the calculus ML_I. We define its *category of contexts* $\mathscr{C}_\mathrm{S}$ to have as objects, contexts $\Gamma, \Delta, \ldots$, in S, considered modulo α-conversion and definitional equality (so we identify Γ and Δ whenever $\vdash \Gamma = \Delta\ \mathsf{ctxt}$ is derivable), and as morphisms, $\Gamma \to \Delta$, judgements $x : \Gamma \vdash f(x) : \Delta$, considered modulo α-conversion and definitional equality (so we identity $f, g : \Gamma \to \Delta$ whenever $x : \Gamma \vdash f(x) = g(x)$ is derivable). To avoid further repetition, we introduce the convention that any further categorical structures we define should also be interpreted modulo α-equivalence and definitional equality. The identity map on Γ is given by $x : \Gamma \vdash x : \Gamma$; whilst composition is given by substitution of terms. Note that $\mathscr{C}_\mathrm{S}$ has a terminal object, which is given by the empty context ( ).

For each context Γ, we now define the category $\mathscr{T}_\mathrm{S}(\Gamma)$ of *types-in-context*-Γ, whose objects A are judgements $x : \Gamma \vdash A(x)\ \mathsf{type}$ and whose morphisms $A \to B$ are judgements $x : \Gamma,\ y : A(x) \vdash f(x, y) : B(x)$. Each morphism $f : \Gamma \to \Delta$ of $\mathscr{C}_\mathrm{S}$ induces a functor $\mathscr{T}_\mathrm{S}(f) : \mathscr{T}_\mathrm{S}(\Delta) \to \mathscr{T}_\mathrm{S}(\Gamma)$ that sends a type A in context Δ to the type f^*A in context Γ given by $x : \Gamma \vdash A(f(x))\ \mathsf{type}$. The assignation $f \mapsto \mathscr{T}_\mathrm{S}(f)$ is itself functorial in f, so we obtain an indexed category $\mathscr{T}_\mathrm{S}(-) : \mathscr{C}_\mathrm{S}^{\mathrm{op}} \to \mathbf{Cat}$, which, *via* the Grothendieck construction, we may equally well view as a split fibration $p : \mathscr{T}_\mathrm{S} \to \mathscr{C}_\mathrm{S}$. We refer to this as the *fibration of types over contexts*.

Explicitly, objects of $\mathscr{T}_{\mathrm{S}}$ are pairs (Γ, A) of a context and a type in that context, and morphisms $(\Gamma, A) \to (\Delta, B)$ are pairs (f, g) of a context morphism $f : \Gamma \to \Delta$ together with a judgement $x : \Gamma,\ y : A(x) \vdash g(x, y) : B(f(x))$. The chosen cartesian lifting of a morphism $f : \Gamma \to \Delta$ at an object (Δ, B) is given by $(f, \iota) : (\Gamma, f^*B) \to (\Delta, B)$, where ι denotes the judgement $x : \Gamma,\ y : B(fx) \vdash y : B(fx)$. Now, for each object (Γ, A) of $\mathscr{T}_{\mathrm{S}}$, we have the extended context $\big(x : \Gamma,\ y : A(x)\big)$, which we denote by $\Gamma.A$. We also have the judgement $x : \Gamma,\ y : A(x) \vdash x : \Gamma$, corresponding to a context morphism $\pi_A : \Gamma.A \to \Gamma$, which we call the *dependent projection* associated to A. In fact, the assignation $(\Gamma, A) \mapsto \pi_A$ provides the action on objects of a functor $E : \mathscr{T}_{\mathrm{S}} \to \mathscr{C}_{\mathrm{S}}^{\mathbf{2}}$ (where $\mathbf{2}$ denotes the arrow category $0 \to 1$), whose action on maps sends the morphism $(f, g) : (\Gamma, A) \to (\Delta, B)$ of $\mathscr{T}_{\mathrm{S}}$ to the morphism

$$\begin{array}{ccc} \Gamma.A & \xrightarrow{f.g} & \Delta.B \\ {\scriptstyle \pi_A}\downarrow & & \downarrow{\scriptstyle \pi_B} \\ \Gamma & \xrightarrow[f]{} & \Delta \end{array} \tag{2}$$

of $\mathscr{C}_{\mathrm{S}}^{\mathbf{2}}$, where $f.g$ denotes the judgement $x : \Gamma,\ y : A \vdash (f(x), g(x, y)) : \Delta.B$.

We can make two observations about this functor E. First, it is fully faithful, which says that every morphism $h : \Gamma.A \to \Delta.B$ fitting into a square like (2) is of the form $f.g$ for a unique $(f, g) : (\Gamma, A) \to (\Delta, B)$. Second, for a cartesian morphism $(f, \iota) : (\Gamma, f^*B) \to (\Delta, B)$, the corresponding square (2) is a pullback square. Indeed, given an arbitrary commutative square

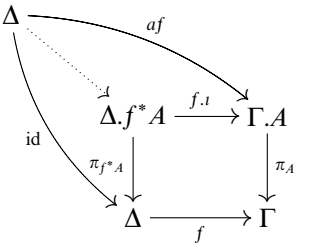
$$\begin{array}{ccc} \Lambda & \xrightarrow{k} & \Delta.B \\ {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle \pi_\Delta} \\ \Gamma & \xrightarrow[f]{} & \Delta \end{array}$$

commutativity forces k to be of the form $z : \Lambda \vdash (fhz, k'z) : \Delta.B$ for some judgement $z : \Lambda \vdash k'(z) : B(fhz)$, so the required factorisation $\Lambda \to \Gamma.f^*B$ is given by the judgement $z : \Lambda \vdash (hz, k'z) : \Gamma.f^*B$. We may abstract away from the above situation as follows. We define a *full split comprehension category* (*cf.* Jacobs (1993)) to be given by a category $\mathscr{C}$ with a specified terminal object, together with a split fibration $p : \mathscr{T} \to \mathscr{C}$ and a full and faithful functor $E : \mathscr{T} \to \mathscr{C}^{\mathbf{2}}$ rendering the triangle

$$\begin{array}{ccc} \mathscr{T} & \xrightarrow{E} & \mathscr{C}^{\mathbf{2}} \\ & {\scriptstyle p}\searrow \quad \swarrow{\scriptstyle \mathrm{cod}} & \\ & \mathscr{C} & \end{array}$$

commutative, and sending cartesian morphisms in $\mathscr{T}$ to pullback squares in $\mathscr{C}^{\mathbf{2}}$. The preceding discussion shows that we may associate to any suitable dependent type theory S a full split comprehension category $\mathfrak{C}(\mathrm{S})$, which we will refer to as the *classifying comprehension category* of S.

Notation 3.1.1. We will make use of the notation developed above in arbitrary comprehension categories $(p : \mathscr{T} \to \mathscr{C}, E : \mathscr{T} \to \mathscr{C}^{\mathbf{2}})$. Thus we write chosen cartesian liftings as $(f, \iota) : (\Gamma, f^*B) \to (\Delta, B)$, and write the image of $(\Gamma, A) \in \mathscr{T}$ under E as $\pi_A : \Gamma.A \to \Gamma$. We will find it convenient to develop a little more notation. Given $\Gamma \in \mathscr{C}$ and $A \in \mathscr{T}(\Gamma)$, we call a map $a : \Gamma \to \Gamma.A$ satisfying $\pi_A a = \mathrm{id}_\Gamma$ a *global section* of A, and denote it by $a \in_\Gamma A$. Given also a morphism $f : \Delta \to \Gamma$ of $\mathscr{C}$, we write $f^*a \in_\Delta f^*A$ for the section of π_{f^*A} induced by the universal property of pullback in the following diagram:



(3)

3.2. *A 2-category of types*

We will now extend the classifying comprehension category $\mathbb{C}(\mathrm{S})$ defined above to a classifying comprehension 2-category. We will not need the full strength of two-dimensional type theory, ML_2, for this. Rather, for the rest of this section we fix an arbitrary dependently typed theory S that admits the structural rules required in the previous subsection together with the identity type rules from Table 1 and the discrete identity rules of Table 2. Our first task will be to construct a 2-category of closed types in S. We will do this by enriching the category $\mathscr{T}_{\mathrm{S}}(\ )$ of closed types with 2-cells derived from the 2-category of *strict internal groupoids* in S. A strict internal groupoid in S is given by a closed type A_0, a family $A_1(x, y)$ of types over $x, y : A_0$ and operations of unit, composition and inverse:

$$\begin{aligned} x : A_0 &\vdash \mathrm{id}_x : A_1(x, x) \\ x, y, z : A_0,\ p : A_1(x, y),\ q : A_1(y, z) &\vdash q \circ p : A_1(x, z) \\ x, y : A_0,\ p : A_1(x, y) &\vdash p^{-1} : A_1(y, x) \end{aligned}$$

that obey the usual five groupoid axioms up to definitional equality. For instance, the left unit axiom requires that

$$x, y : A_0,\ p : A_1(x, y) \vdash \mathrm{id}_y \circ p = p : A_1(x, y)$$

should hold. We will generally write that (A_0, A_1) is an internal groupoid in S, leaving the remaining structure understood. Now, an *internal functor* $F : (A_0, A_1) \to (B_0, B_1)$ between internal groupoids is given by judgements

$$\begin{aligned} x : A_0 &\vdash F_0(x) : B_0 \\ x, y : A_0,\ p : A_1(x, y) &\vdash F_1(p) : B_1(F_0 x, F_0 y) \end{aligned}$$

subject to the usual two functoriality axioms (up to definitional equality again); whilst an *internal natural transformation* $\alpha : F \Rightarrow G$ is given by specifying a family of components $x : A_0 \vdash \alpha(x) : B_1(F_0 x, G_0 x)$ subject to the (definitional) naturality axiom.

Proposition 3.2.1. The strict groupoids, functors and natural transformations internal to S form a 2-category **Gpd**(S) that is *locally groupoidal*, in the sense that every 2-cell is invertible.

Proof. Recall that for any category $\mathscr{E}$, we can define a 2-category $\mathbf{Gpd}(\mathscr{E})$ of groupoids internal to that category[†]. In particular, we have the 2-category $\mathbf{Gpd}(\mathscr{C}_\mathrm{S})$ of groupoids internal to the category of contexts of S. Now, each strict internal groupoid $\mathscr{A}$ in S gives rise to such an internal groupoid $\mathscr{A}'$ in $\mathscr{C}_\mathrm{S}$ whose object of objects is the context $(x : A_0)$ and whose object of morphisms is the context $(x : A_0,\ y : A_0,\ p : A_1(x, y))$. We can check that internal functors $\mathscr{A} \to \mathscr{B}$ in S correspond bijectively with internal functors $\mathscr{A}' \to \mathscr{B}'$ in $\mathscr{C}_\mathrm{S}$, and that this correspondence extends to the natural transformations between them. Thus, we may take **Gpd**(S) to be the 2-category whose objects are strict internal groupoids in S, whose hom-categories are given by $\mathbf{Gpd}(\mathrm{S})(\mathscr{A}, \mathscr{B}) := \mathbf{Gpd}(\mathscr{C}_\mathrm{S})(\mathscr{A}', \mathscr{B}')$ and whose remaining structure is inherited from $\mathbf{Gpd}(\mathscr{C}_\mathrm{S})$. Note that every 2-cell of $\mathbf{Gpd}(\mathscr{C}_\mathrm{S})$ is invertible, so the same obtains for **Gpd**(S) □

Our method for obtaining the 2-category of closed types will be to construct a functor $\mathscr{T}_\mathrm{S}(\ ) \to \mathbf{Gpd}(\mathrm{S})$, and to lift the 2-cell structure of **Gpd**(S) along it.

Proposition 3.2.2. We may assign to each closed type A in S a strict internal groupoid (A, Id_A), and the assignation $A \mapsto (A, \mathsf{Id}_A)$ underlies a functor $\mathscr{T}_\mathrm{S}(\ ) \to \mathbf{Gpd}(\mathrm{S})$.

Proof. The proof of this result is essentially due to Hofmann and Streicher (1998). We repeat it because we will need the details. We first show that (A, Id_A) has the structure of a strict internal groupoid. For identities, we take $x : A \vdash \mathrm{id}_x := \mathrm{r}(x) : \mathsf{Id}(x, x)$. For composition, we require a judgement

$$x, y, z : A,\ p : \mathsf{Id}(x, y),\ q : \mathsf{Id}(y, z) \vdash q \circ p : \mathsf{Id}(x, z),$$

and by Id-elimination on p – in the extended form given in equation (1) – it suffices to define this when $y = z$ and $q = r(y)$, for which we take $r(y) \circ p := p$. Similarly, to give the judgement

$$x, y : A,\ p : \mathsf{Id}(x, y) \vdash p^{-1} : \mathsf{Id}(y, x)$$

providing inverses, it suffices to consider the case $x = y$ and $p = \mathrm{r}(x)$, for which we take $\mathrm{r}(x)^{-1} = \mathrm{r}(x)$. We must now check the five groupoid axioms. The first unitality axiom $\mathrm{id}_y \circ p = p$ follows from the Id-computation rule. For the other unitality axiom, it suffices, by discrete identity types, to show that

$$x, y : A,\ p : \mathsf{Id}(x, y) \vdash p \circ \mathrm{r}(x) \approx p : \mathsf{Id}(x, y)$$

[†] One commonly requires the category $\mathscr{E}$ to have pullbacks, but this is inessential.

holds, and by Id-elimination, it suffices to do this in the case $x = y$ and $p = \mathrm{r}(x)$, for which we have that $\mathrm{r}(x) \circ \mathrm{r}(x) = \mathrm{r}(x)$, as required. Likewise, for the associativity axiom, it suffices to show that

$$w, x, y, z : A,\ p : \mathsf{Id}(w, x),\ q : \mathsf{Id}(x, y),\ s : \mathsf{Id}(y, z) \vdash s \circ (q \circ p) \approx (s \circ q) \circ p : \mathsf{Id}(w, z),$$

and again by Id-elimination, it suffices to do this when $y = z$ and $s = \mathrm{r}(y)$, when we have that $\mathrm{r}(y) \circ (q \circ p) = q \circ p = (\mathrm{r}(y) \circ q) \circ p$, as required. Note that again we require the extended form of Id-elimination of equation (1), and in future we will use this without further comment. The invertibility axioms are similar. Suppose now that in addition to A we are given another type B together with a judgement $x : A \vdash f(x) : B$ between them. We will extend this to an internal functor $(f, f^\bullet) : (A, \mathsf{Id}_A) \to (B, \mathsf{Id}_B)$. We define the action on hom-types

$$x, y : A,\ p : \mathsf{Id}(x, y) \vdash f^\bullet(p) : \mathsf{Id}(fx, fy)$$

by Id-elimination on p, since when $x = y$ and $p = \mathrm{r}(x)$, we may take $f^\bullet(\mathrm{r}(x)) := \mathrm{r}(f(x))$. We must now check the functoriality axioms. The fact that $(f, f^\bullet)$ preserves identities follows from the Id-computation rule. To show that it preserves binary composition, it is enough, by discrete identity types, to show that

$$x, y, z : A,\ p : \mathsf{Id}(x, y),\ q : \mathsf{Id}(y, z) \vdash f^\bullet(q \circ p) \approx f^\bullet(q) \circ f^\bullet(p) : \mathsf{Id}(fx, fz)$$

holds. But this follows by Id-elimination on q, since when $y = z$ and $q = r(y)$, we have that $f^\bullet(\mathrm{r}(y) \circ p) = f^\bullet(p) = \mathrm{r}(f(y)) \circ f^\bullet(p) = f^\bullet(\mathrm{r}(y)) \circ f^\bullet(p)$, as required. We must now check that the assignation $f \mapsto (f, f^\bullet)$ is itself functorial. To show that it preserves identities, we must show that for any closed type A,

$$x, y : A,\ p : \mathsf{Id}(x, y) \vdash (\mathrm{id}_A)^\bullet(p) = p : \mathsf{Id}(x, y)$$

holds. By discrete identity types, it suffices to show this up to mere propositional equality, and by Id-elimination, we need only do so in the case when $x = y$ and $p = \mathrm{r}(x)$, when we have that $(\mathrm{id}_A)^\bullet(\mathrm{r}(x)) = \mathrm{r}(\mathrm{id}_A(x)) = \mathrm{r}(x)$, as required. To show that $f \mapsto (f, f^\bullet)$ respects composition, we must show that for maps of closed types $f : A \to B$ and $g : B \to C$, the judgement

$$x, y : A,\ p : \mathsf{Id}(x, y) \vdash (gf)^\bullet(p) = g^\bullet(f^\bullet(p)) : \mathsf{Id}(gfx, gfy)$$

holds. Again, it suffices to do this only up to propositional equality, and this only in the case where $x = y$ and $p = \mathrm{r}(x)$, so we have that $(gf)^\bullet(\mathrm{r}(x)) = \mathrm{r}(g(f(x))) = g^\bullet(\mathrm{r}(f(x)) = g^\bullet(f^\bullet(\mathrm{r}(x))$, as required. □

Corollary 3.2.3. The category $\mathscr{T}_\mathrm{S}(\ )$ of closed types in S may be extended to a locally groupoidal 2-category $\mathfrak{T}_\mathrm{S}(\ )$ whose 2-cells $\alpha : f \Rightarrow g : A \to B$ are given by judgements $x : A \vdash \alpha(x) : \mathsf{Id}_B(fx, gx)$.

Proof. If we view $\mathscr{T}_\mathrm{S}(\ )$ as a 2-category with only identity 2-cells, the functor of the previous proposition may be seen as a 2-functor $\mathscr{T}_\mathrm{S}(\ ) \to \mathbf{Gpd}(\mathrm{S})$. We can factorise this 2-functor as a composite

$$\mathscr{T}_\mathrm{S}(\ ) \to \mathfrak{T}_\mathrm{S}(\ ) \to \mathbf{Gpd}(\mathrm{S})$$

whose first part is bijective on objects and 1-cells and whose second part is fully faithful on 2-cells. We now define $\mathfrak{T}_{\mathrm{S}}(\ )$ to be the intermediate 2-category in this factorisation. We must check that this definition agrees with the description of $\mathfrak{T}_{\mathrm{S}}(\ )$ given above. Clearly, this is so for the objects and morphisms, whilst for the 2-cells, we must show that for any $f, g : A \to B$, each judgement $x : A \vdash \alpha(x) : \mathsf{Id}_B(fx, gx)$ satisfies the axiom for an internal natural transformation $\alpha : (f, f^\bullet) \Rightarrow (g, g^\bullet)$. By discrete identity types, this amounts to validating the judgement

$$x, y : A,\ p : \mathsf{Id}_A(x, y) \vdash g^\bullet(p) \circ \alpha(x) \approx \alpha(y) \circ f^\bullet(p) : \mathsf{Id}_B(fx, gy),$$

and by Id-elimination on p, it suffices to do this in the case where $x = y$ and $p = \mathrm{r}(x)$, for which we have that $g^\bullet(\mathrm{r}(x)) \circ \alpha(x) = \mathrm{r}(g(x)) \circ \alpha(x) = \alpha(x) = \alpha(x) \circ \mathrm{r}(f(x)) = \alpha(x) \circ f^\bullet(\mathrm{r}(x))$, as required. □

Corollary 3.2.4. For any context Γ in S, the category $\mathscr{T}_{\mathrm{S}}(\Gamma)$ of types-in-context-Γ may be extended to a locally groupoidal 2-category $\mathfrak{T}_{\mathrm{S}}(\Gamma)$ in which 2-cells $\alpha : f \Rightarrow g$ are judgements $x : \Gamma,\ y : A \vdash \alpha(x, y) : \mathsf{Id}_B(f(x, y),\ g(x, y))$.

Proof. We consider the *slice theory* S/Γ, whose closed types are the types of S in context Γ. It is easy to see that S/Γ admits the same inference rules as S – and, in particular, has discrete identity types – so that the result follows upon identifying $\mathfrak{T}_{\mathrm{S}}(\Gamma)$ with $\mathfrak{T}_{\mathrm{S}/\Gamma}(\ )$. □

3.3. *A 2-category of contexts*

In this section, we generalise the construction of the 2-category of closed types in order to construct a 2-category of contexts. The method will be a direct transcription of the one used in the previous section, but in order for it to make sense, we need to extend the identity type constructor to a 'meta-constructor', which operates on entire contexts rather than single types.

