THE IDENTITY TYPE WEAK FACTORISATION SYSTEM

NICOLA GAMBINO AND RICHARD GARNER

Abstract. We show that the classifying category $\mathcal{C}(\mathbb{T})$ of a dependent type theory $\mathbb{T}$ with axioms for identity types admits a non-trivial weak factorisation system. After characterising this weak factorisation system explicitly, we relate it to the homotopy theory of groupoids.

1. Introduction

From the point of view of mathematical logic and theoretical computer science, Martin-Löf's axioms for identity types [24] admit a conceptually clear explanation in terms of the propositions-as-types paradigm [14, 21, 27]. The fundamental idea behind this explanation is that, for any two elements a, b of a type A, we have a new type $\mathrm{Id}_A(a, b)$, whose elements are to be thought of as proofs that a and b are equal. Yet, identity types determine a highly complex structure on each type, which is far from being fully understood. A glimpse of this structure reveals itself as soon as we start applying the construction of identity types iteratively: not only do we have proofs of equality between two elements of a type, but also of proofs of equality between such proofs, and so on. The difficulty of isolating the structure determined by identity types is closely related to the problem of describing a satisfactory category-theoretic semantics for them. For example, the semantics arising from locally cartesian closed categories [9, 29] validates not only the axioms for identity types, but also additional axioms, known as the reflection rules, which make identity types essentially trivial. One approach to obtain a semantics of identity types that does not validate the reflection rules is to consider categories equipped with a weak factorisation system [2].

Our aim here is to advance our understanding of the categorical structure implicit in the axioms for identity types. We do so by providing further evidence of a close connection with the notion of a weak factorisation system. Our main result states that if $\mathbb{T}$ is a dependent type theory

Date: April 14, 2008.

with the axioms for identity types, then its classifying category $\mathcal{C}(\mathbb{T})$ admits a non-trivial weak factorisation system, which we shall refer to as the *identity type weak factorisation system*. This result should be regarded as analogous to the fundamental result exhibiting the structure of a cartesian closed category on the classifying category of the simply-typed lambda calculus [19, 28]. As such, it provides also a contribution to the development of a functorial semantics [20] for dependent type theories with identity types [2, 5, 7, 10, 15].

A remarkable feature of the identity type weak factorisation system is that its definition does not involve the notion of identity types, but only a canonical class of maps in $\mathcal{C}(\mathbb{T})$, to which we shall refer as *dependent projections*. Indeed, the axioms for identity types are used only to verify that the appropriate axioms hold. After having established the existence of the identity type weak factorisation system, we will provide an explicit characterisation of its classes of maps. This will lead us to two applications. The first establishes an unusual stability property of the identity type weak factorisation system; the second provides further insight into the relationship between dependent type theories with identity types and the category of groupoids, which we denote **Gpd**.

The idea of relating identity types and groupoids dates back to the discovery of the groupoid model of type theory [12]. Indeed, Hofmann and Streicher noticed that the axioms for identity types allow us to equip each type with the structure of a groupoid. We develop this idea in three directions. First, we generalise it by exhibiting a groupoid structrure on each context, which we think of as a family of types, rather than on a single type. Secondly, we extend this construction to a functor $\mathcal{F} \colon \mathcal{C}(\mathbb{T}) \to \mathbf{Gpd}$. Finally, we use the functor $\mathcal{F}$ to relate the identity type weak factorisation system to the natural Quillen model structure on the category of groupoids [1, 17], by showing how the identity type weak factorisation system is mapped into the weak factorisation system determined by injective equivalences and Grothendieck fibrations in **Gpd**.

2. Identity types

2.1. **The syntax of dependent type theories.** The dependent type theories that we consider allow us to make judgements of four forms:

$$A \in \mathrm{Type}\,, \quad a \in A\,, \quad A = B \in \mathrm{Type}\,, \quad a = b \in A\,. \tag{1}$$

They assert, respectively, that A is a type, that a is an element of A, that A and B are definitionally equal types, and that a and b are definitionally equal elements of A. We speak of *definitional equality*, rather than just equality, since the axioms for identity types will provide us with a second notion of equality, which we will refer to as *propositional equality*. The distinction between definitional and propositional equality plays a fundamental role for our purposes. As usual, the judgements in (1) may also be made relative to *contexts*, which consist of lists of variable declarations of the form

$$\Phi = \big(x_0 \in A_0, x_1 \in A_1(x_0), \ldots, x_n \in A_n(x_0, \ldots, x_{n-1})\big)\,. \tag{2}$$

For the context in (2) to be well-formed, it is necessary that the variables $x_0, \ldots, x_n$ are distinct, and that the following sequence of judgements is derivable

$$\begin{gathered} A_0 \in \mathrm{Type}\,, \\ (x_0 \in A_0)\ A_1(x_0) \in \mathrm{Type}\,, \\ \ldots \\ (x_0 \in A_0\,, \ldots\,, x_{n-1} \in A_{n-1}(x_0, \ldots, x_{n-2}))\ A_n(x_0, \ldots, x_{n-1}) \in \mathrm{Type}\,. \end{gathered}$$

From now on, whenever we mention contexts, we implicitly assume that they are well-formed. We write $(\Phi)\ J$ to express that a judgement J holds under the assumptions in Φ. The axioms for a dependent type theory will be stated here as deduction rules of the form

$$\frac{(\Phi_1)\ J_1 \quad \cdots \quad (\Phi_n)\ J_n}{(\Phi)\ J}\,.$$

The dependent type theories that we discuss here are always be assumed to include the basic axioms stated in Appendix A. For more information on dependent type theories, see [21, 23].

2.2. **The classifying category.** We briefly recall the definition of the classifying category $\mathcal{C}(\mathbb{T})$ of a dependent type theory $\mathbb{T}$. For a context Φ as in (2) and a context $\Psi = (y_0 \in B_0, \ldots, y_m \in B_m(y_0, \ldots, y_{m-1}))$, a *context morphism* $f\colon \Phi \to \Psi$ consists of a sequence $f = (b_0, \ldots, b_m)$ such that the following judgements are derivable

$$\begin{gathered} (\Phi)\ b_0 \in B_0\,, \\ (\Phi)\ b_1 \in B_1(b_0)\,, \\ \ldots \\ (\Phi)\ b_m \in B_m(b_0, \ldots, b_{m-1})\,. \end{gathered}$$

We can define an equivalence relation on contexts by considering two contexts to be equivalent if they coincide up to renaming of their free variables and up to componentwise definitional equality. Similarly, we can define an equivalence relation on context morphisms by considering two context morphisms from Φ to Ψ to be equivalent if they concide up to renaming of the free variables in Φ and up to pointwise definitional equality. The classifying category $\mathcal{C}(\mathbb{T})$ can then be defined as having equivalence classes of contexts as objects, and equivalence classes of context morphisms as maps. Composition is defined via substitution; and identity maps are defined in the evident way [25]. When working with $\mathcal{C}(\mathbb{T})$, we implicitly identify contexts and context morphisms up to the equivalence relations defined above, without introducing additional notation. The empty context, written ( ) here, is a terminal object in $\mathcal{C}(\mathbb{T})$.

2.3. **Dependent contexts and dependent elements.** To work with the category $\mathcal{C}(\mathbb{T})$ and its slices, it will be convenient to have some abbreviations for manipulating contexts, as developed in [6]. Let us consider a fixed context Γ. For a sequence of variable declarations as in (2), we write $(\Gamma)\ \Phi \in \mathrm{Cxt}$ to abbreviate the following sequence of judgements

$$\begin{gathered}(\Gamma)\ A_0 \in \mathrm{Type}\,,\\ (\Gamma, x_0 \in A_0)\ A_1(x_0) \in \mathrm{Type}\,,\\ \dots\\ (\Gamma, x_0 \in A_0, \dots, x_{n-1} \in A_{n-1}(x_0, \dots, x_{n-1}))\ A_n(x_0, \dots, x_{n-1}) \in \mathrm{Type}\,.\end{gathered}$$

When these are derivable, we say that Φ is a *dependent context* relative to Γ. Dependent contexts relative to the empty context are simply contexts. If we have a dependent context Φ relative to Γ, we obtain a new context (Γ, Φ) by concatenation, and an evident map $(\Gamma, \Phi) \to \Gamma$, projecting away the variables that are in Φ. Maps of this form will be referred to as *dependent projections*. The class of dependent projections may be understood as the closure under composition of the class of *display maps* [15, 32]. As we will see in Section 4, dependent projections play a crucial role in the definition of the identity type weak factorisation system. For a context Φ relative to Γ, as above, and a sequence $a = (a_0, a_1, \dots, a_n)$, we write $(\Gamma)\ a \in \Phi$ to abbreviate the following sequence of judgements

$$\begin{gathered}(\Gamma)\ a_0 \in A_0\,,\\ (\Gamma)\ a_1 \in A_1(a_0)\,,\\ \dots\\ (\Gamma)\ a_n \in A_n(a_0, \dots, a_{n-1})\,.\end{gathered}$$

When these can be derived, we say that a is a *dependent element* of Φ relative to Γ, and obtain a map $a\colon \Gamma \to (\Gamma, \Phi)$ over Γ. The expressions $(\Gamma)\ \Phi \in \mathrm{Cxt}$ and $(\Gamma)\ a \in \Phi$ should be understood as counterparts of the first and second judgement in (1). It is also possible to introduce expressions $(\Gamma)\ \Phi = \Psi \in \mathrm{Cxt}$ and $(\Gamma)\ a = b \in \Phi$ that correspond to the third and fourth judgement in (1), respectively, so that the evident counterparts of the rules in Appendix A hold. The details are essentially straightforward, and hence omitted.

