Synthetic 1-Categories in Directed Type Theory

THORSTEN ALTENKIRCH, University of Nottingham, United Kingdom
JACOB NEUMANN, University of Nottingham, United Kingdom

The field of *directed type theory* seeks to design type theories capable of reasoning synthetically about (higher) categories, by generalizing the symmetric *identity* types of Martin-Löf Type Theory to asymmetric *hom-types*. We articulate the directed type theory of the *category model*, with appropriate modalities for keeping track of variances and a powerful directed-J rule capable of proving results about arbitrary terms of hom-types; we put this rule to use in making several constructions in synthetic 1-category theory. Because this theory is expressed entirely in terms of *generalized algebraic theories*, we know automatically that this directed type theory admits a syntax model and is the first step towards *directed higher observational type theory*.

Additional Key Words and Phrases: semantics, directed type theory, homotopy type theory, category theory,generalized algebraic theories

ACM Reference Format:
Thorsten Altenkirch and Jacob Neumann. 2025. Synthetic 1-Categories in Directed Type Theory. 1, 1 (July 2025), 25 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

One exciting aspect of the emergent field of *homotopy type theory (HoTT)* [28] is the observation that *types are* ∞*-groupoids* [29]. Homotopy type theory can be understood as a *synthetic* theory of ∞-groupoids: all the higher structure is generated by the simple rules for manipulating identity types in Martin-Löf Type Theory [21, 22], permitting efficient reasoning with these complex structures.

Not long after homotopy type theory was established, the search for *directed homotopy type theory*—a synthetic theory of (higher) *categories*—began. In a directed type theory, the identity types of ordinary Martin-Löf Type Theory (which are provably symmetric in the theory, i.e. a witness $p : \mathsf{Id}(t, t')$ can be turned into $p^{-1} : \mathsf{Id}(t', t)$) are replaced by asymmetric *hom-types*. However, building a type theory to effectively work with these hom-types is beset by difficulties, in particular the need to carefully track the *variances* of terms. A common feature of many approaches to directed type theory (e.g. [20, 23, 24]) is to track these variances by adopting some kind of modal typing discipline. However, no consensus ever emerged for exactly how to do this. More recent approaches to directed type theory (such as the work of Riehl and Shulman [25]) avoid these issues by adopting a more indirect approach inspired by simplicial spaces, at the cost of a more elaborate, multi-layered theory.

A possible new approach to directed type theory seems to be on the horizon, drawing from the recent development of *higher observational type theory* [4, 5]. Higher observational type theory, or H.O.T.T., seeks to strike a balance between the properties of "Book HoTT" (as originally articulated in [28]) and cubical type theory [10], particularly with regards to the central axiom of HoTT,

Authors' addresses: Thorsten Altenkirch, University of Nottingham, Nottingham, United Kingdom, thorsten.altenkirch@nottingham.ac.uk; Jacob Neumann, University of Nottingham, Nottingham, United Kingdom, jacob.neumann@nottingham.ac.uk.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2025 Association for Computing Machinery.
XXXX-XXXX/2025/7-ART $15.00
https://doi.org/XXXXXXX.XXXXXXX

Voevodsky's *univalence axiom*. In Book HoTT, identity types are defined inductively, which results in the univalence axiom being impossible to compute with—complicating a key attribute of HoTT, its amenability to computer formalization. Cubical type theory rectifies this situation by instead defining identity types in terms of a special *interval* type, and adding enough machinery to make univalence computable as a *theorem* rather than an axiom (though at the expense of Book HoTT's intuitive simplicity). Higher observational type theory seeks to build homotopy type theory around *observational* identities, in particular turning univalence into a *definition*, preserving both the computational and intuitive character of homotopy type theory. H.O.T.T.'s definitional univalence suggests a clear directed analogue, making *directed higher observational type theory* an appealing prospect. Though H.O.T.T. remains to be fully worked out, it's clear that *second-order generalized algebraic theories (SOGATs)* [7, 17, 26, 27] provide the appropriate setting for formulating this theory, as the language of SOGATs provide a *higher-order abstract syntax* ideal for handling formal languages with variable binders.

Ordinary (first-order) generalized algebraic theories (GATs) [8] have played a prominent role in the semantics of type theory: Dybjer's *categories with families (CwFs)* [11] are a GAT articulating the basic mechanics of type theory, and provide a highly flexible and modular approach to interpreting type theories. Articulating the semantics of type theory as a GAT comes with numerous advantages: the universal algebraic features of GATs—such as homomorphisms, displayed models, products and coproducts of models, free and cofree models—are well-understood [18]. In particular, every GAT has an initial model, the *syntax model*, which can be constructed as a quotient inductive-inductive type [16]. Finally, GATs have the advantage of being (relatively) straightforward to formalize, as they make all the relevant operations and equations explicit. Anticipating SOGAT and higher observational treatments of directed type theory, we begin by articulating directed type theory as a generalized algebraic theory.

1.1 Related Work

The present work draws most closely from Hofmann and Streicher's work on the groupoid model [15]; in particular, we develop a directed analogue of the groupoid model—the *category model*—and adapt the groupoid models main constructs (dependent types and identity types) to the directed setting. We also closely follow the kind of metatheoretic arguments made there, and develop a directed analogue of their *universe extensionality*, an early articulation of univalence. We also draw from the closely-related *setoid model* of [2, 13].

Among directed type theories, the present work draws some constructs from the theory of Licata and Harper [20], particularly their treatment of the *opposite category* construction as a modality on contexts and context extension, as well as their treatment of Π-types in the directed setting. Our *directed $\mathcal{J}$-rule* for eliminating hom-types is similar to one of the eliminators given by North [23], though with the critical difference mentioned below. Our theory, like North's, is "1-dimensional" in the sense of Licata-Harper in that we maintain *judgmental* equality as a symmetric notion, as opposed to "2-dimensional" theories [1, 20, 24] which introduce a theory of *directed reductions*. All these theories adopt a modal typing discipline for handling variances, as do we, unlike the theories of [19, 25, 31] and [30], which adopt approaches akin to simplicial and cubical type theories, respectively.

The commonality between the setoid, groupoid, preorder, and category model described in Section 2 is made much more general [18][16], where it is shown that any GAT gives rise to a CwF of algebras, algebra morphisms, displayed algebras, and sections. The *modal* features of the present work are most likely instances of the modal type theory of [12].

1.2 Contribution and Organization

We articulate a directed type theory satisfying the following constraints.

(1) It is presented as a generalized algebraic theory.
(2) It is 1-dimensional in the sense of Licata-Harper: there are no 'directed reductions' introduced judgmentally.
(3) It is *deeply-polarized*: there is a modal typing discipline to keep track of variances, which operates not just on types but on contexts, substitutions, and context extension.
(4) The directed J-rule (directed path induction) permits reasoning about arbitrary terms of hom-types.
(5) Hom-types can be iterated,[1] expressing synthetic higher categorical structure (though in the present work we only consider 1-category theoretic structure).

To our knowledge, there is no existing type theory satisfying all these criteria.

Starting with Section 2, we adopt a semantics-driven approach by investigating a particular model, the *category model* and abstracting its key features into a series of abstract notions of model (Section 3). These notions are all GATs (indeed, CwFs with additional structure), and therefore each give rise to a syntax model. Our main notion is that of a *Directed CwF (DCwF)*, a generalized algebraic theory of directed types with adequate polarity structure to properly track variances.

Achieving (4) while maintaining a modal typing discipline requires a novel approach. In the typing rules of existing directed type theories (including ours), the endpoint terms t and t' of a hom-type $\mathsf{Hom}(t, t')$ are assigned opposite variances: t negative and t' positive. However, this poses a difficulty for typing the identity morphism $\mathsf{refl}_t : \mathsf{Hom}(t, t)$ since t must assume both variances. North [23] solves this by restricting t to be a term of a *core type* (interpreted semantically by groupoids), but the consequent J-rule only operates on hom-terms with a core endpoint, not arbitrary ones. Our solution instead uses groupoid *contexts* rather than groupoid *types*.

In Section 4, we show that this is a viable framework for conducting synthetic category theory. In this section, we adopt an informal style reminiscent of [28], showing how this theory can be operated and how the groupoid context can be carefully maintained by a simple syntactic rule. We use our directed J-rule to give several basic constructions in synthetic (1-)category theory.

Finally, we consider the directed universe of sets in the category model, which serves as the category of sets. The existence of a directed universe allows us to make the metatheoretic argument that the syntax of DCwFs cannot prove the symmetry of hom-types (i.e. this is a genuinely *directed* type theory) or the uniqueness of homs (analogous to Hofmann and Streicher's proof that the groupoid model refutes the uniqueness of identity proofs). We conclude by sketching several possible routes for further study.

1.3 Metatheory and Notation

Throughout, we work in an informal type-theoretic metatheory, using pseudo-Agda notation to specify GATs, make category-theoretic constructions, and define terms in the syntax of Directed CwFs. We use the notations

$$(x : X) \to P(x) \qquad \text{and} \qquad (x : X) \times P(x)$$

for the dependent function and dependent sum types, respectively. When defining dependent functions, we'll enclose arguments in curly brackets to indicate that they're implicit. Any variables appearing free are also assumed to be implicitly universally quantified. We sometimes use underscores to indicate where the arguments to a function are written. When defining an instance T of a construct given as a **record** type, we'll often omit the names of specific components, referring to

[1] In contrast to e.g. [20], where homs-between-homs and homs-between-homs-between-homs is not possible to express.

all of them as just T (matching the category-theoretic convention of referring to both the object- and morphism-parts of a functor F by just F).

We use $=$ to mean *definitional* or *judgmental* equality in our metatheory, whereas $\equiv$ means *propositional* equality (though there's no reason they couldn't coincide, i.e. in an extensional metatheory). We tacitly make use of appropriate extensionality principles (particularly function extensionality) for both notions of equality, and the uniqueness of identity proofs for $\equiv$. We write Prop for the type of h-propositions in our metatheory, i.e. those P such that $p \equiv p'$ for all $p, p' : P$.

We assume basic familiarity with category theory. The set of objects of a category Γ is denoted $|\Gamma|$, the set of Γ-morphisms from γ_0 to γ_1 is denoted $\Gamma\ [\gamma_0, \gamma_1]$, and identities are written as id. The *discrete* groupoid/category on a set X is the category whose objects are elements of X and whose morphisms from x_0 to x_1 are inhabitants of the identity type $x_0 \equiv x_1$. The *opposite category* construction is understood to be definitionally involutive, i.e. $|\Gamma^{\mathrm{op}}|$ is defined to be $|\Gamma|$ and $\Gamma^{\mathrm{op}}\ [\gamma_0, \gamma_1]$ is defined to be $\Gamma\ [\gamma_1, \gamma_0]$, and thus

$$(\Gamma^{\mathrm{op}})^{\mathrm{op}} = \Gamma.$$

2 The Category Interpretation of Type Theory

As mentioned, generalized algebraic theories (GATs) are a desirable formalism for expressing models of type theory, particularly when modelling numerous extensions to a 'basic' type theory. When a theory is given as a GAT, all operations and equations are made clear and explicit, making it easier to compare and contrast similar theories. The theory of **Categories with Families (CwFs)** (originally defined by Dybjer [11]) present the fundamental operations of type theory—contexts, variables, terms, types, and substitutions—encoded as a GAT; upon this basic framework, an endless variety of different type theories can be studied.

The main components of a CwF are given in Fig. 1: a category Con of *contexts*, whose morphisms are called *substitutions*; a presheaf Ty on Con and a dependent presheaf Tm over Ty; and a *context extension* operation guaranteeing that Tm is *locally representable* (in the sense of [7]). The last line says that there is an isomorphism (natural in Δ) between the type of pairs (σ, t) with $\sigma : \mathsf{Sub}\ \Delta\ \Gamma$ and $t : \mathsf{Tm}(\Delta, A[\sigma])$ and the type of substitutions from Δ to $\Gamma \triangleright A$. The left-to-right direction of this isomorphism is denoted $\langle \_, \_ \rangle$ and the opposite direction as $\mathsf{p} \circ \_, \mathsf{v}[\_]$, so

$$\tau \equiv \langle \mathsf{p} \circ \tau, \mathsf{v}[\tau] \rangle \qquad \text{and} \qquad \sigma \equiv \mathsf{p} \circ \langle \sigma, t \rangle \qquad \text{and} \qquad t \equiv \mathsf{v}[\langle \sigma, t \rangle]$$

for any σ, t as above and $\tau : \mathsf{Sub}\ \Delta\ (\Gamma \triangleright A)$.

