Decomposing Identity Types into Null Types

Yuta Yamamoto

Homotopy type theory (HoTT) is a variant of dependent type theory which admits a *homotopy theoretic* interpretation. It not only can reason about homotopy theory but also serves as a foundation for mathematics under the identification of proposition-as-types. Its primitive type formers include the identity type, Σ-type, Π-type, and universes, and various objects can be constructed from these.

It is a natural question to ask whether these type formers can be decomposed into "more primitive" ones, which makes the theory more understandable. As a first step toward such a direction, we focus on the identity type, which is particularly interesting because it consists of *homotopies* and plays an important role in HoTT.

As the main contribution, we devise a restricted type theory where identity types are derivable. We use the semantic intuition of chain complexes to introduce a type former called a *null type*, intended to consist of *nullhomotopies*, together with additive structures on all types. We also show that the syntax is modeled by any additive category together with a comonad, which includes our intended model of chain complexes.

1 Introduction

The recent discovery of a close connection between Martin-Löf type theory and homotopy theory has given rise to a field called *homotopy type theory* (HoTT) [43, 32]. It is an extension of Martin-Löf type theory based on the observation that we can regard types as ∞-groupoids and identity types as their path spaces. An ∞-groupoid is like a topological space, but the notion of *paths* or *homotopies* plays a central role and the collection of paths between any endpoints again forms an ∞-groupoid called a *path space*. This results in infinite towers of path spaces, hence the name. HoTT not only can reason about the theory of ∞-groupoids or more generally homotopy theory but also serves as a foundation for mathematics under the identification of propositions-as-types. The primitive syntax of HoTT includes identity types, Σ-types, Π-types, and universes, and various propositions can be constructed from these.

There are various approaches to improvements to the syntax of HoTT, including the *cubical approach* [11, 2, 3], the current most active area among such. This study adds another idea for an improvement to HoTT. The question we begin with is whether the primitive type formers in HoTT are really "primitive". If not, decomposing them into "more primitive" ones makes the theory more understandable. As a first step toward such a direction, we focus on the identity type because it is particularly interesting in that it consists of homotopies and plays an important role in HoTT.

In this study, we devise a restricted type theory based on the idea regarding types as *chain complexes*. We introduce a type former called a *null type*, intended to consist of *nullhomotopies*, together with additive structures on all types. The reasons for considering such syntax are as follows. First, the identity type is derivable from the null type, which may provide a hint to improve HoTT because the inference rules for the null type are much simpler than those for the identity type. Second, as a chain complex is a common mathematical tool for studying (co)homology and spectra, our theory may be a step towards synthetic, computer-checked reasoning about them. Third, a dependent type theory with additives, which is not defined before, is interesting in its own right because our way of endowing every type with an abelian group structure using context morphisms is generally applicable to other algebraic structures.

To appear in EPTCS.

a: A *homotopy* between paths

b: A *nullhomotopy* of a loop

Figure 1: Homotopy and nullhomotopy

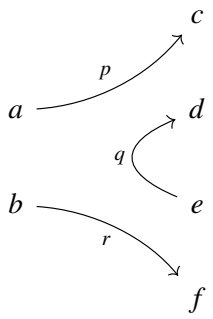


Figure 2: 0-cells $a+b$ and $c+d-e+f$ identified by a 1-cell $p+q+r$

A geometric consideration we begin with is the following; a homotopy α between paths p and q, as in Figure 1a, can also be viewed as a nullhomotopy, i.e., a continuous deformation from the trivial loop, of the loop $q \cdot p^{-1}$ as in Figure 1b. Syntactically, this corresponds to the following theorem in HoTT:

$$\mathrm{Id}_{\mathrm{Id}_A(a,b)}(p,q) \simeq \mathrm{Null}_a(q \cdot p^{-1}), \tag{1}$$

where $(-) \simeq (-)$, $(-) \cdot (-)$, and $(-)^{-1}$ are equivalence between types, a composition of paths, and an inverse of a path, respectively, and $\mathrm{Null}_a(l)$ is defined to be $\mathrm{Id}_{\mathrm{Id}_A(a,a)}(\mathrm{refl}_A(a), l)$. This observation suggests a type theory where the null type is primitive instead of the identity type. The identity type should be recovered by regarding the above theorem as the definition. However, there is a problem that the "definition" (1) does not apply to general terms p and q because the composition and inverse operations are defined only on identity terms, i.e., terms whose types are the identity types. This is where the chain complex comes in. Loosely speaking, a chain complex is like a space, but one can take sums or additives of its points (0-cells), paths (1-cells), and higher cells. Every cell is associated with a lower cell called a *boundary* instead of having two endpoints. A cell l that is a boundary of other higher cell α is said to be *nullhomotopic* or *null*. The cell α is called a *nullhomotopy* and is like a "proof object" of the nullness of l. Two cells are identified if their difference is null; the nullhomotopy is a proof object of an identification of the cells as in Figure 2. Keeping these in mind, we introduce another syntax modeled by chain complexes, the *additives* $a+b$, $-a$, and 0, which enable us to define the identity type between terms $a : A$ and $b : A$ as

$$\mathrm{Id}_A(a,b) := \mathrm{Null}(b + (-a)).$$

A natural candidate for the model of our type theory is the chain complex model defined in [43, 39]. The syntax corresponding to the nullhomotopy should be the null type. However, there is a problem

$$\frac{\vdash A \quad \Gamma \vdash a : A}{\Gamma \vdash \mathrm{Null}_A(a)}$$

$$\frac{\Gamma \vdash a : A, p : \mathrm{Null}_A(a) \qquad x : A, y : \mathrm{Null}_A(x) \vdash C}{\Gamma \vdash \mathrm{elim}_{\mathrm{Null}}(a,p) : C[a/x, p/y]}$$

Figure 3: Null types

concerning the interpretation of the additives; due to the type dependency, even if the term $a+b$ is well-formed, the types of a and b need not be the same, thus the interpretation of them need not be parallel, i.e., they need not have the same domain and codomain, obstructing the naive interpretation using the additive structures on hom-sets. To avoid the problem, we use the idea of local universes [25] in the framework of the natural model [5] to construct a model where such terms a and b are interpreted not as parallel morphisms but as parallel *liftings*. As a result, we show that our theory is modeled by any additive category with a comonad, which includes our intended model of chain complexes.

The plan of the paper is as follows. The first part, Section 2, presents the syntactical aspects of our type theory. The formal inference rules of null types and additives are presented. The derivability of the identity types is proved. The second part, Section 3, describes the semantics.

2 Syntax

This section presents the restricted type theory which contains the *null types* and the *additives*. We also show that it can recover identity types. The rest of the paper is based on the fragment of Martin-Löf type theory that contains only the basic structural rules, which is standard but included in Appendix A for completeness.

2.1 Null Types

The *null type* is very much like the identity type. While the identity type intuitively expresses a homotopy between two terms, the null type should express a nullhomotopy of a term. We extend the theory by the following grammar:

$$\begin{aligned} A, B, C &::= \cdots \\ &\quad \mid \mathrm{Null}_A(a) && \text{type} \\ a, b, c &::= \cdots \\ &\quad \mid \mathrm{elim}_{\mathrm{Null}}(a,b) && \text{term} \end{aligned}$$

To motivate our rules for null types, recall that the rules for identity types state that they are inductive types generated by the canonical element $\mathrm{refl}(x) : \mathrm{Id}_A(x,x)$. Similarly, one can think of null types as inductive types generated by no element. The rules for null types are listed in Figure 3. These are called the formation and elimination rules, respectively.