Remark 2.3.1. Let Φ be a context, or a dependent context, as in (2). We also write $(x \in \Phi)$ to denote Φ itself. We then write $(x \in \Phi\,, y \in \Phi)$ for the result of duplicating Φ and renaming the variables in the second copy to avoid clashes, so as to obtain the context

$$\big(x_0 \in A_0, x_1 \in A_1(x_0), \ldots, x_n \in A_n(x_0, \ldots, x_{n-1}),\\ y_0 \in A_0, y_1 \in A_1(y_0), \ldots, y_n \in A_n(y_0, \ldots, y_{n-1})\big)\,.$$

2.4. **The axioms for identity types.** The axioms for identity types are stated in Table 1. The axiom in (3) is referred to as the *formation rule* for identity types. It asserts that if A is a type and a, b are elements of A, then $\mathrm{Id}_A(a,b)$ is a type. We omit the subscript in expressions of the form $\mathrm{Id}_A(a,b)$ when no confusion arises. If there exists an element $p \in \mathrm{Id}(a,b)$, then we say that a and b are *propositionally equal*. The axiom in (4) is referred to as the *introduction rule* for identity types. Elements of the form $\mathrm{r}(a) \in \mathrm{Id}(a,a)$ are referred to as *reflexivity elements*. The axioms in (5) and (6) are referred to as the *elimination rule* and the *computation rule*, respectively. For brevity, we have omitted from their premisses the judgement

$$\big(x \in A,\; y \in A,\; u \in \mathrm{Id}(x,y),\; \Theta(x,y,u)\big)\; C(x,y,u) \in \mathrm{Type}\,.$$

Here we are assuming that $\Theta(x,y,u)$ is a dependent context relative to the context $(x \in A, y \in A, u \in \mathrm{Id}(x,y))$. We have highlighted the variables x, y, u in $\Theta(x,y,u)$ in order to describe their role in the deduction rules without using substitution. All the axioms in Table 1 should be understood as being relative to a context Γ that is common to both the premisses and the conclusion of the rules, which we leave implicit for brevity. Hence, all the constructions that we perform in $\mathcal{C}(\mathbb{T})$ may equally well take place in one of its slices. The presence of the implicit context Γ plays a role only when stating further axioms, which we assume but do not spell out, expressing commutation laws between the syntax of identity types and the substitution operation. For example, the first of these axioms allows us to derive a rule of the form

$$\frac{(x \in A_0)\ A_1(x) \in \mathrm{Type} \quad (x \in A_0)\ a_1(x) \in A_1(x) \quad (x \in A)\ b_1(x) \in A_1(x) \quad a_0 \in A_0}{\mathrm{Id}_{A_1(x)}(a_1(x), b_1(x))[a_0/x] = \mathrm{Id}_{A_1(a_0)}(a_1(a_0), b_1(a_0)) \in \mathrm{Type}}\,.$$

Since these axioms are essentially straightforward, we prefer to omit them. However, let us point out that the assumption of these axioms ensures that all the constructions that we perform, when regarded as taking place in one of the slices of $\mathcal{C}(\mathbb{T})$, are stable under pullbacks.

$$\frac{A \in \mathrm{Type} \qquad a \in A \qquad b \in A}{\mathrm{Id}_A(a,b) \in \mathrm{Type}} \tag{3}$$

$$\frac{a \in A}{\mathrm{r}(a) \in \mathrm{Id}_A(a,a)} \tag{4}$$

$$\frac{p \in \mathrm{Id}_A(a,b) \qquad \big(x \in A, \Theta(x,x,\mathrm{r}(x))\big)\ d(x) \in C(x,x,\mathrm{r}(x))}{(\Theta(a,b,p))\ \mathrm{J}(a,b,p,d) \in C(a,b,p)} \tag{5}$$

$$\frac{a \in A \qquad (x \in A, \Theta(x,x,\mathrm{r}(x)))\ d(x) \in C(x,x,\mathrm{r}(x))}{(\Theta(a,a,\mathrm{r}(a))\ \mathrm{J}(a,a,\mathrm{r}(a),d) = d(a) \in C(a,a,\mathrm{r}(a))} \tag{6}$$

TABLE 1. Deduction rules for identity types.

Remark 2.4.1. The elimination and computation rules, as stated in Table 1, generalise the standard elimination and computation rules for identity types [24]. The latter can be obtained from the former by restricting the context $\Theta(x, y, u)$ to be empty. The reason for adopting the generalised rules instead of the standard ones is related to our preference for working without assuming the axioms for Π-types. Without Π-types, the standard rules are quite weak, since they do not seem to imply the Leibniz rule for propositional equality [24], whereas our generalised rules suffice, as shown in Lemma 2.5.1. Furthermore, the generalised rules become derivable from the standard ones in the presence of Π-types, so that our development applies also to dependent type theories with standard axioms for identity types and Π-types.

Remark 2.4.2. We do not assume the rules in (7), to which we refer as the *reflection rules*.

$$\frac{p \in \mathrm{Id}(a,b)}{a = b \in A} \qquad \frac{p \in \mathrm{Id}(a,b)}{\mathrm{r}(a) = p \in \mathrm{Id}(a,b)} \tag{7}$$

The first reflection rule was shown to be independent from the axioms for identity types in [12]. Note that the judgement $\mathrm{r}(a) \in \mathrm{Id}(a, b)$, that is presupposed by the conclusion of the second reflection rule, is derivable by the first reflection rule and the standard rules concerning substitution, as given in Appendix A. The reflection rules are generally avoided, since they imply that propositional equality and definitional equality collapse into equivalent notions, which has the effect of destroying the decidability of type-checking, one of the fundamental properties of dependent type theories [11]. As shown in [15, Proposition 10.1.3] extending [31, Theorem 1.1], the reflection rules are equivalent to the following rule

$$\frac{p \in \mathrm{Id}(a,b) \quad (x \in A, y \in A, u \in \mathrm{Id}(x,y), \Theta(x,y,u))\ e(x,y,u) \in C(x,y,u)}{(\Theta(a,b,p))\ J(a,b,p,[x]e(x,x,\mathrm{r}(x))) = e(a,b,p) \in C(a,b,p)}$$

Here and in the following, expressions of the form $[x]f$ denote λ-abstractions in the metatheory. If the computation rule for identity types in (5) is understood as a version of the β-rule, then this rule can be understood as a version of the η-rule.

2.5. **Identity contexts.** One fundamental fact for our development is that, as shown in [8], the axioms for identity types can be used to construct what we will refer to as *identity contexts*. More precisely, there are explicit definitions such that all the rules in Table 2 are derivable. Because of their similarity with the axioms for identity types, we refer to (8) as the *formation rule*, to (9) as the *introduction rule*, to (10) as the *elimination rule*, and to (11) as the *computation rule* for identity contexts. When stating these rules, we leave again implicit a context Γ, to which all the notions are assumed to be relative. For example, in the introduction rule Φ may be assumed to be a context relative to Γ. When stating the elimination and computation rules, we are assuming that Φ has the form in (2) and using the notational conventions set in Remark 2.3.1. For a context Φ and $a, b \in \Phi$, we refer to a context of the form $\mathrm{Id}_\Phi(a,b)$ as an *identity context*. As before, we have omitted the judgement

$$\big(x \in \Phi, y \in \Phi, u \in \mathrm{Id}_\Phi(x,y), \Theta(x,y,u)\big)\ \Omega(x,y,u) \in \mathrm{Cxt}$$

from the premisses of the elimination and computation rules. From now on, we omit the subscript from expressions of the form $\mathrm{Id}_\Phi(a,b)$ if no confusion arises. It will be convenient to fix some terminology. When we use the elimination rule as in Table 2, we will say that we are applying the elimination rule on $p \in \mathrm{Id}(a,b)$. We then refer to the relative context $\Omega(x,y,u)$ as the *eliminating context*, and to d as the *eliminating family*.

Lemma 2.5.1 states a very useful property, to which we refer as the *Leibniz rule for contexts*, which is going to be used repeatedly in what follows. To prove it, we make use of the general formulation of the elimination and computation rules that we adopt here.