Two paradigm examples of CwFs are the *setoid model* of [2, 13] and the *groupoid model* of [15]. In the former, the contexts are setoids (i.e. sets equipped with equivalence relations), the types are families of setoids (functorially) indexed over their context setoid, and terms are given by the appropriate notion of *section* of their type (see the Coq formalization of [3] for a precise definition). The groupoid model is quite similar: contexts are groupoids, types are families of groupoids functorially indexed over their context groupoid, and terms are the appropriate notion of section. Indeed, we can view the groupoid model as generalizing the setoid model: a setoid can be viewed as a groupoid whose hom-sets are *subsingletons* (or *propositions*, in the terminology of [28]), sets with at most one element. In other words, the groupoid model is what results when the assumption of *proof-irrelevance* is dropped from the setoid model.

Both these models provide interpretations for numerous type formers, in particular the dependent types, identity types, and universes characteristic of Martin-Löf Type Theory [21, 22]. The difference in these models is reflected in the type theories they interpret: while both models permit arbitrary iteration of the identity type former (expressing identities between identities, and identities between identities between identities, and so on), these iterated identity types become trivial more quickly in the setoid model. More precisely, the setoid model validates the *uniqueness of identity proofs*

```
record CwF : Set where
  field
  -- Category of contexts
  Con : Set
  Sub : Con → Con → Set
  id : Sub Γ Γ
  _∘_ : Sub Δ Γ → Sub Θ Δ → Sub Θ Γ

  -- The empty context (terminal object)
  • : Con
  ! : (Γ : Con) → Sub Γ •

  -- Presheaf of types
  Ty : Con → Set
  _[_] : Ty Γ → Sub Δ Γ → Ty Δ

  -- Presheaf of terms
  Tm : (Γ : Con) → Ty Γ → Set
  _[_] : Tm(Γ,A) → (σ : Sub Δ Γ) → Tm(Δ, A[σ])

  -- Context extension
  _▷_ : (Γ : Con) → Ty Γ → Con
  ⟨_,_⟩ : (σ : Sub Δ Γ) × Tm(Δ, A[σ])  ≅  Sub Δ (Γ▷A) : (p∘_, v[_])
```

Fig. 1. Main components of a CwF

principle, meaning that any two terms of an identity type, $p, q : \mathsf{Tm}(\Gamma, \mathsf{Id}(x, y))$ are themselves identical, $\mathsf{UIP}(p, q) : \mathsf{Tm}(\Gamma, \mathsf{Id}(p, q))$. The groupoid model famously violates this principle: in the type theory of the groupoid model, there are types (in particular, the universe of sets) which are not *h-sets*, i.e. they possess terms which are proved identical by multiple, *distinct* identity proofs.

This provides a roadmap for how we might develop a model of directed type theory. Since directed type theory can be described as "dependent type theory, but with *asymmetric* identity types", this leads us to suspect that models of directed type theory will result if we simply drop the assumption of symmetry from the setoid and groupoid models. A setoid without symmetry is a preorder, and a groupoid without symmetry is a category. A close inspection of the definition of the groupoid model reveals that nothing in its interpretation of *just the CwF structure* requires symmetry (i.e. that morphisms are invertible), and thus we can define the preorder model of type theory and the **category model of type theory**. The category model is given by Fig. 2, minus the definition of context extension (which will be discussed more below). This is just a generalization of the groupoid model, obtained by dropping symmetry: contexts are categories (and substitutions are functors), types are families of categories, and terms are sections. We wont focus on the preorder model here, but leave it to future work to develop the directed analogue of setoid-model-specific considerations. Instead, we'll highlight those features of the category model which are relevant for modelling directed type theory, before abstracting those features into the notion of a *directed CwF* in the next section.

```
Con = Cat

Sub Γ Δ = Cat [ Γ, Δ ]

• : Cat
• = 𝟙  -- the singleton category, with one object, ∗, and only the identity morphism

-- A : Ty Γ means A : Γ → Cat
record Ty (Γ : Con) : Set where
 field
  obj : |Γ| → Cat
  map : Γ [ γ₀, γ₁ ] → Cat [ obj γ₀, obj γ₁ ]
  fid : map (id_γ) ≡ id_obj(γ)
  fcomp : map (γ₁₂ ∘ γ₀₁) ≡ (map γ₁₂) ∘ (map γ₀₁)

record Tm (Γ : Con) (A : Ty Γ) : Set where
 field
  obj : (γ : |Γ|) → |A(γ)|
  map : (γ₀₁ : Γ [ γ₀, γ₁ ]) → (A γ₁) [A γ₀₁ (obj γ₀), obj(γ₁)]
  fid : map (id_γ) ≡ id_obj(γ)
  fcomp : map (γ₁₂ ∘ γ₀₁) ≡ (map γ₁₂) ∘ (A γ₁₂ (map γ₀₁))
```

Fig. 2. The CwF structure of the category model, excluding context extension.

While the basic CwF structure of the groupoid model doesn't require symmetry (i.e. that all morphisms are invertible), its interpretations of further type formers certainly do. After all, our hope is that by passing from the groupoid model to the category model, the symmetric identity types of the former will become asymmetric *hom-types* in the latter. Consider the semantics of the identity type former in the groupoid model. Here, and henceforth, we define a type (in this case $\mathsf{Id}(t, t')$) by giving its object- and morphism-parts, which are denoted obj and map in Fig. 2, but here are both written as just $\mathsf{Id}(t, t')$.

```
-- Taken from [15, Section 4.10]
Id : Tm(Γ, A) → Tm(Γ, A) → Ty Γ
 (Id(t,t')) γ = (A γ) [t γ, t' γ]      -- Discrete groupoid

 (Id(t,t')) (γ₀₁ : Γ[ γ₀ , γ₁ ]) : (A γ₀) [t γ₀, t' γ₀] → (A γ₁) [t γ₁, t' γ₁]
 (Id(t,t')) γ₀₁ x₀ = (t' γ₀₁) ∘ (A γ₀₁ x₀) ∘ (t γ₀₁)⁻¹
```

Here, the fact that $A(\gamma_1)$ is a *groupoid* is used in an essential way (we must take the inverse of $t'(\gamma_{01})$), and hence this definition doesn't work in the category model. But notice the following: the term t is in the "negative" position (the domain) and the term t' is in the "positive" position. Fittingly, we only use the *inverse* of $t(\gamma_{01})$—never $t(\gamma_{01})$ itself—and only use $t'(\gamma_{01})$ but not its inverse. This observation will provide the key to adapting this definition for the category model.

What is needed is for t to be a *contravariant* term of type A, while keeping t' as *covariant*. This difference can be articulated in the category model, using a fundamental construct from category

theory: *opposite categories*. A type $A : \mathsf{Ty}\ \Gamma$ in the category model consists of a family of categories $A(\gamma)$ for each object $\gamma : |\Gamma|$ and a functor $A(\gamma_{01}) : \mathsf{Cat}[A(\gamma_0), A(\gamma_1)]$ for each morphism $\gamma_{01} : \Gamma[\gamma_0, \gamma_1]$. Given such a family of categories A, we can form a new family A^-, where $A^-(\gamma)$ is defined as the opposite category of $A(\gamma)$. This extends to the morphism part as well, because any functor $f : \mathsf{Cat}\ [C, D]$ can be viewed as a functor on their opposites, $f : \mathsf{Cat}\ [C^{\mathrm{op}}, D^{\mathrm{op}}]$. Alternatively, we could view A as a functor $\Gamma \to \mathsf{Cat}$, and define A^- to be the composition of A with the endofunctor $(\_)^{\mathrm{op}} : \mathsf{Cat} \to \mathsf{Cat}$. We can state generally that the category model validates the following rule:

$$\frac{A : \mathsf{Ty}\ \Gamma}{A^- : \mathsf{Ty}\ \Gamma.}$$

If $t : \mathsf{Tm}(\Gamma, A^-)$, this means that the object part of t will still send objects $\gamma : |\Gamma|$ to objects of $A(\gamma)$, since $A(\gamma)$ and $A^-(\gamma)$ have the same objects. But observe the type of its morphism part:

$$t : (\gamma_{01} : \Gamma\ [\gamma_0, \gamma_1]) \to (A\ \gamma_1)\ [t\ \gamma_1, A\ \gamma_{01}\ (t\ \gamma_0)].$$

This is precisely what we need to articulate the definition of hom-types in the category model: see Fig. 3. This definition is almost exactly the same as the semantics of Id in the groupoid model, but with t changed to be a term of A^-, thus eliminating the need for the categories $A(\gamma_i)$ to be groupoids. Here's the hom-type formation, expressed as a rule:

$$\frac{t : \mathsf{Tm}(\Gamma, A^-) \quad t' : \mathsf{Tm}(\Gamma, A)}{\mathsf{Hom}(t, t') : \mathsf{Ty}\ \Gamma.}$$

The type annotation of t as a "negative" term and the implicit annotation of t' as "positive" serve as a kind of *modal typing discipline* for keeping track of the *variances* of terms.

```
Hom : Tm(Γ, A⁻) → Tm(Γ, A) → Ty Γ
  (Hom(t,t')) γ = (A γ) [t γ, t' γ] -- Discrete category

  (Hom(t,t')) (γ01 : Γ[ γ0 , γ1 ]) : (A γ0) [t γ0, t' γ0] → (A γ1) [t γ1, t' γ1]
  (Hom(t,t')) γ01 x0 = (t' γ01) ∘ (A γ01 x0) ∘ (t γ01)
```

Fig. 3. Semantics of the Hom-type former in the category model

For now, we just state the formation rule for hom-types; introducing and eliminating terms of hom-types will require more machinery. To see what kind of machinery, let's instead consider dependent function types. Like with the formation of hom-types, Π-types involve positive and negative "variance": a function is contravariant in its argument and covariant in its result. Therefore, as we might expect, the interpretation of Π-types in the groupoid model ([15, Section 4.6]) makes essential use of the invertability of morphisms in a groupoid. Again, it only comes into play when defining the *morphism* part of the interpretation: the object part (reproduced in Fig. 5) defines for each $\gamma : |\Gamma|$ an auxiliary type B_γ in context $A(\gamma)$, and then specifies the category $\Pi(A, B)\ \gamma$ with terms $\theta : \mathsf{Tm}(A(\gamma), B_\gamma)$ as objects. This works fine in the category model. However, defining the morphism part of $\Pi(A, B)$ requires a kind of negative variance *deeper* than the shallow contravariance of A^-: in the type $\mathsf{Hom}(t, t')$ it was a *term* that occurred negatively (t), in the type $\Pi(A, B)$ it's a *type* that occurs negatively.

To make sense of this, we must consider the *opposite category* operation, not just as acting on each $A(\gamma)$ in a family of categories over a context Γ, but as acting on the contexts themselves. In

the category model, we have the following rules.

$$\frac{\Gamma : \mathsf{Con}}{\Gamma^- : \mathsf{Con}} \qquad \frac{\sigma : \mathsf{Sub}\ \Delta\ \Gamma}{\sigma^- : \mathsf{Sub}\ \Delta^-\ \Gamma^-}$$

That is, we can negate contexts and substitutions as well as types: Γ^- is interpreted as Γ^{op}, and this operation is (*co*variantly) lifted onto functors as before. Now consider the difference in the morphism parts of terms with these different kinds of variance.

```
-- t : Tm(Γ, A) where A : Ty Γ
t γ01 : (A γ1) [ A γ01 (t γ0), t γ1 ]
-- t : Tm(Γ, A⁻) where A : Ty Γ
t γ01 : (A γ1) [ t γ1, A γ01 (t γ0) ]
-- t : Tm(Γ⁻, A) where A : Ty Γ⁻
t γ01 : (A γ0) [ t γ0, A γ01 (t γ1) ]
```

This is why we referred to this as "shallow" and "deep" negation: the difference between the first two is that we've flipped around each $A(\gamma)$, whereas in the third term, the dependence of A on Γ has itself been flipped around (A is now contravariant, so $A\ \gamma_{01}$ takes objects of $A(\gamma_1)$ to objects of $A(\gamma_0)$). It is this latter kind of contravariance that describes A's position in $\Pi(A, B)$.