Several comments about the rules in Figure 3 are in order. First, since we do not have an introduction rule, we also do not have (β)-rule. Second, we assume that the type A is closed. The geometric intuition is as the following; while HoTT regards a type as a space, we regard it as a chain complex. A term $p : \mathrm{Null}_A(a)$ can be seen as a 1-cell p on A witnessing the nullness of a 0-cell a. Similarly, a term

$\alpha : \mathrm{Null}_{\mathrm{Null}_A(0)}(l)$ can be seen as a 2-cell α on A witnessing the nullness of a 1-cell l on A (a term $0 : A$ is called an *additive*, which we describe later). The closedness assumption requires that one cannot construct a type like $\Gamma \vdash \mathrm{Null}_{\mathrm{Null}_A(a)}(p)$ unless a is 0. Intuitively, this expresses the fact that asking whether the cell is null or not is meaningful only if the cell is *closed*, i.e., its boundary is 0. This is analogous to the fact that asking whether two terms are identical or not is meaningful only if they have the same type.

2.2 Additives

Now, we present another new syntactical construct, the additives. They represent an additive structure of a category and plays an essential role in defining identity types. The rules are a bit tricky. A possible attempt is to introduce a "rule":

$$\frac{\Gamma \vdash a : A, b : A}{\Gamma \vdash a + b : A.} \tag{2}$$

However, the "rule" (2) contradicts our intuition. In the presence of type dependencies, the term $\Gamma \vdash a : A$ is not necessarily interpreted as a morphism unlike the case of the simply typed situation. For example, the locally cartesian category model [35, 17], or local universes model [25], treats terms $a : A$ not as mere morphisms but as *sections* or *liftings* for the diagram that the type $\Gamma \vdash A$ represents. The problem is that even if hom-sets have abelian group structures, the set of sections or liftings does not have such structures in general. Geometric intuition also suggests that the "rule" (2) is false. Consider the case when A is of the form $\mathrm{Null}_C(c)$. Then, terms $a : \mathrm{Null}_C(c)$ and $b : \mathrm{Null}_C(c)$ are cells on a space C whose boundaries are c. If we take the sum of the cells a and b, we have a cell $a + b$ whose boundary is $c + c$ because the boundary operator works linearly. In particular, we want the type of a term $a + b$ to be $\mathrm{Null}_C(c + c)$, not $\mathrm{Null}_C(c)$.

A solution is to use context morphisms, the syntactical counterpart of morphisms; given context morphisms $\Gamma \vdash \sigma \Rightarrow \Delta, \tau \Rightarrow \Delta$, we want their additive $\Gamma \vdash \sigma + \tau \Rightarrow \Delta$. The following "rule" formalizes the above intuition:

$$\frac{\Gamma \vdash \sigma \Rightarrow \Delta, \tau \Rightarrow \Delta \quad \Delta \vdash A \quad \Gamma \vdash a : A[\sigma], b : A[\tau]}{\Gamma \vdash a +_{(\Delta \vdash A), \sigma, \tau} b : A[\sigma +_\Delta \tau],} \tag{3}$$

where $\sigma + \tau$ is a derived operator defined elementwise. Although (3) coincides with our intuition, it contains too much "junk" information, and in particular, it prevents us from the identification between terms necessary for recovering identity types. To deal with this difficulty, a technical notion called a *normal type* is introduced. The basic observation is that information contained in subscripts is excessive; any term of the form $a : A[\sigma]$ can be rewritten to $a : \bar{A}[\sigma']$ where $\bar{A}$ can be thought of as a kind of "normalization" of A. The rules of additives are restricted for the case where the type A is normal so that we have minimum information for type-checking.

Definition 4. *A type $\Delta \vdash A$ is said to be normal, written $\Delta \vdash^{normal} A$, if variables in Δ occur in A in that order from left to to right and each occurs exactly once. To denote a normal type $\Delta \vdash^{normal} A$, we often omit the context and simply write A since Δ is clear from A.*

Now, we extend the grammar of terms with the following:

$$\begin{aligned} a, b, c ::= {} & \cdots \\ & \mid 0_{(\Delta \vdash A)} \mid a +_{(\Delta \vdash A), \sigma, \tau} b \mid -_{(\Delta \vdash A), \sigma} a. \end{aligned}$$

Here, the type $\Delta \vdash A$ is intended to be normal. Context morphisms σ and τ and contexts Δ in subscripts are often omitted. Normal types A in subscripts are also omitted when they are clear. The rules for

$$\frac{\Gamma \vdash \qquad \Delta \vdash^{\mathrm{normal}} A}{\Gamma \vdash 0_{(\Delta \vdash A)} : A[0_\Delta]}$$

$$\frac{\Gamma \vdash \sigma \Rightarrow \Delta, \tau \Rightarrow \Delta \quad \Delta \vdash^{\mathrm{normal}} A \quad \Gamma \vdash a : A[\sigma], b : A[\tau]}{\Gamma \vdash a +_{(\Delta \vdash A),\sigma,\tau} b : A[\sigma +_\Delta \tau]}$$

$$\frac{\Gamma \vdash \sigma \Rightarrow \Delta \qquad \Delta \vdash^{\mathrm{normal}} A \qquad \Gamma \vdash a : A[\sigma]}{\Gamma \vdash -_{(\Delta \vdash A),\sigma} a : A[-_\Delta \sigma]}$$

Figure 4: Additives

$$\frac{\Gamma \vdash \qquad \Delta \vdash A}{\Gamma \vdash 0_{\bar{A}} : A[0_\Delta]}$$

$$\frac{\Gamma \vdash \sigma \Rightarrow \Delta, \tau \Rightarrow \Delta \qquad \Delta \vdash A \qquad \Gamma \vdash a : A[\sigma], b : A[\tau]}{\Gamma \vdash a +_{\bar{A}} b : A[\sigma +_\Delta \tau]}$$

$$\frac{\Gamma \vdash \sigma \Rightarrow \Delta \qquad \Delta \vdash A \qquad \Gamma \vdash a : A[\sigma]}{\Gamma \vdash -_{\bar{A}} a : A[-_\Delta \sigma]}$$

Figure 5: Derived rules for additives

additives are listed in Figure 4, together with the derived rules of additives in Figure 5, where 0_Δ, $+_\Delta$, and $-_\Delta$ are defined elementwise. For example, 0_Δ is defined as:

$$0_{()} := ()$$
$$0_{\Delta,x:A} := (0_\Delta, 0_{\bar{A}}).$$

We also assume the following axioms of abelian groups:

$$a + (b + c) = (a + b) + c$$
$$0 + a = a$$
$$-a + a = 0$$
$$a + b = b + a$$

and an equation

$$\mathrm{elim}_{\mathrm{Null}}(0,0) = 0. \tag{5}$$

Remark 6. *We regard the variables of type A in the subscript are bound by those of* Δ*, i.e., substitutions do not affect the subscript and we identify terms whose subscripts are* α*-equivalent.*

The derivability of the rules in Figure 5 follows from the following lemma:

Lemma 7. *For each type* $\Delta \vdash A$*, there exists a unique (up to renaming of variables) normal type* $\Delta' \vdash^{\mathrm{normal}} \bar{A}$ *together with* $\Delta \vdash \sigma \Rightarrow \Delta'$ *such that* $\bar{A}[\sigma] = A$*.*

Proof. Fix $\mathbb{N}$-indexed family of variables $x_1, x_2, \ldots$. For each type A, define $\bar{A}$ to be a type obtained by replacing each occurrence of variables by $x_1, x_2, \ldots$ from left to right. □

Equation (5) can be thought of as a computational rule for the null type. Since this presentation is not "orthogonal" in that the syntax of null types relies on that of additives, it is tempting to recover the orthogonality by regarding all of the rules of additives as part of the structural rules and assuming "preservation of zero":

$$M[0_\Delta] = 0, \tag{8}$$

or more generally, "linearity of substitutions":

$$\begin{aligned} M[0_\Delta] &= 0 \\ M[\sigma+\tau] &= M[\sigma]+M[\tau] \\ M[-\sigma] &= -M[\sigma], \end{aligned}$$

which are natural to consider since they correspond to the linearity of post-compositions in an additive category, and its dual concept of the linearity of pre-compositions always holds syntactically because substitutions always commute with any term constructor.