(8) $$\frac{a \in \Phi \quad b \in \Phi}{\mathrm{Id}_\Phi(a,b) \in \mathrm{Cxt}}$$

(9) $$\frac{a \in \Phi}{\mathrm{r}(a) \in \mathrm{Id}_\Phi(a,a)}$$

(10) $$\frac{p \in \mathrm{Id}_\Phi(a,b) \quad (x \in \Phi, \Theta(x,x,\mathrm{r}(x))\ d(x) \in \Omega(x,x,\mathrm{r}(x))}{(\Theta(a,b,p))\ \mathrm{J}(a,b,p,d) \in \Omega(a,b,p)}$$

(11) $$\frac{a \in \Phi \quad (x \in \Phi, \Theta(x,x,\mathrm{r}(x)))\ d(x) \in \Omega(x,x,\mathrm{r}(x))}{(\Theta(a,a,\mathrm{r}(a)))\ \mathrm{J}(a,a,\mathrm{r}(a),d) = d(a) \in \Omega(a,a,\mathrm{r}(a))}$$

TABLE 2. Deduction rules for identity contexts.

Lemma 2.5.1. *We can derive a rule of the form*

$$\frac{p \in \mathrm{Id}(a,b) \qquad (x \in \Phi)\ \Omega(x) \in \mathrm{Cxt} \qquad e \in \Omega(a)}{p_!(e) \in \Omega(b)}$$

such that

$$\frac{a \in \Phi \quad e \in \Omega(a)}{(\mathrm{r}(a))_!(e) = e \in \Omega(a)}$$

holds.

Proof. We use elimination over $p \in \mathrm{Id}(a,b)$ with

$$\big(x \in \Phi, y \in \Phi, u \in \mathrm{Id}(x,y), z \in \Omega(x)\big)\ \Omega(y) \in \mathrm{Cxt}$$

as the eliminating context. Since we have

$$(x \in \Phi, z \in \Omega(x))\ z \in \Omega(x)$$

the elimination rule allows us to derive

$$(z \in \Omega(a))\ \mathrm{J}(a,b,p,[x]z) \in \Omega(b)\,.$$

The required term $p_!(e)$ is defined as the result of substituting $e \in \Omega(a)$ for $z \in \Omega(a)$ in the expression $\mathrm{J}(a,b,p,[x]z)$, so that

$$p_!(e) = \mathrm{J}(a,b,p,[x]e) \in \Omega(b)$$

The second rule is an immediate consequence of this definition and the computation rule. □

The Leibniz rule allows us to give a description of the identity contexts by induction on their length. For a context of length $n = 1$, the definition is straightforward: if $\Phi = (x \in A)$, then elements of Φ are the same as elements of the type A, and the identity context corresponding to $a, b \in \Phi$ is given by

$$\mathrm{Id}_\Phi(a, b) = (u \in \mathrm{Id}_A(a, b)) .$$

Assuming we know how to define the identity contexts associated to contexts of length n, we can describe the identity contexts associated to a context of length $n + 1$ as follows. Let us assume to have a context Φ of length n as in (2), and consider a context Φ' of length $n + 1$ of the form

$$\Phi' = (x \in \Phi, x_{n+1} \in A_{n+1}(x)) . \tag{12}$$

By definition, elements $a', b' \in \Phi'$ have the form $a' = (a, a_{n+1})$ and $b' = (b, b_{n+1})$, where $a, b \in \Phi$, $a_{n+1} \in A_{n+1}(a)$, and $b_{n+1} \in A_{n+1}(b)$. Their associated identity context has the form

$$\mathrm{Id}(a', b') = \big(u \in \mathrm{Id}_\Phi(a, b), u_{n+1} \in \mathrm{Id}_{A_{n+1}(b)}(u_!(a_{n+1}), b_{n+1})\big)$$

where we have $u_!(a_{n+1}) \in A_{n+1}(b)$ by the following application of the Leibniz rule

$$\frac{u \in \mathrm{Id}_\Phi(a, b) \qquad (x \in \Phi)\ A_{n+1}(x) \in \mathrm{Type} \qquad a_{n+1} \in A_{n+1}(a)}{u_!(a_{n+1}) \in A_{n+1}(b)} .$$

In the following, we will need only this description of the identity contexts and the rules in Table 2. Hence, we do not describe the other syntactic constructs involved in them, and refer the reader to [8] for details.

3. The fundamental groupoid of a context

3.1. **Contexts as spaces.** The axioms for identity types allow us to think of a context Φ as a space, as we explain below. To emphasize this perspective, we refer to elements of Φ as *points*. Given two points $a, b \in \Phi$, we refer to elements of $\mathrm{Id}(a, b)$ as *paths* from a to b. Following this idea, the context $\mathrm{Id}(a, b)$ can be thought of as the space of all paths from a to b. Now that we have points and paths, we may want to speak of homotopies between paths. For this, it suffices to apply the formation rule as follows:

$$\frac{p_0 \in \mathrm{Id}(a, b) \quad p_1 \in \mathrm{Id}(a, b)}{\mathrm{Id}(p_0, p_1) \in \mathrm{Cxt}} .$$

Elements $\theta \in \mathrm{Id}(p_0, p_1)$ will be referred to as *homotopies* from p_0 to p_1, and two paths $p_0 \in \mathrm{Id}(a, b)$ and $p_1 \in \mathrm{Id}(a, b)$ will be said to be *homotopic* if there exists an homotopy between them. Our aim is to pursue this analogy and to define a groupoid having the points of Φ as objects and homotopy equivalence classes of paths as maps. Because of the analogy with the definition of the fundamental groupoid of a space [22], we refer to this groupoid as the *fundamental groupoid* of a context. In order to carry this over, we need to show that paths can be composed, that there are constant paths, and that the groupoid axioms hold. All of this follows from the axioms for identity types via the rules for identity contexts of Table 2.

3.2. **The fundamental groupoid construction.** Lemma 3.2.1 can the understood as expressing that paths can be composed, that there is a trival path on each point, and that paths can be reversed. From now on, we assume to work with a fixed context Φ as in (2). We use freely the notation introduced in Remark 2.3.1.

Lemma 3.2.1. *We can derive rules of the form*

$$\frac{p \in \mathrm{Id}(a,b) \quad q \in \mathrm{Id}(b,c)}{q \circ p \in \mathrm{Id}(a,c)} \qquad \frac{a \in \Phi}{1_a \in \mathrm{Id}(a,a)} \qquad \frac{p \in \mathrm{Id}(a,b)}{p^{-1} \in \mathrm{Id}(b,a)}$$

such that

$$\frac{p \in \mathrm{Id}(a,b)}{1_b \circ p = p \in \mathrm{Id}(a,b)} \qquad \frac{a \in \Phi}{(1_a)^{-1} = 1_a \in \mathrm{Id}(a,a)}$$

hold.

Proof. For the first rule, we apply the Leibniz rule as follows

$$\frac{q \in \mathrm{Id}(b,c) \quad (x \in \Phi)\ \mathrm{Id}(a,x) \in \mathrm{Cxt} \quad p \in \mathrm{Id}(a,b)}{q_!(p) \in \mathrm{Id}(a,c)}$$

and define $q \circ p =_{\mathrm{def}} q_!(p) \in \mathrm{Id}(a,c)$. For the second rule, we use the introduction rule, and simply define $1_a =_{\mathrm{def}} \mathrm{r}(a) \in \mathrm{Id}(a,a)$. For the third rule, we apply the elimination rule on $p \in \mathrm{Id}(a,b)$, with

$$(x \in \Phi, y \in \Phi, u \in \mathrm{Id}(x,y))\ \mathrm{Id}(y,x) \in \mathrm{Cxt}$$

as the eliminating context, and $\mathrm{r}(x) \in \mathrm{Id}(x,x)$ as the eliminating family. Hence, we can define

$$p^{-1} \underset{\mathrm{def}}{=} \mathrm{J}(a,b,p,[x]\,\mathrm{r}(x)) \in \mathrm{Id}(b,a).$$

The computation rule implies the other rules. □

Lemma 3.2.2. *Let* $p_0, p_1 \in \mathrm{Id}(a,b)$ *and* $q_0, q_1 \in \mathrm{Id}(b,c)$. *We can derive rules of the form*

$$\frac{\phi \in \mathrm{Id}(p_0,p_1) \quad \psi \in \mathrm{Id}(q_0,q_1)}{\psi \circ \phi \in \mathrm{Id}(q_0 \circ p_0, q_1 \circ p_1)} \qquad \frac{\phi \in \mathrm{Id}(p_0,p_1)}{\phi^* \in \mathrm{Id}({p_0}^{-1}, {p_1}^{-1})}$$

such that

$$\frac{\phi \in \mathrm{Id}(p_0,p_1) \quad q \in \mathrm{Id}(b,c)}{1_q \circ \phi = \phi \in \mathrm{Id}(q \circ p_0, q \circ p_1)} \qquad \frac{p \in \mathrm{Id}(a,b)}{(1_p)^* = 1_p \in \mathrm{Id}(p,p)}$$

hold.