We need another ingredient to state the Π-type formation rule: *negative context extension*. In the type $\Pi(A, B)$, A appears negatively, i.e. we want A to depend negatively on Γ, i.e. $A : \mathsf{Ty}(\Gamma^-)$; but B appears positively—we want B to depend *covariantly* on Γ, plus a variable of type A. In the theory of CwFs, a type "depending on a variable" of another type is encoded by context extension. In the category model, we in fact have *two* context extension operations (see Fig. 4), corresponding to the two ways a type can depend on a context. The positive context extension operator, $\triangleright^+$, obeys the usual isomorphism discussed above. For the negative extension, the isomorphism becomes:

$$(\sigma : \mathsf{Sub}\ \Delta\ \Gamma) \times (\mathsf{Tm}(\Delta^-, A[\sigma^-]^-)) \cong \mathsf{Sub}\ \Delta\ (\Gamma \triangleright^- A) \tag{1}$$

for any $A : \mathsf{Ty}\ \Gamma^-$. We'll write $\langle \_,\_\ \_ \rangle$ for the left-to-right direction of this isomorphism, and write $\mathsf{p}_{-,A} : \mathsf{Sub}\ (\Gamma \triangleright^- A)\ \Gamma$ and $\mathsf{v}_{-,A} : \mathsf{Tm}((\Gamma \triangleright^- A)^-, A[\mathsf{p}^-_{-,A}]^-)$ for the data obtained from applying the right-to-left direction to the identity morphism on $\Gamma \triangleright^- A$. With this, we have everything needed to give the semantics of the Π-type former; this is done in Fig. 5. As expected for Π-types, we have an isomorphism between $\mathsf{Tm}(\Gamma, \Pi^+(A, B))$ and $\mathsf{Tm}(\Gamma \triangleright^- A, B)$, the application and lambda-abstraction rules. This is omitted here for reasons of space, but included in the appendix with accompanying calculations; see Fig. 13.

```
_▷⁺_ : (Γ : Con) → Ty Γ → Con
  |Γ ▷⁺ A| = (γ : |Γ|) × |A γ|
  (Γ ▷⁺ A) [ (γ0, a0) , (γ1, a1) ] = (γ01 : Γ [ γ0, γ1 ]) × (A γ1) [ A γ01 a0, a1 ]

_▷⁻_ : (Γ : Con) → Ty Γ⁻ → Con
  |Γ ▷⁻ A| = (γ : |Γ|) × |A γ|
  (Γ ▷⁻ A) [ (γ0, a0) , (γ1, a1) ] = (γ01 : Γ [ γ0, γ1 ]) × (A γ1) [ a0 , A γ01 a1 ]
```

Fig. 4. Semantics of context extension in the category model

```
Π : (A : Ty Γ⁻) → Ty(Γ ▷⁻ A) → Ty Γ
 |Π(A,B) γ| = Tm(A(γ), B_γ)
  where
   B_γ : Ty(A γ)
    B_γ a = B(γ,a)
    B_γ (x : (A γ)[ a , a' ]) = B(id_γ,x)

 record (Π(A,B) γ)[_,_] : (θ θ' : Tm(A(γ), B_γ)) → Set where
    component : (a : |A γ|) → (B(γ,a)) [ θ a , θ' a ]
    naturality : (x : (A γ) [ a , a' ]) → (θ' x) ∘ B(γ,x)(component a) ≡ (component a') ∘ (θ x)

 Π(A,B) γ01 : Tm(A(γ0), B_γ0) →  Tm(A(γ1), B_γ1)
  (Π(A,B) γ01 θ0) (a1 : |A γ1|) = B(γ01, id_(A γ01 a1)) (θ0(A γ01 a1))
  (Π(A,B) γ01 θ0) (x1 : A γ1 [a1 , a1']) = B(γ01, id_(A γ01 a1') (θ0(A γ01 x1))
```

Fig. 5. Semantics of the Π-type former in the category model

Let's now return to the key feature of directed type theory, hom-types. Above, we gave just the formation rule for hom-types, but said nothing of how to introduce or eliminate terms of this type. Stating the introduction rule, the term refl inhabiting $\mathsf{Hom}(t, t)$ for each term t proves rather subtle. The difficulty stems from the mixed-variance problem mentioned in the introduction: since our formation rule demands the domain term t be of type A^- and the codomain term t' to be of type A, it's not immediately clear how to make $\mathsf{Hom}(t, t)$ well-formed. There is, in general, no way to coerce terms of type A into terms of type A^- or vice versa, and we have no rule permitting us to use a term in both variances.

As mentioned in the introduction, the solution to this problem presented in [23] is to use *core types*. This solution consists of asserting a new type A^0 for each A, equipped with coercions $\mathsf{Tm}(\Gamma, A^0) \to \mathsf{Tm}(\Gamma, A)$ and $\mathsf{Tm}(\Gamma, A^0) \to \mathsf{Tm}(\Gamma, A^-)$. Then, for a term $t : \mathsf{Tm}(\Gamma, A^0)$, it makes sense to write $\mathsf{Hom}(t, t)$, as t can be coerced to both the positive and negative modality, in order to fit the Hom formation rule. From there, a directed J-rule can be stated for eliminating hom terms. The issue with this solution is that it forces homs to have core endpoints: the directed J-rule can *only* be used to prove claims about homs anchored at a core term, and it's not clear that proofs about arbitrary homs can be made. For the synthetic category-theoretic claims we study below, this will prove to be an unacceptable restriction.

A solution which avoids this shortcoming is revealed by considering hom-types in the empty context. In the empty context, a type A is the same thing as a category, and a term of type A is the same thing as an object of type A (we silently coerce between $\mathbb{1} \to X$ and X). So then there's no difference between terms of type A and terms of type A^-, since a category and its opposite have the same objects. Therefore, in the empty context, there is no mixed-variance problem, and we can state the introduction rule for refl_t simply by coercing t to be positive and negative as needed.

This doesn't extend to arbitrary contexts: as we saw above, terms $t : \mathsf{Tm}(\Gamma, A^-)$ and $t' : \mathsf{Tm}(\Gamma, A)$ have different morphism parts. But here's the key observation: if Γ is a *groupoid*, then we can still coerce between A and A^-: given $t : \mathsf{Tm}(\Gamma, A^-)$, we can obtain $-t : \mathsf{Tm}(\Gamma, A)$, and vice-versa. The definition is given in Fig. 6; there (and henceforth), we use $\Gamma : \mathsf{NeutCon}$ to indicate that Γ is a groupoid, and therefore can invert Γ-morphisms as needed. So, rather than introduce a new

type A^0 whose terms can be either positive or negative, we have instead have identified those contexts—neutral contexts—where terms of the familiar types A and A^- can be inter-converted. Given this, we can introduce refl:

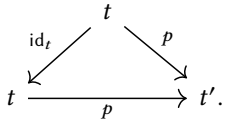
$$\frac{\Gamma : \mathsf{NeutCon} \quad A : \mathsf{Ty}\ \Gamma \quad t : \mathsf{Tm}(\Gamma, A^-)}{\mathsf{refl}_t : \mathsf{Tm}(\Gamma, \mathsf{Hom}(t, -t)).}$$

We only need to assert refl_t for t of type A^-, because the analogous rule for t' of type A can be derived: given $t' : \mathsf{Tm}(\Gamma, A)$, we observe that $t' = -(-t')$, so $\mathsf{refl}_{-t'} : \mathsf{Tm}(\Gamma, \mathsf{Hom}(-t', t'))$.

```
– : {Γ : NeutCon}{A : Ty Γ} → Tm(Γ,A) → Tm(Γ,A⁻)
 –t' γ = t' γ
 –t' γ₀₁ = A γ₀₁ (t'(γ₀₁⁻¹))
– : {Γ : NeutCon}{A : Ty Γ} → Tm(Γ,A⁻) → Tm(Γ,A)
 –t γ = t γ
 –t γ₀₁ = A γ₀₁ (t(γ₀₁⁻¹))
```

Fig. 6. Semantics of neutral-context coercion in the category model

Let's conclude this section by giving an eliminator for our hom-type, known as the *directed J-rule* or *directed path induction*. Following [15, Section 4.10], we study directed path induction in the empty context first, which can then be extended to an arbitrary neutral context. Given $A : \mathsf{Ty}\bullet$ and $t : \mathsf{Tm}(\bullet, A^-)$ and some $M : \mathsf{Ty}(\bullet \triangleright^+ A \triangleright^+ \mathsf{Hom}(t, \mathsf{v}))$, our goal is to be able to prove $M[t', p]$ for arbitrary t' and p, just by supplying a term m of $M[-t, \mathsf{refl}_t]$. Translated into the category model semantics: A is a category, t and t' are objects of A, M is a functor from the coslice category t/A into Cat, p is an A-morphism from t to t', and m is an object of the category $M(t, \mathsf{id}_t)$. The key observation is that p is then *also* a morphism in the coslice category from (t, id) to (t', p).



Therefore,

$$M(p) : \mathsf{Cat}\ [M(t, \mathsf{id}_t), M(t', p)]$$

and so the object part of this functor turns objects of $M(t, \mathsf{id}_t)$ into objects of $M(t', p)$, that is, it turns terms $m : \mathsf{Tm}(\bullet, M[-t, \mathsf{refl}])$ into terms

$$(\mathsf{J}_{t,M}\ m)\ [t', p] : \mathsf{Tm}(\bullet, M[t', p]).$$

And, since $M(\mathsf{id})$ is the identity functor, we have the β law, saying that $\mathsf{J}_{t,M}\ m\ [-t, \mathsf{refl}_t] \equiv m$. The general law replaces $\bullet$ with an arbitrary neutral context:

$$\frac{\begin{array}{cc}\Gamma : \mathsf{NeutCon} & A : \mathsf{Ty}\ \Gamma \\ t : \mathsf{Tm}(\Gamma, A^-) & M : \mathsf{Ty}(\Gamma \triangleright^+ A \triangleright^+ \mathsf{Hom}(t[\mathsf{p}_A], \mathsf{v})) \\ m : \mathsf{Tm}(\Gamma, M[-t, \mathsf{refl}_t]) & \end{array}}{\mathsf{J}_{t,M}\ m\ : \mathsf{Tm}(\Gamma \triangleright^+ A \triangleright^+ \mathsf{Hom}(t[\mathsf{p}_A], \mathsf{v}), M)} \tag{2}$$

but the category model interpretation—see Fig. 7—essentially follows this same idea. If M doesn't need to depend on the term of type $\mathsf{Hom}(t, \mathsf{v})$, then we can instead use the simpler rule

$$\frac{\begin{array}{cc}\Gamma\colon \mathsf{NeutCon} & A\colon \mathsf{Ty}\ \Gamma \\ t\colon \mathsf{Tm}(\Gamma, A^-) & M\colon \mathsf{Ty}(\Gamma \triangleright^+ A) \quad m\colon \mathsf{Tm}(\Gamma, M[-t])\end{array}}{\mathsf{J}_{t,M}\ m\ :\ \mathsf{Tm}(\Gamma \triangleright^+ A \triangleright^+ \mathsf{Hom}(t[\mathsf{p}_A], \mathsf{v}), M[\mathsf{p}_{\mathsf{Hom}(t[\mathsf{p}_A],\mathsf{v})}])} \tag{3}$$

Note that the dependence on $\mathsf{Hom}(t[\mathsf{p}_A], \mathsf{v})$ is preserved in the conclusion, even if M ignores it. In Section 4 we put this rule to use in synthetic category theory constructions and proofs.