Proposition 3.3.1. The following inference rules are definable in S:

$$\frac{\Phi\ \mathsf{ctxt} \qquad a, b : \Phi}{\mathsf{Id}_\Phi(a, b)\ \mathsf{ctxt}}\ \mathsf{Id\text{-}FORM'} \qquad \frac{\Phi\ \mathsf{ctxt} \qquad a : \Phi}{\mathrm{r}(a) : \mathsf{Id}_\Phi(a, a)}\ \mathsf{Id\text{-}INTRO'}$$

$$\frac{\begin{array}{c} x, y : \Phi,\ p : \mathsf{Id}_\Phi(x, y),\ \Delta \vdash \Theta(x, y, p)\ \mathsf{ctxt} \\ x : \Phi,\ \Delta[x, x, \mathrm{r}(x)/x, y, p] \vdash d(x) : \Theta(x, x, \mathrm{r}(x)) \end{array}}{x, y : \Phi,\ p : \mathsf{Id}_\Phi(x, y),\ \Delta \vdash \mathrm{J}_d(x, y, p) : \Theta(x, y, p)}\ \mathsf{Id\text{-}ELIM'}$$

$$\frac{\begin{array}{c} x, y : \Phi,\ p : \mathsf{Id}_\Phi(x, y),\ \Delta \vdash \Theta(x, y, p)\ \mathsf{ctxt} \\ x : \Phi,\ \Delta[x, x, \mathrm{r}(x)/x, y, p] \vdash d(x) : \Theta(x, x, \mathrm{r}(x)) \end{array}}{x : \Phi,\ \Delta[x, x, \mathrm{r}(x)/x, y, p] \vdash \mathrm{J}_d(x, x, \mathrm{r}(x)) = d(x) : \Theta(x, x, \mathrm{r}(x))}\ \mathsf{Id\text{-}COMP'}$$

In order to prove this result, we will make use of the following well-known consequence of the identity type rules.

Proposition 3.3.2 (The Leibniz rule). Given A type and $x : A \vdash B(x)$ type in S, the following rules are derivable:

$$\frac{a_1, a_2 : A \qquad p : \mathsf{Id}(a_1, a_2) \qquad b_2 : B(a_2)}{p^*(b_2) : B(a_1)}\ \textsf{Id-SUBST}$$

$$\frac{a : A \qquad b : B(a)}{\mathrm{r}(a)^*(b) = b : B(a)}\ \textsf{Id-SUBST-COMP}$$

Proof. By Id-elimination on p, it suffices to derive the first rule in the case where $a_1 = a_2$ and $p = \mathrm{r}(a_1)$, in which case we can take $\mathrm{r}(a_1)^*(b) := b$. The second rule now follows from the Id-computation rule. □

The key idea behind the proof of Proposition 3.3.1 can be illustrated by considering a context $\Phi = (x : A,\ y : B(x))$ of length 2. The corresponding identity context Id_Φ will be given by

$$\mathsf{Id}_\Phi\big((x, y), (x', y')\big) := \big(p : \mathsf{Id}_A(x, x'),\ q : \mathsf{Id}_{B(x)}(y, p^* y')\big).$$

We use substitution along the first component p to make the second component q type-check. This can be seen as a type-theoretic analogue of the Grothendieck construction for fibrations. Indeed, it is possible to show that there is a propositional isomorphism between this identity context Id_Φ and the identity type $\mathsf{Id}_{\Sigma(A,B)}$. Thus, in principle, there is no need to introduce identity contexts. However, we prefer to do so in order to obtain a cleaner separation between the identity rules and the Σ-type rules.

Proof of Proposition 3.3.1 The proof has two stages: first, we define the generalised Id-inference rules in the special case where the context Φ has length 1; then we use these to define them in the general case. We will reduce syntactic clutter by proving our results only in the case where the postcontext Δ is empty: the reader may readily supply the annotations for the general case. For the first part of the proof, we suppose we are given a context $\Phi = (x : A)$ of length 1. The inference rules Id-FORM′ and Id-INTRO′ for Φ are just the usual Id-formation and Id-introduction rules for A. However, Id-ELIM′ corresponds to the following generalised elimination rule:

$$\frac{x, y : A,\ p : \mathsf{Id}(x, y) \vdash \Theta(x, y, p)\ \mathsf{ctxt} \qquad x : A \vdash d(x) : \Theta(x, x, \mathrm{r}(x))}{x, y : A,\ p : \mathsf{Id}(x, y) \vdash \mathrm{J}_d(x, y, p) : \Theta(x, y, p)} \tag{4}$$

with Id-COMP′ stating that $\mathrm{J}_d(a, a, \mathrm{r}(a)) = d(a)$. We will define the elimination rule by induction on the length n of the context Θ. When $n = 0$, this is trivial, and when $n = 1$, we use the usual identity elimination rule. So suppose now that we have defined the rule for all contexts Θ of length n, and consider a context $x, y : A,\ p : \mathsf{Id}(x, y) \vdash \Theta(x, y, p)\ \mathsf{ctxt}$ of length $n + 1$. Thus Θ is of the form

$$\Theta(x, y, p) = (u : \Lambda(x, y, p),\ v : D(x, y, p, u))$$

for some context Λ of length n and type D. It follows that to make a judgement $x : A \vdash d(x) : \Theta(x, x, \mathrm{r}(x))$ is equally well to make a pair of judgements

$$\begin{aligned} x : A &\vdash d_1(x) : \Lambda(x, x, \mathrm{r}(x)) \\ x : A &\vdash d_2(x) : D\big(x, x, \mathrm{r}(x), d_1(x)\big). \end{aligned} \tag{5}$$

By the induction hypothesis, we may apply the elimination rule (4) for the context Λ with eliminating family d_1 to deduce the existence of a term

$$x, y : A,\ p : \mathsf{Id}(x, y) \vdash \mathrm{J}_{d_1}(x, y, p) : \Lambda(x, y, p) \tag{6}$$

satisfying $\mathrm{J}_{d_1}(x, x, \mathrm{r}(x)) = d_1(x)$. Now we consider the dependent type

$$x, y : A,\ p : \mathsf{Id}(x, y) \vdash C(x, y, p) := D\big(x, y, p, \mathrm{J}_{d_1}(x, y, p)\big)\ \mathsf{type}. \tag{7}$$

We have that $C(x, x, \mathrm{r}(x)) = D\big(x, x, \mathrm{r}(x), \mathrm{J}_{d_1}(x, x, \mathrm{r}(x))\big) = D\big(x, x, \mathrm{r}(x), d_1(x)\big)$ and hence from (5) we can derive the judgement

$$x : A \vdash d_2(x) : C\big(x, x, \mathrm{r}(x)\big). \tag{8}$$

Now applying the standard Id-elimination rule to (7) and (8) yields a judgement

$$x, y : A,\ p : \mathsf{Id}(x, y) \vdash \mathrm{J}_{d_2}(x, y, p) : D\big(x, y, p, \mathrm{J}_{d_1}(x, y, p)\big) \tag{9}$$

satisfying $\mathrm{J}_{d_2}(x, x, \mathrm{r}(x)) = d_2(x)$. But to give (6) and (9) is equally well to give a dependent element $x, y : A,\ p : \mathsf{Id}(x, y) \vdash \mathrm{J}_d(x, y, p) : \Theta(x, y, p)$, and the respective computation rules for J_{d_1} and J_{d_2} now imply the computation rule for J_d. This completes the first part of the proof.

We now construct the generalised inference rules for an arbitrary context Φ. Once again the proof will be by induction, this time on the length of Φ. For the base case, the only context of length 0 is $(\ )$, the empty context. For this, we take the identity context $\mathsf{Id}_{(\ )}$ to be the empty context also. The introduction rule is vacuous, whilst the elimination rule requires us to provide, for each closed context Θ and global element $d : \Theta$, a global element $\mathrm{J}_d : \Theta$ satisfying the computation rule $\mathrm{J}_d = d : \Theta$. Thus we simply take $\mathrm{J}_d := d$ and are done. Suppose now that we have defined identity contexts for all contexts of length n, and consider a context $\Phi = \big(x_1 : \Lambda, x_2 : D(x_1)\big)$ of length $n + 1$. In order to define Id_Φ, we first apply the induction hypothesis to Λ in order to define its Leibniz rule. Thus, given $x : \Lambda \vdash \Upsilon(x)\ \mathsf{ctxt}$, we may define a judgement

$$x, y : \Lambda,\ p : \mathsf{Id}_\Lambda(x, y),\ z : \Upsilon(y) \vdash p^*(z) : \Upsilon(x)$$

satisfying $\mathrm{r}(x)^*(z) = z : \Upsilon(x)$. The proof is as in Proposition 3.3.2. Now, to give the formation rule for Id_Φ is equally well to give a judgement

$$x_1 : \Lambda,\ y_1 : D(x_1),\ x_2 : \Lambda,\ y_2 : D(x_2) \vdash \mathsf{Id}_\Phi(x_1, y_1, x_2, y_2)\ \mathsf{ctxt},$$

which we do by setting

$$\mathsf{Id}_\Phi(x_1, y_1, x_2, y_2) := \big(p : \mathsf{Id}_\Lambda(x_1, x_2),\ q : \mathsf{Id}_{D(x_1)}(y_1, p^* y_2)\big).$$

Next, in order to define the introduction rule for Id_Φ, is equally well to give judgements

$$x : \Lambda,\ y : D(x) \vdash r_1(x, y) : \mathsf{Id}_\Lambda(x, x)$$
$$x : \Lambda,\ y : D(x) \vdash r_2(x, y) : \mathsf{Id}_{D(x)}\big(y,\ r_1(x, y)^*(y)\big),$$

which we do by setting $r_1(x, y) := \mathrm{r}(x)$ and $r_2(x, y) := \mathrm{r}(y)$, where for the second of these we make use of the fact that $\mathsf{Id}_{D(x)}\big(y, \mathrm{r}(x)^*(y)\big) = \mathsf{Id}_{D(x)}(y, y)$. In order to define the elimination rule for Id_Φ, we first define a context dependent on $x_1, x_2 : \Lambda$ and $p : \mathsf{Id}_\Lambda(x_1, x_2)$ by

$$\Delta(x_1, x_2, p) := \big(y_1 : D(x_1),\ y_2 : D(x_2),\ q : \mathsf{Id}_{D(x_1)}(y_1, p^* y_2)\big).$$

We may then write the premisses of the elimination rule for Id_Φ as

$$x_1, x_2 : \Lambda,\ p : \mathsf{Id}_\Lambda(x_1, x_2),\ z : \Delta(x_1, x_2, p) \vdash \Theta(x_1, x_2, p, z)\ \mathsf{ctxt} \tag{10}$$

and

$$x : \Lambda,\ y : D(x) \vdash d(x, y) : \Theta(x, x, \mathrm{r}(x), y, y, \mathrm{r}(y)). \tag{11}$$

We would like to apply the elimination rule for Id_Λ (with postcontext Δ) to equation (10). In order to do this, we need to exhibit a generating family

$$x : \Lambda,\ z : \Delta(x, x, \mathrm{r}(x)) \vdash d'(x, z) : \Theta(x, x, \mathrm{r}(x), z), \tag{12}$$

which is equivalently a family

$$x : \Lambda,\ y_1, y_2 : D(x),\ q : \mathsf{Id}_{D(x)}(y_1, y_2) \vdash d'(x, y_1, y_2, q) : \Theta(x, x, \mathrm{r}(x), y_1, y_2, q)$$

since we have that $\mathrm{r}(x)^*(y_2) = y_2$. But we may obtain such a family by applying the generalised elimination rule (4) for $\mathsf{Id}_{D(x)}$ to the dependent context

$$x : \Lambda,\ y_1, y_2 : D(x),\ q : \mathsf{Id}_{D(x)}(y_1, y_2) \vdash \Theta(x, x, \mathrm{r}(x), y_1, y_2, q)\ \mathsf{ctxt}$$

with eliminating family (11). This yields a judgement (12) as required, whilst the computation rule says that $d'(x, y, y, \mathrm{r}(y)) = d(x, y)$. Now applying the elimination rule for Id_Λ to (10) and (12) yields a judgement

$$x_1, x_2 : \Lambda,\ p : \mathsf{Id}_\Lambda(x_1, x_2),\ z : \Delta(x_1, x_2, p) \vdash \mathrm{J}_{d'}(x_1, x_2, p, z) : \Theta(x_1, x_2, p, z)$$

of the correct form to provide the conclusion of the elimination rule for Id_Φ. From the computation rule for Id_Λ, this will satisfy $\mathrm{J}_{d'}(x, x, \mathrm{r}(x), z) = d'(x, z)$, and thus, in particular, we get

$$\mathrm{J}_{d'}(x, x, \mathrm{r}(x), y, y, \mathrm{r}(y)) = d'(x, y, y, \mathrm{r}(y)) = d(x, y),$$

which gives us the computation rule for Id_Φ. □

Using Proposition 3.3.1, we can now construct the 2-category of contexts in S by mimicking the developments of Section 3.2. We first define a *strict groupoid context* in S to be given by a context Γ_0 together with a dependent family $x, y : \Gamma_0 \vdash \Gamma_1(x, y)\ \mathsf{ctxt}$ of hom-contexts, and operations of unit, composition and inverse satisfying the groupoid axioms as before. It is still the case that any groupoid context induces an internal groupoid object in the category of contexts $\mathscr{C}_\mathrm{S}$, so, with the obvious definition of functor and natural transformation, we obtain a 2-category **GpdCtxt**(S) of groupoid contexts in S. Following Proposition 3.2.2, we now define a functor $\mathscr{C}_\mathrm{S} \to$ **GpdCtxt**(S) sending Γ to $(\Gamma, \mathsf{Id}_\Gamma)$. A small

subtlety we must check in order for this to go through is that S does not just have discrete identity types, but also discrete identity *contexts*. But this follows by a straightforward induction on the length of a context. Thereafter, the argument of Proposition 3.2.3 carries over to give the following corollary.

Corollary 3.3.3. The category $\mathscr{C}_S$ of contexts in S may be extended to a locally groupoidal 2-category $\mathfrak{C}_S$ whose 2-cells $\alpha : f \Rightarrow g : \Gamma \to \Delta$ are judgements $x : \Gamma \vdash \alpha(x) : \mathsf{Id}_\Delta(fx, gx)$.

We conclude this section with a simple observation.

Proposition 3.3.4. The 2-category $\mathfrak{C}_S$ has a 2-terminal object given by the empty context ( ).

Proof. It is clear that every context Γ admits a unique morphism $! : \Gamma \to ( )$, which makes ( ) a terminal object. For it to be 2-terminal, we must also show that for any 2-cell $\alpha : ! \Rightarrow ! : \Gamma \to ( )$ we have $\alpha = \mathrm{id}_!$. But this follows because we defined $\mathsf{Id}_{( )} := ( )$ in the proof of Proposition 3.3.1. □

3.4. *A 2-fibration of types over contexts*

The next stage in our development will be to extend the fibration of types over contexts to a 2-fibration of types over contexts. In Section 3.1, we built the one-dimensional fibration by first defining an indexed category of types over contexts, and then applying the Grothendieck construction. It turns out that in the two-dimensional case the indexed 2-category of types over contexts has a structure so elaborate (it is given by a trihomomorphism $\mathfrak{T}_S(-) : \mathfrak{C}_S^{\mathrm{coop}} \to \mathbf{Gray}$, where **Gray** is the tricategory of 2-categories, 2-functors, pseudo-natural transformations, and modifications – see Gordon *et al.* (1995)) that it is significantly less work to construct the associated 2-fibration directly. We begin by recalling the definition of a 2-fibration from Hermida (1999). Of the several equivalent formulations given there, the most convenient for our purposes is given by the following definition.

Definition 3.4.1 (*cf.* Hermida (1999, Theorem 2.8)). Let **E** and **B** be 2-categories. We say that a 2-functor $p : \mathbf{E} \to \mathbf{B}$ is a *cloven 2-fibration* if the following four conditions are satisfied:

(i) The underlying ordinary functor of p is a cloven fibration of categories.
(ii) Each cartesian 1-cell $f : y \to z$ of **E** has the following two-dimensional universal property: whenever we are given a 2-cell $\alpha : g \Rightarrow h : x \to z$ of **E** together with a factorisation

$$p(\alpha) = \; p(x) \underset{l}{\overset{k}{\Downarrow \gamma}} p(y) \xrightarrow{p(f)} p(z),$$

we may lift this to a unique factorisation

$$\alpha = \quad x \;\underset{l'}{\overset{k'}{\Downarrow \gamma'}}\; y \xrightarrow{\;f\;} z$$

satisfying $p(\gamma') = \gamma$.

(iii) For each $x, y \in \mathbf{E}$, the induced functor $p_{x,y} : \mathbf{E}(x,y) \to \mathbf{B}(px, py)$ is a cloven fibration of categories.

(iv) For each $x, y, z \in \mathbf{E}$ and $f : x \to y$, the functor $(-) \cdot f : \mathbf{E}(y,z) \to \mathbf{E}(x,z)$ preserves cartesian liftings of 2-cells.

We say further that a cloven 2-fibration is *globally split* if its underlying fibration of categories in (i) is a split fibration.

We will now show that the split fibration $p : \mathscr{T}_{\mathrm{S}} \to \mathscr{C}_{\mathrm{S}}$ of types over contexts extends to a globally split 2-fibration $p : \mathfrak{T}_{\mathrm{S}} \to \mathfrak{C}_{\mathrm{S}}$. The first step will be to construct the total 2-category $\mathfrak{T}_{\mathrm{S}}$. Before doing this we prove a useful lemma.

Lemma 3.4.2. For a dependent projection $\pi_A : \Gamma.A \to \Gamma$ of $\mathfrak{C}_{\mathrm{S}}$, its lifting to an internal functor $(\pi_A, \pi_A{}^{\bullet})$, as defined in Proposition 3.2.2, satisfies

$$(x,y),\ (x',y') : \Gamma.A,\ (p,q) : \mathsf{Id}_{\Gamma.A}\big((x,y),\ (x',y')\big) \ \vdash\ \pi_A{}^{\bullet}(p,q) = p : \mathsf{Id}_{\Gamma}(x,x').$$

Proof. By discrete identity types, it suffices to show that $\pi_A{}^{\bullet}(p,q) \approx p$, and by Id-elimination on $\Gamma.A$, we need only consider the case where $x = x'$, $y = y'$, $p = \mathrm{r}(x)$ and $q = r(y)$. But here, by definition of $\pi_A{}^{\bullet}$, we have $\pi_A{}^{\bullet}(\mathrm{r}(x), \mathrm{r}(y)) = \mathrm{r}(\pi_A(x,y)) = \mathrm{r}(x)$, as required. □

Proposition 3.4.3. The category $\mathscr{T}_{\mathrm{S}}$ defined in Section 3.1 extends to a locally groupoidal 2-category $\mathfrak{T}_{\mathrm{S}}$ whose 2-cells $(\alpha, \beta) : (f,g) \Rightarrow (f',g') : (\Gamma, A) \to (\Delta, B)$ are given by pairs of judgements

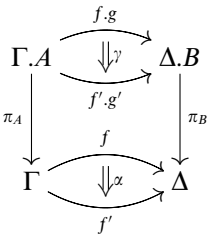
$$\begin{aligned} x : \Gamma \ &\vdash\ \alpha(x) : \mathsf{Id}_{\Delta}(fx, f'x) \\ x : \Gamma,\ y : A(x) \ &\vdash\ \beta(x,y) : \mathsf{Id}_{B(fx)}\big(g(x,y),\ \alpha(x)^{*}(g'(x,y))\big). \end{aligned} \tag{13}$$

Proof. If we view $\mathscr{T}_{\mathrm{S}}$ as a 2-category with only identity 2-cells, the functor $E : \mathscr{T}_{\mathrm{S}} \to \mathscr{C}_{\mathrm{S}}^{\mathbf{2}}$ defined in Section 3.1 may be viewed as a 2-functor $\mathscr{T}_{\mathrm{S}} \to \mathfrak{C}_{\mathrm{S}}^{\mathbf{2}}$. We can factorise this 2-functor as a composite

$$\mathscr{T}_{\mathrm{S}} \to \mathfrak{T}_{\mathrm{S}} \to \mathfrak{C}_{\mathrm{S}}^{\mathbf{2}} \tag{14}$$

whose first part is bijective on objects and 1-cells and whose second part is bijective on 2-cells. We claim that the intermediate 2-category is the $\mathfrak{T}_{\mathrm{S}}$ of the Proposition. Clearly, it has the correct objects and 1-cells, whilst for the 2-cells, we must show that given maps $(f,g),(f',g') : (\Lambda, A) \to (\Delta, B)$ of $\mathscr{T}_{\mathrm{S}}$, pairs of judgements as in (13) are in bijection with

diagrams



(15)

in $\mathfrak{C}_S$ satisfying $\pi_B\gamma = \alpha\pi_A$. For a diagram like (15), the 2-cell $\gamma : f.g \Rightarrow f'.g'$ corresponds, by the definition of $\mathsf{Id}_{\Delta.B}$ given in Proposition 3.3.1, to a pair of judgements

$$\begin{aligned} &x : \Gamma,\ y : A(x) \vdash \gamma_1(x,y) : \mathsf{Id}_\Delta(fx, f'x) \\ &x : \Gamma,\ y : A(x) \vdash \gamma_2(x,y) : \mathsf{Id}_{B(fx)}\big(g(x,y),\ \gamma_1(x,y)^*(g'(x,y))\big), \end{aligned} \tag{16}$$

whilst the equality $\pi_B\gamma = \alpha\pi_A$ corresponds to the validity of the judgement

$$x : \Gamma,\ y : A(x) \vdash \alpha(x) = \pi_B{}^\bullet(\gamma(x,y)) : \mathsf{Id}_\Delta(fx, f'x).$$

But by Lemma 3.4.2, we have $\pi_B{}^\bullet(\gamma(x,y)) = \gamma_1(x,y)$, so $\alpha(x) = \gamma_1(x,y)$, and we may identify (16) with (13) upon taking $\beta := \gamma_2$. □

Corollary 3.4.4. The fully faithful functor $E : \mathscr{T}_S \to \mathscr{C}_S^{\mathbf{2}}$ of Section 3.1 extends to a 2-fully faithful (that is, bijective on 1- and 2-cells) 2-functor $E : \mathfrak{T}_S \to \mathfrak{C}_S^{\mathbf{2}}$.