Proof. For the first rule, use elimination over $\psi \in \mathrm{Id}(q_0,q_1)$ and Lemma 3.2.1. For the second rule, use elimination over $\phi \in \mathrm{Id}(p_0,p_1)$ and Lemma 3.2.1. □

Lemma 3.2.3. *We can derive rules of the form*

$$\frac{p \in \mathrm{Id}(a,b) \qquad q \in \mathrm{Id}(b,c) \qquad r \in \mathrm{Id}(c,d)}{\alpha_{p,q,r} \in \mathrm{Id}((r \circ q) \circ p, r \circ (q \circ p))}$$

$$\frac{p \in \mathrm{Id}(a,b)}{\phi_p \in \mathrm{Id}(1_b \circ p, p)} \qquad \frac{p \in \mathrm{Id}(a,b)}{\psi_p \in \mathrm{Id}(p \circ 1_a, p)}$$

$$\frac{p \in \mathrm{Id}(a,b)}{\sigma_p \in \mathrm{Id}(p^{-1} \circ p, 1_a)} \qquad \frac{p \in \mathrm{Id}(a,b)}{\tau_p \in \mathrm{Id}(p \circ p^{-1}, 1_b)}$$

such that

$$\frac{p \in \mathrm{Id}(a,b) \qquad q \in \mathrm{Id}(b,c)}{\alpha_{p,q,1_c} = 1_{q \circ p} \in \mathrm{Id}(q \circ p, q \circ p)}$$

$$\frac{a \in \Phi}{\phi_{1_a} = 1_{1_a} \in \mathrm{Id}(1_a, 1_a)} \qquad \frac{a \in \Phi}{\psi_{1_a} = 1_{1_a} \in \mathrm{Id}(1_a, 1_a)}$$

$$\frac{a \in A}{\sigma_{1_a} = 1_{1_a} \in \mathrm{Id}(1_a, 1_a)} \qquad \frac{a \in A}{\tau_{1_a} = 1_{1_a} \in \mathrm{Id}(1_a, 1_a)}$$

hold.

Proof. For $\alpha_{p,q,r}$ use elimination over $r \in \mathrm{Id}(c,d)$. For ϕ_p we can define ϕ_p to be 1_p by Lemma 3.2.1. For ψ_p use elimination over $p \in \mathrm{Id}(a,b)$. For σ_p and τ_p use elimination over $p \in \mathrm{Id}(a,b)$. □

Let $a, b \in \Phi$. An application of Lemma 3.2.1, taking Φ therein to be $\mathrm{Id}_\Phi(a, b)$, shows that homotopy of paths is a reflexive, symmetric, and transitive relation on the set of paths from a to b. We write $[a, b]$ for the quotient set, and $[p]\colon a \to b$ for the equivalence class of a path $p \in \mathrm{Id}(a, b)$. We can now define the fundamental groupoid $\mathcal{F}(\Phi)$ associated to Φ. The objects of $\mathcal{F}(\Phi)$ are the elements of Φ. The maps from a to b in $\mathcal{F}(\Phi)$ are equivalence classes of paths $[p]\colon a \to b$. Composition, identities, and inverses in $\mathcal{F}(\Phi)$ are defined by letting

$$[q] \circ [p] =_{\mathrm{def}} [q \circ p]\,, \quad 1_a =_{\mathrm{def}} [1_a]\,, \quad [p]^{-1} =_{\mathrm{def}} [p^{-1}]\,.$$

These operations are well-defined by Lemma 3.2.2. To establish the axioms for a category, we need to show

$$[r \circ (q \circ p)] = [(r \circ q) \circ p]\,, \quad [1_b \circ p] = [p]\,, \quad [p \circ 1_a] = [p] \tag{13}$$

and to establish the additional axioms for a groupoid, we need to show

$$[p^{-1} \circ p] = [1_a]\,, \quad [p \circ p^{-1}] = [1_b]\,. \tag{14}$$

For (13), we need homotopies $\alpha \in \mathrm{Id}(r \circ (q \circ p), (r \circ q) \circ p)$, $\phi \in \mathrm{Id}(1_b \circ p, p)$, and $\psi \in \mathrm{Id}(p \circ 1_a, p)$. For (14), we need homotopies $\sigma \in \mathrm{Id}(p^{-1} \circ p, 1_a)$ and $\tau \in \mathrm{Id}(p \circ p^{-1}, 1_b)$. All of these are provided by Lemma 3.2.3.

3.3. **Functorial aspects.** We write **Gpd** for the category having small groupoids as objects and functors as maps.

Proposition 3.3.1. *Let $\mathbb{T}$ be a dependent type theory with axioms for identity types. The function mapping a context Φ to its fundamental groupoid $\mathcal{F}(\Phi)$ extends to a functor $\mathcal{F}\colon \mathcal{C}(\mathbb{T}) \to$ **Gpd**.*

Proof. We need to define a functor $\mathcal{F}(f)\colon \mathcal{F}(\Phi) \to \mathcal{F}(\Psi)$ for every context morphism $f\colon \Phi \to \Psi$. On objects, $\mathcal{F}(f)$ sends $a \in \mathcal{F}(\Phi)$ to $fa \in \mathcal{F}(\Psi)$. On maps, $\mathcal{F}(f)$ sends $[p]\colon a \to b$ to $[f(p)]\colon f(a) \to f(b)$, where $f(p) \in \mathrm{Id}(f(a), f(b))$ is defined using the elimination rule by letting

$$f(p) = \mathrm{J}(a, b, p, [x]1_{fx}) \in \mathrm{Id}(f(a), f(b))\,, \tag{15}$$

so that

$$f(1_a) = 1_{fa} \in \mathrm{Id}(f(a), f(a))\,. \tag{16}$$

It is routine to check that the action of $\mathcal{F}(f)$ on maps is well-defined. To show that $\mathcal{F}(f)$ is a functor amounts to verifying the equations

$$[f(q \circ p)] = [f(q) \circ f(p)]\,, \qquad [f(1_a)] = [1_{f(a)}]\,.$$

For the first equation, elimination on $q \in \mathrm{Id}(b, c)$ can be used to exhibit the required homotopy between $f(q \circ p)$ and $f(q) \circ f(p)$. For the second equation, use (16). We have therefore defined $\mathcal{F}\colon \mathcal{C}(\mathbb{T}) \to$ **Gpd** on objects and maps. Thus, it remains to check that it is a functor. We begin by checking

$$\mathcal{F}(g \circ f) = \mathcal{F}(g) \circ \mathcal{F}(f)\,.$$

It is clear that $\mathcal{F}(g \circ f)$ and $\mathcal{F}(g) \circ \mathcal{F}(f)$ have the same action on objects. To show that they coincide on maps, we need to show that we have a homotopy between $g(f(p))$ and $(gf)(p)$ for every $p \in \mathrm{Id}(a,b)$. By (15), we have

$$g(f(p)) = \mathrm{J}(f(a), f(b), f(p), [x]\, 1_{g(x)}) \in \mathrm{Id}(gf(a), gf(b))$$

and

$$(gf)(p) = \mathrm{J}(a, b, p, [x]\, 1_{gf(x)}) \in \mathrm{Id}(gf(a), gf(b))\,.$$

The required homotopy can be obtained by the elimination rule on $p \in \mathrm{Id}(a,b)$. To conclude the proof, it suffices to check that

$$\mathcal{F}(1_\Phi) = 1_{\mathcal{F}(\Phi)}\,.$$

As before, it is clear that $\mathcal{F}(1_\Phi)$ is the identity on objects. To check that it is the identity on maps, it suffices to show that $[p] \colon a \to b$ and $[1_\Phi(p)] \colon a \to b$ are the same equivalence classes, which can be proved by the elimination rule on $p \in \mathrm{Id}(a,b)$. □

4. The identity type weak factorisation system

4.1. **Weak factorisation systems.** Let us recall the notion of a weak factorisation system [4]. For this, we need some terminology and notation. Let $\mathcal{E}$ be a category. Given maps $f \colon A \to B$ and $g \colon C \to D$ in $\mathcal{E}$, we say that f has the *left lifting property* with respect to g, or that g has the *right lifting property* with respect to f, if every commutative diagram of the form

$$\begin{array}{ccc} A & \xrightarrow{h} & C \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle g} \\ B & \xrightarrow[k]{} & D \end{array}$$

has a *diagonal filler*, that is to say is a map $j \colon B \to C$ making the diagram

$$\begin{array}{ccc} A & \xrightarrow{h} & C \\ {\scriptstyle f}\downarrow & \overset{j}{\nearrow} & \downarrow{\scriptstyle g} \\ B & \xrightarrow[k]{} & D \end{array}$$

commute. We write $f \pitchfork g$ to denote this situation. For a class of maps $\mathcal{M}$, we define $\mathcal{M}^{\pitchfork}$ to be the class of maps having the right lifting property with respect to every map in $\mathcal{M}$. Similarly, we define ${}^{\pitchfork}\mathcal{M}$ to be the class of maps having the left lifting property with respect to every map in $\mathcal{M}$. A *weak* *factorisation system* on $\mathcal{E}$ consists of a pair of classes of maps $(\mathcal{A}, \mathcal{B})$ such that the following hold.

(1) Every map f admits a factorisation $f = pi$ with $i \in \mathcal{A}$ and $p \in \mathcal{B}$.

(2) $\mathcal{A}^{\pitchfork} = \mathcal{B}$ and $\mathcal{A} = {}^{\pitchfork}\mathcal{B}$.