$$\begin{array}{l}
\mathsf{J} : (\mathsf{t} : \mathsf{Tm}(\Gamma, \mathsf{A}^-)) \to (\mathsf{M} : \mathsf{Ty}\ (\Gamma \triangleright^+ \mathsf{A} \triangleright^+ \mathsf{Hom}(\mathsf{t}[\mathsf{p}_A],\mathsf{v}))) \\
\quad \to \mathsf{Tm}(\Gamma, \mathsf{M}[-\mathsf{t},\mathsf{refl}_\mathsf{t}]) \to \mathsf{Tm}(\Gamma \triangleright^+ \mathsf{A} \triangleright^+ \mathsf{Hom}(\mathsf{t}[\mathsf{p}_A],\mathsf{v}), \mathsf{M}) \\
\quad (\mathsf{J}_{\mathsf{t},\mathsf{M}}\ \mathsf{m}) : (\gamma : |\Gamma|) \to (\mathsf{a} : |\mathsf{A}\ \gamma|) \to (\mathsf{x} : (\mathsf{A}\ \gamma)\ [\mathsf{t}\ \gamma\ ,\ \mathsf{a}]) \to |\ \mathsf{M}(\gamma, \mathsf{a}, \mathsf{x})\ | \\
\quad (\mathsf{J}_{\mathsf{t},\mathsf{M}}\ \mathsf{m})\ \gamma\ \mathsf{a}\ \mathsf{x} = \mathsf{M}\ (\mathsf{id}_\gamma, \mathsf{x}, \rho)\ (\mathsf{m}\ \gamma)\ \ \mathit{-- \rho : x \circ A\ id_\gamma\ id_{\mathsf{t}\gamma} \circ t\ id_\gamma \equiv x} \\
\\
\quad (\mathsf{J}_{\mathsf{t},\mathsf{M}}\ \mathsf{m}) : (\gamma_{01} : \Gamma\ [\gamma_0,\gamma_1]) \to (\mathsf{a}_{01} : \mathsf{A}\gamma_1\ [\mathsf{A}\ \gamma_{01}\ \mathsf{a}_0, \mathsf{a}_1]) \to (\varphi_{01} : \mathsf{a}_{01} \circ (\mathsf{A}\ \gamma_{01}\ \mathsf{x}_0) \circ \mathsf{t}(\gamma_{01}) \equiv \mathsf{x}_1) \\
\qquad \to \mathsf{M}(\gamma_1, \mathsf{a}_1, \mathsf{x}_1)[\ \mathsf{M}(\gamma_{01},\mathsf{a}_{01},\varphi_{01})\ ((\mathsf{J}_{\mathsf{t},\mathsf{M}}\ \mathsf{m})\ \gamma_0\ \mathsf{a}_0\ \mathsf{x}_0), ((\mathsf{J}_{\mathsf{t},\mathsf{M}}\ \mathsf{m})\ \gamma_1\ \mathsf{a}_1\ \mathsf{x}_1)\ ] \\
\quad (\mathsf{J}_{\mathsf{t},\mathsf{M}}\ \mathsf{m})\ \gamma_{01}\ \mathsf{a}_{01}\ \rho_{01} = \mathsf{M}\ (\mathsf{id}_{\gamma_1}, \mathsf{x}_1, \rho_1)\ (\mathsf{m}\ \gamma_{01})\ \mathit{-- \rho_1 : x_1 \circ A\ id_{\gamma_1}\ id_{\mathsf{t}\gamma_1} \circ t\ id_{\gamma_1} \equiv x_1} \\
\\
\mathsf{J}\beta : (\mathsf{J}_{\mathsf{t},\mathsf{M}}\mathsf{m})\ [-\mathsf{t},\mathsf{refl}_\mathsf{t}] \equiv \mathsf{m}
\end{array}$$

Fig. 7. Semantics of directed path induction in the category model

3 Directed Categories with Families

The aim of the present work is not just to establish the category model as a suitable interpretation of directed type theory, but to abstract the category model to a general, abstract notion of 'model' of directed type theory. Specifically, we wish to present this model notion as a generalized algebraic theory, that is, as a CwF with further structure. We do so in several stages, progressively capturing more of the structure described in the previous section. In addition to making the complex and multifaceted notion of 'directed CwF' more digestible, this approach will also give us several intermediate notions, each of which is worthy of further study in its own right. First, we encapsulate the 'negation' structure.

Definition 3.1 (Polarized CwF). A **polarized category with families (PCwF)** consists of a CwF $\mathcal{C} = (\mathsf{Con}, \mathsf{Ty}, \mathsf{Tm}, \triangleright, \ldots)$ equipped with the following operations.

- An endofunctor $(\_)^-\colon \mathsf{Con} \to \mathsf{Con}$ such that $(\Gamma^-)^- \equiv \Gamma$ and $(\sigma^-)^- \equiv \sigma$ for all Γ and σ
- A natural transformation $(\_)^-\colon \mathsf{Ty} \to \mathsf{Ty}$ such that $(A^-)^- \equiv A$ for all A.

So a PCwF is just a CwF equipped with context-, substitution-, and type-negation involutions. The fact that the type-negation operation is a natural transformation just says that it is stable under substitution, i.e. $A[\sigma]^- \equiv A[\sigma^-]$. Now, notably absent from this definition is the negative context extension operation $\triangleright^-$; by this definition, a PCwF only has the positive one. This is because the negative operation is, in fact, definable: in the category model, the following equation holds for any Γ and any $A\colon \mathsf{Ty}\ \Gamma$:

$$(\Gamma \triangleright^+ A)^- = \Gamma^- \triangleright^- A^-. \tag{4}$$

Here we use the fact that $(\Gamma^-)^- = \Gamma$,[2] and hence $A : \mathsf{Ty}\ (\Gamma^-)^-$, making the right-hand side well-formed. Consequently, we can turn this equation around to *define* negative context extension: for $A : \mathsf{Ty}(\Gamma^-)$, let $\Gamma \triangleright^- A$ be $(\Gamma^- \triangleright^+ A^-)^-$. The isomorphism characterizing $\triangleright^-$ (Equation 1) can then be proved as a consequence of the one for $\triangleright^+$.

Also absent from Definition 3.1 is any mechanism connecting the context/substitution negation endofunctor to the type-negation operation. It's unclear if this ought to be rectified, or if their connection is just a peculiarity of the category model. Not every CwF fits the same mold of "contexts are structures, types are families of structures", so it's not possible to require in general that the type-negation operation is just post-composition with the context-negation functor. There *are* suitably abstract ways of connecting the two—for instance, we can note that the category model is *democratic* in the sense of [9, Defn. 3] : there is an isomorphism K between contexts Γ and closed types; this isomorphism is compatible with both negation operations, in that $K(\Gamma^-) = K(\Gamma)^-$. However, we don't need need such strong assumptions for the results of Section 4, so we omit them from the general definition of PCwFs.

Of course, the category model and the preorder model are both examples of PCwFs, where the negation is the 'opposite' construction. But so are the groupoid and setoid models. Indeed, the groupoid model is a *sub-PCwF* of the category model: a groupoid Γ is a category, and so it makes perfect sense to take the opposite category of Γ, obtaining Γ^-, which is also a groupoid. What makes the groupoid model a peculiar instance of a PCwF is that $\Gamma \cong \Gamma^-$ for every Γ, and $A(\gamma) \cong A^-(\gamma)$ for every A. It is what we'll call a *symmetric PCwF*. The setoid model is also a symmetric PCwF, but strictly so: there, $\Gamma \equiv \Gamma^-$. The situation exemplified by the groupoid/category and setoid/preorder models—a symmetric sub-PCwF of another PCwF—is what we capture in our next notion.

Definition 3.2 (Neutral-Polarized CwF). A **sub-PCwF** $\mathcal{D}$ of a PCwF $\mathcal{C}$ consists of predicates $\mathsf{D}_{\mathsf{Con}} : \mathsf{Con} \to \mathsf{Prop}$ and $\mathsf{D}_{\mathsf{Ty}} : \{\Gamma : \mathsf{Con}\} \to \mathsf{Ty}\ \Gamma \to \mathsf{Prop}$ such that

- $\mathsf{D}_{\mathsf{Con}}\ \bullet$;
- if $\mathsf{D}_{\mathsf{Con}}(\Gamma)$, then $\mathsf{D}_{\mathsf{Con}}(\Gamma^-)$;
- if $\mathsf{D}_{\mathsf{Ty}}(A)$, then $\mathsf{D}_{\mathsf{Ty}}(A^-)$;
- if $A : \mathsf{Ty}\ \Gamma$ is such that $\mathsf{D}_{\mathsf{Ty}}(A)$, then $\mathsf{D}_{\mathsf{Ty}}(A[\sigma])$ for any $\sigma : \mathsf{Sub}\ \Delta\ \Gamma$; and
- if $\mathsf{D}_{\mathsf{Con}}(\Gamma)$ and $\mathsf{D}_{\mathsf{Ty}}(A)$, then $\mathsf{D}_{\mathsf{Con}}(\Gamma \triangleright A)$.

We indicate a sub-PCwF by $\mathcal{D} = (\mathsf{DCon}, \mathsf{DTy})$ to indicate that DCon is the subcategory of $\mathsf{D}_{\mathsf{Con}}$-contexts, and DTy is the subpresheaf of D_{Ty}-types.

A **neutral-polarized category with families (NPCwF)** consists of a PCwF $\mathcal{C}$ and a sub-PCwF $\mathcal{N} = (\mathsf{NeutCon}, \mathsf{NeutTy})$ such that

- NeutCon is symmetric: every $\Gamma : \mathsf{NeutCon}$ comes equipped with an $e : \mathsf{Sub}\ \Gamma\ \Gamma^-$ such that $e^- : \mathsf{Sub}\ \Gamma^-\ \Gamma$ is an inverse of e; moreover, this isomorphism is natural in the sense that, for $\Delta, \Gamma : \mathsf{NeutCon}$ and $\sigma : \mathsf{Sub}\ \Delta\ \Gamma$, we have $\sigma \equiv e^-_\Gamma \circ \sigma^- \circ e_\Delta$;
- if either $\Gamma : \mathsf{NeutCon}$ and $A : \mathsf{Ty}\ \Gamma$ or $\Gamma : \mathsf{Con}$ and $A : \mathsf{NeutTy}\ \Gamma$, there are coercion operations $- : \mathsf{Tm}(\Gamma, A) \to \mathsf{Tm}(\Gamma, A^-)$ and $- : \mathsf{Tm}(\Gamma, A^-) \to \mathsf{Tm}(\Gamma, A)$ such that $-(-t) \equiv t$ for all t;
- for every $\Gamma : \mathsf{NeutCon}$ and $A : \mathsf{Ty}\Gamma^-$, there is an isomorphism

$$(e \triangleright A) : \Gamma \triangleright^+ A[e] \cong \Gamma \triangleright^- A$$

  such that
  – $\mathsf{p}_{-,A} \circ (e \triangleright A) \equiv \mathsf{p}_{A[e]}$

[2] In the category model, this holds as a definitional equality, though in Definition 3.1 we only asserted it propositionally, since we're defining a GAT.

– for every $\Delta, \Gamma : \mathsf{NeutCon}$, $\sigma : \mathsf{Sub}\ \Delta\ \Gamma$, $A : \mathsf{Ty}(\Gamma^-)$ and every $a : \mathsf{Tm}(\Delta, A[e \circ \sigma])$,[3]

$$(e \triangleright A) \circ \langle \sigma, {}_+\, a \rangle \equiv \langle \sigma, {}_-\, -a[e^-] \rangle\,.$$

In the third bullet point, note that A is *not* assumed to be in $\mathsf{NeutTy}\ \Gamma^-$; if it were, then $\Gamma^- \triangleright A : \mathsf{NeutCon}$ and then we could use the isomorphism e for $\Gamma^- \triangleright A$ and the coercion operators to construct this. With A being an arbitrary type, this is a genuine addition to the theory. The requirements of this point are somewhat *ad-hoc*: these were just the principles needed in Section 4 to be able to operate effectively with neutral contexts (and all are provable in the category model). Perhaps a more mature version of this theory will place more requirements on NPCwFs, but this is all the neutral-polar structure needed here.