However, we do not assume them for several reasons. First, it complicates type-checking. Second, they make the theory degenerate, i.e., we can prove that any closed term $\vdash a : A$ is equal to 0 because we have $a = a[()] = a[0_{()}] = 0$, which obstructs Π-types, higher inductive types, and universes. Indeed, it is interesting to consider the system in which structures of types are not necessarily preserved by compositions because such categories do exist, i.e., there are categories together with some structures on hom-sets (not limited to the abelian group structures and hence not being the model of our syntax) that are not preserved by compositions in general. Examples include the (bi-)category **Prof** of profunctors, which has cartesian closed categorical structures on each hom-sets. The existence of such categories suggests the possibility of modifying our theory to admit many other interesting models other than additive categories. However, since not only post-compositions but also pre-compositions do not preserve the structures in **Prof**, to admit such models, it should require fundamental changes to the system so that it does not treat variable substitutions in a usual manner, which is not explored further here. Examples of additives are in order.

Example 9. *Given terms $\Gamma \vdash p : \mathrm{Null}_A(a), q : \mathrm{Null}_A(b)$, we have a term $\Gamma \vdash p+q : \mathrm{Null}_A(a+b)$. Similarly, for a term $\Gamma \vdash p : \mathrm{Null}_A(a)$, we have $\Gamma \vdash -p : \mathrm{Null}_A(-a)$ and a zero term $\Gamma \vdash 0 : \mathrm{Null}_A(0)$.*

Example 10. *Given a closed type $\vdash A$, we can form its loop space $\vdash \Omega A := \mathrm{Null}_A(0)$. It will be interpreted by the loop chain complex (Definition 34) in our model. Note that one does not need explicitly to take a base point to form a loop space, which may be a possible advantage for our theory to reason about stable homotopy theory.*

Example 11. *For any closed type $\vdash A$, the set of terms of a type A carries the structure of an abelian group. In particular, the following rules are derivable:*

$$\frac{\Gamma \vdash \quad \vdash A}{\Gamma \vdash 0 : A} \qquad \frac{\vdash A \quad \Gamma \vdash a : A, b : A}{\Gamma \vdash a+b : A} \qquad \frac{\vdash A \quad \Gamma \vdash a : A}{\Gamma \vdash -a : A}$$

Remark 12. *The additives are presented using context morphisms and the abelian group structure on closed types are derived as in Example 11. Assuming Σ-types, one can do the converse. That is, we can take the additives of context morphisms $\Gamma \vdash \sigma \Rightarrow \Delta$ via its representative term $\Gamma \vdash \tilde{\sigma} : \tilde{\Delta}$. However, we did not include other type formers and chose to present the rules for additives using context morphisms to concentrate on our concern.*

2.3 Identity Types

The main feature of this theory is that the identity type is a derived concept. Our aim in this subsection is to recover the identity type using the null types and the additives.

The identity type allows us to form a type $\Gamma \vdash \mathrm{Id}_A(a,b)$ for any terms $\Gamma \vdash a : A, b : A$. To express such a type, let us first consider a simple case. Consider a closed type A and terms $\Gamma \vdash a : A, b : A$, then we can simply define the identity type by

$$\Gamma \vdash \mathrm{Id}_A(a,b) := \mathrm{Null}_A(b + (-a)) \tag{13}$$

The intuition is that cells a and b are identified when their difference is null. Now we want to extend it for a dependent type $\Gamma \vdash A$. To this end, note that given terms $\Gamma \vdash a : A, b : A$, we have a term $\Gamma \vdash (b - a) := (b +_{\bar{A}} (-_{\bar{A}} a)) : A[0_\Gamma]$, which leads to the following:

Definition 14. *For terms $\Gamma \vdash a : A, b : A$, we define their identity type to be*

$$\Gamma \vdash \mathrm{Id}_A(a,b) := \mathrm{Null}_{A[0_\Gamma]}(b - a)$$

Note that the type $A[0_\Gamma]$ is closed so that one can form its null type. When the type A is already closed, the above definition coincides with (13). Now, we prove our main result stating that the type defined above indeed behaves like the identity type:

Theorem 15. *The introduction, elimination, and (β) rule are admissible for the identity types defined in Definition 14.*

Proof. We can define a reflexivity term by

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash \mathrm{refl}(a) := 0 : \mathrm{Id}_A(a,a)}$$

since $\mathrm{Id}_A(a,a) = \mathrm{Null}_{A[0_\Gamma]}(0)$ is a closed type.

To define the elimination term, assume we have the following terms and a type:

$$\begin{aligned} \Gamma \vdash a : A, b : A, p : \mathrm{Id}_A(a,b) \\ \Gamma, x : A, y : A, z : \mathrm{Id}_A(x,y) \vdash C \\ \Gamma, x : A \vdash c : C[x/y, \mathrm{refl}(x)/z]. \end{aligned}$$

First, applying a substitution

$$\begin{aligned} y : A[0_\Gamma], z : \mathrm{Id}_A(0,y) \vdash (0_\Gamma, 0, y, z) \\ \Rightarrow \Gamma, x : A, y : A, z : \mathrm{Id}_A(x,y) \end{aligned}$$

to C, we obtain a type

$$y : A[0_\Gamma], z : \mathrm{Id}_{A[0_\Gamma]}(0,y) \vdash C[(0_\Gamma, 0, y, z)],$$

but since $\mathrm{Id}_{A[0_\Gamma]}(0,y)$ is equal to $\mathrm{Null}_{A[0_\Gamma]}(y)$, we have

$$\Gamma \vdash \mathrm{elim}_{\mathrm{Null}}(b - a, p) : C[(0_\Gamma, 0, b - a, p)] \tag{16}$$

by the elimination rule of null type. Now, by applying substitution to c, we have

$$\Gamma \vdash c[a/x] : C[a/x, a/y, \mathrm{refl}(a)/z]. \tag{17}$$

Adding (16) and (17) yields the desired term

$$\Gamma \vdash \mathrm{elim}_{\mathrm{Id}}(a,b,p,c) := c[a/x] + \mathrm{elim}_{\mathrm{Null}}(b-a,p) : C.$$

(β) rule is proved by

$$\begin{aligned}\mathrm{elim}_{\mathrm{Id}}(a,a,\mathrm{refl}(a),c) &= c[a/x] + \mathrm{elim}_{\mathrm{Null}}(a-a,\mathrm{refl}(a)) \\ &= c[a/x] + \mathrm{elim}_{\mathrm{Null}}(0,0) \\ &= c[a/x] + 0 \\ &= c[a/x]\end{aligned}$$

This completes the proof. □

A natural question is that whether the converse direction of Theorem 15 holds, i.e., whether we can define null types using identity types. Indeed, assuming additives, we can define a type:

$$\frac{\Gamma \vdash a : A \quad \vdash A}{\Gamma \vdash \mathrm{Null}_A(a) := \mathrm{Id}_A(0,a).}$$

Then, proving the elimination rule for the null type amounts to constructing a term e in:

$$\frac{x : A, y : \mathrm{Id}_A(0,x) \vdash C}{x : A, y : \mathrm{Id}_A(0,x) \vdash e : C}$$

and since we have a term $\vdash 0 : C[0/x, \mathrm{refl}(0)/y]$, it suffices to have the based path induction (Figure 9).

Then, it is also natural to ask whether our theory can derive based path induction. We have a partial answer:

Theorem 18. *The based path induction is admissible for the identity types defined in Definition 14 if the "base" term a preserves zero, i.e., $a[0_\Gamma] = 0$. In particular, if we assume the "preservation of zero"* (8)*, the based path induction is admissible.*

Proof. Similar to that of Theorem 15. □

Theorem 15 can be thought of as a corollary of Theorem 18 because the usual path induction is an instance of a based path induction where the "base" is chosen to be x, which obviously preserves zero without any assumption.