We refer to (1) as the Factorisation Axiom and to (2) as the Weak Orthogonality Axiom. For more information on weak factorisation systems, see [16, Appendix D].

$$\frac{\Phi \in \mathrm{Cxt}}{(x \in \Phi, y \in \Phi)\ \mathrm{Id}_\Phi(x,y) \in \mathrm{Cxt}}$$

$$\frac{\Phi \in \mathrm{Cxt}}{(x \in \Phi)\ \ \mathrm{r}(x) \in \mathrm{Id}_\Phi(x,x)}$$

$$\frac{(x \in \Phi, \Theta(x,x,\mathrm{r}(x)))\ d(x) \in \Omega(x,x,\mathrm{r}\,x)}{(x \in \Phi, y \in \Phi, u \in \mathrm{Id}_\Phi(x,y), \Theta(x,y,u))\ \mathrm{J}(x,y,u,d) \in \Omega(x,y,u)}$$

$$\frac{(x \in \Phi, \Theta(x,x,\mathrm{r}(x)))\ d(x) \in \Omega(x,x,\mathrm{r}(x))}{\big((x \in \Phi, \Theta(x,x\,\mathrm{r}(x)))\big)\ \mathrm{J}(x,x,\mathrm{r}(x),d) = d(x) \in \Omega(x,x,\mathrm{r}(x))}$$

Table 3. Variable-based rules for identity contexts.

4.2. **The identity type weak factorisation system.** Let $\mathbb{T}$ be a dependent type theory and consider its classifying category $\mathcal{C}(\mathbb{T})$. Recall from Section 2.3 that a *dependent projection* is a context morphism of the form $\Gamma, \Phi \to \Gamma$, obtained by forgetting the variables in Φ. We write $\mathcal{J}$ for the set of dependent projections in $\mathcal{C}(\mathbb{T})$. Our main result is the following.

Theorem 4.2.1. *Let $\mathbb{T}$ be a dependent type theory. If $\mathbb{T}$ includes the axioms for identity types, then the pair $(\mathcal{A}, \mathcal{B})$, where $\mathcal{A} = {}^{\pitchfork}\mathcal{J}$ and $\mathcal{B} =_{\mathrm{def}} \mathcal{A}^{\pitchfork}$, forms a weak factorisation system on $\mathcal{C}(\mathbb{T})$.*

Let us emphasize that identity types are not involved in the definition of the classes of maps $\mathcal{A}$ and $\mathcal{B}$. They are, however, essential for the proof that these classes of maps satisfy the axioms for a weak factorisation system. As usual in the proof of the existence of a weak factorisation system, the difficulties are concentrated in one particular step of the proof. Lemma 4.2.2 is the key step in our case, with the proof of Theorem 4.2.1 following from it by standard arguments in the theory of weak factorisation systems. To prove Lemma 4.2.2, it is convenient to work with the equivalent formulation of the rules for identity contexts given in Table 3. The equivalence between the sets of rules in Table 2 and in Table 3 follows by standard properties of substitution. As before, we use the notational convention stipulated in Remark 2.3.1.

Lemma 4.2.2. *Every map f admits a factorisation $f = pi$, where $i \in \mathcal{A}$ and p is a dependent projection.*

Proof. For $f \colon \Phi \to \Psi$, define $\mathrm{Id}(f) =_{\mathrm{def}} \big(x \in \Phi, y \in \Psi, u \in \mathrm{Id}_\Psi(fx, y)\big)$. The required factorisation is defined as follows:

$$\Phi \xrightarrow{i_f} \mathrm{Id}(f) \xrightarrow{p_f} \Psi\,, \tag{17}$$

where $i_f =_{\mathrm{def}} (x, fx, 1_{fx})$ and $p_f = (y)$. Apart from the ordering of the variable declarations $x \in \Phi$ and $y \in \Psi$, which is clearly unessential, p_f is a dependent projection, as required. Hence, we only need to show that $i_f \in \mathcal{A}$. This amounts to showing that it has the left lifting property with respect to all the dependent projections. This amounts to providing diagonal fillers for every diagram of the form

$$\begin{array}{ccc} \Phi & \longrightarrow & (\Lambda, \Xi) \\ \downarrow & & \downarrow \\ \mathrm{Id}(f) & \longrightarrow & \Lambda \end{array}$$

Since dependent projections are closed under pullback [25, Lemma 6.3.2], it suffices to show that we can define a diagonal filler for every diagram of the form

$$\begin{array}{ccc} \Phi & \longrightarrow & (\mathrm{Id}(f), \Omega) \\ \downarrow & & \downarrow \\ \mathrm{Id}(f) & = & \mathrm{Id}(f) \end{array}$$

The right-hand side dependent projection gives us a dependent context $\Omega(x, y, u)$ relative to $(x \in \Phi, y \in \Psi, u \in \mathrm{Id}(f))$. By the commutativity of the diagram, the top horizontal map gives us a dependent element $d(x) \in \Omega(x, fx, 1_{fx})$ relative to $(x \in \Phi)$. We can derive

$$(\Gamma, y_0 \in \Psi, y_1 \in \Psi, v \in \mathrm{Id}(y_0, y_1), \Theta(y_0, y_1, v))\ \Omega(x, y_1, v \circ u) \in \mathrm{Cxt}\,,$$

where $\Gamma =_{\mathrm{def}} (x \in \Phi)$ and $\Theta(y_0, y_1, v) = (u \in \mathrm{Id}(fx, y_0), z \in \Omega(x, y_0, u))$. By the definitional equality $1_y \circ u = u \in \mathrm{Id}(f(x), y_0)$, proved in Lemma 3.2.1, we have

$$\big(\Gamma, y \in \Psi, \Theta(y, y, 1_y)\big)\ z \in \Omega(x, y, 1_y \circ u)\,,$$

By the elimination rule applied to $v \in \mathrm{Id}(y_0, y_1)$, we obtain

$$(\Gamma, y_0 \in \Psi, y_1 \in \Psi, v \in \mathrm{Id}(y_0, y_1), \Theta(y_0, y_1, v)) \\ m(y_0, y_1, v, u, z) \in \Omega(x, y_1, v \circ u)$$

where $m(y_0, y_1, v, u, z) =_{\mathrm{def}} \mathrm{J}(y_0, y_1, v, [\_]z)$. We then obtain

$$(x \in \Phi, y \in \Psi, u \in \mathrm{Id}(fx, y), z \in \Omega(x, fx, 1_{fx}) \\ n(x, y, u, z) \in \Omega(x, y, u \circ 1_{fx})$$

where $n(x, y, u, z) =_{\mathrm{def}} m(fx, y, u, 1_{fx}, z)$. We can now substitute $d(x)$ for z and obtain

$$(x \in \Phi, y \in \Psi, u \in \mathrm{Id}(fx, y))\ n(x, y, u, d(x)) \in \Omega(x, y, u \circ 1_{fx})$$

Let us now recall that by Lemma 3.2.3 we have $\psi_u \in \mathrm{Id}(u \circ 1_{fx}, u)$. Hence, we can apply Lemma 2.5.1 and obtain

$$(x \in \Phi, y \in \Psi, u \in \mathrm{Id}(fx, y))\ (\psi_u)_!(n(x, y, u, d(x))) \in \Omega(x, y, u)$$

We claim that $j =_{\mathrm{def}} (x, y, u, (\psi_u)_!(n(x, y, u, d(x))))$ provides the required filler, fitting in the diagram

$$\begin{array}{ccc} \Phi & \longrightarrow & (\mathrm{Id}(f), \Omega) \\ \downarrow & \overset{j}{\nearrow} & \downarrow \\ \mathrm{Id}(f) & = & \mathrm{Id}(f) \end{array}$$

The commutativity of the bottom triangle is evident. The commutativity of the top triangle follows by the chain of definitional equalities

$$\begin{aligned} (\psi_{1_{fx}})_!(n(x, fx, 1_{fx}, d(x))) &= n(x, fx, 1_{fx}, d(x)) \\ &= m(fx, fx, 1_{fx}, 1_{fx}, d(x)) \\ &= \mathrm{J}(fx, fx, 1_{fx}, [\_]d(x)) \\ &= d(x). \end{aligned}$$

Here, we used Lemma 3.2.1, the definitions of n and m, and the elimination rule for identity types. The commutativity of the bottom triangle is immediate. □

Proof of Theorem 4.2.1. The Factorisation Axiom follows from Lemma 4.2.2, since $\mathcal{J} \subseteq \mathcal{B}$. The very definition of $\mathcal{B}$ implies that $\mathcal{A} = {}^{\pitchfork}\mathcal{B}$. For the Weak Orthogonality Axiom, we reason as follows. To show that $\mathcal{A}^{\pitchfork} = \mathcal{B}$, observe that $\mathcal{J} \subseteq \mathcal{B}$. Hence, we have ${}^{\pitchfork}\mathcal{B} \subseteq {}^{\pitchfork}\mathcal{J}$, and so ${}^{\pitchfork}\mathcal{B} \subseteq \mathcal{A}$. Thus, we only need to prove that $\mathcal{A} \subseteq {}^{\pitchfork}\mathcal{B}$. For this, observe that every map in $\mathcal{B}$ is a retract of a dependent projection by Lemma 4.2.2. This follows from