Proof. We take E to be the second half of the factorisation in (14). □

We now define $p : \mathfrak{T}_S \to \mathfrak{C}_S$ to be the composite of the 2-functor E of the previous Proposition with the codomain 2-functor $\mathfrak{C}_S^{\mathbf{2}} \to \mathfrak{C}_S$. Explicitly, p is the 2-functor sending (Γ, A) to Γ, (f, g) to f and (α, β) to α. We intend to show that p is a (globally split) 2-fibration, and will do so by making using of two further properties of the 2-functor $E : \mathfrak{T}_S \to \mathfrak{C}_S^{\mathbf{2}}$. The first of these generalises directly the one-dimensional situation described in Section 3.1. Its proof is less straightforward than one might think.

Proposition 3.4.5. For each $(\Delta, B) \in \mathfrak{T}_S$ and $f : \Gamma \to \Delta$ in $\mathfrak{C}_S$, the following pullback square in $\mathfrak{C}_S$ is also a 2-pullback:

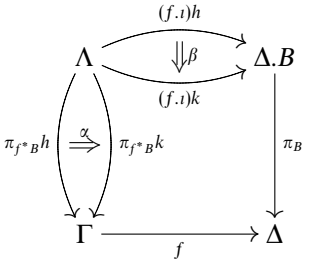
$$\begin{array}{ccc} \Gamma.f^*B & \xrightarrow{f.\iota} & \Delta.B \\ {\scriptstyle \pi_{f^*B}}\downarrow & & \downarrow{\scriptstyle \pi_B} \\ \Gamma & \xrightarrow[f]{} & \Delta. \end{array} \tag{17}$$

Proof. We begin by introducing a piece of local notation: for the duration of this proof, we will write applications of the Leibniz rule as

$$\frac{a_1, a_2 : A \qquad p : \mathsf{Id}(a_1, a_2) \qquad b_2 : B(a_2)}{\mathsf{subst}_B(p, b_2) : B(a_1)}\ \mathsf{Id\text{-}SUBST}$$

We do this in order to make explicit the family B in which substitution is occurring. Now, to say that (17) is not just a pullback but also a 2-pullback is to say that whenever we

are given maps $h, k : \Lambda \to \Gamma.f^*B$ and 2-cells



(18)

in $\mathfrak{C}_S$ satisfying $f\alpha = \pi_B\beta$, we can find a unique 2-cell $\gamma : h \Rightarrow k : \Lambda \to \Gamma.f^*B$ satisfying $\pi_{f^*B} \circ \gamma = \alpha$ and $f.\iota \circ \gamma = \beta$. In order to show this, we will first need to understand how $f.\iota$ lifts to an internal functor

$$(f.\iota,\ (f.\iota)^\bullet) : (\Gamma.f^*B,\ \mathsf{Id}_{\Gamma.f^*B}) \to (\Delta.B,\ \mathsf{Id}_{\Delta.B}).$$

So suppose we are given elements (x_1, y_1) and $(x_2, y_2) : \Gamma.f^*B$. Now, a typical element $(p, q) : \mathsf{Id}_{\Gamma.f^*B}\big((x_1, x_2),\ (y_1, y_2)\big)$ is given by a pair of judgements

$$p : \mathsf{Id}_\Gamma(x_1, y_1) \qquad \text{and} \qquad q : \mathsf{Id}_{B(fx_1)}\big(x_2,\ \mathsf{subst}_{f^*B}(p, y_2)\big). \tag{19}$$

This is sent by $(f.\iota)^\bullet$ to some element $(u, v) : \mathsf{Id}_{\Delta.B}\big((fx_1, y_1),\ (fy_1, y_2)\big)$, or, equally well, a pair of judgements

$$u : \mathsf{Id}_\Gamma(fx_1, fy_1) \qquad \text{and} \qquad v : \mathsf{Id}_{B(fx_1)}\big(x_2,\ \mathsf{subst}_B(u, y_2)\big). \tag{20}$$

Since we have $\pi_B \circ f.\iota = f \circ \pi_{f^*B}$, we have by Lemma 3.4.2 that

$$u = \pi_B{}^\bullet(u, v) = (\pi_{\mathscr{B}} \circ f.\iota)^\bullet(p, q) = (f \circ \pi_{f^*B})^\bullet(p, q) = f^\bullet(p).$$

So it only remains to describe v. We will do this by reduction to a special case. Suppose we have $x_2 = \mathsf{subst}_{f^*B}(p, y_2)$ and $q = \mathrm{r}(\mathsf{subst}_{f^*B}(p, y_2))$. We denote the corresponding v by

$$\theta(p, y_2) : \mathsf{Id}_{B(fx_1)}\big(\mathsf{subst}_{f^*B}(p, y_2),\ \mathsf{subst}_B(f^\bullet(p), y_2)\big). \tag{21}$$

Note that in the case where $x_1 = y_1$ and $p = \mathrm{r}(x_1)$, we have by Id-computation that $\theta(\mathrm{r}(x_1), y_2) = \mathrm{r}(y_2)$. We now use (21) to describe the general case. We claim that given p and q as in (19), the corresponding v as in (20) satisfies

$$v = \theta(p, y_2) \circ q : \mathsf{Id}_{B(fx_1)}\big(x_2,\ \mathsf{subst}_B(f^\bullet(p), y_2)\big).$$

Now, by discrete Id-types, it suffices to show this up to propositional equality, and by Id-elimination on $\Gamma.f^*B$, this only in the case where $x_1 = y_1$, $p = \mathrm{r}(x_1)$, $x_2 = y_2$ and $q = \mathrm{r}(x_2)$. Here, by definition of $(f.\iota)^\bullet$ and Id-computation, we have on the one hand that $v = \mathrm{r}(x_2)$, but on the other that $\theta(\mathrm{r}(x_1), x_2) \circ \mathrm{r}(x_2) = \mathrm{r}(x_2) \circ \mathrm{r}(x_2) = \mathrm{r}(x_2)$, as required. This completes the proof of the claim.

We are now ready to show that (17) is a 2-pullback. So suppose we are given maps $h, k : \Lambda \to \Gamma.f^*B$ and 2-cells α, β as in (18). To give h is to give judgements $x : \Lambda \vdash h_1(x) : \Gamma$ and $x : \Lambda \vdash h_2(x) : B(fh_1x)$, and correspondingly for k, whilst to give α and β as in (18)

satisfying $f\alpha = \pi_B\beta$ is to give judgements

$$x : \Lambda \vdash \alpha(x) : \mathsf{Id}_\Gamma(h_1x, k_1x)$$
$$x : \Lambda \vdash \beta_1(x) : \mathsf{Id}_\Delta(fh_1x, fk_1x)$$
$$x : \Lambda \vdash \beta_2(x) : \mathsf{Id}_{B(fh_1x)}(h_2x,\ \mathsf{subst}_B(\beta_1x,\ k_2x))$$

satisfying

$$x : \Lambda \vdash f^\bullet(\alpha x) = \pi_B{}^\bullet(\beta_1x, \beta_2x) : \mathsf{Id}_\Delta(fh_1x, fk_1x).$$

By Lemma 3.4.2, we have that $\pi_B{}^\bullet(\beta_1x, \beta_2x) = \beta_1(x)$, so to give (18) satisfying $f\alpha = \pi_B\beta$, is equally well to give a pair of judgements

$$x : \Lambda \vdash \alpha(x) : \mathsf{Id}_\Gamma(h_1x, k_1x)$$
$$x : \Lambda \vdash \beta_2(x) : \mathsf{Id}_{B(fh_1x)}(h_2x,\ \mathsf{subst}_B(f^\bullet\alpha x, k_2x)).$$

From this we are required to find a unique 2-cell $\gamma : h \Rightarrow k : \Lambda \to \Gamma.f^*B$ satisfying $\pi_{f^*B} \circ \gamma = \alpha$ and $(f.\imath) \circ \gamma = \beta$, or, equally well, a pair of judgements

$$x : \Lambda \vdash \gamma_1(x) : \mathsf{Id}_\Gamma(h_1x, k_1x)$$
$$x : \Lambda \vdash \gamma_2(x) : \mathsf{Id}_{B(fh_1x)}(h_2x,\ \mathsf{subst}_{f^*B}(\gamma_1x, k_2x))$$

satisfying $(\pi_{f^*B})^\bullet(\gamma_1x, \gamma_2x) = \alpha(x)$ and $(f.\imath)^\bullet(\gamma_1x, \gamma_2x) = (f^\bullet\alpha x, \beta_2x)$.

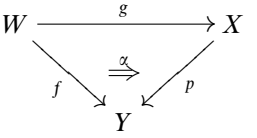
Now, by Lemma 3.4.2, we have $(\pi_{f^*B})^\bullet(\gamma_1x, \gamma_2x) = \gamma_1(x)$, so we must take $\gamma_1 := \alpha$. But from our investigations above, we have

$$(f.\imath)^\bullet(\gamma_1x, \gamma_2x) = (f.\imath)^\bullet(\alpha x, \gamma_2x) = \big(f^\bullet(\alpha x),\ \theta(\alpha x, k_2x) \circ \gamma_2(x)\big),$$

which tells us that we must have $\gamma_2(x) := \theta(\alpha x, k_2x)^{-1} \circ \beta_2(x)$. Uniqueness of γ then follows easily. □

The second property of E we will consider has no one-dimensional analogue as it involves the inherently 2-categorical notion of an *isofibration*.

Definition 3.4.6. Let **K** be a 2-category. A morphism $p : X \to Y$ in **K** is said to be a *cloven isofibration* if for every invertible 2-cell

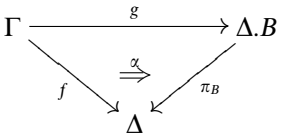


(22)

we are given a choice of 1-cell $s_\alpha : W \to X$ and 2-cell $\sigma_\alpha : s_\alpha \Rightarrow g$ satisfying $p \circ s_\alpha = f$ and $p \circ \sigma_\alpha = \alpha$; and these choices are natural in W, in the sense that if we are also given $k : W' \to W$, we have $s_{\alpha k} = s_\alpha \circ k$ and $\sigma_{\alpha k} = \sigma_\alpha \circ k$. A cloven isofibration is said to be *normal* if for any $g : W \to X$, we have $s_{\mathrm{id}_{pg}} = g$ and $\sigma_{\mathrm{id}_{pg}} = \mathrm{id}_g$.

Proposition 3.4.7. Every dependent projection $\pi_B : \Delta.B \to \Delta$ in $\mathfrak{C}_S$ may be equipped with the structure of a normal isofibration.

Proof. Suppose we are given an invertible 2-cell



(23)

of $\mathfrak{C}_S$. We must find a 1-cell $s_\alpha : \Gamma \to \Delta.B$ and 2-cell $\sigma_\alpha : s_\alpha \Rightarrow g$ satisfying $\pi_B \circ s_\alpha = f$ and $\pi_B \circ \sigma_\alpha = \alpha$. Now, to give a 2-cell as in (23) is equally well to give judgements

$$\begin{aligned} x : \Gamma \vdash f(x) : \Delta \qquad & x : \Gamma \vdash g_1(x) : \Delta \\ x : \Gamma \vdash g_2(x) : B(g_1 x) \qquad & x : \Gamma \vdash \alpha(x) : \mathsf{Id}(fx, g_1 x). \end{aligned}$$

So we may take $s_\alpha : \Gamma \to \Delta.B$ to be given by the pair of judgements

$$x : \Gamma \vdash f(x) : \Delta \quad \text{and} \quad x : \Gamma \vdash (\alpha x)^*(g_2 x) : B(fx), \tag{24}$$

and take $\sigma_\alpha : s_\alpha \Rightarrow g$ to be given by the pair of judgements

$$\begin{gathered} x : \Gamma \vdash \alpha(x) : \mathsf{Id}(fx, g_1 x) \\ x : \Gamma,\ y : A(x) \vdash \mathrm{r}\big((\alpha x)^*(g_2 x)\big) : \mathsf{Id}\big((\alpha x)^*(g_2 x),\ (\alpha x)^*(g_2 x)\big). \end{gathered} \tag{25}$$

Given further $k : \Lambda \to \Gamma$, the equalities $s_{\alpha k} = s_\alpha \circ k$ and $\sigma_{\alpha k} = \sigma_\alpha \circ k$ correspond precisely to the stability of (24) and (25) under substitution in x. Thus π_B is a cloven isofibration, and it just remains to check normality. But when α is an identity 2-cell, we have $f(x) = g_1(x)$ and $\alpha(x) = \mathrm{r}(g_1(x))$; so by the Leibniz computation rule, (24) reduces to g and (25) to id_g, as required. □

We will refer to the isofibration structure described in Proposition 3.4.7 as the *canonical isofibration structure* on a dependent projection.

Remark 3.4.8. Proposition 3.4.7 provides a link between the 2-categorical semantics of this paper and the homotopy-theoretic semantics espoused in Awodey and Warren (2009). The key idea of that paper is that a suitable environment for modelling intensional type theory should be a category equipped with a *weak factorisation system* $(\mathscr{L}, \mathscr{R})$ in the sense of Bousfield (1977), whose right-hand class of maps $\mathscr{R}$ is used to model dependent projections. Now, any finitely complete 2-category carries a weak factorisation system $(\mathscr{L}, \mathscr{R})$ where $\mathscr{R}$ is the class of normal isofibrations: it forms one half of what Gambino (2008, Section 4) calls the 'dual of the natural model structure on a 2-category'. Thus our two-dimensional semantics fits naturally into the framework outlined in Awodey and Warren (2009).

This result can also be seen as a special case of a result obtained in Gambino and Garner (2008). The main result of that paper is that the classifying category of any intensional type theory may be equipped with a weak factorisation system whose right class of maps is generated by the dependent projections, and it is shown (Lemma 13) that the maps in this right class are 'type-theoretic normal isofibrations'. Our Proposition 3.4.7 can be seen as a two-dimensional collapse of this result.

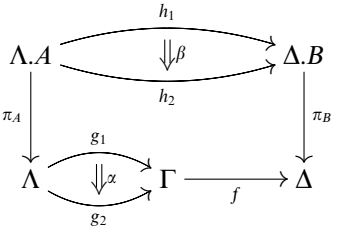
Using Propositions 3.4.5 and 3.4.7, we may now show the following proposition.

Proposition 3.4.9. The 2-functor $p : \mathfrak{T}_S \to \mathfrak{C}_S$ is a globally split 2-fibration.

Proof. We check the four clauses in Definition 3.4.1:

(i) This is immediate, since the underlying ordinary functor of $p : \mathfrak{T}_S \to \mathfrak{C}_S$ is the split fibration $p : \mathscr{T}_S \to \mathscr{C}_S$.

(ii) It suffices to consider a chosen cartesian lifting $(f, \iota) : (\Gamma, f^*B) \to (\Delta, B)$ of $\mathfrak{T}_S$. Taking advantage of the 2-fully faithfulness of $E : \mathfrak{T}_S \to \mathfrak{C}_S^{\mathbf{2}}$, we may express the property we are to verify as follows: for each diagram

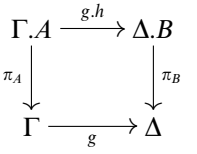


in $\mathfrak{C}_S$ with $\pi_B\beta = f\alpha\pi_A$, there is a unique factorisation

$$\beta = \Lambda.A \overset{h'_1}{\underset{h'_2}{\Downarrow \beta'}} \Gamma.f^*B \xrightarrow{f.\iota} \Delta.B$$

with $\pi_{f^*B}\beta' = \beta\pi_A$. But this follows without difficulty from the fact that diagram (17) is a 2-pullback.

(iii) Assuming we are given (Γ, A) and (Δ, B) in $\mathfrak{T}_S$, we are required to show that the functor $\mathfrak{T}_S\big((\Gamma, A), (\Delta, B)\big) \to \mathfrak{C}_S(\Gamma, \Delta)$ is a fibration. Using once more the 2-fully faithfulness of E, it suffices to show that for each commutative square



in $\mathfrak{C}_S$ and 2-cell $\alpha : f \Rightarrow g$, we can find a 1-cell $k : \Gamma.A \to \Delta.B$ and a 2-cell $\beta : k \Rightarrow g.h$ satisfying $\pi_B k = f\pi_A$ and $\pi_B\beta = \alpha\pi_A$. This follows using the canonical isofibration structure of π_B.

(iv) We must show that each $(-) \cdot f : \mathfrak{T}_S(y, z) \to \mathfrak{T}_S(x, z)$ preserves cartesian liftings of 2-cells. As every 2-cell of $\mathfrak{T}_S$ is invertible, and hence cartesian, this is automatic. □

We conclude this section by considering the pullback stability of the canonical isofibration structures of Proposition 3.4.7. To this end, consider a square like (17). Both vertical arrows π_B and π_{f^*B} have their canonical isofibration structures. But we also have a second isofibration structure on π_{f^*B}, which is obtained by pulling back the canonical structure of π_B along f. A careful examination of the proof of Proposition 3.4.7 reveals that *these two structures on π_{f^*B} need not coincide*. In other words, the canonical isofibration structures of Proposition 3.4.7 are not necessarily stable by pullbacks. At first glance, this may

appear surprising, since stability by pullbacks tends to be an automatic consequence of stability under substitution. However, a more careful analysis shows that in this case, stability under substitution corresponds to a more restricted form of pullback stability, which we now describe.

Suppose we are given $\Delta \in \mathfrak{C}_S$, $A \in \mathfrak{T}_S(\Delta)$ and $B \in \mathfrak{T}_S(\Delta.A)$. We can view the dependent projection $\pi_B : \Delta.A.B \to \Delta.A$ not only as a map of $\mathfrak{C}_S$, but also as a map

$$\begin{array}{ccc} \Delta.A.B & \xrightarrow{\pi_B} & \Delta.A \\ & {\scriptstyle \pi_A \pi_B} \searrow \quad \swarrow {\scriptstyle \pi_A} & \\ & \Delta & \end{array} \tag{26}$$

of $\mathfrak{C}_S/\Delta$. It is easy to see that the forgetful 2-functor $\mathfrak{C}_S/\Delta \to \mathfrak{C}_S$ creates normal isofibrations, so (26) is canonically a normal isofibration in $\mathfrak{C}_S/\Delta$. Suppose we are now given a morphism $f : \Gamma \to \Delta$ of $\mathfrak{C}_S$. By pulling back (26) along f, we obtain the map

$$\begin{array}{ccc} \Gamma.f^*A.f^*B & \xrightarrow{\pi_{f^*B}} & \Gamma.f^*A \\ & {\scriptstyle \pi_{f^*A}\pi_{f^*B}} \searrow \quad \swarrow {\scriptstyle \pi_{f^*A}} & \\ & \Gamma & \end{array} \tag{27}$$

of $\mathfrak{C}_S/\Gamma$ (note that we are abusing notation slightly: we should really write the left-hand vertex as $\Gamma.f^*A.(f.\imath)^*B$), and this now has two isofibration structures on it: the one induced by the canonical isofibration structure on π_{f^*B} and the one obtained by pulling back the isofibration structure of (26). The following Proposition now tells us that these two isofibration structures on (27) *do* coincide.

Proposition 3.4.10. Suppose we are given $\Delta \in \mathfrak{C}_S$, $A \in \mathfrak{T}_S(\Delta)$ and $B \in \mathfrak{T}_S(\Delta.A)$ and $f : \Gamma \to \Delta$ as above. With reference to the 2-pullback square

$$\begin{array}{ccc} \Gamma.f^*A.f^*B & \xrightarrow{f.\imath.\imath} & \Delta.A.B \\ {\scriptstyle \pi_{f^*B}} \downarrow & & \downarrow {\scriptstyle \pi_B} \\ \Gamma.f^*A & \xrightarrow[f.\imath]{} & \Delta.A \end{array} \tag{28}$$

the canonical isofibration structure on π_{f^*B} *qua* map of $\mathfrak{C}_S/\Gamma$ agrees with the pullback along f of the canonical isofibration structure on π_B *qua* map of $\mathfrak{C}_S/\Delta$.