3 Semantics

So far we have defined the syntax of our type theory. In this section, we show that an additive category with a comonad gives rise to a model of our type theory, which includes a chain complex model as a special case. The very basics concerning chain complexes are included in Appendix B. We use the *local universes* construction [25] to obtain our model. The framework we use to interpret type dependencies is a *natural model* defined by Awodey [5]. We begin by recalling its definition.

Definition 19. *For a category $\mathscr{C}$, let $\hat{\mathscr{C}}$ be the presheaf category of $\mathscr{C}$ and $\mathrm{y} : \mathscr{C} \to \hat{\mathscr{C}}$ be the Yoneda embedding. For any presheaf $F \in \hat{\mathscr{C}}$ and an object $\Gamma \in \mathscr{C}$, we identify elements of $F(\Gamma)$ and morphisms $\mathrm{y}\Gamma \to F$ in $\hat{\mathscr{C}}$ by Yoneda lemma. Then, a natural model consists of:*

- *a category $\mathscr{C}$ with a terminal object* 1*;*

- *a morphism* $\mathrm{typeof} : \mathrm{Tm} \to \mathrm{Ty}$ *in* $\hat{\mathscr{C}}$*;*
- *for any object* $\Gamma \in \mathscr{C}$ *and element* $A \in \mathrm{Ty}(\Gamma)$*, an object* $\Gamma.A \in \mathscr{C}$*, a morphism* $p_{\Gamma,A} : \Gamma.A \to \Gamma$ *in* $\mathscr{C}$*, and an element* $q_{\Gamma,A} \in \mathrm{Tm}(\Gamma.A)$ *such that the diagram*

$$\begin{array}{ccc} \mathrm{y}\,\Gamma.A & \xrightarrow{q_{\Gamma,A}} & \mathrm{Tm} \\ {\scriptstyle \mathrm{y}\,p_{\Gamma,A}}\downarrow & \lrcorner & \downarrow{\scriptstyle \mathrm{typeof}} \\ \mathrm{y}\,\Gamma & \xrightarrow[A]{} & \mathrm{Ty} \end{array}$$

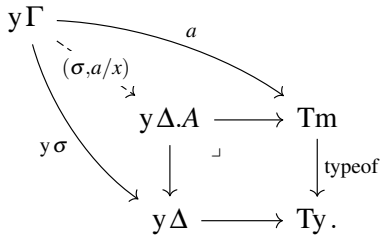
is a pullback square.

Remark 20. *In [5], Awodey defined a natural model to be a morphism between presheaves that is representable in the sense of Grothendieck. However, in the above definition, we moreover assume that the specific representation is given as a structure. This presentation is equivalent to category with families [12].*

Given a natural model $\mathrm{typeof} : \mathrm{Tm} \to \mathrm{Ty}$ in $\hat{\mathscr{C}}$, we regard contexts $\Gamma \vdash$ as objects $\Gamma \in \mathscr{C}$, context morphisms $\Gamma \vdash \sigma \Rightarrow \Delta$ as morphisms $\sigma : \Gamma \to \Delta$, types $\Gamma \vdash A$ as elements $A \in \mathrm{Ty}(\Gamma)$, and terms $\Gamma \vdash a : A$ as liftings of $\Gamma \vdash A$ as in:

$$\begin{array}{ccc} & & \mathrm{Tm} \\ & {\scriptstyle a}\nearrow & \downarrow{\scriptstyle \mathrm{typeof}} \\ \mathrm{y}\,\Gamma & \xrightarrow[A]{} & \mathrm{Ty}\,. \end{array}$$

A substitution is interpreted as the precomposition. The validity of basic structural rules can be verified as the following. We interpret empty context $()$ as $1 \in \mathscr{C}$ and context extension $\Gamma, x : A$ as $\Gamma.A$. The context morphism $\Gamma \vdash () \Rightarrow ()$ is the unique morphism to the terminal object and the context morphism $\Gamma \vdash (\sigma, a/x) \Rightarrow (\Delta, x : A)$ is defined by the unique morphism:



Now, we present the *local universes* construction of a natural model from a category with a class of morphisms. Note that we do not require that the the class of morphisms to be pullback-stable as in [5] which is not necessary for our purpose.

Definition 21. *Given a category* $\mathscr{C}$ *with a terminal object* $1 \in \mathscr{C}$*, chosen pullbacks, and a class of morphisms* $\mathfrak{M} \subseteq \mathrm{Mor}(\mathscr{C})$*, we construct a morphism* $\mathrm{typeof}_{\mathfrak{M}} : \mathrm{Tm}_{\mathfrak{M}} \to \mathrm{Ty}_{\mathfrak{M}}$ *in* $\hat{\mathscr{C}}$ *by:*

$$\begin{aligned} \mathrm{Tm}_{\mathfrak{M}}(\Gamma) &:= \{\langle a, d\rangle \in \mathrm{Ob}(\Gamma\backslash\mathscr{C}) \times \mathfrak{M} \mid \mathrm{cod}\, a = \mathrm{dom}\, d\} \\ \mathrm{Ty}_{\mathfrak{M}}(\Gamma) &:= \{\langle b, d\rangle \in \mathrm{Ob}(\Gamma\backslash\mathscr{C}) \times \mathfrak{M} \mid \mathrm{cod}\, b = \mathrm{cod}\, d\} \\ \mathrm{typeof}_{\mathfrak{M}}(\Gamma) &:= \langle a, d\rangle \mapsto \langle d \circ a, d\rangle. \end{aligned}$$

Theorem 22. *Given a category* $\mathscr{C}$ *with a terminal object* $1 \in \mathscr{C}$*, chosen pullbacks, and a class of morphisms* $\mathfrak{M} \subseteq \mathrm{Mor}(\mathscr{C})$ *a morphism* $\mathrm{typeof}_{\mathfrak{M}} : \mathrm{Tm}_{\mathfrak{M}} \to \mathrm{Ty}_{\mathfrak{M}}$ *in* $\hat{\mathscr{C}}$ *is a natural model.*

Proof. For $\Gamma \in \mathscr{C}$ and $A = \langle |A| : \Gamma \to A_0, d_A : A_1 \to A_0 \rangle \in \mathrm{Ty}_{\mathfrak{M}}(\Gamma)$, we define $\Gamma.A$, $p_{\Gamma,A}$ and $q_{\Gamma,A}$ by the following pullback square

$$\begin{array}{ccc} \Gamma.A & \xrightarrow{q'_A} & A_1 \\ {\scriptstyle p'_A}\downarrow & \lrcorner & \downarrow{\scriptstyle d_A} \\ \Gamma & \xrightarrow[|A|]{} & A_0 \end{array}$$

and

$$\begin{aligned} p_{\Gamma,A} &:= p'_A \\ q_{\Gamma,A} &:= \langle q'_A, d_A \rangle. \end{aligned}$$

They fit into a pullback square

$$\begin{array}{ccc} \mathrm{y}\,\Gamma.A & \xrightarrow{q_{\Gamma,A}} & \mathrm{Tm}_{\mathfrak{M}} \\ {\scriptstyle \mathrm{y}\,p_{\Gamma,A}}\downarrow & \lrcorner & \downarrow{\scriptstyle \mathrm{typeof}_{\mathfrak{M}}} \\ \mathrm{y}\,\Gamma & \xrightarrow[A]{} & \mathrm{Ty}_{\mathfrak{M}}. \end{array}$$

For detail, see [5]. □

It would be helpful to describe the interpretation of type theory in terms of $\mathscr{C}$ and $\mathfrak{M}$; we regard contexts $\Gamma \vdash$ as objects in $\mathscr{C}$, context morphisms $\Gamma \vdash \sigma \Rightarrow \Delta$ as morphisms in $\mathscr{C}$, types $\Gamma \vdash A$ as cospans

$$\begin{array}{ccc} & & A_1 \\ & & \downarrow{\scriptstyle d_A} \\ \Gamma & \xrightarrow[|A|]{} & A_0 \end{array}$$

in $\mathscr{C}$, and terms $\Gamma \vdash a : A$ as liftings

$$\begin{array}{ccc} & & A_1 \\ & {\scriptstyle a}\nearrow & \downarrow{\scriptstyle d_A} \\ \Gamma & \xrightarrow[|A|]{} & A_0 \end{array}$$

in $\mathscr{C}$. Substitutions correspond to precompositions.