Let us also note that a common feature in directed type theories (e.g. [23, 24]) is to include *core types*, i.e. an operation of the form $(\_)^0 : \mathsf{Ty}\ \Gamma \to \mathsf{NeutTy}\ \Gamma$. In the category model, this is interpreted as applying the *core groupoid* construction to each category $A(\gamma)$, producing a family of groupoids indexed over Γ. We might as well have a deep version too, operating on contexts $(\_)^0 : \mathsf{Con} \to \mathsf{NeutCon}$. We won't need these features for the present work (and therefore don't endeavor to axiomatize them), but, once again, it's quite possible that it would be fruitful to add them in the future.

Recall that CwFs are not a single notion of model for a single type theory, but rather that CwFs encode the basic structural operations of type theory, upon which innumerable different type theories can be specified by defining the desired term- and type-formers. We have arrived at the same point in our development of a semantics for directed type theory: the notion of NPCwF consists solely of structural components, but nothing that actually allows for the construction of interesting types and terms. So let's rectify this by giving the directed analogue of the standard core of undirected type theory: identity types, dependent types, and universes.

We start with the directed analogue of identity types, hom-types.

Definition 3.3 (Directed CwF). A **directed CwF (DCwF)** is a NPCwF equipped with the following structure:

- a type former

$$\mathsf{Hom} : \{\Gamma : \mathsf{Con}\}\{A : \mathsf{Ty}\ \Gamma\} \to \mathsf{Tm}(\Gamma, A^-) \to \mathsf{Tm}(\Gamma, A) \to \mathsf{Ty}\ \Gamma$$

  which is stable under substitution:

$$\mathsf{Hom}(t, t')[\sigma] \equiv \mathsf{Hom}(t[\sigma], t'[\sigma]);$$

- in any $\Gamma : \mathsf{NeutCon}$, a term $\mathsf{refl}_t : \mathsf{Hom}(t, -t)$ for each term $t : \mathsf{Tm}(\Gamma, A^-)$, also stable under substitution by $\sigma : \mathsf{Sub}\ \Delta\ \Gamma$ for $\Delta : \mathsf{NeutCon}$; and
- a term former J as given in Equation 2, also appropriately stable under substitution.

For any $\Gamma : \mathsf{Con}$ and $A : \mathsf{NeutTy}\ \Gamma$, write $\mathsf{Id}(t, t')$ for $\mathsf{Hom}(t, t')$.

The naming of Hom versus Id is suggestive: the types in a DCwF are supposed to function like synthetic categories (with Hom encoding their morphisms), and the neutral types are synthetic groupoids, whose homs are symmetric like an identity type. This point is best illustrated by the following claim.

Proposition 3.4. *Every DCwF has an operation*

$$\mathsf{symm} : \{\Gamma : \mathsf{NeutCon}\}\{A : \mathsf{NeutTy}\}\{t : \mathsf{Tm}(\Gamma, A^-)\}\{t' : \mathsf{Tm}(\Gamma, A)\}$$
$$\to \mathsf{Tm}(\Gamma, \mathsf{Id}(t, t')) \to \mathsf{Tm}(\Gamma, \mathsf{Id}(-t', -t))$$

[3]Note that $e_\Gamma \circ \sigma \equiv \sigma^- \circ e_\Delta$, so $a : \mathsf{Tm}(\Delta, A[\sigma^- \circ e_\Delta])$, hence why the right-hand side of the following equation is well-formed. The need for e_Δ and the need to negate $a[e^-]$ is why Δ must be neutral here.

Proof. By the following construction in the DCwF syntax:

```
symm : {Γ : NeutCon}{A : NeutTy Γ}{t : Tm(Γ, A⁻)}{t' : Tm(Γ, A)}
  → Tm(Γ, Id(t, t')) → Tm(Γ, Id(–t', –t))
symm p = (J_{t,S} refl_t)[ t', p ]
  where
    S : Ty (Γ ▷ A)
    S = Id(–v , –t)
```

□

This proof relies on the neutrality of A in a very subtle, but critical way: in the definition of the type family S, the variable term $\mathsf{v} : \mathsf{Tm}(\Gamma \triangleright A, A[\mathsf{p}_A])$ is negated, so that it is of type $A[\mathsf{p}_A]^-$ and therefore able to stand as the first argument to Id. But this is only possible if $\Gamma \triangleright A : \mathsf{NeutCon}$ because term-negation is only defined in neutral contexts. This reasoning will prove important for the style of reasoning we employ in Section 4, so we isolate it as a principle.

Principle (Var Neg). For $\Gamma : \mathsf{NeutCon}$, the variable term

$$\mathsf{v} : \mathsf{Tm}(\Gamma \triangleright^+ A, A[\mathsf{p}_A])$$

can only be negated (i.e. forming $-\mathsf{v}$) if $A : \mathsf{NeutTy}\ \Gamma$ (and likewise for $\triangleright^-$).

In Section 5, we'll argue that there's *no* way to construct this symmetry term (for arbitrary DCwFs[4]) if A is not assumed to be neutral.

Before proceeding, it's worth explaining what is "the DCwF syntax" mentioned in the proof above. This is where it becomes relevant that DCwFs are presented as generalized algebraic theories: as mentioned in the introduction, [16] proves that any GAT has an initial syntax model. Therefore, any construction done in the syntax model (such as the construction of symm above) can be interpreted into *any* DCwF. This is why a syntactic construction was adequate to prove a claim about *all* DCwFs in the foregoing proof. In the next section, our proofs will all be syntactic, and thereby apply to arbitrary DCwFs.

Let us make an important observation about the syntax of DCwFs. An important criterion for our theory is that hom-types can be *iterated*, that is, our syntax allows for the formation of homs between homs, and homs between homs between homs, and so on. The iteration of identity types is, after all, how homotopy type theory is able to serve as a synthetic language for higher groupoids; and since hom types are iterable in the DCwF syntax, it is a synthetic language for higher categories. However, a given model may be *truncated*, in that the higher structure may become trivial after a certain point. This is the case with the groupoid model: while its types do not all obey the *uniqueness of identity proofs (UIP)* principle and are therefore not mere *h-sets*, they do obey "UIP, one level up": in the groupoid model, identity proofs of identity proofs *are* unique.

The same happens in the category model: in general, there may be terms $p : \mathsf{Tm}(\Gamma, \mathsf{Hom}(t, t')^-)$ and $q : \mathsf{Tm}(\Gamma, \mathsf{Hom}(t, t'))$ but no term of type $\mathsf{Hom}(p, q)$. But if there is such a term, there is exactly one. So the (1-)category model, unsurprisingly, can only model 1-categories. But there's another sense in which the category model structure trivializes "one level up": all the hom-types are interpreted as discrete categories, which are necessarily *groupoids*. So, while $\mathsf{Hom}(t, t')$ is still a synthetic category, it's actually a synthetic *setoid*. This is appropriate for doing synthetic 1-category theory: it makes sense that the hom-types are trivial *as categories*: to do 1-category

[4]with some nontrivial amount of structure.

theoretic arguments, we wish to speak of *identities* between parallel morphisms, not further category-theoretic structure.

We encapsulate DCwFs like this into a definition for further study.

Definition 3.5. A $(1,1)$-**truncated DCwF** is a DCwF such that

- $\mathsf{Hom}(t, t')$ is a *neutral type* for any terms t, t'; and
- UIP holds for identities of hom-terms:

$$\mathsf{UIP}^1 : (\alpha : \mathsf{Tm}(\Gamma, \mathsf{Id}(p,q)^-)) \to (\beta : \mathsf{Tm}(\Gamma, \mathsf{Id}(p,q))) \to \mathsf{Tm}(\Gamma, \mathsf{Id}(\alpha, \beta))$$

The numbering follows the well-known indexing of (n,m)-categories (see e.g. [6, Defn. 8]) to refer to ∞-categories where all parallel k-morphisms are equal when $k > n$ and all k-morphisms are invertible for $k > m$. We could define (n,m)-truncated DCwFs for arbitrary n and m (for instance, the preorder model would be $(0,1)$-truncated, the groupoid model $(1,0)$-truncated, etc.), but that would take us too far afield. For the present work, we will work with $(1,1)$-truncated DCwFs, and develop the theory of synthetic 1-category theory (i.e. synthetic $(1,1)$-category theory) in that language. The practical consequence of working in the syntax of $(1,1)$-truncated DCwFs is that we only have one "layer" of homs, and the type $\mathsf{Hom}(t, t')$ itself is neutral, i.e. its homs are symmetric identity types.

To conclude this section, we state the Π-type former in PCwFs. This is just the appropriately-polarized analogue of [14, Defn. 3.15], and is approximately the same rule for Π-types in [20].

Definition 3.6. A PCwF supports **polarized** Π**-types** if it comes equipped with a type former

$$\Pi : (A : \mathsf{Ty}\ \Gamma^-) \to \mathsf{Ty}(\Gamma \triangleright^- A) \to \mathsf{Ty}\ \Gamma$$

which is stable under substitution, along with a natural isomorphism

$$\mathsf{lam} : \mathsf{Tm}(\Gamma \triangleright^- A, B) \cong \mathsf{Tm}(\Gamma, \Pi(A,B)) : \mathsf{app}.$$

The β-law is that $\mathsf{app} \circ \mathsf{lam} \equiv \mathsf{id}$ and the η-law the other way around. The differences between polarized Π-types and the familiar Π-types of undirected type theory are pretty minimal when operating in neutral contexts: for instance, when instantiating to non-dependent functions $A \to B$, the application operator defined by $f\ \$\ t = (\mathsf{app}\ f)[\mathsf{id}, \_\ t]$ has type

$$\_\$\_ : \mathsf{Tm}(\Gamma, A \to B) \to \mathsf{Tm}(\Gamma^-, A^-) \to \mathsf{Tm}(\Gamma, B).$$

If $\Gamma : \mathsf{NeutCon}$, then we can take terms from $\mathsf{Tm}(\Gamma^-, A)$, $\mathsf{Tm}(\Gamma, A[e])$ and $\mathsf{Tm}(\Gamma, A[e]^-)$ and use the negation operator and the e^- substitution to get into $\mathsf{Tm}(\Gamma^-, A^-)$ for the purposes of applying functions. In the next section, we push this bureaucracy into the background and proceed informally.

4 Synthetic Category Theory

In this section, we work in an arbitrary $(1,1)$-truncated DCwF with polarized Π-types by only working in the syntax. In the main body of the text, we'll adopt an informal type theoretic style (inspired by [28]). We assume that we're working in some neutral context Γ, though we don't explicitly reference Γ. We'll write $t : A$ to indicate $t : \mathsf{Tm}(\Gamma, A)$. In what follows, we'll use the letters $p, q, r, s, t, u, v, w, f, g$ to name terms (of various types) in Γ, whereas the letters x, y, z will be the names of *variables* obtained by extending Γ. We'll have to be careful to abide by the variable negation rule:

Principle (Var Neg). An expression e can only be negated if all the variables occurring in it are of neutral types.

We suppress the distinction between Γ and Γ^-, since we can substitute back and forth with e behind the scenes, as needed. Negative context extension will just behave like positive extension by a negative type: recall that

$$\mathsf{v}_{-,A} : \mathsf{Tm}((\Gamma \triangleright^- A)^-, A[\mathsf{p}^-_{-,A}]^-)$$

i.e.

$$\mathsf{v}_{-,A} : \mathsf{Tm}(\Gamma^- \triangleright^+ A^-, A[\mathsf{p}_{A^-}]^-)$$

so, if we're suppressing the distinction between Γ and Γ^-, then this is just a variable x of A^-. Accordingly, we'll apply functions like this:

$$\frac{f : \prod_{(x:\, A^-)} B(x) \quad t : A^-}{f(t) : B(t)}$$

and form them like this:

$$\frac{x : A^- \vdash e : B(x)}{(\lambda(x : A^-) \to e) : \prod_{x:\, A^-} B(x).}$$

Finally, here's our principle of directed path induction:

Principle (Directed Path Induction). For every $t : A^-$, if $M(x, y)$ is a type family depending on $x : A$ and $y : \mathsf{Hom}(t, x)$, then, for each

$$m : M(-t, \mathsf{refl}_t),$$

we get an

$$\mathsf{ind}_M(m, x, y) : M(x, y)$$

for all x, y.