Lemma 4.2.2 via the Retract Argument [13, Lemma 1.1.9]. Hence, if a map has the left lifting property with respect to all dependent projections, then it has the left lifting property with respect to all maps in $\mathcal{B}$. The required inclusion $\mathcal{A} \subseteq {}^{\pitchfork}\mathcal{B}$ follows. $\square$

Let us illustrate what happens when we apply the factorisation of Lemma 4.2.2 to the identity map $1_\Phi \colon \Phi \to \Phi$. The factorisation in (17) becomes

$$\Phi \xrightarrow{i} \mathrm{Id}(\Phi) \xrightarrow{p} \Phi\,.$$

where $\mathrm{Id}(\Phi)$ is defined as the context $(y \in \Phi, x \in \Phi, u \in \mathrm{Id}(x,y))$, and the maps i and p are defined by letting $i =_{\mathrm{def}} (x, x, 1_x)$ and $p =_{\mathrm{def}} (y)$, respectively. Let us consider a diagram of the form

$$\begin{array}{ccc} \Phi & \longrightarrow & (\mathrm{Id}(\Phi), \Omega) \\ {\scriptstyle i}\downarrow & & \downarrow \\ \mathrm{Id}(\Phi) & = & \mathrm{Id}(\Phi) \end{array}$$

Here, Ω is a context relative to $\mathrm{Id}(\Phi)$. By the commutativity of the diagram, the top horizontal map gives us a dependent element $d(x) \in \Omega(x, x, \mathrm{r}(x))$ relative to $(x \in \Phi)$. By the elimination rule, we can deduce that $\mathrm{J}(x, y, u, d) \in \Omega(x, y, u)$. We can therefore define a filler j

$$\begin{array}{ccc} \Phi & \longrightarrow & (\mathrm{Id}(\Phi), \Omega) \\ {\scriptstyle i}\downarrow & \nearrow{\scriptstyle j} & \downarrow \\ \mathrm{Id}(\Phi) & = & \mathrm{Id}(\Phi) \end{array}$$

by letting $j =_{\mathrm{def}} (x, y, u, \mathrm{J}(x, y, u, d))$. The commutativity of the top triangle follows from the computation rule, while the commutativity of the bottom triangle is immediate. This is the key idea underpinning the semantics of identity types in weak factorisation systems introduced in [2].

5. Characterisation and applications

5.1. **Characterisation of the weak factorisation system.** We provide an explicit characterisation of the maps in the classes $\mathcal{A}$ and $\mathcal{B}$ of the identity type weak factorisation system established in Theorem 4.2.1. For this, we introduce some terminology, which is inspired by concepts of 2-dimensional category theory [3, 18, 30]. We define a context morphism $f \colon \Phi \to \Psi$ to be a *type-theoretic injective equivalence* if we can derive a jugdement

$$(y \in \Psi)\ s(y) \in \Phi$$

such that we can derive also judgements of the form

$$(x \in \Phi)\ x = s(f(x)) \in \Phi\,, \tag{18}$$

$$(y \in \Psi)\ \varepsilon_y \in \mathrm{Id}_\Psi(f(s(y)), y)\,, \tag{19}$$

$$(x \in \Phi)\ \varepsilon_{f(x)} = 1_{f(x)} \in \mathrm{Id}_\Psi(f(x), f(x))\,. \tag{20}$$

We say that $f \colon \Phi \to \Psi$ is a *type-theoretic normal isofibration* if we can derive a judgement

$$\big(x \in \Phi, y \in \Psi, u \in \mathrm{Id}(f(x), y)\big)\ j(x, y, u) \in \Phi$$

such that we can also derive judgements of the following form

$$\big(x \in \Phi, y \in \Psi, u \in \mathrm{Id}(f(x), y)\big)\ fj(x, y, u) = y \in \Psi\,, \tag{21}$$

$$(x \in \Phi)\ j(x, fx, 1_{fx}) = x \in \Phi\,. \tag{22}$$

Although the identity type weak factorisation system does not seem to be functorial, we can follow the argument used to characterise the maps of a functorial weak factorisation system in [26, §2.4] to establish Lemma 5.1.1.

Lemma 5.1.1. *Let $f \colon \Phi \to \Psi$ be a context morphism.*

- *(i) It holds that $f \in \mathcal{A}$ if and only if f is a type-theoretic injective equivalence.*
- *(ii) It holds that $f \in \mathcal{B}$ if and only if f is a type-theoretic normal isofibration.*

Proof. Recall that by Lemma 4.2.2 every map $f \colon \Phi \to \Psi$ admits a factorisation

$$\Phi \xrightarrow{i_f} \mathrm{Id}(f) \xrightarrow{p_f} \Psi\,,$$

where $i_f \in \mathcal{A}$ and $p_f \in \mathcal{B}$. Let us prove (i). Define $\mathcal{A}'$ to be the class of maps $f \colon \Phi \to \Psi$ such that the commutative diagram

$$\begin{array}{ccc} \Phi & \xrightarrow{i_f} & \mathrm{Id}(f) \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle p_f} \\ \Psi & = & \Psi \end{array} \tag{23}$$

has a diagonal filler. We claim that $\mathcal{A} = \mathcal{A}'$. To show $\mathcal{A} \subseteq \mathcal{A}'$, let $f \colon \Phi \to \Psi$ be in $\mathcal{A}$. The diagram in (23) has a diagonal filler since $f \in \mathcal{A}$ and $p_f \in \mathcal{B}$. To show $\mathcal{A}' \subseteq \mathcal{A}$, let $f \colon \Phi \to \Psi$ be in $\mathcal{A}'$, and assume to have a diagonal filler $j \colon \Psi \to \mathrm{Id}(f)$ for the diagram in (23). We can then exhibit f as a retract of i_f by the diagram

$$\begin{array}{ccccc} \Phi & = & \Phi & = & \Phi \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle i_f} & & \downarrow{\scriptstyle f} \\ \Psi & \xrightarrow[j]{} & \mathrm{Id}(f) & \xrightarrow[p_f]{} & \Psi \end{array}$$

We have that $i_f \in \mathcal{A}$. Since the class $\mathcal{A}$, being defined by a weak orthogonality condition, is closed under retracts, we have $f \in \mathcal{A}$. To conclude the proof, it is sufficient to observe that f is an injective equivalence if and only if $f \in \mathcal{A}'$. This involves unfolding the definition of the context $\mathrm{Id}(f)$.
For the proof of (ii), let $\mathcal{B}'$ be the class of maps $f \colon \Phi \to \Psi$ such that the commutative diagram

$$\begin{array}{ccc} \Phi & = & \Phi \\ {\scriptstyle i_f}\downarrow & & \downarrow{\scriptstyle f} \\ \mathrm{Id}(f) & \xrightarrow[p_f]{} & \Psi \end{array}$$

has a diagonal filler. The rest of argument follows along the lines of the one used to establish (i) and hence we omit it. □

5.2. **Pullback stability of injective equivalences.** The first application concerns a closure property of the identity type weak factorisation system which does not seem to be shared by many other examples of weak factorisation systems. Recall that, even if $\mathcal{C}(\mathbb{T})$ is not complete, it does admit pullbacks along dependent projections [25, §6].

Proposition 5.2.1. *Pullbacks of maps in $\mathcal{A}$ along maps in $\mathcal{J}$ are in $\mathcal{A}$.*

Proof. By Lemma 5.1.1, the claim follows once we show that, for a pullback diagram of form

$$\begin{array}{ccc} (x \in \Phi, z \in \Omega(f(x)) & \longrightarrow & (x \in \Phi) \\ {\scriptstyle g}\downarrow & & \downarrow{\scriptstyle f} \\ (y \in \Psi, z \in \Omega(y)) & \longrightarrow & (y \in \Psi) \end{array}$$

where $g = (f(x), z)$, if f is an injective equivalence, then so is g. Since f is an injective equivalence, we may assume to have

$$(y \in \Psi)\ s(y) \in \Phi$$

and the judgements in (18), (19), (20). Our first step in showing that g is an injective equivalence will be to construct a judgement

$$(y \in \Psi, z \in \Omega(y))\ t(y, z) \in (x \in \Phi,\ z \in \Omega(f(x))$$

satisfying $(x \in \Phi,\, z \in \Omega(fx))\ tg(x,z) = (x,z) \in (x \in \Phi, z \in \Omega(x))$. Now, to give t is equivalently to give judgements

$$(y \in \Psi,\, z \in \Omega(y))\ t_1(y,z) \in \Phi\,,$$

$$(y \in \Psi,\, z \in \Omega(y))\ t_2(y,z) \in \Omega(f(t_1(y,z))).$$

So we define $t_1(y,z) =_{\text{def}} s(y)$. Now must give an element $t_2(y,z) \in \Omega(fsy)$. We obtain this by substituting $z \in \Omega(y)$ along $\varepsilon_y^{-1} \in \mathrm{Id}(y, fsy)$ using the Leibniz rule:

$$(y \in \Psi,\, z \in \Omega(y))\ \ t_2(y,z) =_{\text{def}} (\varepsilon_y^{-1})_!(z) \in \Omega(f(t_1(y,z)))\,.$$