Example 23. *Let $\mathscr{C}$ be a category with a terminal object 1 and chosen pullbacks. Then, a comonad (L, ε, δ) on $\mathscr{C}$ induces a natural model $\mathrm{typeof}_{\varepsilon}$ where the natural transformation ε is regarded as a class of morphisms $\varepsilon \subseteq \mathrm{Mor}(\mathscr{C})$ of its components.*

In particular, it includes our desired model, namely the category $\mathrm{Ch}(R\mathrm{Mod})$ of chain complexes with a comonad N, where a (chain) homotopy between f and g corresponds to the following commutative diagram

$$\begin{array}{ccc} A & \xrightarrow{\alpha} & NB \\ & {\scriptstyle g-f}\searrow & \downarrow{\scriptstyle \varepsilon_B} \\ & & B. \end{array}$$

See Appendix B for detail.

Now we prove the following two main results:

Theorem 24. *For a category $\mathscr{C}$ with a terminal object 1, chosen pullbacks, and a comonad (L, ε, δ), the natural model* $\mathrm{typeof}_\varepsilon$ *models null types.*

Proof. For the formation rule, given a term $\Gamma \vdash a : A$ of a closed type $\vdash A$, we define the null type $\Gamma \vdash \mathrm{Null}_A(a)$ to be a cospan

$$\begin{array}{ccc} & & L(x:A) \\ & & \downarrow{\scriptstyle \varepsilon_{x:A}} \\ \Gamma & \xrightarrow[(a/x)]{} & x:A \end{array}$$

where (a/x) denotes the context morphism $\Gamma \vdash (a/x) \Rightarrow x : A$. For the elimination rule, given a type $x : A, y : \mathrm{Null}_A(x) \vdash C$, we define the term $x : A, y : \mathrm{Null}_A(x) \vdash \mathrm{elim}_{\mathrm{Null}}(x, y) : C$ to be a lifting

$$\begin{array}{ccccc} L(x:A) & \xrightarrow{\delta_{x:A}} & LL(x:A) & \xrightarrow{L(|C|\circ\varphi^{-1})} & L(C_0) \\ \uparrow{\scriptstyle \varphi} & & & & \downarrow{\scriptstyle \varepsilon_{C_0}} \\ x:A, y:\mathrm{Null}_A(x) & & \xrightarrow[|C|]{} & & C_0 \end{array}$$

where φ is the canonical isomphism. To check that the diagram commutes is routine. □

Theorem 25. *For an additive category $\mathscr{C}$ with a zero object 0, chosen pullbacks, and a class of morphisms $\mathfrak{M} \subseteq \mathrm{Mor}(\mathscr{C})$, the natural model* $\mathrm{typeof}_{\mathfrak{M}}$ *models additives.*

Proof. We define the interpretation of $a +_{(\Delta \vdash A), \sigma, \tau} b$ by the sum of the two diagrams:

$$\begin{array}{ccccc} & & & & A_1 \\ & & \overset{a}{\nearrow} & & \downarrow{\scriptstyle d_A} \\ \Gamma & \xrightarrow[\sigma]{} & \Delta & \xrightarrow[|A|]{} & A_0 \end{array}$$

$$\begin{array}{ccccc} & & & & A_1 \\ & & \overset{b}{\nearrow} & & \downarrow{\scriptstyle d_A} \\ \Gamma & \xrightarrow[\tau]{} & \Delta & \xrightarrow[|A|]{} & A_0 \end{array}$$

Zeros and inverses are interpreted similarly. □

As direct corollaries of the above two theorems, we have the following. The comonad L in some sense controls the intensionality/extensionality of the interpretation.

Corollary 26. *The category* $\mathrm{Ch}(R\mathrm{Mod})$ *with a comonad N induces a natural model that interprets our type theory. Null types are interpreted by nullhomotopies and identity types are interpreted by homotopies.*

Corollary 27. *The category* $\mathrm{Ch}(R\mathrm{Mod})$ *with the constant comonad* 0 *induces a natural model that interprets our type theory. Null types are interpreted by strict nullness and identity types are interpreted by strict equalities, i.e., the inhabitation of the type* $\mathrm{Null}_A(a)$ *asserts that* $a = 0$ *in the model and the inhabitation of the type* $\mathrm{Id}_A(a, b)$ *asserts that* $a = b$ *in the model.*

4 Related and Future Work

Hofmann and Streicher [19] showed that the identity types can be interpreted by groupoids, which implies that the *uniqueness of identity proofs* is not derivable in the syntax. Later, it was shown in [6, 43, 39] that the identity types can be interpreted in a homotopy theoretic setting, which includes the category of chain complexes as an instance. This has given rise to the study of the higher dimensional aspects of type theory, e.g., [13, 14, 38, 24, 4]. Voevodsky's simplicial sets model which validates the *univalence axiom* [41, 22, 42] enhanced the study of this field and has given rise to a line of works of HoTT and the study of reasoning about homotopy theory within HoTT, e.g., [36, 8, 40, 20]. The current most active area among such is the one using the cubical approach [11, 2, 3], which is motivated to provide computational content to univalence. They use the semantic intuition of cubical sets and introduce the *path type*, which replaces the identity type. The development of the study of spectra and (co)homology in HoTT can be found in [10, 15, 9].

In the construction of Voevodsky's simplicial sets model, universes played two distinct roles: firstly to provide a type universe and secondly to obtain coherence of the model. Lumsdaine and Warren's *local universes model* [25] disentangled those two aspects and provided a general method to obtain coherence without a universe. Awodey defined *natural model* as a general and flexible framework for interpreting type theory. He summarized and applied the local universes model to the natural model.

A natural question is whether our syntax admits other syntactical elements. Although we can check that the Σ-type is consistent because it is valid in our model, how about Π-types, universes, and higher inductive types? Since any additive category cannot have interesting type-theoretic structures, it seems hopeless for our theory in its current form to admit another syntax. The first approach to this problem is to modify the syntax to weaken the additive structures and to admit models other than additive categories. Note that the structure of abelian groups is excessive to recover identity types from null types; one only requires the "subtractions" $b - a$. The second approach is to consider *linear* Π-types since the chain complex model is closed for a different notion of monoidal categorical structure. Such a theory should borrow an idea from a *bunched logic* [30] to deal with the situation where we have another concatenation operation of contexts, namely, the one corresponding to the tensor product.

Another question is how much of the homotopy theory of chain complexes does our theory capture. Many examples showing the practical uses of the theory should be investigated. Since our model $\mathrm{Ch}(R\mathrm{Mod})$ is *stable* as a model category, another natural question is whether is it possible to extend our syntax to obtain something like *stable homotopy type theory*. In recent work [34], Riley et al. defined an extension of homotopy type theory with an axiom that asserts the stability of spectra. Pressing forward this work, we want to obtain a type theory that has stability as a theorem, not as an axiom, which is desirable from the computational point of view. Since null types are interpreted by a comonad, a natural direction is to modify our type theory in light of the *modality* and construct a syntax that is based on a line of works [31, 7, 33, 1, 37], which establishes links between modalities and homotopy type theory. It should require us to modify the judgmental structure of our type theory, in particular presentations with double contexts or different judgment forms for modal assumptions.