So, for instance, the construction of symmetry above (the proof of Proposition 3.4) would be expressed informally as follows: given a neutral type A and a term $t : A^-$, define a type family over $x : A, y : Id(t, x)$ by

$$S(x, y) = \mathsf{Id}(-x, -t)$$

We have not violated (Var Neg) because $x : A$ and A is neutral. We have a term of type $S(-t, \mathsf{refl}_t)$, i.e. $\mathsf{Id}(t, -t)$, namely refl_t. So therefore we get $S(x, y)$ for arbitrary x, y. If we have a particular $t' : A$ and $p : \mathsf{Id}(t, t')$, we can put

$$\mathsf{symm}\ p = \mathsf{ind}_S(\mathsf{refl}_t, t', p).$$

Again, we emphasize that it is (Var Neg) which prevents this argument from working for non-neutral types, as desired. Below, we are more casual with our application of directed path induction (e.g. not defining the type family explicitly) in cases where (Var Neg) is not a concern.

With that, we can proceed to the informal constructions. Along the way, the explicit constructions in the DCwF syntax are carried out in the accompanying figures.

4.1 Composition of Homs

Formal development: Fig. 8

As mentioned, a type A in directed type theory is supposed to be a *synthetic category*. The terms $t : A$ represent objects, and the terms $p : \mathsf{Hom}(t, t')$ represent morphisms. For this to truly be category theory, however, we must be able to compose morphisms. We'll write composition in diagrammatic order: given $t, u : A^-$ and $v' : A$, we should be able to compose $p : \mathsf{Hom}(t, -u)$ with $q : \mathsf{Hom}(u, v')$ to get $p \cdot q : \mathsf{Hom}(t, v')$. We do this by directed path induction on q, by putting

$$p \cdot \mathsf{refl}_u = p.$$

The refl terms serve as the identity morphisms of the category: by the above, we know that $p \cdot \mathsf{refl}_u \equiv p$, and thus $\mathsf{refl}_p : \mathsf{Id}(p \cdot \mathsf{refl}_u, p)$. As for the other unit law, we must again use directed path induction: since

$$\mathsf{refl}_u \cdot \mathsf{refl}_u \equiv \mathsf{refl}_u,$$

we have that $\mathsf{refl}_{\mathsf{refl}_u} : \mathsf{Id}(\mathsf{refl}_u \cdot \mathsf{refl}_u, \mathsf{refl}_u)$, and, by induction we get a term

$$\mathsf{r{-}unit}\ q = \mathsf{ind}(\mathsf{refl}_{\mathsf{refl}_u}, v, q) : \mathsf{Id}(\mathsf{refl}_u \cdot q, q)$$

for each $q : \mathsf{Hom}(u, v')$.

Finally, we get that the composition operation is associative. Given $t, u, v : A^-$ and $w' : A$ as well as $p : \mathsf{Hom}(t, -u)$, $q : \mathsf{Hom}(u, -v)$, and $r : \mathsf{Hom}(v, w')$, we construct

$$\mathsf{assoc}\ p\ q\ r : \mathsf{Id}(p \cdot (q \cdot r), (p \cdot q) \cdot r)$$

by directed path induction on r. If $r = \mathsf{refl}_v$, then $q \cdot r \equiv q$ and $(p \cdot q) \cdot r \equiv p \cdot q$. Thus, we have

$$\mathsf{refl}_{p \cdot q} : \mathsf{Id}(p \cdot (q \cdot \mathsf{refl}_v), (p \cdot q) \cdot \mathsf{refl}_v)$$

and then the induction carries through, and we get assoc $p\ q\ r$ as desired.

```
C : {t : Tm(Γ, A⁻)} → Ty (Γ ▷⁺ A ▷⁺ Hom(t'[pA], v))
C = Hom(t[pA],vA)

_·_ : {t t' : Tm(Γ, A⁻)}{t'' : Tm(Γ, A)} → Tm(Γ, Hom(t,-t')) → Tm(Γ, Hom(t',t''))
  → Tm(Γ, Hom(t,t''))
p · q = (JJt',C p) [t'',q]

r-unit : (q : Tm(Γ, Hom(t',t''))) → Tm(Γ, Id(reflt' · q, q))
r-unit q = (JJt',R reflrefl)[t'',q]
  where
    R : Ty (Γ ▷⁺ A ▷⁺ Hom(t'[pA], vA))
    R = Id((JJt',C reflt'), vHom(t'[pA],vA))

l-unit : (p : Tm(Γ, Hom(t,-t'))) → Tm(Γ, Id(p · reflt', p))
l-unit p = reflp    -- by Jβ, p · refl ≡ p

assoc : (p : Tm(Γ, Hom(t,-t'))) → (q : Tm(Γ, Hom(t',-t''))) → (r : Tm(Γ, Hom(t'',t''')))
  → Tm(Γ, Id(p · (q · r), (p · q) · r))
assoc p q r = (JJt',S reflp·q)[t''',r]
  where
    S : Ty (Γ ▷⁺ A ▷⁺ Hom(t[pA], vA))
    S = Id((JJt,C (p · q)), vHom(t[pA],vA))
```

Fig. 8. Composition of Homs

4.2 Synthetic Functors

Formal development: Fig. 9, Fig. 10, Fig. 11, Fig. 12

If types A, B are synthetic categories, it should come as no surprise that terms $f : A \to B$ are synthetic *functors*. The object part is given by the usual function application, but the variances are somewhat mixed: if $t : A^-$, then we can say $f(t) : B$. However, we can still apply f to a term $t' : A$, we just have to put a minus on t', i.e. $f(-t')$.

Unlike usual ("analytic") category theory, we don't have to explicitly define the morphism part of a functor; any term of type $A \to B$ we can write down will come with a morphism part for free. To obtain this morphism part, again we use directed path induction: given an $f : A \to B$ and some $t : A^-$, we can define a B-morphism

$$\mathsf{map}\ f\ p : \mathsf{Hom}(-f(t), f(-t'))$$

for every $t' : A^-$ and $p : \mathsf{Hom}(t, t')$ by putting

$$\mathsf{map}\ f\ \mathsf{refl}_t = \mathsf{refl}_{-f(t)} \quad : \mathsf{Hom}(-f(t), f(t)).$$

By definition, this operation preserves identities (sending refl to refl), and respects composition: if we have $t, u : A^-$ and $p : \mathsf{Hom}(t, -u)$, then, since $\mathsf{map}\ f\ \mathsf{refl}_u \equiv \mathsf{refl}_{-f(u)}$ and $p \cdot \mathsf{refl}_u \equiv p$ and $(\mathsf{map}\ f\ p) \cdot \mathsf{refl}_{-f(u)} \equiv \mathsf{map}\ f\ p$, we have

$$\mathsf{refl}_{(\mathsf{map}\ f\ p)} : \mathsf{Id}(\mathsf{map}\ f\ (p \cdot \mathsf{refl}_u), (\mathsf{map}\ f\ p) \cdot (\mathsf{map}\ f\ \mathsf{refl}_t)).$$

By induction, we get an identity between $\mathsf{map}\ f\ (p \cdot q)$ and $(\mathsf{map}\ f\ p) \cdot (\mathsf{map}\ f\ q)$ for arbitrary q.

Let us also note that functors are also composable: given $f : A \to B$ and $g : B \to C$, we get the usual

$$g \circ f = \lambda(x : A^-) \to g(-f(x)).$$

Of course, we can prove $\mathsf{map}\ (g \circ f)\ p$ equal to $\mathsf{map}\ g\ (\mathsf{map}\ f\ p)$ by directed path induction, using the following observations:

$$\begin{aligned} \mathsf{map}\ f\ \mathsf{refl}_t &\equiv \mathsf{refl}_{-f(t)} \\ \mathsf{map}\ g\ \mathsf{refl}_{-f(t)} &\equiv \mathsf{refl}_{-g(-f(t))} \\ \mathsf{map}\ (g \circ f)\ \mathsf{refl}_t &\equiv \mathsf{refl}_{-(g \circ f)(t)}. \end{aligned}$$

```
_$_ : {Γ : NeutCon}{A : Ty Γ⁻}{B : Ty Γ}
  → (Tm(Γ, A → B)) → Tm(Γ⁻,A⁻) → Tm(Γ,A)
f $ t = (app f)[ id ,_ t ]

map : {Γ : NeutCon}{A : Ty Γ⁻}{B : Ty Γ}(f : Tm(Γ, A → B)){t : Tm(Γ⁻,A⁻)}{t' : Tm(Γ⁻,A)}
  → (Tm(Γ⁻, Hom(t,t'))) → Tm(Γ, Hom( -(f $ t) , f $ (-t')))
map f p = (J_{t[e],MAP} refl_{-(f $ t)})[t'[e]]
  where
    MAP : Ty (Γ ▷ A[e])
    MAP = Hom( (-(f $ t))[p_{A[e]}], (app f)[e ▷ A] )
```

Fig. 9. Morphism part of Functors

$$
\begin{aligned}
&\mathsf{Hom}(-(f\ \$\ t), -(-(f\ \$\ t)))\\
&\equiv \mathsf{Hom}(-(f\ \$\ t), f\ \$\ t)\\
&= \mathsf{Hom}(-(f\ \$\ t), (\mathsf{app}\ f)[\mathsf{id}_{,-}\ t])\\
&\equiv \mathsf{Hom}(-(f\ \$\ t), (\mathsf{app}\ f)[(e \triangleright A) \circ \langle \mathsf{id}_{,+}\ -t[e]\rangle]) \qquad (*)\\
&\equiv \mathsf{Hom}(-(f\ \$\ t), (\mathsf{app}\ f)[(e \triangleright A)][\mathsf{id}_{,+}\ -t[e]])\\
&\equiv \mathsf{Hom}(-(f\ \$\ t)[\mathsf{p}_{A[e]}, (\mathsf{app}\ f)[(e \triangleright A)])[\mathsf{id}_{,+}\ -t[e]]\\
&= \mathsf{MAP}[\mathsf{id}_{,+}\ -t[e]]
\end{aligned}
$$

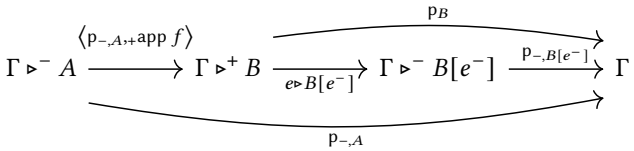
(*) $\langle \mathsf{id}_{,-}\ t\rangle \equiv (e \triangleright A) \circ \langle \mathsf{id}_{,+}\ -t[e]\rangle$ by Definition 3.2 since Γ neutral.