Proof. As in the proof of Proposition 3.4.5, we will use subst notation in applications of the Leibniz rule in order to make clear the dependent family in which the substitution is taking place. Now, to prove the Proposition, it suffices to show the following. Suppose we are given an invertible 2-cell

$$\begin{array}{ccc} \Lambda & \xrightarrow{k} & \Gamma.f^*A.f^*B \\ & {\scriptstyle h} \searrow \quad \overset{\alpha}{\Rightarrow} \quad \swarrow {\scriptstyle \pi_{f^*B}} & \\ & \Gamma.f^*A & \end{array} \tag{29}$$

of $\mathfrak{C}_S/\Gamma$ (that is, one satisfying $\pi_{f^*A}\alpha = \mathrm{id}_{\pi_{f^*A}h}$). Writing $\alpha' := f.\imath.\imath \circ \alpha$ and $k' := f.\imath.\imath \circ k$, we must show that

$$s_{\alpha'} = f.\imath.\imath \circ s_\alpha : \Lambda \to \Delta.A.B \quad \text{and} \quad \sigma_{\alpha'} = f.\imath.\imath \circ \sigma_\alpha : s_{\alpha'} \Rightarrow k', \tag{30}$$

where we obtain $(s_\alpha, \sigma_\alpha)$ from the canonical isofibration structure on π_{f^*B} and $(s_{\alpha'}, \sigma_{\alpha'})$ from that on π_B. So suppose we are given a 2-cell as in (29), with h, k and α given as follows:

$$\begin{aligned}
&x : \Lambda \vdash h(x) := (h_1 x, h_2 x) : \Gamma.f^*A \\
&x : \Lambda \vdash k(x) := (h_1 x, k_2 x, k_3 x) : \Gamma.f^*A.f^*B \\
&x : \Lambda \vdash \alpha(x) := (\mathrm{r}h_1 x, \alpha_2 x) : \mathsf{Id}_{\Gamma.f^*A}\big((h_1 x, h_2 x), (h_1 x, k_2 x)\big).
\end{aligned}$$

We first compute the pair $(s_\alpha, \sigma_\alpha)$. The map $s_\alpha : \Lambda \to \Gamma.f^*A.f^*B$ is given by

$$x : \Lambda \vdash \big(h_1 x,\ h_2 x,\ \mathsf{subst}_{[u,v]B(fu,v)}((\mathrm{r}h_1 x, \alpha_2 x), k_3 x)\big) : \Gamma.f^*A.f^*B,$$

which, by unfolding the inductive description of the Id-elimination rule given in the proof of Proposition 3.3.1, is equal to

$$x : \Lambda \vdash \big(h_1 x,\ h_2 x,\ \mathsf{subst}_{[v]B(fh_1 x,v)}(\alpha_2 x, k_3 x)\big) : \Gamma.f^*A.f^*B. \tag{31}$$

The corresponding 2-cell $\sigma_\alpha : s_\alpha \Rightarrow k$ is now given by

$$x : \Lambda \vdash \big(\mathrm{r}h_1 x,\ \alpha_2 x,\ \mathrm{r}(\mathsf{subst}_{[v]B(fh_1 x,v)}(\alpha_2 x, k_3 x))\big) : \mathsf{Id}(s_\alpha x, kx). \tag{32}$$

Next we compute the pair $(s_{\alpha'}, \sigma_{\alpha'})$. By the proof of Proposition 3.4.5, we have

$$\begin{aligned}
\alpha'(x) &:= (f.\imath)^\bullet(\mathrm{r}h_1 x, \alpha_2 x) \\
&= (f^\bullet \mathrm{r}h_1 x, \theta(\mathrm{r}h_1 x, k_2 x) \circ \alpha_2 x) \\
&= (\mathrm{r}fh_1 x, \mathrm{r}(k_2 x) \circ \alpha_2 x) = (\mathrm{r}fh_1 x, \alpha_2 x).
\end{aligned}$$

Thus the morphism $s_{\alpha'} : \Gamma \to \Delta.A.B$ is given by

$$x : \Lambda \vdash \big(fh_1 x,\ h_2 x,\ \mathsf{subst}_{[u,v]B(u,v)}((\mathrm{r}fh_1 x, \alpha_2 x), k_3 x)\big) : \Delta.A.B,$$

which, by unfolding the description of Id-elimination, is definitionally equal to

$$x : \Lambda \vdash \big(fh_1 x,\ h_2 x,\ \mathsf{subst}_{[v]B(fh_1 x,v)}(\alpha_2 x, k_3 x)\big) : \Delta.A.B. \tag{33}$$

The corresponding 2-cell $\sigma_{\alpha'} : s_{\alpha'} \Rightarrow k'$ is now given by

$$x : \Lambda \vdash \big(\mathrm{r}fh_1 x,\ \alpha_2 x,\ \mathrm{r}(\mathsf{subst}_{[v]B(fh_1 x,v)}(\alpha_2 x, k_3 x))\big) : \mathsf{Id}(s_{\alpha'} x, k'x). \tag{34}$$

It remains to verify the equalities in (30). The first equality follows immediately from inspection of (31) and (33). For the second, we will need a calculation. Suppose we are given (x, y, s) and $(x, z, t) : \Lambda.f^*A.f^*B$ together with identity proofs $p : \mathsf{Id}(y, z)$ and $q : \mathsf{Id}(s, \mathsf{subst}_{[v]B(fx,v)}(p, t))$. We claim that

$$(f.\imath.\imath)^\bullet(\mathrm{r}(x), p, q) = (\mathrm{r}(fx), p, q) : \mathsf{Id}_{\Delta.A.B}\big((fx, y, s), (fx, z, t)\big). \tag{35}$$

By discrete identity types, it suffices to prove this up to propositional equality, and by applying Id-elimination twice, first on p and then on q, it suffices for this to show that, given $(x, y, s) : \Gamma.f^*A.f^*B$, we have

$$(f.\imath.\imath)^\bullet(\mathrm{r}x, \mathrm{r}y, \mathrm{r}k) \approx (\mathrm{r}fx, \mathrm{r}y, \mathrm{r}k) : \mathsf{Id}_{\Delta.A.B}\big((fx, y, s), (fx, y, s)\big).$$

But this follows by the Id-computation rule and the definition of $(f.\imath.\imath)^\bullet$. Thus we have (35) as claimed. We now use this to affirm the second equality in (30). Given $x : \Lambda$, we have

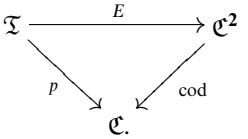
$$\begin{aligned}(f.\imath.\imath \circ \sigma_\alpha)(x) &= (f.\imath.\imath)^\bullet\big(\mathrm{r}h_1 x,\ \alpha_2 x,\ \mathrm{r}(\mathsf{subst}_{[v]B(fh_1x,v)}(\alpha_2 x, k_3 x))\big)\\ &= \big(\mathrm{r}fh_1 x,\ \alpha_2 x,\ \mathrm{r}(\mathsf{subst}_{[v]B(fh_1x,v)}(\alpha_2 x, k_3 x))\big)\\ &= \sigma_{f.\imath\circ\alpha}(x)\\ &= \sigma_{\alpha'}(x).\end{aligned}$$

□

Remark 3.4.11. Although it may seem somewhat technical, the result we have just proved is absolutely crucial for obtaining a sound notion of two-dimensional model. Without it, our models would not necessarily be sound for the rules expressing stability of the elimination rules under change of ambient context. It is not just the identity type rules that would be afflicted either: we will see in Sections 5.2–5.5 below that Proposition 3.4.10 is used in verifying the stability under substitution of *all* the type-theoretic elimination rules. One of the key issues in giving higher-dimensional and homotopy-theoretic semantics for intensional type theory will be finding an appropriate counterpart of this Proposition.

3.5. *Comprehension 2-categories*

We may abstract away from the syntactic investigations of the preceding sections as follows. We define a *full split comprehension 2-category* $\mathbb{C}$ to be given by the following data: a locally groupoidal 2-category $\mathfrak{C}$ with a specified 2-terminal object; a globally split 2-fibration $p : \mathfrak{T} \to \mathfrak{C}$, with $\mathfrak{T}$ also locally groupoidal; and a 2-fully faithful 2-functor $E : \mathfrak{T} \to \mathfrak{C}^{\mathbf{2}}$ rendering the triangle



commutative. Moreover, the 2-functor E should:

— send cartesian morphisms in $\mathfrak{T}$ to 2-pullback squares in $\mathfrak{C}$;
— send each object of $\mathfrak{T}$ to a normal isofibration in $\mathfrak{C}$; and
— satisfy the stability conditions of Proposition 3.4.10.

The preceding development shows that we may associate a full split comprehension 2-category to each dependent type theory S satisfying the rules for identity types in

Table 1 and the discreteness rules of Table 2. We denote this comprehension 2-category by $\mathbb{C}(\mathrm{S})$, and call it the *classifying comprehension 2-category* of S.

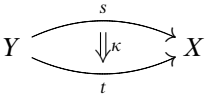
4. Categorical models for ML_2: logical aspects

4.1. *Identity types*

In this section, we will examine the structure induced on the syntactic comprehension 2-category of the previous section by the logical rules of two-dimensional type theory. Once again we consider a fixed dependently typed calculus S, which we now suppose to admit all of the rules in Tables 1, 2 and 3. We begin by investigating the identity types. Given how deeply intertwined these have been with the construction of the syntactic comprehension 2-category, it is perhaps unsurprising that their characterisation is rather intrinsic. It will be given in terms of the 2-categorical notion of arrow object. Given a 2-category $\mathbf{K}$, an *arrow object* for $X \in \mathbf{K}$ is given by an object $Y \in \mathbf{K}$ such that 1-cells into Y correspond naturally to 2-cells into X. That is, we have an isomorphism of categories

$$\mathbf{K}(A, Y) \cong \mathbf{K}(A, X)^{\mathbf{2}} \tag{36}$$

2-natural in A. In particular, under the bijection (36), the identity map $\mathrm{id}_Y : Y \to Y$ corresponds to a 2-cell

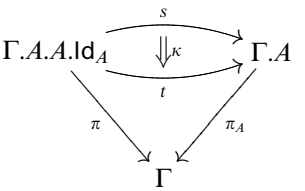


and 2-naturality of (36) says that any other such 2-cell into X factors uniquely through κ. In the language of enriched category theory (Kelly 1982), an arrow object is a certain kind of (weighted) limit, namely a *power* (or sometimes a *cotensor*) with the category $\mathbf{2}$. For a full treatment of 2-categorical limits, see Kelly (1989).

We now introduce a small abuse of notation. Given $\Gamma \in \mathfrak{C}_\mathrm{S}$ and $A \in \mathfrak{T}_\mathrm{S}(\Gamma)$, we write $\Gamma.A.A$ for the context $\big(x : \Gamma, y : A(x), z : A(x)\big)$, rather than the more correct $\Gamma.A.\pi_A^*A$, and write π_1 and π_2 for the context morphisms $\Gamma.A.A \to \Gamma.A$ projecting onto the first or second copy of A.

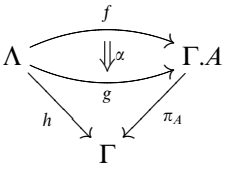
Proposition 4.1.1. For every context Γ and type $\Gamma \vdash A\ \mathsf{type}$ in S, the context $\Gamma.A.A.\mathsf{Id}_A$, together with the projections $\pi_1\pi_{\mathsf{Id}_A}, \pi_2\pi_{\mathsf{Id}_A} : \Gamma.A.A.\mathsf{Id}_A \to \Gamma.A$, can be made into an arrow object for $\Gamma.A$ in the slice 2-category $\mathfrak{C}_\mathrm{S}/\Gamma$.

Proof. We write $s := \pi_1\pi_{\mathsf{Id}_A}$ and $t := \pi_2\pi_{\mathsf{Id}_A}$. We are to find a 2-cell



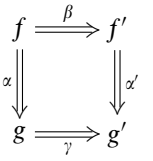
(37)

in $\mathfrak{C}_S$ that is over Γ in the sense that $\pi_A s = \pi_A t = \pi$ and $\pi_A \kappa = \mathrm{id}_\pi$, and such that any other 2-cell



(38)

over Γ factors through κ *via* a unique morphism $\bar{\alpha} : \Lambda \to \Gamma.A.A.\mathsf{Id}_A$. The universal property of κ also has a two-dimensional aspect. Suppose we are given a commutative diagram



(39)

of 1- and 2-cells $\Lambda \to \Gamma.A$ over Γ. Then we should be able to find a unique 2-cell $\delta : \bar{\alpha} \Rightarrow \bar{\alpha}' : \Lambda \to \Gamma.A.A.\mathsf{Id}_A$ with $\beta = s\delta$ and $\gamma = t\delta$. We begin by defining κ as in (37). For this we are required to give a judgement

$$x : \Gamma,\ y, z : A(x),\ p : \mathsf{Id}(y, z) \vdash \kappa(x, y, z, p) : \mathsf{Id}(y, z),$$

which we do by taking $\kappa(x, y, z, p) := p$. We now verify the universal property of κ. Suppose we are given an α as in (38). Then the commutativity conditions $\pi_A f = \pi_A g = h$ mean that f and g correspond to judgements

$$x : \Lambda \vdash f(x) : A(hx) \qquad \text{and} \qquad x : \Lambda \vdash g(x) : A(hx),$$

and thus, by Lemma 3.4.2, the condition $\pi_A \alpha = \mathrm{id}_h$ allows us to view α as a judgement $x : \Lambda \vdash \alpha(x) : \mathsf{Id}(fx, gx)$. We now define the morphism $\bar{\alpha} : \Lambda \to \Gamma.A.A.\mathsf{Id}_A$ to be given by the term $x : \Lambda \vdash (hx, fx, gx, \alpha x) : \Gamma.A.A.\mathsf{Id}_A$. It is immediate from the definition of κ that $\kappa\bar{\alpha} = \alpha$, and, moreover, that if $\kappa m = \alpha$ for some $m : \Lambda \to \Gamma.A.A.\mathsf{Id}_A$, we have $\bar{\alpha} = m$. We still need to verify the two-dimensional universal property of κ. So suppose we are given 1- and 2-cells as in (39). We are required to define a 2-cell $\delta : \bar{\alpha} \Rightarrow \bar{\alpha}' : \Lambda \to \Gamma.A.A.\mathsf{Id}_A$ satisfying $s\delta = \beta$ and $t\delta = \gamma$. In order to satisfy these last two requirements, δ, if it exists, must be given by a judgement

$$x : \Lambda \vdash (\mathrm{r}hx, \beta x, \gamma x, \delta_4 x) : \mathsf{Id}_{\Gamma.A.A.\mathsf{Id}_A}\big((hx, fx, gx, \alpha x), (hx, f'x, g'x, \alpha' x)\big)$$

for some $x : \Lambda \vdash \delta_4(x) : \mathsf{Id}_{\mathsf{Id}(fx,gx)}\big(\alpha x,\ (\beta x, \gamma x)^*(\alpha' x)\big)$. By discrete identity types, this is only possible if in fact $\alpha(x) = (\beta x, \gamma x)^*(\alpha' x)$, whereupon we can take $\delta_4(x) = \mathrm{r}(\alpha x)$. We claim that in fact $(\beta x, \gamma x)^*(\alpha' x) = (\gamma x)^{-1} \circ (\alpha' x \circ \beta x)$, so we will be done if we can show that $\alpha(x) = (\gamma x)^{-1} \circ (\alpha' x \circ \beta x)$, and this follows from the equation $\gamma\alpha = \alpha'\beta$ using the groupoid laws for Id_A. Thus it remains only to prove the claim, which will follow from the more general result that

$$x : \Gamma,\ y, z, y', z' : A(x),\ p : \mathsf{Id}(y, y'),\ q : \mathsf{Id}(z, z'),\ s : \mathsf{Id}(y', z') \vdash (p, q)^*(s) = q^{-1} \circ (s \circ p) : \mathsf{Id}(y, z).$$

By discrete identity types, it suffices to prove this up to propositional equality, and by Id-elimination on p and q, it suffices to consider the case where $y = y'$, $z = z'$, $p = \mathrm{r}(y)$ and $q = \mathrm{r}(z)$, where we have that $(\mathrm{r}(y), \mathrm{r}(z))^*(s) = s = \mathrm{r}(z)^{-1} \circ (s \circ \mathrm{r}(y))$, as required. □

Proposition 4.1.2 (Stability for identity types). Let Γ, Δ be contexts in S, let $f : \Gamma \to \Delta$ be a context morphism, and let $x : \Delta \vdash B(x)\ \mathsf{type}$. Then the comparison morphism

$$\Gamma.f^*B.f^*B.(f.\imath.\imath)^*(\mathsf{Id}_B) \to \Gamma.f^*B.f^*B.\mathsf{Id}_{f^*B}$$

induced by the universal property of Id_{f^*B} is an identity.

Proof. The claim follows immediately from the stability of identity types under substitution. □

4.2. *Digression on 2-categorical adjoints*

Our characterisation of the remaining type constructors of ML_2 will be given in terms of weak 2-categorical adjoints. We therefore break off at this point in order to give a brief summary of the 2-categorical notions necessary for this characterisation. Let $\mathbf{K}$ be a 2-category. By a *retract equivalence* in $\mathbf{K}$, we mean a pair of objects $x, y \in \mathbf{K}$, a pair of morphisms $i : x \to y$ and $p : y \to x$ satisfying $pi = \mathrm{id}_x$, and an invertible 2-cell $\theta : \mathrm{id}_y \Rightarrow ip$ satisfying $\theta i = \mathrm{id}_i$ and $p\theta = \mathrm{id}_p$. In these circumstances, we may call i an *injective equivalence* – with the understanding that the extra data (p, θ) is provided as part of this assertion – or call p a *surjective equivalence* (with the same understanding). Given now a 2-functor $U : \mathbf{K} \to \mathbf{L}$ and an object $x \in \mathbf{L}$, we define a *retract bireflection* of x along U to be an object $Fx \in \mathbf{K}$ and morphism $\eta_x : x \to UFx$ such that for each $y \in \mathbf{K}$, the functor

$$\mathbf{K}(Fx, y) \xrightarrow{U_{Fx,y}} \mathbf{L}(UFx, Uy) \xrightarrow{(-)\circ\eta_x} \mathbf{L}(x, Uy)$$

is a surjective equivalence of categories. By a *left retract biadjoint* F for U, we mean a choice for every $x \in \mathbf{L}$ of a retract bireflection Fx of x along U. Note that if F is a left retract biadjoint for U, then the assignation $x \mapsto Fx$ will not in general extend to a 2-functor $F : \mathbf{L} \to \mathbf{K}$; rather, it gives a *pseudo-functor*, which preserves identities and composition only up to invertible 2-cells. Similarly, the maps $\eta_x : x \to UFx$ do not provide components of a 2-natural transformation $\eta : \mathrm{id}_{\mathbf{L}} \Rightarrow UF$ but merely of a *pseudo-natural* transformation, whose naturality squares commute only up to invertible 2-cells. We could give a definition of left retract biadjoint in terms of a pseudo-functor $\mathbf{K} \to \mathbf{L}$ and unit and counit transformations η and ϵ satisfying weakened versions of the triangle laws (see Street (1974, Section 1) for the details), but the above description is both simpler and, as we will see, closer to the type theory. In fact, the above definitions admit a further simplification, using the observation that the surjective equivalences of categories are precisely those functors $F : \mathscr{C} \to \mathscr{D}$ that are fully faithful and whose object function $\mathrm{ob}\, F : \mathrm{ob}\, \mathscr{C} \to \mathrm{ob}\, \mathscr{D}$ is a split epimorphism.

Proposition 4.2.1. To give a retract bireflection of $x \in \mathbf{L}$ along $U : \mathbf{K} \to \mathbf{L}$ is to give an object $Fx \in \mathbf{K}$ and map $\eta_x : x \to UFx$, together with, for each $f : x \to Uy$ in $\mathbf{L}$, a choice of map $\bar{f} : Fx \to y$ in $\mathbf{K}$ satisfying $U\bar{f} \circ \eta_x = f$, all subject to the requirement that, for

every $h,k : Fx \to y$ in $\mathbf{K}$ and every $\alpha : Uh \circ \eta_x \Rightarrow Uk \circ \eta_x$ in $\mathbf{L}$, there is a unique $\bar{\alpha} : h \Rightarrow k$ with $U\bar{\alpha} \circ \eta_x = \alpha$.

Given a 2-functor $U : \mathbf{K} \to \mathbf{L}$ and $x \in \mathbf{L}$ as before, we have the dual notion of *retract bicoreflection* of x along U: this being given by an object $Gx \in \mathbf{K}$, together with a morphism $\epsilon_x : UGx \to x$ such that for each $y \in \mathbf{K}$, the functor

$$\mathbf{K}(y, Gx) \xrightarrow{U_{y,Gx}} \mathbf{L}(Uy, UGx) \xrightarrow{\epsilon_x \circ (-)} \mathbf{L}(Uy, x)$$

is a surjective (not injective!) equivalence of categories. Now a *right retract biadjoint* for U is given by a choice for every $x \in \mathbf{L}$ of a retract bicoreflection along U. As before, we have an elementary characterisation of retract bicoreflections.

Proposition 4.2.2. To give a retract bicoreflection of $x \in \mathbf{L}$ along $U : \mathbf{K} \to \mathbf{L}$ is to give an object $Gx \in \mathbf{K}$ and map $\epsilon_x : UGx \to x$, together with, for each $f : Uy \to x$ in $\mathbf{L}$, a choice of map $\bar{f} : y \to Gx$ in $\mathbf{K}$ satisfying $\epsilon_x \circ U\bar{f} = f$, all subject to the requirement that, for every $h,k : y \to Gx$ in $\mathbf{K}$ and every $\alpha : \epsilon_x \circ Uh \Rightarrow \epsilon_x \circ Uk$ in $\mathbf{L}$, there is a unique $\bar{\alpha} : h \Rightarrow k$ with $\epsilon_x \circ U\bar{\alpha} = \alpha$.