References

[1] Natasha Alechina, Michael Mendler, Valeria de Paiva, and Eike Ritter. Categorical and kripke semantics for constructive S4 modal logic. In Laurent Fribourg, editor, *Computer Science Logic, 15th International Workshop, CSL 2001. 10th Annual Conference of the EACSL, Paris, France, September 10-13, 2001,*

Proceedings, volume 2142 of *Lecture Notes in Computer Science*, pages 292–307. Springer, 2001. URL: https://doi.org/10.1007/3-540-44802-0_21, doi:10.1007/3-540-44802-0_21.

[2] Carlo Angiuli, Guillaume Brunerie, Thierry Coquand, Kuen-Bang Hou (Favonia), Robert Harper, and Daniel R. Licata. Syntax and models of cartesian cubical type theory. 2019. Unpublished draft. URL: https://github.com/dlicata335/cart-cube.

[3] Carlo Angiuli, Kuen-Bang Hou (Favonia), and Robert Harper. Cartesian cubical computational type theory: Constructive reasoning with paths and equalities. In Dan R. Ghica and Achim Jung, editors, *27th EACSL Annual Conference on Computer Science Logic, CSL 2018, September 4-7, 2018, Birmingham, UK*, volume 119 of *LIPIcs*, pages 6:1–6:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. URL: https://doi.org/10.4230/LIPIcs.CSL.2018.6, doi:10.4230/LIPIcs.CSL.2018.6.

[4] Peter Arndt and Krzysztof Kapulkin. Homotopy-theoretic models of type theory. In C.-H. Luke Ong, editor, *Typed Lambda Calculi and Applications - 10th International Conference, TLCA 2011, Novi Sad, Serbia, June 1-3, 2011. Proceedings*, volume 6690 of *Lecture Notes in Computer Science*, pages 45–60. Springer, 2011. URL: https://doi.org/10.1007/978-3-642-21691-6_7, doi:10.1007/978-3-642-21691-6_7.

[5] Steve Awodey. Natural models of homotopy type theory. *Mathematical Structures in Computer Science*, 28(2):241–286, 2018. URL: https://doi.org/10.1017/S0960129516000268, doi:10.1017/S0960129516000268.

[6] Steve Awodey and Michael A Warren. Homotopy theoretic models of identity types. In *Mathematical proceedings of the cambridge philosophical society*, volume 146, pages 45–55. Cambridge University Press, 2009.

[7] Nick Benton and Philip Wadler. Linear logic, monads and the lambda calculus. In *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, USA, July 27-30, 1996*, pages 420–431. IEEE Computer Society, 1996. URL: https://doi.org/10.1109/LICS.1996.561458, doi:10.1109/LICS.1996.561458.

[8] Guillaume Brunerie. On the homotopy groups of spheres in homotopy type theory. *Computing Research Repository*, abs/1606.05916, 2016. URL: http://arxiv.org/abs/1606.05916, arXiv:1606.05916.

[9] Guillaume Brunerie, Axel Ljungström, and Anders Mörtberg. Synthetic Integral Cohomology in Cubical Agda. In Florin Manea and Alex Simpson, editors, *30th EACSL Annual Conference on Computer Science Logic (CSL 2022)*, volume 216 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:19, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/opus/volltexte/2022/15731, doi:10.4230/LIPIcs.CSL.2022.11.

[10] Evan Cavallo. Synthetic cohomology in homotopy type theory, 2015.

[11] Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. Cubical type theory: A constructive interpretation of the univalence axiom. In Tarmo Uustalu, editor, *21st International Conference on Types for Proofs and Programs, TYPES 2015, May 18-21, 2015, Tallinn, Estonia*, volume 69 of *LIPIcs*, pages 5:1–5:34. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. URL: https://doi.org/10.4230/LIPIcs.TYPES.2015.5, doi:10.4230/LIPIcs.TYPES.2015.5.

[12] Peter Dybjer. Internal type theory. In Stefano Berardi and Mario Coppo, editors, *Types for Proofs and Programs, International Workshop TYPES'95, Torino, Italy, June 5-8, 1995, Selected Papers*, volume 1158 of *Lecture Notes in Computer Science*, pages 120–134. Springer, 1995. URL: https://doi.org/10.1007/3-540-61780-9_66, doi:10.1007/3-540-61780-9_66.

[13] Nicola Gambino and Richard Garner. The identity type weak factorisation system. *Theoretical computer science*, 409(1):94–109, 2008. URL: https://doi.org/10.1016/j.tcs.2008.08.030, doi:10.1016/j.tcs.2008.08.030.

[14] Richard Garner. Two-dimensional models of type theory. *Mathematical Structures in Computer Science*, 19(4):687–736, 2009. URL: https://doi.org/10.1017/S0960129509007646, doi:10.1017/S0960129509007646.

[15] Robert Graham. Synthetic homology in homotopy type theory. *Computing Research Repository*, abs/1706.01540, 2017. URL: http://arxiv.org/abs/1706.01540, arXiv:1706.01540.

[16] Allen Hatcher. *Algebraic Topology*. Algebraic Topology. Cambridge University Press, 2002. URL: https://pi.math.cornell.edu/~hatcher/AT/AT.pdf.

[17] Martin Hofmann. On the interpretation of type theory in locally cartesian closed categories. In Leszek Pacholski and Jerzy Tiuryn, editors, *Computer Science Logic, 8th International Workshop, CSL '94, Kazimierz, Poland, September 25-30, 1994, Selected Papers*, volume 933 of *Lecture Notes in Computer Science*, pages 427–441. Springer, 1994. URL: https://doi.org/10.1007/BFb0022273, doi:10.1007/BFb0022273.

[18] Martin Hofmann. Syntax and semantics of dependent types. In *Extensional Constructs in Intensional Type Theory*, pages 13–54. Springer, 1997.

[19] Martin Hofmann and Thomas Streicher. The groupoid interpretation of type theory. *Twenty-five years of constructive type theory (Venice, 1995)*, 36:83–111, 1998.

[20] Kuen-Bang Hou et al. *Higher-dimensional types in the mechanization of homotopy theory*. PhD thesis, US Army, 2017.

[21] Mark Hovey. *Model categories*. Number 63. American Mathematical Society, 2007.

[22] Chris Kapulkin and Peter LeFanu Lumsdaine. The simplicial model of univalent foundations (after voevodsky). *arXiv preprint arXiv:1211.2851*, 2012.

[23] Masaki Kashiwara and Pierre Schapira. *Categories and sheaves*. Grundlehren der mathematischen Wissenschaften; 332. Springer, 2006.

[24] Peter LeFanu Lumsdaine. Weak omega-categories from intensional type theory. *Logical Methods in Computer Science*, 6(3), 2010. URL: https://doi.org/10.2168/LMCS-6(3:24)2010, doi:10.2168/LMCS-6(3:24)2010.

[25] Peter LeFanu Lumsdaine and Michael A. Warren. The local universes model: An overlooked coherence construction for dependent type theories. *ACM Transactions on Computational Logic*, 16(3):23:1–23:31, 2015. URL: https://doi.org/10.1145/2754931, doi:10.1145/2754931.

[26] Per Martin-Löf. An intuitionistic theory of types: Predicative part. In *Studies in Logic and the Foundations of Mathematics*, volume 80, pages 73–118. Elsevier, 1975.

[27] Per Martin-Löf. On the meanings of the logical constants and the justifications of the logical laws. *Nordic journal of philosophical logic*, 1(1):11–60, 1996. URL: http://docenti.lett.unisi.it/files/4/1/1/6/martinlof4.pdf.

[28] Per Martin-Löf and Giovanni Sambin. *Intuitionistic type theory*, volume 9. Bibliopolis Naples, 1984.