Fig. 10. Calculation that $\mathsf{refl}_{-(f\ \$\ t)} : \mathsf{MAP}[-t[e]]$ in Fig. 9

```
_∘_ : {Γ : NeutCon}{A : Ty Γ⁻}{B C : Ty Γ} →
   Tm(Γ, A → B) → Tm(Γ, B[e⁻] → C) → Tm(Γ, A → C)
g ∘ f = lam( (app g)[ e ▷ B[e⁻] ][ p_{-,A} ,+ app f ] )
```

Fig. 11. Composition of Functions

The diagram



commutes, since $p_{-,B[e^-]} \circ (e \triangleright B[e^-]) \equiv \mathsf{p}_B$ by Definition 3.2 and $\mathsf{p}_B \circ \langle \mathsf{p}_{-,A},+\ \mathsf{app}\ f\rangle \equiv \mathsf{p}_{-,A}$ by Fig. 1. Thus, since $\mathsf{app}\ g : \mathsf{Tm}(\Gamma \triangleright^- B[e^-], C[\mathsf{p}_{-,B[e^-]}])$,

$$(\mathsf{app}\ g)[e \triangleright B[e^-]][\mathsf{p}_{-,A},+\ \mathsf{app}\ f] : \mathsf{Tm}(\Gamma \triangleright^- A, C[\mathsf{p}_{-,A}]).$$

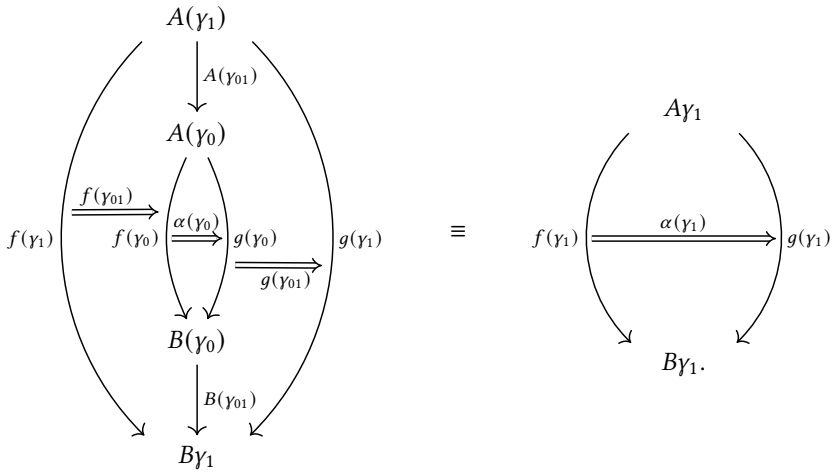
Fig. 12. Auxiliary calculations for Fig. 11

5 Further observations about the Category Model

As the previous section showed, the syntax of 1-truncated Directed CwFs provides a nice setting for some very basic constructions in synthetic category theory. However, further expansion of the DCwF syntax is needed to be able to capture the full range of constructions in category theory. In this section, we'll observe some constructions that can be made (and some equivalences that hold) in the category model, which require further study to be internalized into the DCwF syntax.

Probably the most significant omission from the synthetic category theory of the previous section is natural transformations. There are some natural transformations expressible in the theory as written, because natural transformations are, as we might hope and expect, homs between functors. That is, the type $A \to B$ is a synthetic category: given $f : \mathsf{Tm}(\Gamma, (A \to B)^-)$ and $g : \mathsf{Tm}(\Gamma, A \to B)$, we can form the type $\mathsf{Hom}(f, g)$. In the category model, these are interpreted as natural transformations from f to g, but all dependent over the context Γ. Both f and g sends objects $\gamma : |\Gamma|$ to functors from $A(\gamma)$ to $B(\gamma)$; if $\alpha : \mathsf{Tm}(\Gamma, \mathsf{Hom}(f, g))$, then α sends γ to a natural transformation

$f(\gamma) \to g(\gamma)$. On morphisms $\gamma_{01} : \Gamma\ [\gamma_0, \gamma_1]$, $\alpha(\gamma_{01})$ is a witness to the fact that



Presently, the only such natural transformations expressible in the DCwF syntax are identities (e.g $\mathsf{refl}_f : \mathsf{Tm}(\Gamma, \mathsf{Hom}(f, -f))$). There is not a way to work with the actual components of a natural transformation, or to define a natural transformation *by* its components. What we would like to be able to do is write terms of type

$$\prod_{x\,:\,A^-} \mathsf{Hom}(-f(x), g(x))$$

and then prove that they are automatically natural by directed path induction, since $\alpha_t \cdot \mathsf{map}\ g\ \mathsf{refl}_t \equiv \mathsf{map}\ f\ \mathsf{refl}_t \cdot \alpha_t$. The issue is that this type violates (Var Neg): we're not allowed to write $-f(x)$ for a variable x without knowing that A is neutral. If A *is* neutral, then we're only capturing natural transformations between functors whose domain is a groupoid, which is a significant restriction. Thus, we have two important open questions: how to make natural transformations definable component-wise in the syntax (ideally using Π-types), and how to express in the syntax that the type of such transformations is equivalent to the hom-type between the two functors.

Another important feature we add are *universes*. The category model comes equipped with several type universes, most significantly the universe of sets. More precisely, we can regard the category Set as a closed type in the category model. The operation $\mathsf{El} : \mathsf{Tm}(\bullet, \mathsf{Set}) \to \mathsf{Ty}\ \bullet$ takes a set X and views it as a discrete category. We can then define

$$\mathsf{Hom\text{–}to\text{–}func} : \{X : \mathsf{Tm}(\bullet, \mathsf{Set}^-)\}\{Y : \mathsf{Tm}(\bullet, \mathsf{Set})\} \to \mathsf{Tm}(\bullet, \mathsf{Hom}(X, Y)) \to \mathsf{Tm}(\bullet, \mathsf{El}(X) \to \mathsf{El}(Y))$$

by directed path induction: $\mathsf{Hom\text{–}to\text{–}func}\ \mathsf{refl}_X$ should be the identity function $\mathsf{lam}\ \mathsf{v} : \mathsf{Tm}(\bullet, \mathsf{El}(X) \to \mathsf{El}(X))$. We can use this to state the following principle.

Principle (External Directed Univalence). Hom–to–func is a bijection.

Really, Hom–to–func is the identity function, since terms of hom-types in the empty context are just the morphisms of the category, the morphisms of Set are functions, and a functor between discrete categories is just a function between their objects. Spelling out the category model semantics, we see that every function is sent to itself. Sufficiently internalized, this principle of Directed Univalence serves as the directed analogue of Hofmann and Streicher's *universe extensionality* axiom [15, Section 5.4]. Further work is required to better develop the theory of isomorphisms in the synthetic category theory, and to compare this principle of directed univalence to existing ones (e.g. [19]).

Let us conclude by observing that the existence of a universe allows for metatheoretic reasoning as well, specifically negative proofs about what *cannot* be done in the syntax. We can view the directed universe Set as a source of nontrivial directedness: if we affirm Section 5, then Set cannot possibly be a neutral type. We show that, in DCwFs equipped with a directed univalent universe Set, hom-types must be asymmetric in general. That is, we cannot construct a term symm like in Proposition 3.4 for non-neutral types in the syntax DCwF+Set (the initial model of the GAT of DCwFs with a universe Set). We do so the same way Hofmann and Streicher [15] proved that ordinary Martin-Löf Type Theory couldn't prove the Uniqueness of Identity Proofs: by countermodel. Hofmann and Streicher's countermodel was the groupoid model, and, of course, ours is the category model.

Proposition 5.1. *There cannot be an operation*

$$\mathsf{symm}' : \{\Gamma : \mathsf{NeutCon}\}\{A : \mathsf{Ty}\}\{t : \mathsf{Tm}(\Gamma, A^-)\}\{t' : \mathsf{Tm}(\Gamma, A)\} \\ \to \mathsf{Tm}(\Gamma, \mathsf{Hom}(t, t')) \to \mathsf{Tm}(\Gamma, \mathsf{Hom}(-t', -t))$$

*definable in the syntax of DCwFs+*Set.

Proof. If the syntax model of DCwF+Set had such an operation symm', then, by initiality, so too would every DCwF with Set, in particular the category model. But then for any $X : \mathsf{Tm}(\bullet, \mathsf{Set}^-)$ and $Y : \mathsf{Tm}(\bullet, \mathsf{Set})$ and $f : \mathsf{Tm}(\bullet, \mathsf{Hom}(X, Y))$, we would obtain $\mathsf{symm}'\ f : \mathsf{Tm}(\bullet, \mathsf{Hom}(Y, X))$. But this is absurd, because the function $? : \emptyset \to \mathbb{1}$ is a term of type $\mathsf{El}(\emptyset) \to \mathsf{El}(\mathbb{1})$ in the category model, and, by Section 5, corresponds to a term of type $\mathsf{Hom}(\emptyset, \mathbb{1})$, but there cannot be any terms of $\mathsf{Hom}(\mathbb{1}, \emptyset)$, because the set of terms of this type is in bijection with the set of functors $\mathsf{El}(\mathbb{1})$ to $\mathsf{El}(\emptyset)$, of which there are none. □

Basically the same argument will show the *uniqueness of homs* principle—that for any hom terms $p : \mathsf{Tm}(\Gamma, \mathsf{Hom}(t, t')^-)$ and $q : \mathsf{Tm}(\Gamma, \mathsf{Hom}(t, t'))$, there is a witness of $\mathsf{Id}(p, q)$—is violated in the category model (a counterexample being Set-homs from the two-element set to itself), and therefore not provable in the syntax of DCwFs+Set. So we can conclude that the difference between $(1, 1)$-truncated DCwFs, $(1, 0)$-truncated DCwFs, and $(0, 1)$-truncated CwFs is reflected internally in the syntax.

6 Conclusion and Future Work

We have laid the laid the foundation for a generalized algebraic theory of directed types, and began to conduct synthetic category theory in that setting. Our semantics-forward approach was to study the category model first, and extract its key features into a series of abstract definitions—the GATs of polarized CwFs, neutral-polar CwFs, directed CwFs, $(1, 1)$-truncated directed CwFs, and directed CwFs with features like polarized dependent types and a directed univalent set universe. Working within the directed type theory of these models, we found that it was possible to work informally with the powerful directed path induction principle to make basic constructions in category theory, with our careful discipline about variable negation preventing the directed type theory from collapsing into undirected type theory.

Much remains to be done. The category theory of Section 4 serves as a proof-of-concept, but needs to be fleshed out into a full theory. As mentioned, work is needed to articulate natural transformations in the theory; our current investigations concern possible generalization of Π-types to di-variant *end types* which address some of the above-mentioned variance issues with natural transformations. For reasons of space, we omitted dependent sum types from the theory. But with them added, much of basic category theory should be expressible in this language, such as isomorphisms, (co)slice categories, (co)limits, exponentials, and perhaps some basic topos theory. Better development of the category of sets should put representability and some properties of

presheaf categories into reach, though internal statement and proof of the Yoneda Lemma will likely rely on the resolution of the above-mentioned dilemma regarding natural transformations. We also leave it to future work to study whether this theory can capture higher category theory by weakening or dropping the assumption of 1-truncation, and, if so, how it compares to existing synthetic higher category frameworks, such as [25].

There are further avenues for developing the type theory of DCwFs. Two important metatheoretic results about the syntax of DCwFs currently being pursued are *canonicity* and *normalization*. Moreover, we would like to verify the correctness of these results by formalizing them in a computer proof assistant. A further goal would be to implement the syntax of DCwFs as a computer proof language itself, hopefully with syntax nearly as convenient as the constructions of Section 4, and formalize larger swaths of category theory in it.

As mentioned in the introduction, a motivation for the present work's focus on generalized algebraic theories is the possibility of expressing it in a *second-order* generalized algebraic theory, following [4, 7, 17, 26, 27]. Since our notions of PCwFs and NPCwFs include explicit operations on contexts (as seen in the substructural character of the (Var Neg) rule), it's clear that either an extension to the SOGAT signature language of [17], and/or a partial internalization of the first-order theory into the second-order theory—à la [4]—will be necessary. A related question is whether the $(\_)^-$ operation (or a neutralization operation interpreted in the category model by core groupoids) on types and contexts can be viewed as a modality in the sense of [12]. Also, as mentioned, this work is oriented towards *higher observational type theory*, and further study of the observational equivalences of this theory (e.g. a characterization of the Hom-types of Π-types) is needed.

Finally, the present framework provides a setting for studying *directed higher-inductive types*—inductively-defined types with both term constructors and *hom constructors*. Some simple examples would be the directed interval (as is studied in [25, 30]) and a directed analogue of the *circle* type [28, Section 6.2]. These examples are modelled by the category model, and can therefore be soundly added to the present theory. Higher examples (such as directed versions of higher tori and spheres) would require more careful metatheoretic work to justify, but could perhaps lead to a number of interesting considerations.

References

[1] Benedikt Ahrens, Paige Randall North, and Niels van der Weide. 2023. Bicategorical type theory: semantics and syntax. *Mathematical Structures in Computer Science* 33, 10 (2023).

[2] Thorsten Altenkirch. 1999. Extensional equality in intensional type theory. In *Proceedings. 14th Symposium on Logic in Computer Science (Cat. No. PR00158)*. IEEE, 412–420.