4.3. *Unit types*

Our first application of the 2-categorical adjoint notions developed above will be to the unit types of S – which we recall is an arbitrary dependent type theory admitting all the rules listed in Tables 1, 2 and 3. In the following result, we use $E(\Gamma) : \mathfrak{T}_S(\Gamma) \to \mathfrak{C}_S/\Gamma$ to denote the 2-functor obtained by restricting $E : \mathfrak{T}_S \to \mathfrak{C}_S$ to the fibre over $\Gamma \in \mathfrak{C}_S$.

Proposition 4.3.1. For each context Γ of S, the object $\mathbf{1}_\Gamma \in \mathfrak{T}_S(\Gamma)$ given by $\Gamma \vdash \mathbf{1}\ \mathsf{type}$ provides a retract bireflection of $\mathrm{id}_\Gamma : \Gamma \to \Gamma$ along the 2-functor $E(\Gamma) : \mathfrak{T}_S(\Gamma) \to \mathfrak{C}_S/\Gamma$.

Proof. The unit of the bireflection $\eta_\Gamma : \Gamma \to \Gamma.\mathbf{1}_\Gamma$ (over Γ) is given by the judgement $x : \Gamma \vdash \star : \mathbf{1}$. Now, given a morphism $f : \Gamma \to \Gamma.A$ over Γ, which is equally well a judgement $x : \Gamma \vdash f(x) : A(x)$, we obtain a factorisation $\bar{f} : \Gamma.\mathbf{1}_\Gamma \to \Gamma.A$ over Γ by $\mathbf{1}$-elimination, taking $\bar{f}$ to be the term $x : \Gamma,\ z : \mathbf{1} \vdash \mathrm{U}_{f(x)}(z) : A(x)$. That this satisies $\bar{f}\eta_\Gamma = f$ is now precisely the computation rule $x : \Gamma \vdash \mathrm{U}_{f(x)}(\star) = f(x)$. We still need to check that for maps $h,k : \Gamma.\mathbf{1} \to \Gamma.A$ over Γ, every 2-cell $\alpha : h\eta_\Gamma \Rightarrow k\eta_\Gamma$ over Γ is of the form $\bar{\alpha}\eta_\Gamma$ for a unique $\bar{\alpha} : h \Rightarrow k$. Now, to give h, k and α is to give judgements

$$\begin{gathered} x : \Gamma,\ z : \mathbf{1} \vdash h(x,z) : A(x) \\ x : \Gamma,\ z : \mathbf{1} \vdash k(x,z) : A(x) \\ x : \Gamma \vdash \alpha(x) : \mathsf{Id}\big(h(x,\star), k(x,\star)\big), \end{gathered}$$

from which we must determine $x : \Gamma,\ z : \mathbf{1} \vdash \bar{\alpha}(x,z) : \mathsf{Id}\big(h(x,z), k(x,z)\big)$. We do this by $\mathbf{1}$-elimination, taking $\bar{\alpha}(x,z) := \mathrm{U}_{\alpha(x)}(z)$. The equality $\bar{\alpha}\eta_\Gamma = \alpha$ now follows from the $\mathbf{1}$-computation rule. We still need to check uniqueness of $\bar{\alpha}$. So suppose we are given $x : \Gamma,\ z : \mathbf{1} \vdash \beta(x,z) : \mathsf{Id}\big(h(x,z), k(x,z)\big)$ satisfying $\beta(x,\star) = \alpha(x)$. We must show that $\beta(x,z) = \bar{\alpha}(x,z)$. By discrete identity types, it suffices to show this up to propositional equality, and by $\mathbf{1}$-elimination, this only in the case where $z = \star$, for which we have that $\beta(x,\star) = \alpha(x) = \bar{\alpha}(x,\star)$, as required. □

Proposition 4.3.2 (Stability for unit types). For each $k : \Gamma \to \Delta$ in $\mathfrak{C}$, we have $k^*(\mathbf{1}_\Delta) = \mathbf{1}_\Gamma$ and $\eta_\Gamma = k^*(\eta_\Delta) : \Gamma \to \Gamma.\mathbf{1}_\Gamma$, and for each $f : \Delta \to \Delta.B$ over Δ, we have $k^*(\bar{f}) = \overline{k^*(f)} : \Gamma.\mathbf{1}_\Gamma \to \Gamma.k^*B$.

Proof. The claim follows from the stability of unit types under substitution. □

Remark 4.3.3. Note carefully what the previous result does *not* say: it does not say that for a context morphism $k : \Gamma \to \Delta$, the comparison map $\mathbf{1}_\Gamma \to k^*\mathbf{1}_\Delta$ of $\mathfrak{T}_S(\Gamma)$ is an identity. Indeed, this map will in general only be *isomorphic* to the identity, since it corresponds to the judgement $x : \Gamma,\ z : \mathbf{1} \vdash \mathrm{U}_\star(z) : \mathbf{1}$.

4.4. *Dependent sum types*

We next consider the dependent sum types.

Proposition 4.4.1. For each context Γ and type $\Gamma \vdash A$ type of S, the 2-functor $\Delta_A := \mathfrak{T}_S(\pi_A) : \mathfrak{T}_S(\Gamma) \to \mathfrak{T}_S(\Gamma.A)$ has a left retract biadjoint Σ_A.

Proof. We must provide, for each $B \in \mathfrak{T}_S(\Gamma.A)$, a retract bireflection $\Sigma_A(B)$ of B along Δ_A. So we take $\Sigma_A(B) \in \mathfrak{T}_S(\Gamma)$ to be given by the judgement $\Gamma \vdash \Sigma(A,B)$ type (where for readability we suppress explicit mention of dependencies on the variables in Γ), and the unit map $\eta : B \to \Delta_A\Sigma_A(B)$ of $\mathfrak{T}_S(\Gamma.A)$ to be given by the judgement $\Gamma,\ y : A,\ z : B(y) \vdash \langle y, z\rangle : \Sigma(A,B)$. Now given a type $C \in \mathfrak{T}_S(\Gamma)$ and a map $f : B \to \Delta_A C$ of $\mathfrak{T}_S(\Gamma.A)$, we must provide a morphism $\bar{f} : \Sigma_A(B) \to C$ of $\mathfrak{T}_S(\Gamma)$ satisfying $\Delta_A(\bar{f}) \circ \eta = f$. But to give f is to give a judgement $\Gamma,\ y : A,\ z : B(y) \vdash f(y,z) : C$, whilst to give $\bar{f}$ is to give a judgement $\Gamma,\ s : \Sigma(A,B) \vdash \bar{f}(s) : C$. Thus, using Σ-elimination, we may define $\bar{f}(s) := \mathrm{E}_f(s)$. The equality $\Delta_A(\bar{f}) \circ \eta = f$ follows by the Σ-computation rule. We still need to show, given two morphisms $h, k : \Sigma_A(B) \to D$ in $\mathfrak{T}_S(\Gamma)$, that each 2-cell $\alpha : \Delta_A(h) \circ \eta \Rightarrow \Delta_A(k) \circ \eta$ is of the form $\Delta_A(\bar{\alpha}) \circ \eta$ for a unique $\bar{\alpha} : h \Rightarrow k$, but this follows by an argument analogous to that given in the proof of Proposition 4.3.1. □

Whilst Proposition 4.4.1 is very natural from a categorical perspective, it fails to capture the full strength of the elimination rule for dependent sums (even though it requires the full strength of that elimination rule in its proof). In order to do this, we need the following result.

Proposition 4.4.2. Suppose we are given a context Γ in S and types $\Gamma \vdash A$ type and $\Gamma,\ x : A \vdash B(x)$ type, and consider the morphism

$$\begin{array}{ccc} \Gamma.A.B & \xrightarrow{\ i\ } & \Gamma.\Sigma_A(B) \\ {\scriptstyle \pi_B}\downarrow & & \downarrow{\scriptstyle \pi_{\Sigma_A(B)}} \\ \Gamma.A & \xrightarrow[\ \pi_A\ ]{} & \Gamma \end{array} \tag{40}$$

in $\mathfrak{C}_S^2$ corresponding to the unit morphism $\eta : B \to \Delta_A\Sigma_A(B)$ in $\mathfrak{T}_S(\Gamma.A)$. The map i appearing in this diagram is an injective equivalence in $\mathfrak{C}_S/\Gamma$.

Proof. We construct a pseudoinverse retraction for i over Γ as follows. The map $p : \Gamma.\Sigma_A(B) \to \Gamma.A.B$ over Γ is given by the projections out of the sum

$$\Gamma,\ s : \Sigma(A,B) \vdash s.1 : A$$
$$\Gamma,\ s : \Sigma(A,B) \vdash s.2 : B(s.1)$$

(where again, we suppress explicit mention of the dependency on Γ). We define these by Σ-elimination on s, the first being given by $s.1 := \mathrm{E}_{[y,z]y}(s)$ and the second by $s.2 := \mathrm{E}_{[y,z]z}(s)$. The equality $pi = \mathrm{id}_{\Gamma.A.B}$ follows from the Σ-computation rule. We must now give a 2-cell $\theta : \mathrm{id}_{\Gamma.\Sigma_A(B)} \Rightarrow ip$, which is equally well a judgement

$$\Gamma,\ s : \Sigma(A,B) \vdash \theta(s) : \mathsf{Id}(s, \langle s.1, s.2\rangle).$$

By Σ-elimination on s, it suffices to define θ when $s = \langle y, z\rangle$, so we have $\langle s.1, s.2\rangle = \langle\langle y,z\rangle.1, \langle y,z\rangle.2\rangle = \langle y,z\rangle$, and we can take $\theta(\langle y,z\rangle) = \mathrm{r}(\langle y,z\rangle)$. The equality $\theta i = \mathrm{id}_i$ now follows by the Σ-computation rule, and it only remains to verify that $p\theta = \mathrm{id}_p$. Now, $p\theta$ corresponds to the judgement

$$\Gamma,\ s : \Sigma(A,B) \vdash p^\bullet(\theta(s)) : \mathsf{Id}_{\Gamma.A.B}\big((s.1, s.2), (s.1, s.2)\big),$$

and we must show that in fact $p^\bullet(\theta(s)) = \mathrm{r}(p(s))$. By discrete identity types, it suffices to show this up to propositional equality, and, by Σ-elimination, this only when $s = \langle y,z\rangle$. But we calculate that $p^\bullet(\theta(\langle y,z\rangle)) = p^\bullet(\mathrm{r}(\langle y,z\rangle)) = \mathrm{r}(p(\langle y,z\rangle))$, as required. □

Proposition 4.4.3 (Stability for dependent sums). Given $k : \Gamma \to \Lambda$ in $\mathfrak{C}_\mathrm{S}$, $A \in \mathfrak{T}(\Lambda)$ and $B \in \mathfrak{T}(\Lambda.A)$, we have that $k^*(\Sigma_A(B)) = \Sigma_{k^*A}(k^*B)$ and $k^*(\eta_{A,B}) = \eta_{k^*A,k^*B}$, and for each $f : B \to \Delta_A C$ in $\mathfrak{T}_\mathrm{S}(\Lambda.A)$, that $k^*\bar{f} = \overline{k^*f} : \Sigma_{k^*A}(k^*B) \to k^*C$. Moreover, reindexing along k sends the injective equivalence structure on $i_{A,B}$ to the injective equivalence structure on i_{k^*A,k^*B}.

Proof. The claim follows from the stability of dependent sum types under substitution. □

4.5. *Dependent product types*

Finally, we turn to the categorical characterisation of dependent product types in S.

Proposition 4.5.1. For each context Γ and type $\Gamma \vdash A\ \mathsf{type}$ of S, the weakening 2-functor $\Delta_A : \mathfrak{T}_\mathrm{S}(\Gamma) \to \mathfrak{T}_\mathrm{S}(\Gamma.A)$ has a right retract biadjoint Π_A.

Proof. Once again, we suppress any explicit mention of dependencies on the variables in Γ. We must provide, for each $B \in \mathfrak{T}_\mathrm{S}(\Gamma.A)$, a retract bicoreflection $\Pi_A(B)$ of B along Δ_A. For this we take $\Pi_A(B) \in \mathfrak{T}_\mathrm{S}(\Gamma)$ to be given by the judgement $\Gamma \vdash \Pi(A,B)\ \mathsf{type}$, and the counit map $\epsilon : \Delta_A\Pi_A(B) \to B$ of $\mathfrak{T}_\mathrm{S}(\Gamma.A)$ to be given by $\Gamma,\ m : \Pi(A,B),\ y : A \vdash m \cdot y : B(y)$. Now, given a type $C \in \mathfrak{T}_\mathrm{S}(\Gamma)$ and a map $f : \Delta_A C \to B$ of $\mathfrak{T}_\mathrm{S}(\Gamma.A)$, we are required to provide a morphism $\bar{f} : C \to \Pi_A(B)$ of $\mathfrak{T}_\mathrm{S}(\Gamma)$ satisfying $\epsilon \circ \Delta_A(\bar{f}) = f$. So if f is the judgement $\Gamma,\ y : C,\ z : A \vdash f(y,z) : B(y)$, we take $\bar{f}$ to be the judgement $\Gamma,\ y : C \vdash \lambda z.\, f(y,z) : \Pi(A,B)$. The equality $\epsilon \circ \Delta_A(\bar{f}) = f$ follows by the β-rule.

It remains to show, given two morphisms $h, k : D \to \Pi_A(B)$ in $\mathfrak{T}_S(\Gamma)$, that each 2-cell $\alpha : \epsilon \circ \Delta_A(h) \Rightarrow \epsilon \circ \Delta_A(k)$ can be written in the form $\epsilon \circ \Delta_A(\bar{\alpha})$ for a unique $\bar{\alpha} : h \Rightarrow k$. It is here that we will make crucial use of function extensionality. So, to give h, k and α is to give judgements $\Gamma, C \vdash h : \Pi(A, B)$; $\Gamma, C \vdash k : \Pi(A, B)$, and $\Gamma, C,\ z : A \vdash \alpha(z) : \mathsf{Id}(h \cdot z, k \cdot z)$ (where we now suppress explicit mention of the dependency on C), so we may define the 2-cell $\bar{\alpha} : h \Rightarrow k$ by applying the rule Π-EXT of Table 3 to obtain the judgement $\Gamma, C \vdash \mathsf{ext}(h, k, \alpha) : \mathsf{Id}(h, k)$. We must now check that $\epsilon \circ \Delta_A(\bar{\alpha}) = \alpha$. Recall from Section 2.3 the operation

$$\frac{m, n : \Pi(A, B) \qquad p : \mathsf{Id}(m, n) \qquad a : A}{p * a : \mathsf{Id}(m \cdot a, n \cdot a)}$$

given by $p * a := \mathrm{J}_{[x]\mathrm{r}(x \cdot a)}(m, n, p)$. It is easy to see that $*$ is just the lifting of ϵ to identity types, so $\epsilon \circ \Delta_A(\bar{\alpha})$ corresponds to the judgement

$$\Gamma,\ C,\ z : A \vdash \mathsf{ext}(h, k, \alpha) * z : \mathsf{Id}(h \cdot z,\ k \cdot z).$$

But by the rule Π-EXT-APP of Table 3, we have $\mathsf{ext}(h, k, \alpha) * z = \alpha(z)$, as required. We still need to check uniqueness of $\bar{\alpha}$. So, supposing given $\Gamma,\ C \vdash \beta : \mathsf{Id}(h, k)$ satisfying $\beta * z = \alpha(z)$, we must show that $\beta = \bar{\alpha}$. Now, because $\beta * z = \alpha(z) = \bar{\alpha} * z$, we have

$$\Gamma,\ C,\ z : A \vdash \mathsf{ext}\big(h, k, [z]\, \beta * z\big) = \mathsf{ext}\big(h, k, [z]\, \bar{\alpha} * z\big) : \mathsf{Id}(h, k).$$

Thus we will be done if we can show that

$$\Gamma,\ C,\ m, n : \Pi(A, B),\ k : \mathsf{Id}(m, n) \vdash \mathsf{ext}(m, n, [z]\, k * z) = k : \mathsf{Id}(m, n)$$

holds. By discrete identity types, it suffices to do this up to propositional equality, and by Id-elimination, this only in the case where $m = n$ and $k = \mathrm{r}(m)$. So we will be done if we can show that

$$\Gamma,\ C,\ m : \Pi(A, B) \vdash \mathsf{ext}(m, m, [z]\, \mathrm{r}(m \cdot z)) \approx \mathrm{r}(m) : \mathsf{Id}(m, m)$$

holds. But this follows immediately from the rule Π-EXT-COMP. □

Proposition 4.5.2 (Stability for dependent products). Given $k : \Gamma \to \Lambda$ in $\mathfrak{C}_S$, $A \in \mathfrak{T}(\Lambda)$ and $B \in \mathfrak{T}(\Lambda.A)$, we have $k^*(\Pi_A(B)) = \Pi_{k^*A}(k^*B)$ and $k^*(\epsilon_{A,B}) = \epsilon_{k^*A, k^*B}$; and for each $f : \Delta_A C \to B$ in $\mathfrak{T}_S(\Lambda.A)$, we have $k^*\bar{f} = \overline{k^*f} : k^*C \to \Pi_{k^*A}(k^*B)$.

Proof. The claim follows from the stability of dependent product types under substitution. □

4.6. *Models of two-dimensional type theory*

We abstract away from the preceding results as follows.

Definition 4.6.1. Let $\mathbb{C} = (p : \mathfrak{T} \to \mathfrak{C},\ E : \mathfrak{C} \to \mathfrak{T}^{\mathbf{2}})$ be a full split comprehension 2-category in the sense of Section 3.5. Then:

— We say that $\mathbb{C}$ has *equality* if, for every $\Gamma \in \mathfrak{C}$ and $A \in \mathfrak{T}(\Gamma)$, there is an object $\mathsf{Id}_A \in \mathfrak{T}(\Gamma.A.A)$ such that $\Gamma.A.A.\mathsf{Id}_A$, together with its two projections onto $\Gamma.A$,

underlies an arrow object for $\Gamma.A$ in $\mathfrak{C}/\Gamma$, and these arrow objects satisfy the stability properties of Proposition 4.1.2.

— We say that $\mathbb{C}$ has *units* if, for every $\Gamma \in \mathfrak{C}$, the map $\mathrm{id}_\Gamma : \Gamma \to \Gamma$ admits a retract bireflection $\mathbf{1}_\Gamma$ along $E(\Gamma) : \mathfrak{T}(\Gamma) \to \mathfrak{C}_S/\Gamma$, and these bireflections satisfy the stability properties of Proposition 4.3.2.

— We say that $\mathbb{C}$ has *sums* if, for every $\Gamma \in \mathfrak{C}$ and $A \in \mathfrak{T}(\Gamma)$, the 2-functor $\Delta_A := \mathfrak{T}(\pi_A) : \mathfrak{T}(\Gamma) \to \mathfrak{T}(\Gamma.A)$ admits a retract left biadjoint Σ_A, and these biadjoints satisfy the conditions of Proposition 4.4.2 and the stability properties of Proposition 4.4.3.

— We say that $\mathbb{C}$ has *products* if, for every $\Gamma \in \mathfrak{C}$ and $A \in \mathfrak{T}(\Gamma)$, the 2-functor $\Delta_A : \mathfrak{T}(\Gamma) \to \mathfrak{T}(\Gamma.A)$ admits a retract right biadjoint Π_A, and these biadjoints satisfy the stability properties of Proposition 4.5.2.

— We say that $\mathbb{C}$ is a *model of two-dimensional type theory* if it has equality, units, sums and products.

Thus, the results of this section can be summarised by saying that, for any dependent type theory S satisfying the rules of Tables 1, 2 and 3, the classifying comprehension 2-category $\mathbb{C}(S)$ is a model of two-dimensional type theory. With an eye to future applications, we conclude this section by gathering together in one place a list of the structures required for a two-dimensional model of type theory.

Definition 4.6.2. A two-dimensional model of type theory $\mathbb{C}$ is given by:

— A locally groupoidal 2-category $\mathfrak{C}$ of *contexts*, with a specified 2-terminal object.

— A locally groupoidal 2-category $\mathfrak{T}$ of *types-in-context*.

— A globally split 2-fibration $p : \mathfrak{T} \to \mathfrak{C}$ in the sense of Definition 3.4.1. Spelling this out, this means that p is a 2-functor such that:

(i) The underlying ordinary functor of p is a split fibration of categories.

(ii) For every cartesian 1-cell $f : y \to z$ of $\mathfrak{C}$ and every 2-cell $\alpha : g \Rightarrow h : x \to z$ of $\mathfrak{T}$, any factorisation of $p(\alpha)$ through $p(f)$ may be lifted uniquely to a factorisation of α through f.