[29] Bengt Nordström, Kent Petersson, and Jan M Smith. *Programming in Martin-Löf's type theory*, volume 200. Oxford University Press, 1990.

[30] Peter W. O'Hearn and David J. Pym. The logic of bunched implications. *The Bulletin of Symbolic Logic*, 5(2):215–244, 1999. URL: https://doi.org/10.2307/421090, doi:10.2307/421090.

[31] Frank Pfenning and Rowan Davies. A judgmental reconstruction of modal logic. *Mathematical Structures in Computer Science*, 11(4):511–540, August 2001. URL: https://doi.org/10.1017/S0960129501003322, doi:10.1017/S0960129501003322.

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

[33] Egbert Rijke, Michael Shulman, and Bas Spitters. Modalities in homotopy type theory. *Logical Methods in Computer Science*, 16(1), 2020. URL: https://doi.org/10.23638/LMCS-16(1:2)2020, doi:10.23638/LMCS-16(1:2)2020.

[34] Mitchell Riley, Eric Finster, and Daniel R. Licata. Synthetic spectra via a monadic and comonadic modality, 2021. arXiv:2102.04099.

[35] Robert A. G. Seely. Locally cartesian closed categories and type theory. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 95, pages 33–48. Cambridge University Press, 1984.

[36] Michael Shulman. Univalence for inverse diagrams and homotopy canonicity. *Mathematical Structures in Computer Science*, 25(5):1203–1277, 2015. URL: https://doi.org/10.1017/S0960129514000565, doi:10.1017/S0960129514000565.

[37] Michael Shulman. Brouwer's fixed-point theorem in real-cohesive homotopy type theory. *Mathematical Structures in Computer Science*, 28(6):856–941, 2018. URL: https://doi.org/10.1017/S0960129517000147, doi:10.1017/S0960129517000147.

[38] Benno van den Berg and Richard Garner. Types are weak ω-groupoids. *Proceedings of the London Mathematical Society*, 102(2):370–394, 2011.

[39] Benno van den Berg and Richard Garner. Topological and simplicial models of identity types. *ACM Transactions on Computational Logic*, 13(1), January 2012. URL: https://doi.org/10.1145/2071368.2071371, doi:10.1145/2071368.2071371.

[40] Floris van Doorn. On the formalization of higher inductive types and synthetic homotopy theory. *Computing Research Repository*, abs/1808.10690, 2018. URL: http://arxiv.org/abs/1808.10690, arXiv:1808.10690.

[41] Vladimir Voevodsky. Notes on type systems, 2011. URL: http://www.math.ias.edu/~vladimir/Site3/Univalent_Foundations_files/expressions_current.pdf.

[42] Vladimir Voevodsky. The equivalence axiom and univalent models of type theory. (talk at cmu on february 4, 2010), 2014. arXiv:1402.5556.

[43] Michael A Warren. *Homotopy theoretic aspects of constructive type theory*. PhD thesis, Carnegie Mellon University, 2008.

[44] Charles A Weibel. *An introduction to homological algebra*. Number 38. Cambridge university press, 1994.

A Basic Type Theory

This presentation is standard but included for completeness. For detail, see e.g. [26, 28, 29, 18]. The syntax is given by first indicating the *forms of judgment* [27]. Our presentation of the type theory contains the following forms of judgment:

$\Gamma \vdash$	Γ is a context
$\Gamma \vdash A$	A is a dependent type w.r.t. Γ
$\Gamma \vdash a : A$	a is a term of a type A w.r.t. Γ
$\Gamma \vdash \sigma \Rightarrow \Delta$	σ is a context morphism from Γ to Δ.

Although one often does not include the fourth judgment in the type theory as a first-class citizen, we do so for convenience.

We list the basic structural rules in Figure 6 and the rule of substitutions in Figure 7 where $\mathscr{J}$

$$\frac{}{() \vdash} \qquad \frac{\Gamma \vdash A}{\Gamma, x : A \vdash} \qquad \frac{(x : A) \in \Gamma}{\Gamma \vdash x : A}$$

$$\frac{}{\Gamma \vdash () \Rightarrow ()} \qquad \frac{\Gamma \vdash \sigma \Rightarrow \Delta \quad \Delta \vdash A \quad \Gamma \vdash a : A[\sigma]}{\Gamma \vdash (\sigma, a/x) \Rightarrow \Delta, x : A}$$

Figure 6: Structural rules

denotes any judgment of a type, a term, or a context morphism and $\mathscr{J}[\sigma]$ is a meta-notation that denotes the substitutions defined to act on expressions as usual, i.e., simultaneously replacing variables by

$$\frac{\Gamma \vdash \sigma \Rightarrow \Delta \qquad \Delta \vdash \mathscr{J}}{\Gamma \vdash \mathscr{J}[\sigma]}$$

Figure 7: Substitutions

$$\frac{\Gamma \vdash a : A, b : A}{\Gamma \vdash \mathrm{Id}_A(a,b)} \qquad \frac{\Gamma \vdash a : A}{\Gamma \vdash \mathrm{refl}_A(a) : \mathrm{Id}_A(a,a)}$$

$$\frac{\begin{array}{c}\Gamma \vdash a : A, b : A, p : \mathrm{Id}_A(a,b) \\ \Gamma, x : A, y : A, z : \mathrm{Id}_A(x,y) \vdash C \\ \Gamma, x : A \vdash c : C[x/y, \mathrm{refl}_A(x)/z]\end{array}}{\Gamma \vdash \mathrm{elim}_{\mathrm{Id}}(a,b,p,c) : C[a/x, b/y, p/z]}$$

$$\mathrm{elim}_{\mathrm{Id}}(a,a,\mathrm{refl}_A(a),c) = c[a/x]$$

Figure 8: Identity types

terms, renaming bound variables whenever necessary. The admissibility of the rule of substitutions holds throughout this paper, which can be verified by induction as usual. As an immediate corollary, we have that *exchange*, *weakening*, and *contraction* are admissible.

As in the literature, we often omit the unnecessary braces and trivial substitutions in a context morphism. For example, $t[((),(x/x)),(a/y))]$ is written as $t[a/y]$. We also write multiple judgments over the same context at once, e.g. $\Gamma \vdash a : A, b : B$, and the like.

Remark 28. *We consider types, terms, and context morphisms up to equality. Formally, one can think that we are considering four additional forms of judgment* $\Gamma = \Delta \vdash$, $\Gamma \vdash A = B$, $\Gamma \vdash a = b : A$, *and* $\Gamma \vdash \sigma = \tau \Rightarrow \Delta$, *in addition to the ones presented above and assuming the reflexivity, symmetry, transitivity, and appropriate congruence rules, which we omit.*

Example 29. *For a context* $\Gamma \vdash$, *we have an identity context morphism* $\Gamma \vdash 1_\Gamma \Rightarrow \Gamma$ *defined by*

$$1_{()} := ()$$
$$1_{\Delta,x:A} := (1_\Delta, x/x).$$

Example 30. *For a term* $\Gamma \vdash a : A$, *we have a context morphism* $\Gamma \vdash (1_\Gamma, a/x) \Rightarrow \Gamma, x : A$.

Although we will not include the identity type as a first-class citizen, we list its inference rules in Figure 8 so that we can refer to them when necessary. These four rules are called formation, introduction, elimination, and (β) rules of identity types, respectively. Subscripts are often omitted. The intuition behind these rules is that the identity type is an inductive type generated by the canonical element $\mathrm{refl}(x) : \mathrm{Id}_A(x,x)$. The elimination rule states that if we want to prove C, it suffices to prove it only when z is a canonical element, namely, $\mathrm{refl}(x)$.