[3] Thorsten Altenkirch, Simon Boulier, Ambrus Kaposi, Christian Sattler, and Filippo Sestini. 2021. Constructing a universe for the setoid model.. In *FoSSaCS*. 1–21.

[4] Thorsten Altenkirch, Yorgo Chamoun, Ambrus Kaposi, and Michael Shulman. 2024. Internal parametricity, without an interval. *Proceedings of the ACM on Programming Languages* 8, POPL (2024), 2340–2369.

[5] Thorsten Altenkirch, Ambrus Kaposi, and Michael Shulman. 2022. Towards Higher Observational Type Theory. 28th International Conference on Types for Proofs and Programs (TYPES 2022).

[6] John C. Baez and Michael Shulman. 2007. Lectures on n-Categories and Cohomology. arXiv:math/0608420 [math.CT] https://arxiv.org/abs/math/0608420

[7] Rafaël Bocquet. 2022. External univalence for second-order generalized algebraic theories. *arXiv:2211.07487* (2022).

[8] John Cartmell. 1986. Generalised algebraic theories and contextual categories. *Annals of pure and applied logic* 32 (1986), 209–243.

[9] Simon Castellan, Pierre Clairambault, and Peter Dybjer. 2021. Categories with families: Unityped, simply typed, and dependently typed. *Joachim Lambek: The Interplay of Mathematics, Logic, and Linguistics* (2021), 135–180.

[10] Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. 2018. Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom. In *21st International Conference on Types for Proofs and Programs*.

[11] Peter Dybjer. 1995. Internal type theory. In *International Workshop on Types for Proofs and Programs*. Springer, 120–134.

[12] Daniel Gratzer, GA Kavvos, Andreas Nuyts, and Lars Birkedal. 2020. Multimodal dependent type theory. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*. 492–506.
[13] Martin Hofmann. 1995. A simple model for quotient types. In *International Conference on Typed Lambda Calculi and Applications*. Springer, 216–234.
[14] Martin Hofmann. 1997. Syntax and semantics of dependent types. In *Extensional Constructs in Intensional Type Theory*. Springer, 13–54.
[15] Martin Hofmann and Thomas Streicher. 1995. The groupoid interpretation of type theory. *Twenty-five years of constructive type theory (Venice, 1995)* 36 (1995), 83–111.
[16] Ambrus Kaposi, András Kovács, and Thorsten Altenkirch. 2019. Constructing quotient inductive-inductive types. *Proc. ACM Program. Lang.* 3, POPL, Article 2 (jan 2019), 24 pages. https://doi.org/10.1145/3290315
[17] Ambrus Kaposi and Szumi Xie. 2024. Second-Order Generalised Algebraic Theories: Signatures and First-Order Semantics. In *9th International Conference on Formal Structures for Computation and Deduction (FSCD 2024)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.
[18] András Kovács. 2022. *Type-Theoretic Signatures for Algebraic Theories and Inductive Types*. Ph. D. Dissertation. Eötvös Loránd University.
[19] Nikolai Kudasov, Emily Riehl, and Jonathan Weinberger. 2023. Formalizing the ∞-Categorical Yoneda Lemma. arXiv:2309.08340 [math.CT]
[20] Daniel R Licata and Robert Harper. 2011. 2-Dimensional Directed Dependent Type Theory. (2011).
[21] Per Martin-Löf. 1975. An Intuitionistic Theory of Types: Predicative Part. In *Logic Colloquium '73*, H.E. Rose and J.C. Shepherdson (Eds.). Studies in Logic and the Foundations of Mathematics, Vol. 80. Elsevier, 73–118.
[22] Per Martin-Löf. 1982. Constructive Mathematics and Computer Programming. In *Logic, Methodology and Philosophy of Science VI*, L. Jonathan Cohen, Jerzy Łoś, Helmut Pfeiffer, and Klaus-Peter Podewski (Eds.). Studies in Logic and the Foundations of Mathematics, Vol. 104. Elsevier, 153–175.
[23] Paige Randall North. 2019. Towards a directed homotopy type theory. *Electronic Notes in Theoretical Computer Science* 347 (2019), 223–239.
[24] Andreas Nuyts. 2015. Towards a directed homotopy type theory based on 4 kinds of variance. *Mém. de mast. Katholieke Universiteit Leuven* (2015).
[25] Emily Riehl and Michael Shulman. 2017. A type theory for synthetic ∞-categories. *arXiv preprint arXiv:1705.07442* (2017).
[26] Taichi Uemura. 2021. *Abstract and concrete type theories*. Ph. D. Dissertation. University of Amsterdam.
[27] Taichi Uemura. 2023. A general framework for the semantics of type theory. *Mathematical Structures in Computer Science* 33, 3 (mar 2023). https://doi.org/10.1017/s0960129523000208
[28] The Univalent Foundations Program. 2013. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study. https://homotopytypetheory.org/book
[29] Benno Van Den Berg and Richard Garner. 2011. Types are weak ω-groupoids. *Proceedings of the london mathematical society* 102, 2 (2011), 370–394.
[30] Matthew Z. Weaver and Daniel R. Licata. 2020. A Constructive Model of Directed Univalence in Bicubical Sets. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science* (Saarbrücken, Germany) *(LICS '20)*. Association for Computing Machinery, New York, NY, USA, 915–928. https://doi.org/10.1145/3373718.3394794
[31] Jonathan Weinberger and Ulrik Buchholtz. 2019. Type-theoretic modalities for synthetic $(\infty, 1)$-categories. International Conference on Homotopy Type Theory (HoTT 2019).

A Additional Calculations

The definition of the Π-type former given in Fig. 5 is repeated in Fig. 13, plus the definitions for lambda abstraction and application.

These definitions rely on the following calculations.

- A naturality calculation for the morphism part of lam t':

$$\begin{aligned}
&t'(\gamma_1, x_1) \circ B(\mathrm{id}_{\gamma_1}, x_1)(t'(\gamma_{01}, \mathrm{id}_{A\ \gamma_{01}\ a_1})) \\
&\equiv t'(\mathrm{id}_{\gamma_1} \circ \gamma_{01}, (A\ \gamma_{01}\ x_1) \circ \mathrm{id}_{A\ \gamma_{01}\ a_1}) \\
&\equiv t'(\gamma_{01}, A\ \gamma_{01}\ x_1) \\
&\equiv t'(\gamma_{01} \circ \mathrm{id}_{\gamma_0}, (A\ \mathrm{id}_{\gamma_0}\ \mathrm{id}_{A\ \gamma_{01}\ a_1'}) \circ A\ \gamma_{01}\ x_1) \\
&\equiv t'(\gamma_{01}, \mathrm{id}_{A\ \gamma_{01}\ a_1'}) \circ B(\gamma_{01}, \mathrm{id}_{A\ \gamma_{01}\ a_1'})(t'(\mathrm{id}_{\gamma_0}, A\ \gamma_{01}\ x_1))
\end{aligned} \tag{5}$$

The first and the last equations are the functoriality conditions of t'. The middle equations are the category laws for Γ, $A(\gamma_0)$, and $A(\gamma_1)$, as well as the functoriality of A.

- To see that the morphism part of app f is well-typed, observe that

$$\begin{aligned}
&(f\ \gamma_{01}) : (\Pi(A,B)\ \gamma_1)[\ \Pi(A,B)\ \gamma_{01}\ (f\ \gamma_0),\ f(\gamma_1)\ ],\\
&\text{i.e.}\\
&(f\ \gamma_{01}) : \mathsf{Transform}\ \{\gamma = \gamma_1\}\ (B(\gamma_{01}, \mathsf{id})\ (f\ \gamma_0), f(\gamma_1))\\
&\text{and therefore,}\\
&\mathsf{component}(f\ \gamma_{01})\ a_1\\
&: B(\gamma_1, a_1)[B(\gamma_{01}, \mathsf{id})\ (f\ \gamma_0\ (A\ \gamma_{01}\ a_1)), f\ \gamma_1\ a_1];
\end{aligned} \tag{6}$$

and also that

$$\begin{aligned}
&a_{01} : (A\ \gamma_0)[a_0, A\ \gamma_{01}\ a_1],\\
&\text{and thus, since } (f\ \gamma_0) : \mathsf{Tm}(A(\gamma_0), B_{\gamma_0}),\\
&f\ \gamma_0\ a_{01}\\
&: B(\gamma_0, A\ \gamma_{01}\ a_1)[B(\mathsf{id}_{\gamma_0}, a_{01})\ (f\ \gamma_0\ a_0), f\ \gamma_0\ (A\ \gamma_{01}\ a_1)]\\
&\text{and therefore}\\
&B(\gamma_{01}, \mathsf{id})\ (f\ \gamma_0\ a_{01})\\
&: B(\gamma_1, a_1)[B(\gamma_{01}, a_{01})\ (f\ \gamma_0\ a_0), B(\gamma_{01}, \mathsf{id})\ (f\ \gamma_0\ (A\ \gamma_{01}\ a_1))].
\end{aligned} \tag{7}$$

and thus we can conclude that $\mathsf{component}(f\ \gamma_{01})\ a_1$ can be composed with $B(\gamma_{01}, \mathsf{id})\ (f\ \gamma_0\ a_{01})$, giving a term of the appropriate type.

```
Π : (A : Ty Γ⁻) → Ty(Γ ▷⁻ A) → Ty Γ
  |Π(A,B) γ| = Tm(A(γ), B_γ)
    where
      B_γ : Ty(A γ)
        B_γ a = B(γ,a)
        B_γ (x : (A γ)[ a , a' ]) = B(id_γ,x)

  record (Π(A,B) γ)[_,_] : (θ θ' : Tm(A(γ), B_γ)) → Set where
      component : (a : |A γ|) → (B(γ,a)) [ θ a , θ' a ]
      naturality : (x : (A γ) [ a , a' ]) → (θ' x) ∘ B(γ,x)(component a) ≡ (component a') ∘ (θ x)

  Π(A,B) γ01 : Tm(A(γ0), B_γ0) →  Tm(A(γ1), B_γ1)
    (Π(A,B) γ01 θ0) (a1 : |A γ1|) = B(γ01, id_(A γ01 a1)) (θ0(A γ01 a1))
    (Π(A,B) γ01 θ0) (x1 : A γ1 [a1 , a1']) = B(γ01, id_(A γ01 a1') (θ0(A γ01 x1))

lam : Tm(Γ ▷⁻ A, B) → Tm(Γ, Π(A,B))
  lam t' : (γ : |Γ|) → | (Π(A,B)) γ |
    lam t' γ : (a : |A γ|) → |B(γ,a)|
    lam t' γ a = t'(γ,a)

    lam t' γ : (x : (A γ)[ a , a' ]) → B(γ,a')[ B(id_γ,x) (lam t' γ a), lam t' γ a' ]
    lam t' γ x = t'(id_γ,x)

  lam t' : (γ01 : Γ[ γ0 , γ1 ]) → Transform {γ = γ1} (Π(A,B) γ01 (lam t' γ0), lam t' γ1)
    component(lam t' γ01) : (a1 : |A γ1|)
      → B(γ1,a1)[ B(γ01, id_(A γ01 a1)) (lam t' γ0 (A γ01 a1)) , lam t' γ1 a1 ]
    component(lam t' γ01) a1 = t'(γ01, id_(A γ01 a1))

    naturality(lam t' γ01) (x1 : (A γ1)[ a1 , a1' ]) = (Equation 5)

app : Tm(Γ, Π(A,B)) → Tm(Γ ▷⁻ A, B)
  app f : (γ : |Γ|) → (a : |A γ|) → |B(γ,a)|
  app f γ a = f γ a                          -- (f γ) : (a : |A γ|) → |B_γ a|

  app f : (γ01 : Γ[ γ0 , γ1 ]) → (a01 : (Aγ0)[ a0 , A γ01 a1 ])
    → B(γ1,a1)[ B(γ01,a01) (app f γ0 a0), app f γ1 a1 ]
  app f γ01 a01 = (component(f γ01) a1) ∘ B(γ01, id) (f γ0 a01)
    -- see Equation 6 and Equation 7
```

Fig. 13. Complete semantics of Π-types in the category model