(iii) For each $x, y \in \mathfrak{T}$, the induced functor $p_{x,y} : \mathfrak{T}(x, y) \to \mathfrak{C}(px, py)$ is a fibration of groupoids.

(Note that condition (iv) of Definition 3.4.1 is automatically satisfied since every fibre category is a groupoid).

— A *comprehension* 2-functor $E : \mathfrak{T} \to \mathfrak{C}^{\mathbf{2}}$, equipped with

– For each object $A \in \mathfrak{T}$, a normal isofibration structure on $E(A)$ in the sense of Definition 3.4.6.

and satisfying the following properties:

(i) $\mathrm{cod} \circ E = p$.

(ii) E is 2-fully faithful (that is, an isomorphism on hom-groupoids).

(iii) E sends cartesian morphisms of $\mathfrak{T}$ to 2-pullback squares in $\mathfrak{C}$.

(iv) The normal isofibration structures picked out by E have the stability properties of Proposition 3.4.10.

In describing the remaining, logical, structure, we make free use of the conventions of Notation 3.1.1:

— For every $\Gamma \in \mathfrak{C}$ and $A \in \mathfrak{T}(\Gamma)$, there is given an object $\mathsf{Id}_A \in \mathfrak{T}(\Gamma.A.A)$ and a 2-cell $\kappa : \pi_1 \Rightarrow \pi_2 : \Gamma.A.A.\mathsf{Id}_A \to \Gamma.A$ over Γ that together provide an arrow object (in the sense of Section 4.1) for $\Gamma.A$ in $\mathfrak{C}/\Gamma$.
— For every $\Gamma \in \mathfrak{C}$, there is given a retract bireflection (in the sense of Proposition 4.2.1) $\mathbf{1}_A$ of the object $\mathrm{id}_\Gamma : \Gamma \to \Gamma$ along $E(\Gamma) : \mathfrak{T}(\Gamma) \to \mathfrak{C}_S/\Gamma$.
— For every $\Gamma \in \mathfrak{C}$ and $A \in \mathfrak{T}(\Gamma)$, there are given both left and right retract biadjoints (in the sense of Section 4.2) Σ_A and Π_A for $\mathfrak{T}(\pi_A) : \mathfrak{T}(\Gamma) \to \mathfrak{T}(\Gamma.A)$.
— For every $\Gamma \in \mathfrak{C}$, $A \in \mathfrak{T}(\Gamma)$ and $B \in \mathfrak{T}(\Gamma.A)$, there is given a choice of injective equivalence structure on the canonical morphism $i : \Gamma.A.B \to \Gamma.\Sigma_A B$ defined as in (40).
— The above structures satisfy the stability properties listed in Propositions 4.1.2, 4.3.2, 4.4.3 and 4.5.2.

5. The internal language of a two-dimensional model

5.1. *2-categorical lifting properties*

In this section we prove a converse to the results of the previous two sections. Given a model $\mathbb{C}$ of two-dimensional type theory, we will construct from it a dependent type theory $S(\mathbb{C})$ admitting the rules of Tables 1, 2 and 3. We call this type theory the *internal language* of $\mathbb{C}$. The key to doing this will be to give semantic analogues in $\mathbb{C}$ of each of the logical rules of ML_2. In giving analogues of the elimination rules, we will make use of the 2-categorical lifting property described in Proposition 5.1.1 below. This is again very much in the spirit of Awodey and Warren (2009), since this is fundamentally a result about the weak factorisation system (injective equivalences, normal isofibrations) described in Remark 3.4.8, or, rather, about an algebraic presentation of this weak factorisation system in the style of Grandis and Tholen (2006).

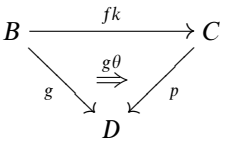
Proposition 5.1.1. Suppose we are given a 2-category $\mathbf{K}$ and a square

$$\begin{array}{ccc} A & \xrightarrow{f} & C \\ {\scriptstyle i}\downarrow & & \downarrow{\scriptstyle p} \\ B & \xrightarrow[g]{} & D \end{array} \tag{41}$$

where i carries the structure of an injective equivalence (*cf.* Section 4.2) and p that of a normal isofibration (*cf.* Definition 3.4.6). From this data we can determine a canonical diagonal filler $j : B \to C$ satisfying $pj = g$ and $ji = f$.

Proof. The injective equivalence structure on i is given by a morphism $k : B \to A$ satisfying $ki = \mathrm{id}_A$ and an invertible 2-cell $\theta : \mathrm{id}_B \Rightarrow ik$ satisfying $\theta i = \mathrm{id}_i$ and $k\theta = \mathrm{id}_k$.

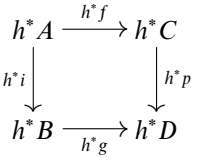
Thus we have an invertible 2-cell



and thus, from the isofibration structure on p, we obtain a map $j := s_{g\theta} : B \to C$ satisfying $pj = g$. It remains to show that $ji = f$. By the definition of isofibration, we have $ji = s_{g\theta} \circ i = s_{g\theta i}$, and since $s_{g\theta i} = s_{g(\mathrm{id}_i)} = s_{\mathrm{id}_{gi}} = s_{\mathrm{id}_{pf}}$, we deduce by normality that $ji = s_{\mathrm{id}_{pf}} = f$, as required. □

We now show that the liftings of the previous Proposition are stable under pullback in a suitable sense. Note that in order for this to make sense, it is crucial that Proposition 5.1.1 gives us a *choice* of filler for each diagram like (41).

Proposition 5.1.2. Suppose we are given a morphism $h : X \to Y$ in a 2-category $\mathbf{K}$ together with a diagram like (41) in the slice $\mathbf{K}/Y$. Suppose we are able to form the 2-pullback of this diagram along h yielding a diagram



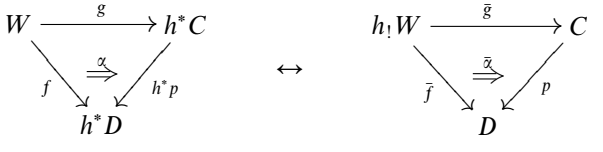
(42)

in $\mathbf{K}/X$. Then the pullback of the canonical filler for (41) along h is equal to the canonical filler for (42), where the injective equivalence structure on h^*i and the isofibration structure on h^*p are those induced by pullback.

Proof. We will first make clear what the induced structures on h^*i and h^*p look like. The injective equivalence data for h^*i is simply given by applying h^* to the corresponding data for i. The normal isofibration structure on h^*p is given as follows, writing $h_! : \mathbf{K}/X \to \mathbf{K}/Y$ for the 2-functor given by postcomposition with h. For any $V \in \mathbf{K}/Y$ whose 2-pullback h^*V along h exists, we have 2-natural bijections of categories

$$\mathbf{K}/Y(h_!U, V) \cong \mathbf{K}/X(U, h^*V). \tag{43}$$

In particular, we have bijections between diagrams of the following two forms:



(44)

So, given an α as on the left of (44), we obtain a lifting for it by:

(i) transposing to obtain a 2-cell $\bar{\alpha}$ as on the right of (44);
(ii) applying the isofibration structure of p to obtain $s_{\bar{\alpha}} : h_!W \to C$ and $\sigma_{\bar{\alpha}} : s_{\bar{\alpha}} \Rightarrow \bar{g}$;
(iii) transposing back using (43) to obtain $s_\alpha : W \to h^*C$ and $\sigma_\alpha : s_\alpha \Rightarrow g$.

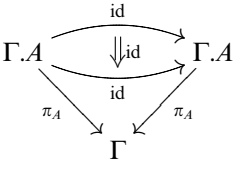
Now consider the case where α in (44) is itself of the form $h^*\beta$ for some $\beta : u \Rightarrow pv : W \to D$ in $\mathbf{K}/Y$. When this is the case, the corresponding $\bar{\alpha}$ is, by naturality, equal to $\beta \circ \epsilon_W$, where $\epsilon_W : h_! h^* W \to W$ is the transpose of id_{h^*W} under the bijection (43). It follows from the definition of isofibration that $s_{\bar{\alpha}} = s_{\beta \circ \epsilon_W} = s_\beta \circ \epsilon_W$, and, similarly, $\sigma_{\bar{\alpha}} = \sigma_\beta \circ \epsilon_W$, so, transposing under (43) and using naturality, we have $s_{h^*\beta} = h^*(s_\beta)$ and $\sigma_{h^*\beta} = h^*(\sigma_\beta)$. Now, according to Proposition 5.1.1, the canonical filler for (42) is given by $s_{(h^*g)(h^*\theta)} = s_{h^*(g\theta)}$, and, by the above argument, this is equal to $h^*(s_{g\theta})$, which is precisely the pullback along h of the canonical filler for (41), as required. □

5.2. *Identity types*

For the rest of the section we fix a model of two-dimensional type theory $\mathbb{C}$. We are going to give semantic analogues of each of the logical constructors of ML_2 in $\mathbb{C}$. We start with the identity types.

5.2.1. *Formation rule.* Given $\Gamma \in \mathfrak{C}$ and $A \in \mathfrak{T}(\Gamma)$, we define the *semantic identity type* on A to be the object $\mathsf{Id}_A \in \mathfrak{T}(\Gamma.A.A)$ whose existence is assured by Definition 4.6.1.

5.2.2. *Introduction rule.* We recall that the object $\Gamma.A.A.\mathsf{Id}_A \in \mathfrak{C}$, together with the maps $s := \pi_1\pi_{\mathsf{Id}_A}$ and $t := \pi_2\pi_{\mathsf{Id}_A} : \Gamma.A.A.\mathsf{Id}_A \to \Gamma.A$, is an arrow object for $\Gamma.A$ in $\mathfrak{C}/\Gamma$. As in Proposition 4.1.1, we write $\kappa : s \Rightarrow t$ for the corresponding universal 2-cell. Applying universality of κ to the 2-cell



in $\mathfrak{C}/\Gamma$, we obtain a morphism $r_A : \Gamma.A \to \Gamma.A.A.\mathsf{Id}_A$ that factorises the diagonal: we have $\pi_{\mathsf{Id}_A} r_A = \delta_A : \Gamma.A \to \Gamma.A.A$. We call this r_A the *semantic introduction rule* for Id_A.

5.2.3. *Elimination and computation rules.* With reference to Table 1, we require semantic analogues of the premisses C and d of the rule Id-ELIM. These are given by an object $C \in \mathfrak{T}(\Gamma.A.A.\mathsf{Id}_A)$ and a map $d : \Gamma.A \to \Gamma.A.A.\mathsf{Id}_A.C$ of $\mathfrak{C}$ making the following diagram commute:

$$\begin{array}{ccc} \Gamma.A & \xrightarrow{d} & \Gamma.A.A.\mathsf{Id}_A.C \\ {\scriptstyle r_A}\downarrow & & \downarrow{\scriptstyle \pi_C} \\ \Gamma.A.A.\mathsf{Id}_A & \xrightarrow[\mathrm{id}]{} & \Gamma.A.A.\mathsf{Id}_A \end{array} \tag{45}$$

To give a semantic analogue of the conclusion J_d satisfying the analogue of the computation rule amounts to giving a filler $J_d : \Gamma.A.A.\mathsf{Id}_A \to \Gamma.A.A.\mathsf{Id}_A.C$ making both sides of (45) commute. Now, by Proposition 3.4.7, π_C is a normal isofibration in $\mathfrak{C}/\Gamma$, so if we can show that r_A is an injective equivalence in $\mathfrak{C}/\Gamma$, we can obtain the required filler J_d by an application of Proposition 5.1.1. To show that r_A is an injective equivalence in $\mathfrak{C}/\Gamma$, we

must first give a retraction of r_A over Γ. We take this to be $t : \Gamma.A.A.\mathsf{Id}_A \to \Gamma.A$ (though we could equally well have chosen s), and we have that $tr_A = \mathrm{id}_{\Gamma.A}$ as required. Next we need a 2-cell $\theta : \mathrm{id} \Rightarrow r_A t$ over Γ satisfying $\theta r_A = \mathrm{id}_{r_A}$ and $t\theta = \mathrm{id}_t$. For this we consider the following diagram of 1- and 2-cells $\Gamma.A.A.\mathsf{Id}_A \to \Gamma.A$:

$$\begin{array}{ccc} s & \xrightarrow{\kappa} & sr_A t \\ {\scriptstyle\kappa}\Downarrow & & \Downarrow{\scriptstyle \mathrm{id}_t} \\ t & \xrightarrow[\mathrm{id}_t]{} & tr_A t \end{array}$$

Because $tr_A = sr_A = \mathrm{id}_{\Gamma.A}$, this diagram is commutative, and thus, by the two-dimensional aspect of the universal property of $\Gamma.A.A.\mathsf{Id}_A$, it is induced by a 2-cell $\theta : \mathrm{id} \Rightarrow r_A t$ over Γ satisfying $s\theta = \kappa$ and $t\theta = \mathrm{id}_t$. It remains to verify that $\theta r_A = \mathrm{id}_{r_A}$. By the uniqueness part of the universal property of $\Gamma.A.A.\mathsf{Id}_A$, it suffices to show that $\kappa \circ \theta r_A = \kappa \circ \mathrm{id}_{r_A}$. But here we have $\kappa\theta = \kappa(r_A t) \circ s\theta = \mathrm{id}_t \circ \kappa = \kappa$, so that $\kappa \circ \theta r_A = \kappa r_A = \kappa \circ \mathrm{id}_{r_A}$, as required.

5.2.4. *Stability rules.* We now verify that the semantic identity rules given above are stable under semantic substitution. So suppose we are given $f : \Delta \to \Gamma$ in $\mathfrak{C}$ together with $A \in \mathfrak{T}(\Gamma)$. We must verify three things:

(i) We must show that reindexing $\Gamma.A.A.\mathsf{Id}_A$ along f yields $\Delta.f^*A.f^*A.\mathsf{Id}_{f^*A}$. This follows immediately from the stability requirements of Proposition 4.1.2.

(ii) We must show that the semantic introduction rule r_{f^*A} is the reindexing along f of r_A. This follows from the fact that arrow object structure on Id_{f^*A} is the reindexing of that on Id_A along f.

(iii) We must show that applications of the semantic elimination rule are stable under substitution. So suppose we are given a diagram like (45). If we view this as a diagram in $\mathfrak{C}/\Gamma$, we can reindex it along f to yield a diagram

$$\begin{array}{ccc} \Delta.f^*A & \xrightarrow{f^*d} & \Delta.f^*A.f^*A.\mathsf{Id}_{f^*A}.Cf \\ {\scriptstyle r_{f^*A}}\downarrow & & \downarrow{\scriptstyle \pi_{Cf}} \\ \Delta.f^*A.f^*A.\mathsf{Id}_{f^*A} & \xrightarrow[\mathrm{id}]{} & \Delta.f^*A.f^*A.\mathsf{Id}_{f^*A} \end{array} \tag{46}$$

in $\mathfrak{C}/\Delta$. We must show that pulling back the assigned filler for (45) along f yields the assigned filler for (46). Now, by the stability properties of Proposition 3.4.10, we know that the isofibration structure on π_{Cf} *qua* map of $\mathfrak{C}/\Delta$ is the one induced by pulling back along f the isofibration structure of π_C *qua* map of $\mathfrak{C}/\Gamma$. Moreover, by the stability of the arrow object structure of Id_A, the injective equivalence structure on r_{f^*A} is the one induced by pulling back that of r_A along f. The result now follows by applying Proposition 5.1.2.

5.2.5. *Remark.* Because $\Gamma.A.A.\mathsf{Id}_A$ is an arrow object in $\mathfrak{C}/\Gamma.A$, in what follows we will pass back and forth without comment between morphisms $h : \Lambda \to \Gamma.A.A.\mathsf{Id}_A$ and 2-cells $\gamma : sh \Rightarrow th : \Lambda \to \Gamma.A$ over Γ.

5.2.6. *Discrete identity rules.* We now show that the semantic identity rules given above satisfy the semantic equivalents of the rules in Table 2. So suppose we are given $\Gamma \in \mathfrak{C}$ and $A \in \mathfrak{T}(\Gamma)$ as before. The semantic analogues of the premisses of the rules in Table 2 are a pair of morphisms $a, b : \Gamma \to \Gamma.A$ of $\mathfrak{C}$ over Γ, together with a 2-cell

$$\Gamma \;\underset{q}{\overset{p}{\Downarrow \alpha}}\; \Gamma.A.A.\mathsf{Id}_A$$

satisfying $sp = sq = a$, $s\alpha = \mathrm{id}_a$, $tp = tq = b$ and $t\alpha = \mathrm{id}_b$. We must show that under these circumstances we have $p = q$ and $\alpha = \mathrm{id}_p$. So consider the following diagram of 1- and 2-cells $\Gamma \to \Gamma.A$:

$$\begin{array}{ccc} sp & \xRightarrow{s\alpha} & sq \\ {\scriptstyle \kappa p}\Downarrow & & \Downarrow{\scriptstyle \kappa q} \\ tp & \xRightarrow[t\alpha]{} & tq \end{array}$$

It is commutative with both sides equal to $\kappa\alpha : sp \Rightarrow tq$, but since $s\alpha = \mathrm{id}_a$ and $t\alpha = \mathrm{id}_b$, we deduce that $\kappa\alpha = \kappa p = \kappa q : a \Rightarrow b$. By the uniqueness part of the universal property of κ, this entails that $p = q : \Gamma \to \Gamma.A.A.\mathsf{Id}_A$. Moreover, we have $\kappa\alpha = \kappa p = \kappa \mathrm{id}_p$, so, again by the uniqueness part of the universal property of κ, we deduce that $\alpha = \mathrm{id}_p$, as required.

5.3. *Unit types*

5.3.1. *Formation rule.* Given $\Gamma \in \mathfrak{C}$, we define the *semantic unit type* at Γ to be the object $\mathbf{1}_\Gamma \in \mathfrak{T}(\Gamma)$, whose existence is assured by Definition 4.6.1.

5.3.2. *Introduction rule.* Recall that $\mathbf{1}_\Gamma$ is a retract bireflection of $\mathrm{id}_\Gamma : \Gamma \to \Gamma$ along the 2-functor $E(\Gamma) : \mathfrak{T}(\Gamma) \to \mathfrak{C}/\Gamma$. So, in particular, we have a unit map $u_\Gamma : \Gamma \to \Gamma.\mathbf{1}_\Gamma$ over Γ, and we call this the *semantic introduction rule* for $\mathbf{1}_\Gamma$.

5.3.3. *Elimination and computation rules.* Suppose we are given $C \in \mathfrak{T}(\Gamma.\mathbf{1}_\Gamma)$ and a map $d : \Gamma \to \Gamma.\mathbf{1}_\Gamma.C$ of $\mathfrak{C}$ fitting into a commutative diagram

$$\begin{array}{ccc} \Gamma & \xrightarrow{d} & \Gamma.\mathbf{1}_\Gamma.C \\ {\scriptstyle u}\downarrow & & \downarrow{\scriptstyle \pi_C} \\ \Gamma.\mathbf{1}_\Gamma & \xrightarrow[\mathrm{id}]{} & \Gamma.\mathbf{1}_\Gamma \end{array}$$

The semantic elimination rule should assign to this data a filler $U : \Gamma.\mathbf{1}_\Gamma \to \Gamma.\mathbf{1}_\Gamma.C$ making both triangles commute. Because π_C is an isofibration in $\mathfrak{C}/\Gamma$, all we need to do is show that u_Γ is an injective equivalence in $\mathfrak{C}/\Gamma$, since then we obtain the desired filler by Proposition 5.1.1. First we must give a retraction for u_Γ over Γ. We take this to be $k := \pi_{\mathbf{1}_\Gamma} : \Gamma.\mathbf{1}_\Gamma \to \Gamma$, which satisfies $ku_\Gamma = \mathrm{id}_\Gamma$ as required. We now give a 2-cell $\theta : \mathrm{id}_{\Gamma.\mathbf{1}_\Gamma} \Rightarrow u_\Gamma k$ satisfying $\theta u_\Gamma = \mathrm{id}_{u_\Gamma}$ and $k\theta = \mathrm{id}_k$. By the two-dimensional aspect of the universal property of $\mathbf{1}_\Gamma$, every 2-cell $\alpha : \mathrm{id}_{\Gamma.\mathbf{1}_\Gamma} \circ u_\Gamma \Rightarrow u_\Gamma k \circ u_\Gamma$ is of the form $\bar{\alpha} \circ u_\Gamma$ for a

unique 2-cell $\bar{\alpha} : \mathrm{id}_{\Gamma.\mathbf{1}_\Gamma} \Rightarrow u_\Gamma k$. But because $\mathrm{id}_{\Gamma.\mathbf{1}_\Gamma} \circ u_\Gamma = \mathrm{id}_{u_\Gamma} = u_\Gamma \circ \mathrm{id}_\Gamma = u_\Gamma k u_\Gamma$, we have, in particular, the 2-cell $\theta := \overline{\mathrm{id}_{u_\Gamma}} : \mathrm{id}_{\Gamma.\mathbf{1}_\Gamma} \Rightarrow u_\Gamma k$, which satisfies $\theta u_\Gamma = \mathrm{id}_{u_\Gamma}$ by definition. That it also satisfies $k\theta = \mathrm{id}_k$ follows from the fact that θ is a 2-cell of $\mathfrak{C}/\Gamma$.