Observe that we have

$$\begin{aligned} tg(x,z) &= t(fx,z) \\ &= (t_1(fx,z), t_2(fx,z)) \\ &= (sfx, (\varepsilon_{fx}^{-1})_!(z)) \\ &= (x, (1_{fx})_!(z)) \\ &= (x,z) \end{aligned}$$

as required. We now come to the second step in the proof, which is to construct a judgement

$$(y \in \Psi,\, z \in \Omega(y))\ \delta_{(y,z)} \in \mathrm{Id}(gt(y,z), (y,z))$$

satisfying $(x \in \Phi,\, z \in \Omega(fx))\ \delta_{(fx,z)} = 1_{(fx,z)} \in \mathrm{Id}((fx,z),(fx,z))$. Now, to give δ is the same as to give a judgement

$$(y \in \Psi,\, z \in \Omega(y))\ \delta(y,z) \in \mathrm{Id}\big((fsy,\, (\varepsilon_y^{-1})_!(z)), (y,z)\big)\,.$$

By the description of identity context in Section 2.5, to give this is equally well to give a pair of judgements

$$(y \in \Psi,\, z \in \Omega(y))\ \delta_1(y,z) \in \mathrm{Id}(fsy, y)$$

$$(y \in \Psi,\, z \in \Omega(y))\ \delta_2(y,z) \in \mathrm{Id}\big(\delta_1(y,z)_!(\varepsilon_y^{-1})_!(z), z\big)$$

So we define $\delta_1(y,z) =_{\text{def}} \varepsilon(y)$. We must now give an element

$$\delta_2(y,z) \in \mathrm{Id}((\varepsilon_y)_!(\varepsilon_y^{-1})_!(z), z)\,.$$

For this, let us show that we can derive a rule of the form

$$\frac{p \in \mathrm{Id}(a,b) \qquad (x \in \Phi)\ \Omega(x) \in \mathrm{Cxt} \qquad e \in \Omega(b)}{\gamma_p(e) \in \mathrm{Id}((p)_!(p^{-1})_!(e), e)}$$

such that

$$\frac{a \in \Phi \quad e \in \Omega(a)}{\gamma_{1_a}(e) = 1_e \in \mathrm{Id}(e,e)}$$

By elimination on $p \in \mathrm{Id}(a,b)$, we define

$$\gamma_p(e) =_{\mathrm{def}} \mathrm{J}(a,b,p,[x]1_e) \in \mathrm{Id}((p)_!(p^{-1})_!(e),e)$$

Indeed, Lemma 3.2.1 implies that for $x \in \Phi$ and $z \in \Omega(x)$, we have

$$\begin{aligned}(1_x)_!(1_x^{-1})_!(z) &= (1_x^{-1})_!(z)\\ &= (1_x)_!(z)\\ &= z\,.\end{aligned}$$

We can then define $\delta_2(y,z) =_{\mathrm{def}} \gamma_{\varepsilon_y}(z)$. This specifies δ, and we now calculate

$$\begin{aligned}\delta(fx,z) &= (\delta_1(fx,z),\delta_2(fx,z))\\ &= (\varepsilon_{fx},\gamma_{\varepsilon_{fx}}(z))\\ &= (1_{fx},\gamma_{1_{fx}}(z))\\ &= (1_{fx},1_z)\\ &= 1_{(fx,z)}\end{aligned}$$

as required. □

5.3. **Relationship with the homotopy theory of groupoids.** Let us recall that the category **Gpd** of groupoids and functors admits a Quillen model structure $(\mathcal{W},\mathcal{C},\mathcal{F})$, in which the class of weak equivalences $\mathcal{W}$ consists of the categorical equivalences, the class of fibrations $\mathcal{F}$ consists of the Grothendieck fibrations, and the class of cofibrations $\mathcal{C}$ consists of the functors that are injective on objects [1, 17]. As a consequence of this, the category **Gpd** admits a first weak factorisation system given by $(\mathcal{W}\cap\mathcal{C},\mathcal{F})$, and a second weak factorisation system given by $(\mathcal{C},\mathcal{W}\cap\mathcal{F})$. We shall be interested in relating the weak factorisation system $(\mathcal{W}\cap\mathcal{C},\mathcal{F})$ on **Gpd** with the identity type weak factorisation system on $\mathcal{C}(\mathbb{T})$. Let us recall that a functor $f\colon A \to B$ between groupoids is a Grothendieck fibration if and only if for every $\beta\colon f(a) \to b$ in B there exists $\alpha\colon a \to a'$ in A such that $f(a') = b$ and $f(\alpha) = \beta$. The required factorisation of a functor $f\colon A \to B$ as an equivalence injective on objects followed by a Grothendieck fibration can be obtained using the familiar *mapping space* construction,

$$A \xrightarrow{i_f} \mathrm{Path}(f) \xrightarrow{p_f} B\,,$$

where $\mathrm{Path}(f)$ is the groupoid whose objects consist of triples (a, b, β), where $a \in A$, $b \in B$, and $\beta\colon f(a) \to b$ in B.

Theorem 5.3.1. *Let $f\colon \Phi \to \Psi$ be a context morphism.*

(i) If $f \in \mathcal{A}$, then $\mathcal{F}(\Phi) \to \mathcal{F}(\Psi)$ is an equivalence injective on objects.

(ii) If $f \in \mathcal{B}$, then $\mathcal{F}(\Phi) \to \mathcal{F}(\Psi)$ is a Grothendieck fibration.

Proof. For part (i), let $f\colon \Phi \to \Psi$ be in $\mathcal{A}$. By Lemma 5.1.1 f is a type-theoretic injective equivalence, so let us assume

$$(y \in \Psi)\ s(y) \in \Phi$$

and the judgements in (18), (19), (20). We show that $\mathcal{F}(s)\colon \mathcal{F}(\Psi) \to \mathcal{F}(\Phi)$ provides a quasi-inverse to $\mathcal{F}(f)\colon \mathcal{F}(\Phi) \to \mathcal{F}(\Psi)$. First of all, we have a natural isomorphism $\mathcal{F}(s) \circ \mathcal{F}(f) \Rightarrow 1_{\mathcal{F}(\Psi)}$ with components given by the maps $[\varepsilon_b]\colon f(s(b)) \to b$. To establish naturality, we need to show that for every $q \in \mathrm{Id}(b_0, b_1)$, there is a homotopy between $q \circ \varepsilon_{b_0}$ and $\varepsilon_{b_1} \circ fs(q)$, which can be proved by elimination on $q \in \mathrm{Id}(b_0, b_1)$. Secondly, we have

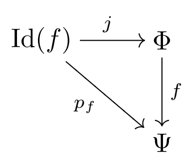
$$\mathcal{F}(s) \circ \mathcal{F}(f) = \mathcal{F}(s \circ f) = \mathcal{F}(1_\Phi) = 1_{\mathcal{F}(\Phi)}\,.$$

which also shows that $\mathcal{F}(f)$ is injective on objects, as required. For part (ii), let $f\colon \Phi \to \Psi$ be a type-theoretic normal isofibration, and assume to have

$$(x \in \Phi, y \in \Psi, u \in \mathrm{Id}(f(x), y))\ j(x, y, u) \in \Phi$$

and judgements as in (21) and (22). By (21), the map $j\colon \mathrm{Id}(f) \to \Phi$ makes the following diagram commute

(24)



To show that $\mathcal{F}(f)\colon \mathcal{F}(\Phi) \to \mathcal{F}(\Psi)$ is Grothendieck fibration, let us consider a map $\beta\colon f(a) \to b$ in $\mathcal{F}(\Psi)$. Let $p \in \mathrm{Id}(f(a), b)$ such that $\beta = [p]$. Note that such a p exists, but it is neither unique nor determined canonically. We then define $a' =_{\mathrm{def}} j(a, b, p)$. Next, we need to define $\alpha\colon a \to a'$ in $\mathcal{F}(\Phi)$. By the description of identity contexts in Section 2.5, we can find an element of the form

$$(1_a, p, \phi) \in \mathrm{Id}(f)\big((a, fa, 1_{fa}), (a, b, p)\big)$$

We define a map $\theta\colon (a, fa, 1_{fa}) \to (a, b, p)$ in $\mathcal{F}(\mathrm{Id}(f))$ by letting $\theta = [(1_a, p, \phi)]$. The required map $\alpha\colon a \to a'$ can then be defined as the result of an application of the functor $\mathcal{F}(j)$ to θ. This has the required domain and codomain, since $a = j(a, f(a), 1_{f(a)}) \in \Phi$ by (22), and $a' = j(a, b, p) \in \Phi$ by the definition set earlier. Furthermore, the commutativity of the diagram in (24) implies that the result of applying $\mathcal{F}(f)$ to α is β, as required. □

We can now compare the factorisations in $\mathcal{C}(\mathbb{T})$ and in **Gpd**.