The elimination rule presented in Figure 8 is sometimes called the *path induction*. The path induction requires us to replace both x and y with the same variable. However, it is sometimes easier to carry out the proof fixing x to some specific element a, which motivates us to the variation called the *based path induction* [32] (Figure 9). It is easy to check that the path induction is derivable from the based path induction, replacing Γ with $(\Gamma, x : A)$ and a with x. As for the converse, although based path induction is known to be derivable from the path induction under the assumption of other type formers (e.g. [32]), the proof without using them is not known to the author's knowledge.

$$\frac{\begin{array}{c}\Gamma \vdash a : A, b : A, p : \mathrm{Id}_A(a,b) \\ \Gamma, y : A, z : \mathrm{Id}_A(a,y) \vdash C \\ \Gamma \vdash c : C[a/y, \mathrm{refl}_A(a)/z]\end{array}}{\Gamma \vdash \mathrm{elim}_{\mathrm{Id}}{}'(a,b,p,c) : C[b/y, p/z]}$$

Figure 9: Based path induction

B Preliminaries on Chain Complexes

We start with the basic definitions concerning chain complexes. References include [44, 16, 23, 21]. In what follows, we fix a commutative ring R.

Definition 31. *Let RMod be the category of R-modules. Then, an (unbounded) chain complex over RMod is a family $(A_n)_{n\in\mathbb{Z}}$ of abelian groups together with morphisms $(\partial_n : A_n \to A_{n-1})_{n\in\mathbb{Z}}$ such that $\partial_n \circ \partial_{n+1} = 0$ for all $n \in \mathbb{Z}$.*

A chain complex A is like a space, with A_n having the n-dimensional information. Intuition is that the elements a in A_n is an n-cell on the space, and $\partial_n a$ is a "boundary" $(n-1)$-cell of a. A morphisms between chain complexes are defined as follows.

Definition 32. *A chain map between unbounded chain complexes A and B consists of a family of morphisms $(f_n : A_n \to B_n)_{n\in\mathbb{Z}}$ such that the diagram*

$$\begin{array}{ccccccc} \cdots & \xrightarrow{\partial^A_{n+1}} & A_n & \xrightarrow{\partial^A_n} & A_{n-1} & \xrightarrow{\partial^A_{n-1}} & \cdots \\ & & \downarrow{\scriptstyle f_n} & & \downarrow{\scriptstyle f_{n-1}} & & \\ \cdots & \xrightarrow{\partial^B_{n+1}} & B_n & \xrightarrow{\partial^B_n} & B_{n-1} & \xrightarrow{\partial^B_{n-1}} & \cdots \end{array}$$

commutes.

These yield an additive (moreover, abelian) category $\mathrm{Ch}(R\mathrm{Mod})$ of chain complexes over RMod. The identities and compositions are defined pointwise. Viewing an R-module M as a chain complex concentrated in degree 0, we abusively write M to denote such a chain complex. We write 0 to denote the zero R-module. It is also a zero object viewed as a chain complex. For R-modules M and N, we write $M \oplus N$ to denote their direct sum. For a family of R-modules $\{M_i\}_i$, we write $\prod_i M_i$ and $\bigoplus_i M_i$ to denote the product and the coproduct of the family, respectively. The following defines a symmetric monoidal closed structure on $\mathrm{Ch}(R\mathrm{Mod})$.

Definition 33. *For $A, B \in \mathrm{Ch}(R\mathrm{Mod})$, a tensor product $A \otimes B$ is defined to be*

$$(A \otimes B)_n := \bigoplus_{i+j=n} A_i \otimes B_j$$
$$\partial(x,y) := (\partial^A_i x, y) + (-1)^i (x, \partial^B_j y)$$

where i and j are the degree of x and y, respectively.

A unit chain complex I is defined to be

$$I := R.$$

An internal hom $[A,B]$ is defined to be

$$[A,B]_n := \prod_{j-i=n} [A_i, B_j]$$
$$\partial f := \partial^B_j \circ f - (-1)^n f \circ \partial^A_{i+1}$$

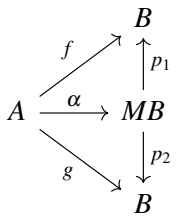
for $f \in [X_i, Y_j]$.

A *loop* and a *suspension* are defined as the following.

Definition 34. *Given a chain complex $A \in \mathrm{Ch}(R\mathrm{Mod})$, its shift by p is a chain complex $A[p]$ defined by $A[p]_n := A_{n+p}$ and $\partial^{A[p]} := (-1)^p \partial^A$. This defines a functor $[p] : \mathrm{Ch}(R\mathrm{Mod}) \to \mathrm{Ch}(R\mathrm{Mod})$. $\Omega(A) := A[1]$ and $\Sigma(A) := A[-1]$ are called loop and suspension of a chain complex A, respectively.*

Now we define the notions of a chain homotopy and a chain nullhomotopy. The fact that the following definitions are equivalent to those in [44, 16, 23] can be verified easily.

Definition 35. *Let $f, g : A \to B$ in $\mathrm{Ch}(R\mathrm{Mod})$ be chain maps. Then, a (chain) homotopy between f and g is a chain map $\alpha : A \to MB$ such that the diagram*



commutes, where $M(-) := [\mathbb{I}, -]$ is an internal hom of the interval $\mathbb{I}$, which is defined by

$$\mathbb{I}_0 := I \oplus I$$
$$\mathbb{I}_1 := I$$
$$\mathbb{I}_{n\neq 0,1} := 0$$
$$\partial^{\mathbb{I}}_1(x) := (x, -x).$$

and p_1 and p_2 are induced by the obvious inclusions $\iota_1, \iota_2 : I \to \mathbb{I}$. Or equivalently, MB can be directly defined as

$$(MB)_n := B_n \oplus B_n \oplus B_{n+1}$$
$$\partial^{MB}(x,y,z) := (\partial^B x, \partial^B y, x - y - \partial^B z)$$

and p_1 and p_2 are the projections.

Intuitively, MB is a "path space" of B, whose points are the paths on B. Then, α is a mapping that continuously assigns to each "point" a in A a "path" αa in B whose "endpoints" are fa and ga. This matches to our intuition of homotopy. Indeed, this notion of a chain homotopy is an instance of a more general concept of a *homotopy in a model category* [21]. Similarly, the notion of a nullhomotopy can be defined as the following.

Definition 36. *Let $f : A \to B$ in $\mathrm{Ch}(R\mathrm{Mod})$ be a chain map. Then, a (chain) nullhomotopy of f is a chain map $\alpha : A \to NB$ such that the diagram*

$$\begin{array}{ccc} A & \xrightarrow{\alpha} & NB \\ & \searrow^{f} & \downarrow{\scriptstyle \varepsilon_B} \\ & & B \end{array}$$

commutes, where $N(-) := [\mathbb{J}, -]$ *is an internal hom of the chain complex* $\mathbb{J}$*, which is defined by*

$$\begin{aligned} \mathbb{J}_0 &:= I \\ \mathbb{J}_1 &:= I \\ \mathbb{J}_{n\neq 0,1} &:= 0 \\ \partial_1^{\mathbb{J}} &:= \mathrm{id}_I. \end{aligned}$$

and ε_B *is induced by the obvious map* $* : I \to \mathbb{J}$*. Or equivalently, NB can be directly defined as*

$$\begin{aligned} (NB)_n &:= B_n \oplus B_{n+1} \\ \partial_n^{NB}(x,y) &:= (\partial_n^B x, x - \partial_{n+1}^B y). \end{aligned}$$

and ε_B *is the projection.*

The famility of morphisms $\{\varepsilon_B\}_B$ constitutes a natural transformation $\varepsilon : N \to 1$, which induces a comonad structure on N:

Lemma 37. *The functor* $N : \mathrm{Ch}(R\mathrm{Mod}) \to \mathrm{Ch}(R\mathrm{Mod})$ *has a comonad structure.*

Proof. The counit is ε. The comultiplication δ is defined by $(\delta_A)_n(x,y) := (x,y,y,0)$. □