5.3.4. *Stability rules.* We must show that the semantic unit rules are stable under semantic substitution. This follows by an argument entirely analogous to that of Section 5.2.4, but using the stability properties of Proposition 4.3.2 rather than Proposition 4.1.2.

5.4. *Sum types*

5.4.1. *Formation rule.* Given $\Gamma \in \mathfrak{C}$, $A \in \mathfrak{T}(\Gamma)$ and $B \in \mathfrak{T}(\Gamma.A)$, we define the *semantic sum type* of A and B to be the object $\Sigma_A(B) \in \mathfrak{T}(\Gamma)$ whose existence is assured by Definition 4.6.1.

5.4.2. *Introduction rule.* $\Sigma_A(B)$ is a retract bireflection of $B \in \mathfrak{T}(\Gamma.A)$ along the 2-functor $\mathfrak{T}(\pi_A) : \mathfrak{T}(\Gamma) \to \mathfrak{T}(\Gamma.A)$, and thus, as in Proposition 4.4.2, we obtain from the unit of this bireflection a map $i : \Gamma.A.B \to \Gamma.\Sigma_A(B)$ of $\mathfrak{C}/\Gamma$. We declare this map to be the *semantic introduction rule* for $\Sigma_A(B)$.

5.4.3. *Elimination and computation rules.* We consider $C \in \mathfrak{T}(\Gamma.\Sigma_A(B))$ and a morphism $d : \Gamma.A.B \to \Gamma.\Sigma_A(B).C$ of $\mathfrak{C}$ fitting into a commutative diagram

$$\begin{array}{ccc} \Gamma.A.B & \xrightarrow{d} & \Gamma.\Sigma_A(B).C \\ {\scriptstyle i}\downarrow & & \downarrow{\scriptstyle \pi_C} \\ \Gamma.\Sigma_A(B) & \xrightarrow[\mathrm{id}]{} & \Gamma.\Sigma_A(B) \end{array}$$

To give the semantic elimination rule satisfying the semantic computation rule, we now have to give a filler $E : \Gamma.\Sigma_A(B) \to \Gamma.\Sigma_A(B).C$ making both triangles commute. We know that π_C is an isofibration in $\mathfrak{C}/\Gamma$, whilst Definition 4.6.1 assures us that i is an injective equivalence in $\mathfrak{C}/\Gamma$, so we obtain the desired filler by applying Proposition 5.1.1.

5.4.4. *Stability rules.* We must show that the semantic rules for dependent sums are stable under semantic substitution. Again, this follows by an argument analogous to that of Section 5.2.4, but this time using the stability properties of Proposition 4.4.3.

5.5. *Product types*

Finally, we give semantic analogues in $\mathbb{C}$ of the rules for the product types. As in the one-dimensional case, there is a slight mismatch here between the syntax and the semantics. This means that, in addition to the right biadjoints to weakening, we will also need to make use of the semantic unit types of Section 5.3 – see Jacobs (1993, Sections 5.1–5.3) for a fuller discussion of this point.

5.5.1. *Formation rule.* For $\Gamma \in \mathfrak{C}$, $A \in \mathfrak{T}(\Gamma)$ and $B \in \mathfrak{T}(\Gamma.A)$, we define the *semantic product type* of A and B to be the object $\Pi_A(B) \in \mathfrak{T}(\Gamma)$ whose existence is assured by Definition 4.6.1.

5.5.2. *Application rule.* $\Pi_A(B)$ is a retract bicoreflection of $B \in \mathfrak{T}(\Gamma.A)$ along $\Delta_A := \mathfrak{T}(\pi_A) : \mathfrak{T}(\Gamma) \to \mathfrak{T}(\Gamma.A)$. The counit of this bicoreflection is a morphism $\epsilon : \Delta_A \Pi_A(B) \to B$ of $\mathfrak{T}(\Gamma.A)$. We define the *semantic application rule* for $\Pi_A(B)$ to be the corresponding morphism $\epsilon : \Gamma.A.\Pi_A(B) \to \Gamma.A.B$ of $\mathfrak{C}/\Gamma.A$.

5.5.3. *Abstraction and β-rules.* For these, we suppose given, as in the premiss of the abstraction rule, a morphism $f : \Gamma.A \to \Gamma.A.B$ over $\Gamma.A$. We are required to produce from this a map $\lambda(f) : \Gamma \to \Gamma.\Pi_A(B)$ over Γ, which, in order for the β-rule to hold, should satisfy $\epsilon \circ \Delta_A(\lambda(f)) = f$. So, consider the unit type $\mathbf{1}_{\Gamma.A} \in \mathfrak{T}(\Gamma.A)$. Applying its universal property to $f : \Gamma.A \to \Gamma.A.B$ yields a morphism $\overline{f} : \Gamma.A.\mathbf{1}_{\Gamma.A} \to \Gamma.A.B$ over $\Gamma.A$ satisfying $\overline{f} \circ u_{\Gamma.A} = f$. We can view $\overline{f}$ as a morphism $\mathbf{1}_{\Gamma.A} \to B$ of $\mathfrak{T}(\Gamma.A)$, which by the stability of unit types under substitution is equally well a morphism $\overline{f} : \Delta_A \mathbf{1}_\Gamma \to B$ of $\mathfrak{T}(\Gamma.A)$. Applying the universal property of $\Pi_A(B)$ to this, we obtain a morphism $\overline{\overline{f}} : \mathbf{1}_\Gamma \to \Pi_A(B)$ of $\mathfrak{T}(\Gamma)$ satisfying $\epsilon \circ \Delta_A(\overline{\overline{f}}) = \overline{f}$. This is equally well a morphism $\Gamma.\mathbf{1}_\Gamma \to \Gamma.\Pi_A(B)$ over Γ, so that we can define the map $\lambda(f) : \Gamma \to \Gamma.\Pi_A(B)$ over Γ to be $\lambda(f) := \overline{\overline{f}} \circ u_\Gamma$. It remains to show that we have $\epsilon \circ \Delta_A(\lambda(f)) = f$, for which we calculate that

$$\epsilon \circ \Delta_A(\lambda(f)) = \epsilon \circ \left(\Delta_A(\overline{\overline{f}}) \circ \Delta_A(u_\Gamma)\right) = \overline{f} \circ u_{\Gamma.A} = f,$$

as required. Here we have used the fact that, by the stability of unit types under substitution, we have $\Delta_A(u_\Gamma) = u_{\Gamma.A}$.

5.5.4. *Function extensionality rules.* We now give semantic analogues of the rules of Table 3. For the first rule, Π-EXT, we suppose given morphisms $m, n : \Gamma \to \Gamma.\Pi_A(B)$ over Γ, together with a 2-cell

$$p : \epsilon \circ \Delta_A(m) \Rightarrow \epsilon \circ \Delta_A(n) : \Gamma.A \to \Gamma.A.B$$

over $\Gamma.A$. We must produce from this a 2-cell $\mathsf{ext}(p) : m \Rightarrow n$. First we apply the universal property of the unit type $\mathbf{1}_\Gamma$ to m and n to obtain morphisms $\overline{m}, \overline{n} : \Gamma.\mathbf{1}_\Gamma \to \Gamma.\Pi_A(B)$ over Γ. These satisfy $m = \overline{m} \circ u_\Gamma$ and $n = \overline{n} \circ u_\Gamma$, and thus we can view p as a 2-cell

$$p : \epsilon \circ \Delta_A(\overline{m}) \circ u_{\Gamma.A} \Rightarrow \epsilon \circ \Delta_A(\overline{n}) \circ u_{\Gamma.A} : \Gamma.A \to \Gamma.A.B,$$

where again we use stability of unit types under pullback to derive that $\Delta_A(u_\Gamma) = u_{\Gamma.A}$. By the two-dimensional aspect of the universal property of $\mathbf{1}_{\Gamma.A}$, we have $p = \overline{p} \circ u_{\Gamma.A}$ for a unique 2-cell

$$\overline{p} : \epsilon \circ \Delta_A(\overline{m}) \Rightarrow \epsilon \circ \Delta_A(\overline{n}) : \Gamma.A.\mathbf{1}_{\Gamma.A} \to \Gamma.A.B.$$

Now, by the two-dimensional aspect of the universal property of $\Pi_A(B)$, we have that $\overline{p} = \epsilon \circ \Delta_A(\overline{\overline{p}})$ for a unique $\overline{\overline{p}} : \overline{m} \Rightarrow \overline{n}$. We now define the 2-cell $\mathsf{ext}(p)$ to be given by $\overline{\overline{p}} \circ u_\Gamma : m \Rightarrow n$.

In order for ext to satisfy the analogue of the rule Π-EXT-COMP, we must show that when $m = n$ and $p = \mathrm{id}_{\epsilon \circ \Delta_A(m)}$, we have $\mathsf{ext}(p) = \mathrm{id}_m$. It is enough for this to show

that (with the above notation) $\overline{\overline{p}} = \mathrm{id}_{\overline{m}} : \overline{m} \Rightarrow \overline{m}$, which, by applying the universal properties of $\Pi_A(B)$ and $\mathbf{1}_{\Gamma.A}$ successively, follows from the fact that $\epsilon \circ \Delta_A(\overline{\overline{p}}) \circ u_{\Gamma.A} = p$ is an identity 2-cell. Finally, we must verify that ext satisfies the analogue of the rule Π-EXT-APP. Recall from Section 4.5 that the operation $*$ appearing in Π-EXT-APP is simply the lifting of $\epsilon : \Gamma.A.\Pi_A(B) \to \Gamma.A.B$ to identity types. From this it follows that we must verify that $\epsilon \circ \Delta_A(\mathsf{ext}(p)) = p : \epsilon \circ \Delta_A(m) \Rightarrow \epsilon \circ \Delta_A(n)$. We calculate that $\epsilon \circ \Delta_A(\mathsf{ext}(p)) = \epsilon \circ \big(\Delta_A(\overline{\overline{p}}) \circ \Delta_A(u_\Gamma)\big) = \overline{p} \circ u_{\Gamma.A} = p$, as required.

5.5.5. *Stability rules.* We must now show that the semantic rules for dependent products are stable under semantic substitution. This follows by an argument analogous to that of Section 5.2.4, though this time we do not need the stability properties of isofibrations (Proposition 3.4.10) at all; instead, we need those for products (Proposition 4.5.2) and also those for units (Proposition 4.3.2).

5.6. *The internal language*

We now define the type theory $\mathrm{S}(\mathbb{C})$ associated to our two-dimensional model $\mathbb{C}$. It is obtained by recursively extending ML_2 with additional inference rules. These inference rules are 'axiom' rules with no premisses, so they may be specified by giving only their conclusion. First we have rules introducing new types:

— For each $A \in \mathfrak{T}(1)$ we add a judgement $\vdash \overline{A}\ \mathsf{type}$.
— For each $A \in \mathfrak{T}(1)$, $B \in \mathfrak{T}(1.A)$, we add a judgement $x : \overline{A} \vdash \overline{B}(x)\ \mathsf{type}$.
— And so on.

Then we have rules introducing new terms:

— For each $A \in \mathfrak{T}(1)$, $a \in_1 A$, we add a judgement $\vdash \overline{a} : \overline{A}$.
— For each $A \in \mathfrak{T}(1)$, $B \in \mathfrak{T}(1.A)$, $b \in_{1.A} B$, we add a judgement $x : \overline{A} \vdash \overline{b}(x) : \overline{B}(x)$.
— And so on.

Here, we use the convention for global sections developed in Notation 3.1.1. Next we have rules identifying the syntactic notions of substitution, weakening, contraction and exchange with their semantic counterparts in $\mathbb{C}$. We give the case of substitution as a representative sample. First we deal with substitution in types:

— For each $A \in \mathfrak{T}(1)$, $B \in \mathfrak{T}(1.A)$, $a \in_1 1.A$, we add a judgement $\vdash \overline{B}(\overline{a}) = \overline{a^* B}\ \mathsf{type}$.
— For each $A \in \mathfrak{T}(1)$, $B \in \mathfrak{T}(1.A)$, $C \in \mathfrak{T}(1.A.B)$, $b \in_{1.A} B$, we add a judgement $x : \overline{A} \vdash \overline{C}(x, \overline{b}(x)) = \overline{b^* C}\ \mathsf{type}$.
— For each $A \in \mathfrak{T}(1)$, $B \in \mathfrak{T}(1.A)$, $C \in \mathfrak{T}(1.A.B)$ and $a \in_1 A$, we add a judgement $y : \overline{B}(\overline{a}) \vdash \overline{C}(\overline{a}, y) = \overline{(a.\imath)^* C}\ \mathsf{type}$.
— And so on.

We now consider substitution in terms:

— For each $A \in \mathfrak{T}(1)$, $B \in \mathfrak{T}(1.A)$, $a \in_1 A$, $b \in_{1.A} B$, we add $\vdash \overline{b}(\overline{a}) = \overline{a^* b} : \overline{B}(\overline{a})$.
— For each $A \in \mathfrak{T}(1)$, $B \in \mathfrak{T}(1.A)$, $C \in \mathfrak{T}(1.A.B)$, $b \in_{1.A} B$, $c \in_{1.A.B} C$, we add $x : \overline{A} \vdash \overline{c}(x, \overline{b}(x)) = \overline{b^* c} : \overline{C}(x, \overline{c}(x))$.

— For each $A \in \mathfrak{T}(1)$, $B \in \mathfrak{T}(1.A)$, $C \in \mathfrak{T}(1.A.B)$, $a \in_1 A$, $c \in_{1.A.B} C$, we add $y : \overline{B}(\overline{a}) \vdash \overline{c}(\overline{a}, y) = \overline{(a.\imath)^* c} : \overline{C}(\overline{a}, y)$.
— And so on.

Finally, we have rules identifying each of the logical rules of ML_2 with its semantic counterpart in $\mathbb{C}$. We only give the case of the identity types; the remainder follow the same pattern. First we have the formation rules:

— For each $A \in \mathfrak{T}(1)$, we add $x, y : \overline{A} \vdash \mathsf{Id}_{\overline{A}}(x, y) = \overline{\mathsf{Id}_A}(x, y)\ \mathsf{type}$.
— For each $A \in \mathfrak{T}(1)$, $B \in \mathfrak{T}(1.A)$, we add

$$x : \overline{A},\ y, z : \overline{B}(x) \vdash \mathsf{Id}_{\overline{B}(x)}(y, z) = \overline{\mathsf{Id}_B}(x, y, z)\ \mathsf{type}.$$

— And so on.

Next we have the introduction rule. Note that for $A \in \mathfrak{T}(\Gamma)$, the semantic introduction rule $r_A : \Gamma \to \Gamma.A.A.\mathsf{Id}_A$ over Γ can be viewed as a global section $r_A \in_\Gamma \delta_A^*(\mathsf{Id}_A)$, where $\delta_A : \Gamma.A \to \Gamma.A.A$ is the diagonal morphism. Thus we may add the following rules:

— For each $A \in \mathfrak{T}(1)$, we add $x : \overline{A} \vdash \mathrm{r}(x) = \overline{r_A}(x) : \mathsf{Id}_{\overline{A}}(x, x)$.
— For each $A \in \mathfrak{T}(1)$, $B \in \mathfrak{T}(1.A)$, we add

$$x : \overline{A},\ y : \overline{B}(x) \vdash \mathrm{r}(y) = \overline{r_B}(x, y) : \mathsf{Id}_{\overline{B}(x)}(y, y).$$

— And so on.

Finally, we come to the identity elimination rule:

— For each $A \in \mathfrak{T}(1)$, $C \in \mathfrak{T}(1.A.A.\mathsf{Id}_A)$ and $d : 1.A \to 1.A.A.\mathsf{Id}_A.C$ as in (45) (which is equally well a global section $d \in_{1.A} r_A^* C$), we add

$$x, y : \overline{A},\ p : \mathsf{Id}_{\overline{A}}(x, y) \vdash \mathrm{J}_{\overline{d}}(x, y, p) = \overline{J_d}(x, y, p) : \overline{C}(x, y, p).$$

— And so on.

Now, in order for the internal language we have set up to be of any use, we require its types and terms to denote unique elements of the model $\mathbb{C}$. The next Proposition tells us that this is the case.

Proposition 5.6.1 (Soundness). For any $B, C \in \mathfrak{T}(1.A_1.A_2 \ldots A_n)$, if the judgement

$$x : \overline{A}_1.\overline{A}_2 \ldots \overline{A}_n \vdash \overline{B}(x) = \overline{C}(x)\ \mathsf{type}$$

is derivable, then $B = C$. Likewise, for global sections $b, c \in_{1.A_1 \ldots A_n} B$, if the judgement

$$x : \overline{A}_1.\overline{A}_2 \ldots \overline{A}_n \vdash \overline{b}(x) = \overline{c}(x) : \overline{B}(x)$$

is derivable, then $b = c$.

Proof. By induction on the derivation of the judgement in question, it suffices to show that the semantic counterpart of each syntactic equality rules is satisfied. For the non-logical equality rules, this is standard (though delicate), and we refer the reader to Hofmann (1995a) or Pitts (2000) for the details (note that we make essential use of the fact that the underlying 1-fibration of $\mathfrak{T} \to \mathfrak{C}$ is split). The other cases we must consider are the computation rules of Tables 1, 2 or 3, and the rules expressing stability of the logical

operations under substitution, and each of these has been dealt with in the preceding sections. □

Remark 5.6.2. Observe that the internal language $\mathrm{S}(\mathbb{C})$ does not give us access to all of the model $\mathbb{C}$: it only allows us to talk about objects of the base 2-category $\mathfrak{C}$ that have the form $1.A_0 \ldots A_n$ (where 1 is the given 2-terminal object). There are two ways around this. We can modify the syntax of our type theory so that contexts and context morphisms are primitive, rather than derived, notions. Then each object or morphism of $\mathfrak{C}$ corresponds directly to a context or context morphism of $\mathrm{S}(\mathbb{C})$. Alternatively, we can keep our type theory the same, and instead work with relative internal languages. Given $\Gamma \in \mathfrak{C}$, the *relative internal language* $\mathrm{S}_\Gamma(\mathbb{C})$ is the type theory whose closed types are objects of $\mathfrak{T}(\Gamma)$, with dependent types being objects of $\mathfrak{T}(\Gamma.A)$, $\mathfrak{T}(\Gamma.A.B)$ and so on. Moreover, because each morphism $\Gamma \to \Delta$ of $\mathfrak{C}$ induces an interpretation (in the sense of Section 5.7 below) $\mathrm{S}_\Delta(\mathbb{C}) \to \mathrm{S}_\Gamma(\mathbb{C})$, we obtain what is in an obvious sense a '$\mathfrak{C}$-indexed type theory'[†].

5.7. *Functorial aspects*

In Sections 3 and 4, we constructed from each type theory S incorporating ML_2 a two-dimensional model $\mathbb{C}(\mathrm{S})$, whilst in the preceding parts of the present section, we have constructed from each two-dimensional model $\mathbb{C}$ a type theory $\mathrm{S}(\mathbb{C})$ incorporating ML_2. It is natural to ask whether these assignations give rise to a *functorial semantics* in the spirit of Lawvere (1968). That is, can we define a syntactic category of type theories and a semantic category of models for which the above assignations underlie an equivalence of categories? We conclude the paper by sketching a positive answer to this question.

We first define a syntactic category **Th**. Its objects are the *generalised algebraic theories* (Cartmell 1986) over ML_2. These are defined inductively by the following three clauses:

(i) Each object of **Th** is a sequent calculus.
(ii) $\mathrm{ML}_2 \in$ **Th**.
(iii) If $\mathrm{S} \in$ **Th**, then so is any extension of S.

Here, an *extension* of S is given by adjoining a set of inference rules, each of which has no premisses, and a conclusion $\mathscr{J}$ that obeys the following requirements:

— If $\mathscr{J}$ is of the form $\Gamma \vdash A$ type, then A must be fresh for S and Γ must be a well-formed context of S.
— If $\mathscr{J}$ is of the form $\Gamma \vdash a : A$, then a must be fresh for S and $\Gamma \vdash A$ type must be derivable in S.
— If $\mathscr{J}$ is of the form $\Gamma \vdash A = B$ type, then $\Gamma \vdash A$ type and $\Gamma \vdash B$ type must be derivable in S
— If $\mathscr{J}$ is of the form $\Gamma \vdash a = b : A$, then $\Gamma \vdash a : A$ and $\Gamma \vdash b : A$ must be derivable in S.

Note that the assignation $\mathbb{C} \mapsto \mathrm{S}(\mathbb{C})$ sends each two-dimensional model to a GAT over ML_2.

[†] A finer analysis shows that this is really a two-dimensional indexing. That is, we have a trihomomorphism $\mathfrak{C}^{\mathrm{coop}} \to$ **Th**, where **Th** is a suitably defined tricategory of two-dimensional theories.