Proposition 5.3.2. *For every context morphism $f\colon \Phi \to \Psi$, we can define an equivalence surjective on objects $\sigma_f\colon \mathcal{F}(\mathrm{Id}(f)) \to \mathrm{Path}(\mathcal{F}(f))$ making the following diagram commute*

$$\begin{array}{ccc} \mathcal{F}(\Phi) & \xrightarrow{\;i_{\mathcal{F}(f)}\;} & \mathrm{Path}(\mathcal{F}(f)) \\ {\scriptstyle \mathcal{F}(i_f)}\downarrow & \nearrow & \downarrow{\scriptstyle p_{\mathcal{F}(f)}} \\ \mathcal{F}(\mathrm{Id}(f)) & \xrightarrow[\;\mathcal{F}(p_f)\;]{} & \mathcal{F}(\Psi) \end{array}$$

Proof. The objects of $\mathcal{F}(\mathrm{Id}(f))$ are triples (a, b, p), where $p \in \mathrm{Id}(f(a), b)$. The objects of $\mathrm{Path}(\mathcal{F}(f))$ are triples a, b, α, where α is an arrow $\alpha\colon f(a) \to b$ in $\mathcal{F}(\Phi)$. Thus, σ_f can be defined as mapping (a, b, p) to $(a, b, [p])$. Direct calculations show the required properties. □

Let us write $\mathbf{J}$ for the groupoid with two objects and an isomorphism between them. As a special case of Proposition 5.3.2, we obtain that for every context Φ, there is a surjective equivalence between

$$\sigma\colon \mathcal{F}(x \in \Phi, y \in \Phi, u \in \mathrm{Id}(x, y)) \to \mathcal{F}(x \in \Phi)^{\mathbf{J}},$$

where $\mathcal{F}(x \in \Phi)^{\mathbf{J}}$ can be seen as the groupoid of isomorphisms in $\mathcal{F}(x \in \Phi)$.

Acknowledgements

The research described here was started during a visit of the first-named author to the Department of Mathematics of the University of Uppsala and discussed further during the workshop Categorical and Homotopical Structures in Proof Theory, held at the Centre de Recerca Matemàtica. We gratefully acknowledge the support of both institutions. The second-named author is supported by a Research Fellowship of St John's College, Cambridge, and a Marie-Curie Intra-European Fellowship, Project no. 040802. The use of Paul Taylor's prooftree package is also acknowledged.

Appendix A. Structural rules for dependent type theories

Weakening and substitution. The rule $(*)$ has the side-condition that the variable x should not appear as a free variable in Γ or Δ. When stating the first two rules below, J stands for an arbitrary judgement.

$$\frac{(\Gamma, \Delta)\ J \quad (\Gamma)\ A \in \mathrm{Type}}{(\Gamma, x \in A, \Delta)\ J}\,(*) \qquad \frac{(\Gamma, x \in A, \Delta)\ J \quad (\Gamma)\ a \in A}{(\Gamma, \Delta[a/x])\ J[a/x]}$$

Reflexivity, symmetry, and transitivity of definitional equality of types.

$$\frac{A \in \mathrm{Type}}{A = A} \qquad \frac{A = B}{B = A} \qquad \frac{A = B \quad B = C}{A = C}$$

Reflexivity, symmetry, and transitivity of definitional equality of objects.

$$\frac{a \in A}{a = a \in A} \qquad \frac{a = b \in A}{b = a \in A} \qquad \frac{a = b \in A \quad b = c \in A}{a = c \in A}$$

Compatibility rules for definitional equality.

$$\frac{a \in A \quad A = B}{a \in B} \qquad \frac{a = b \in A \quad A = B}{a = b \in B}$$

References

[1] D. W. Anderson, *Fibrations and geometric realizations*, Bulletin of the American Mathematical Society **84(5)** (1978), 765–788.

[2] S. Awodey and M. Warren, *Homotopy theoretic models of identity types*, Mathematical Proceedings of the Cambridge Philosophical Society (To appear).

[3] R. Blackwell, G. M. Kelly, and A. J. Power, *Two-dimensional monad theory*, Journal of Pure and Applied Algebra **59** (1989), 1–41.

[4] A. K. Bousfield, *Constructions of factorisation systems in categories*, Journal of Pure and Applied Algebra **9** (1977), 207–220.

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

[6] N. G. de Brujin, *Telescopic mappings in typed lambda calculus*, Information and Computation **91(2)** (1991), 189–204.

[7] P. Dybjer, *Internal type theory*, Types for proofs and programs (S. Berardi and M. Coppo, eds.), Lecture Notes in Computer Science, vol. 1158, Springer, 1997, pp. 120–134.

[8] R. Garner, *Two-dimensional models of type theory*, In preparation.

[9] M. Hofmann, *On the interpretation of type theory of locally cartesian closed categories*, Computer Science Logic 1994 (L. Pacholski and J. Tiuryn, eds.), Lecture Notes in Computer Science, vol. 933, Springer, 1995, pp. 427–441.

[10] ______, *Syntax and semantics of dependent type theories*, Semantics of Logics of Computation (P. Dybjer and A. M. Pitts, eds.), Cambridge University Press, 1997, pp. 79–130.

[11] ______, *Extensional concepts in intensional type theory*, Springer, 1999.

[12] M. Hofmann and T. Streicher, *The groupoid model of type theory*, Twenty-five years of constructive type theory (G. Sambin and J. Smith, eds.), Oxford Logic Guides, vol. 36, Oxford University Press, 1998, pp. 83–111.

[13] M. Hovey, *Model categories*, Mathematical Surveys and Monographs, vol. 63, American Mathematical Society, 1999.

[14] W. A. Howard, *The formulae-as-types notion of construction*, To H. B. Curry: Essays in Combinatory Logic, Lambda Calculus, and Formalism (J. R. Hindley and J. P. Seldin, eds.), Academic Press, 1980, pp. 479–490.

[15] B. Jacobs, *Categorical logic and type theory*, Studies in Logic and the Foundations of Mathematics, vol. 141, Elsevier, 1999.

[16] A. Joyal, *The theory of quasi-categories and its applications*, Quaderns 45, Centre de Recerca Matemàtica, 2008.

[17] A. Joyal and M. Tierney, *Strong stacks and classifying spaces*, Category Theory (Como 1990) (Berlin), Lecture Notes in Mathematics, vol. 1488, Springer, 1991, pp. 213–236.

[18] S. Lack, *Homotopy-theoretic aspects of 2-monads*, Journal of Homotopy and Related Structures (To appear).

[19] J. Lambek and P. J. Scott, *Introduction to higher-order categorical logic*, Cambridge Studies in Advanced Mathematics, vol. 7, Cambridge University Press, 1986.

[20] F. W. Lawvere, *Functorial semantics*, Proc. Nat. Acad. Sci. USA, **50** (1963), 869–872.

[21] P. Martin-Löf, *Intuitionistic type theory*, Blibliopolis, 1984.

[22] J. P. May, *A concise course in algebraic topology*, Chicago Lecture Notes in Mathematics, The University of Chicago Press, 1999.

[23] B. Nordström, K. Petersson, and J. M. Smith, *Programming in Martin-Löf type theory: an introduction*, Oxford University Press, 1990.

[24] ______, *Martin-Löf Type Theory*, Handbook of Logic in Computer Science. Volume V (S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, eds.), Oxford University Press, 2001, pp. 1–37.

[25] A. M. Pitts, *Categorical logic*, Handbook of Logic in Computer Science. Volume V (S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, eds.), Oxford University Press, 2001, pp. 39–128.

[26] J. Rosický and W. Tholen, *Lax factorisation algebras*, Journal of Pure and Applied Algebra **175(3)** (2002), 335–382.

[27] D. S. Scott, *Constructive validity*, Symposium on Automated Demonstration, Lecture Notes in Mathematics, vol. 125, Springer, 1970, pp. 237–275.

[28] ______, *Relating theories of the λ-calculus*, To H. B. Curry: Essays in Combinatory Logic, Lambda Calculus, and Formalism (J. R. Hindley and J. P. Seldin, eds.), Academic Press, 1980, pp. 403–450.

[29] R. A. G. Seely, *Locally cartesian closed categories and type theories*, Mathematical Proceedings of the Cambridge Philosophical Society **95** (1984), 33–48.

[30] R. Street, *Fibrations and Yoneda's lemma in a 2-category*, Category Seminar (Proc. Sem. Sydney 1972/73) (G. M. Kelly and R. H. Street, eds.), Lecture Notes in Mathematics, vol. 420, Springer, 1974, pp. 104–133.

[31] T. Streicher, *Investigations into intensional type theory*, Habilitationsschrift, Ludwig-Maximilians Universität München, 1993.

[32] Paul Taylor, *Practical foundations of mathematics*, Cambridge Studies in Advanced Mathematics, vol. 59, Cambridge University Press, 1999.

Nicola Gambino
Department of Computer Science
University of Leicester
University Road LE1 7RH,
England and Centre de Recerca Matemàtica
Apartat 50, E-08193 Bellaterra, Spain
E-mail address: nicola.gambino@gmail.com

Richard Garner
Department of Mathematics
Uppsala University
Box 480, S-751 06
Uppsala, Sweden
E-mail address: rhgg2@cam.ac.uk