The morphisms of **Th** are equivalence classes of interpretations. Given $\mathrm{S}, \mathrm{T} \in \mathbf{Th}$, an *interpretation* $F : \mathrm{S} \to \mathrm{T}$ is a function F taking derivable judgements of S to derivable judgements of T, subject to the following requirements:

— Each $F(A\ \mathsf{type})$ should have the form $FA\ \mathsf{type}$.
— Each $F(a : A)$ should have the form $Fa : FA$.
— Each $F(A = B\ \mathsf{type})$ should have the form $FA = FB\ \mathsf{type}$.
— Each $F(a = b : A)$ should have the form $Fa = Fb : FA$.

Moreover, if we suppose $F(\Gamma \vdash A\ \mathsf{type})$ has the form $F\Gamma \vdash FA\ \mathsf{type}$, then:

— Each $F(\Gamma,\ x : A \vdash B(x)\ \mathsf{type})$ should have the form $F\Gamma,\ x : FA \vdash FB(x)\ \mathsf{type}$.
— Each $F(\Gamma,\ x : A \vdash b(x) : B(x))$ should have the form $F\Gamma,\ x : FA \vdash Fb(x) : FB(x)$.
— And similarly for the two equality judgement forms.

Finally, we require that F should commute with all the inference rules of ML_2. We give the case of the rule of Id-formation for illustration. Suppose we are given a derivable judgement $\Gamma \vdash A\ \mathsf{type}$ in S. We write its image under F as $F\Gamma \vdash FA\ \mathsf{type}$, and the image of $\Gamma,\ x, y : A \vdash \mathsf{Id}_A(x, y)\ \mathsf{type}$ as $F\Gamma,\ x, y : FA \vdash F\mathsf{Id}_A(x, y)\ \mathsf{type}$. Now the following judgement should be derivable in T:

$$F\Gamma,\ x, y : FA \vdash \mathsf{Id}_{FA}(x, y) = F\mathsf{Id}_A(x, y)\ \mathsf{type}.$$

The equivalence relation we impose on interpretations identifies $F, G : \mathrm{S} \to \mathrm{T}$ if they differ only up to definitional equality in the obvious sense. It is now straightforward to show that GATs and equivalence classes of interpretations form a category **Th**.

Remark 5.7.1. Using the above notion of interpretation, we can now say what it means to give an interpretation of a GAT T in a two-dimensional model $\mathbb{C}$, namely, to give an interpretation (in the above sense) $\mathrm{T} \to \mathrm{S}(\mathbb{C})$. It is easy to check that this accords with the intuitive syntactic notion we would give.

We now define a semantic category **Mod**. Its objects are models of two-dimensional type theory as in Definition 4.6.1. A morphism $F : \mathbb{C} \to \mathbb{C}'$ is given by a pair of 2-functors $F_1 : \mathfrak{C} \to \mathfrak{C}'$ and $F_2 : \mathfrak{T} \to \mathfrak{T}'$ rendering the squares

$$\begin{array}{ccc} \mathfrak{T} & \xrightarrow{F_2} & \mathfrak{T}' \\ {\scriptstyle p}\downarrow & & \downarrow{\scriptstyle p'} \\ \mathfrak{C} & \xrightarrow[F_1]{} & \mathfrak{C}' \end{array} \qquad \text{and} \qquad \begin{array}{ccc} \mathfrak{T} & \xrightarrow{F_2} & \mathfrak{T}' \\ {\scriptstyle E}\downarrow & & \downarrow{\scriptstyle E'} \\ \mathfrak{C}^{\mathbf{2}} & \xrightarrow[F_1{}^{\mathbf{2}}]{} & (\mathfrak{C}')^{\mathbf{2}} \end{array}$$

commutative and preserving all the additional structure on the nose.

Proposition 5.7.2. The assignations $\mathrm{S} \mapsto \mathbb{C}(\mathrm{S})$ and $\mathbb{C} \mapsto \mathrm{S}(\mathbb{C})$ underlie functors $\mathbb{C}(-) : \mathbf{Th} \to \mathbf{Mod}$ and $\mathrm{S}(-) : \mathbf{Mod} \to \mathbf{Th}$.

Proof. Given an interpretation $F : \mathrm{S} \to \mathrm{T}$, we define functors $G_0 : \mathfrak{C}(\mathrm{S}) \to \mathfrak{C}(\mathrm{T})$ and $G_1 : \mathfrak{T}(\mathrm{S}) \to \mathfrak{T}(\mathrm{T})$ by an obvious structural induction over the objects and morphisms of the domain categories. In order to extend these functors to 2-functors, we first show by induction that every object $\Gamma \in \mathfrak{C}(\mathrm{S})$ has an accompanying arrow object given by

the identity context Id_Γ. But now, since F preserves the identity type structure, the corresponding G_0 will preserve these arrow objects, so we may extend G_0 to a 2-functor by regarding each 2-cell of $\mathfrak{C}(\mathrm{S})$ as a 1-cell into an arrow object, mapping this 1-cell over and then turning the resulting 1-cell back into a 2-cell of $\mathfrak{C}(\mathrm{T})$. Because the comprehension 2-functors of $\mathbb{C}(\mathrm{S})$ and $\mathbb{C}(\mathrm{T})$ are 2-fully faithful, this in turn determines the extension of G_1 to a 2-functor. Finally, the fact that F strictly preserves the remaining structure implies that the same is true of (G_0, G_1), so we obtain a morphism of models $\mathbb{C}(\mathrm{S}) \to \mathbb{C}(\mathrm{T})$ as required.

Conversely, given a morphism of models $F : \mathbb{C} \to \mathbb{C}'$, we may define an interpretation $\mathrm{S}(\mathbb{C}) \to \mathrm{S}(\mathbb{C}')$ as follows. By structural induction, every closed type A of $\mathrm{S}(\mathbb{C})$ is definitionally equal to one of the form $\overline{X}$ for some $X \in \mathfrak{T}(1)$, and by Proposition 5.6.1, this X is unique. Thus we may define the image of A under the interpretation to be the type $\overline{G_1(X)}$ of $\mathrm{S}(\mathbb{C}')$. Similarly, every closed term $a : A$ of $\mathrm{S}(\mathbb{C})$ is definitionally equal to one of the form $\overline{x} : \overline{X}$ for a unique map $x : 1 \to 1.X$ of $\mathfrak{C}$, so we may define $F(a)$ to be the term $\overline{G_0(x)} : \overline{G_1(X)}$. This definition extends to types and terms in non-empty contexts in an obvious way. Finally, the fact that our morphism of models preserves all the remaining structure on the nose implies the same for the interpretation just described. □

However, the functors defined in this Proposition do not give rise to an equivalence of categories. There are two reasons for this. The first is the issue raised in Remark 5.6.2. Observe that any two-dimensional model in the image of $\mathbb{C}(-)$ has the property that each object $\Gamma \in \mathfrak{C}$ is of the form $1.A_1 \dots A_n$ for a unique (possibly empty) sequence of objects $A_1 \in \mathfrak{T}(1), \dots, A_n \in \mathfrak{T}(1.A_1 \dots A_{n-1})$. This is the 'tree condition' of Cartmell (1986). Clearly, not every two-dimensional model has this property, so if we are to obtain an equivalence, we must first cut down to the full sub-2-category $\mathbf{Mod}_{tr} \subset \mathbf{Mod}$ on those that do. The second reason we do not obtain an equivalence is more subtle. In order for $\mathbf{Th} \simeq \mathbf{Mod}_{tr}$ to hold, we must certainly have for each $\mathrm{S} \in \mathbf{Th}$ that $\mathrm{S}(\mathbb{C}(\mathrm{S})) \cong \mathrm{S}$. However, this turns out not to be the case: we run into problems with the terms witnessing the elimination rules. As an illustration, we will show that $\mathrm{ML}_2 \ncong \mathrm{S}(\mathbb{C}(\mathrm{ML}_2))$. Because the object ML_2 is initial in $\mathbf{Th}$, there is a unique morphism $F : \mathrm{ML}_2 \to \mathrm{S}(\mathbb{C}(\mathrm{ML}_2))$, so it suffices to show that F is not surjective. First observe that by $\mathbf{1}$-elimination we can derive a judgement

$$z : \mathbf{1} \vdash \mathrm{U}_\star(z) : \mathbf{1} \tag{47}$$

in ML_2. Next note that the judgements of $\mathrm{S}(\mathbb{C}(\mathrm{ML}_2))$ are simply equivalence classes of judgements of ML_2 with respect to definitional equality, so, by passing to the quotient, we obtain from (47) a judgement

$$z : [\mathbf{1}] \vdash [\mathrm{U}_\star](z) : [\mathbf{1}] \tag{48}$$

of $\mathrm{S}(\mathbb{C}(\mathrm{ML}_2))$. The crucial point is that (48) does not coincide with the value of F at the judgement (47). This latter can be described as follows. First we derive a term $z : \mathbf{1} \vdash \phi(z) : \mathsf{Id}_{\mathbf{1}}(z, \star)$ in ML_2 by $\mathbf{1}$-elimination, taking $\phi(z) := \mathrm{U}_{\mathrm{r}(\star)}(z)$. Now, by the description of the semantic unit types given in Section 5.3, we see that applying F to (47) yields (up to definitional equality) the following judgement in $\mathrm{S}(\mathbb{C}(\mathrm{ML}_2))$:

$$z : [\mathbf{1}] \vdash [\phi(z)^*(\star)] : [\mathbf{1}]. \tag{49}$$

Now, if (49) were definitionally equal to (48), we would also have that $z : \mathbf{1} \vdash \mathrm{U}_\star(z) = \phi(z)^*(\star) : \mathbf{1}$ in ML_2, and this is not the case. Hence F applied to (47) does not yield (48), from which it follows by induction over derivable judgements of ML_2 that (48) cannot lie in the image of $F : \mathrm{ML}_2 \to \mathrm{S}(\mathbb{C}(\mathrm{ML}_2))$.

We can resolve this issue in several ways. The first is to change our notion of model so that it accords more closely with the type theory. This is unsatisfactory as we have then reverted to a categorical paraphrasing of type theoretic syntax. A second alternative is to change our notion of type theory so that it accords more closely with the categorical model. This involves removing the elimination rules altogether, and instead taking the Leibniz rule, together with the injective equivalence structures on the introduction terms, as primitives. This is unsatisfactory for a more subtle reason. Whilst it may be reasonably straightforward to give this alternative presentation for two-dimensional type theory, we would find as we moved towards full intensional type theory that it would require more and more intricate sets of rules expressing appropriate coherence properties of our new primitives. The elegant simplicity of intensional type theory would be lost completely.

A third solution, and our preferred one, is to equip our categories of theories and of models with more generous notions of morphism, ones that preserve some of the structure only up to propositional, rather than definitional equality. There is a great deal of scope in how far we go with this. In the present paper, we make only the minimal modifications necessary to obtain the desired equivalence. A fuller treatment would take account of the fact that our models and theories are themselves two-dimensional structures, so that their respective totalities should give rise not merely to equivalent categories, but also to triequivalent **Gray**-categories (=semi-strict 3-categories) in the sense of Gordon *et al.* (1995). Adopting this more comprehensive approach would be necessary if, for instance, we wished to study the 2-category of interpretations of some generalised algebraic theory inside a particular two-dimensional model. However, for our present purposes, we do not need to go this far, so, in the interests of brevity, we do not.

The minimal modification that we will consider is given as follows. On the syntactic side, we define a category $\mathbf{Th}_\psi$ with as objects GATs over ML_2 and as maps $F : \mathrm{S} \to \mathrm{T}$ *pseudo-interpretations*, whose definition generalises that of an interpretation by dropping the requirement that F should preserve each of the following rules up to definitional equality: **1**-ELIM, Id-ELIM, Σ-ELIM, and Π-ABS. One may now think that, in order to justify the name pseudo-interpretation, we should ask for F to preserve these rules at least up to propositional equality, but it turns out that this is unnecessary because this weaker form of preservation is a consequence of the type-theoretic elimination rules.

On the semantic side, we define a category $\mathbf{Mod}_\psi$ whose objects are two-dimensional models and whose maps $F : \mathbb{C} \to \mathbb{C}'$ are *pseudo-morphisms*. These are obtained by relaxing in the definition of morphism of models the requirement that the following structures should be preserved:

— the normal isofibration structures on dependent projections π_A;

— the injective equivalence structures on the maps $i : \Gamma.A.B \to \Gamma.\Sigma_A(B)$ associated to dependent sums;

— the assignations $f \mapsto \overline{f}$ on 1-cells associated to the unit types, dependent sums, and dependent products.

Once again, we do not need to add conditions requiring these pieces of structure to be preserved up to isomorphism, since this will be an automatic consequence of the remaining structure. As before, we write $(\mathbf{Mod}_\psi)_{tr}$ for the full subcategory of $\mathbf{Mod}_\psi$ on those models satisfying the tree condition.

Proposition 5.7.3. The functors $\mathfrak{C}(-)$ and $\mathrm{S}(-)$ extend to functors $\mathbf{Th}_\psi \to (\mathbf{Mod}_\psi)_{tr}$ and $(\mathbf{Mod}_\psi)_{tr} \to \mathbf{Th}_\psi$, respectively.

Proof. The argument of Proposition 5.7.2 carries over almost entirely unmodified. The only subtlety arises in defining the pseudo-morphism of models $\mathfrak{C}(\mathrm{S}) \to \mathfrak{C}(\mathrm{T})$ corresponding to a pseudo-interpretation $F : \mathrm{S} \to \mathrm{T}$. As before, we define functors G_0 and G_1 by induction on the objects and morphisms of $\mathfrak{C}(\mathrm{S})$, but when it comes to extending these to 2-functors, we encounter the problem that the interpretation F, since it no longer preserves the rule Id-ELIM, may not send identity contexts to identity contexts. However, using the fact that Id-ELIM is preserved at least up to *propositional* equality, we may show by induction that F will send an identity context to something isomorphic to an identity context. From this, it follows that G_0 will still preserve arrow objects, so we may continue the argument as before. □

But now we have the following proposition.

Proposition 5.7.4. The functors $\mathfrak{C}(-)$ and $\mathrm{S}(-)$ induce an equivalence of categories $(\mathbf{Mod}_\psi)_{tr} \simeq \mathbf{Th}_\psi$.

Proof. First observe that if $\mathfrak{C}$ is a model satisfying the tree condition, then the contexts and context morphisms of $\mathrm{S}(\mathfrak{C})$ are, up to definitional equality, just the objects and morphisms of $\mathfrak{C}$, whilst the types-in-context of $\mathrm{S}(\mathfrak{C})$ are just the objects of $\mathfrak{T}$. From this it follows that $\mathrm{S}(-)$ is fully faithful. Indeed, given a pseudo-interpretation $F : \mathrm{S}(\mathfrak{C}) \to \mathrm{S}(\mathfrak{C}')$, our observation allows us to define functors $G_0 : \mathfrak{C} \to \mathfrak{C}'$ and $G_1 : \mathfrak{T} \to \mathfrak{T}'$. As in the last proof, the pseudo-interpretation F must send an identity context to something isomorphic to an identity context, from which it follows that G_0 preserves arrow objects, allowing us to extend G_0 and G_1 to 2-functors as before. It is now easy to verify that the resultant pair (G_0, G_1) is a pseudo-morphism, and by examining the proof of Proposition 5.7.2, we see that it is sent by $\mathrm{S}(-)$ to F and that it is the unique pseudo-morphism with this property.

It remains to show that for each $\mathrm{S} \in \mathbf{Th}_\psi$ we have an isomorphism $\mathrm{S} \cong \mathrm{S}(\mathfrak{C}(\mathrm{S}))$ in $\mathbf{Th}_\psi$, and that these are natural in S. Now, up to definitional equality, the judgements of $\mathrm{S}(\mathfrak{C}(\mathrm{S}))$ are the same as those of S, so we obtain mutually inverse assignations between the judgements of the former and those of the latter. Moreover, by following through the constructions of $\mathfrak{C}(-)$ and $\mathrm{S}(-)$, we see that all of the logical structure of $\mathrm{S}(\mathfrak{C}(\mathrm{S}))$ is given as in S, with the possible exception of the rules **1**-ELIM, Id-ELIM, Σ-ELIM and Π-ABS (as seen in the discussion following Proposition 5.7.2). But this says precisely that these mutually inverse assignations are pseudo-interpretations, and thus give rise to a natural isomorphism $\mathrm{S} \cong \mathrm{S}(\mathfrak{C}(\mathrm{S}))$ in $\mathbf{Th}_\psi$, as required. □

Acknowledgements

The author thanks the anonymous referees for their helpful suggestions.

References

Aczel, P. (1994) Notes towards a formalisation of constructive Galois theory. Manuscript, University of Manchester.

Awodey, S. and Warren, M. (2009) Homotopy theoretic models of identity types. *Mathematical Proceedings of the Cambridge Philosophical Society* **146** (1) 45–55.

Bousfield, A. K. (1977) Constructions of factorization systems in categories. *Journal of Pure and Applied Algebra* **9** (2-3) 207–220.

Cartmell, J. (1986) Generalised algebraic theories and contextual categories. *Annals of Pure and Applied Logic* **32** 209–243.

de Bruijn, N. (1991) Telescopic mappings in typed lambda calculus. *Information and Computation* **91** (2) 189–204.

Dybjer, P. (1996) Internal type theory. In: Types for proofs and programs (Torino, 1995). *Springer-Verlag Lecture Notes in Computer Science* **1158** 120–134.

Ehrhard, T. (1988) *Une sémantique catégorique des type dépendents. Application au Calcul des Constructions*, Ph.D. thesis, Université Paris VII.

Gambino, N. (2008) Homotopy limits for 2-categories. *Mathematical Proceedings of the Cambridge Philosophical Society* **145** (1) 43–63.

Gambino, N. and Garner, R. (2008) The identity type weak factorisation system. *Theoretical Computer Science* **409** 94–109.

Garner, R. (2009) On the strength of dependent products in the type theory of Martin-Löf. *Annals of Pure and Applied Logic* **160** 1–12.

Gordon, R., Power, J. and Street, R. (1995) Coherence for tricategories. *Memoirs of the American Mathematical Society* **117** (558).

Grandis, M. and Tholen, W. (2006) Natural weak factorization systems. *Archivum Mathematicum* **42** (4) 397–408.

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

Hofmann, M. (1994) Elimination of extensionality in martin-löf type theory. In: Types for proofs and programs (Nijmegen, 1993). *Springer-Verlag Lecture Notes in Computer Science* **806** 166–190.

Hofmann, M. (1995a) *Extensional concepts in intensional type theory*, Ph.D. thesis, University of Edinburgh.

Hofmann, M. (1995b) On the interpretation of type theory in locally cartesian closed categories. In: Computer science logic (Kazimierz, 1994). *Springer-Verlag Lecture Notes in Computer Science* **933** 427–441.

Hofmann, M. and Streicher, T. (1998) The groupoid interpretation of type theory. In: Twenty-five years of constructive type theory (Venice, 1995). *Oxford Logic Guides* **36** 83–111.

Hyland, J. M. and Pitts, A. (1989) The theory of constructions: categorical semantics and topos-theoretic models. In: Categories in computer science and logic (Boulder, CO, 1987). *Contemporary Mathematics* **92** 137–199.

Jacobs, B. (1993) Comprehension categories and the semantics of type dependency. *Theoretical Computer Science* **107** (2) 169–207.

Jacobs, B. (1999) *Categorical logic and type theory*, Studies in Logic and the Foundations of Mathematics **141**, North-Holland.

Kelly, G. M. (1982) *Basic concepts of enriched category theory*, London Mathematical Society Lecture Note Series **64**, Cambridge University Press.

Kelly, G. M. (1989) Elementary observations on 2-categorical limits. *Bulletin of the Australian Mathematical Society* **39** (2) 301–317.

Kelly, G. M. and Street, R. (1974) Review of the elements of 2-categories. In: Category Seminar (Proc. Sem., Sydney, 1972/1973). *Springer-Verlag Lecture Notes in Mathematics* **420** 75–103.

Lawvere, F. W. (1968) Some algebraic problems in the context of functorial semantics of algebraic theories. In: Reports of the Midwest Category Seminar II. *Springer-Verlag Lecture Notes in Mathematics* **61** 41–61.

Nordström, B., Petersson, K. and Smith, J. M. (1990) *Programming in Martin-Löf's Type Theory*, International Series of Monographs on Computer Science **7**, Oxford University Press.

Pitts, A. M. (2000) Categorical logic. In: Abramsky, S., Gabbay, D. M. and Maibaum, T. S. E. (eds.) *Handbook of Logic in Computer Science, Volume 5. Algebraic and Logical Structures*, Oxford University Press 39–128.

Seely, R. (1984) Locally cartesian closed categories and type theory. *Mathematical Proceedings of the Cambridge Philosophical Society* **95** (1) 33–48.

Street, R. (1974) Fibrations and Yoneda's lemma in a 2-category. In: Category Seminar (Proc. Sem., Sydney, 1972/1973). *Springer-Verlag Lecture Notes in Mathematics* **420** 104–133.

Taylor, P. (1999) *Practical foundations of mathematics*, *Cambridge Studies in Advanced Mathematics* **59**, Cambridge University Press.