MULTIMODAL DEPENDENT TYPE THEORY

DANIEL GRATZER [a], G.A. KAVVOS [b], ANDREAS NUYTS [c], AND LARS BIRKEDAL [a]

[a] Aarhus University
   e-mail address: gratzer@cs.au.dk, birkedal@cs.au.dk

[b] University of Bristol
   e-mail address: alex.kavvos@bristol.ac.uk

[c] Vrije Universiteit Brussel
   e-mail address: andreas.nuyts@vub.be

ABSTRACT. We introduce MTT, a dependent type theory which supports multiple modalities. MTT is parametrized by a mode theory which specifies a collection of modes, modalities, and transformations between them. We show that different choices of mode theory allow us to use the same type theory to compute and reason in many modal situations, including guarded recursion, axiomatic cohesion, and parametric quantification. We reproduce examples from prior work in guarded recursion and axiomatic cohesion, thereby demonstrating that MTT constitutes a simple and usable syntax whose instantiations intuitively correspond to previous handcrafted modal type theories. In some cases, instantiating MTT to a particular situation unearths a previously unknown type theory that improves upon prior systems. Finally, we investigate the metatheory of MTT. We prove the consistency of MTT and establish canonicity through an extension of recent type-theoretic gluing techniques. These results hold irrespective of the choice of mode theory, and thus apply to a wide variety of modal situations.

1. INTRODUCTION

In order to increase the expressivity of Martin-Löf Type Theory (MLTT) we often wish to extend it with unary type operators that we call *modalities* or *modal operators*. Some modal operators arise as shorthands for internally definable structure [RSS20], while others are used as a device for internalising non-definable structure from particular models. In the latter case, we are sometimes even able to prove that a modality cannot be internally expressed—at least not without extensive changes to the judgmental structure of type theory: see e.g. the 'no-go' theorems by [Shu18, §4.1] and [LOPS18]. This paper is concerned with the development of a systematic approach to the judgmental formulation of type theories with multiple interacting modalities.

The addition of a modality to a dependent type theory is a non-trivial exercise. Modal operators often interact with the context of a type or term in a complicated way, and naïve approaches lead to undesirable interplay with other type formers and substitution. However, the consequent gain in expressivity is substantial, and so it is well worth the effort. For example, modalities have been used to express guarded recursive definitions [BMSS12, BGC+16, BGM17, Gua18], parametric quantification [NVD17, ND18], proof irrelevance [Pfe01, AS12, ND18], and to define global operations which cannot be localized to an arbitrary context [LOPS18]. There has also been concerted effort towards the development of a dependent type theory corresponding to Lawvere's *axiomatic cohesion* [Law07], which has many interesting applications [Sch13, SS12, Shu18, GLN+17, Kav19].

Despite this recent flurry of developments, a unifying account of modal dependent type theory has yet to emerge. Faced with a new modal situation, a type theorist must handcraft a brand new system, and then prove the usual battery of metatheorems. This introduces formidable difficulties on two levels. First, an increasing number of these applications are *multimodal*: they involve multiple interacting modalities, which significantly complicates the design of the appropriate judgmental structure. Second, the technical development of each such system is entirely separate, so that one cannot share the burden of proof even between closely related systems. To take a recent example, there is no easy way to transfer the work done in the 80-page-long normalization proof for MLTT⌾ [GSB19a] to a normalization proof for the modal dependent type theory of [BCM+20], even though these systems are only marginally different. Put simply, if one wished to prove that type-checking is decidable for the latter, then one would have to start afresh.

We intend to avoid such duplication in the future. Rather than designing a new dependent type theory for some preordained set of modalities, we will introduce a system that is *parametrized by a mode theory*, i.e. an algebraic specification of a modal situation. This system, which we call MTT, solves both problems at once. First, by instantiating it with different mode theories we will show that it can capture a wide class of situations. Some of these, e.g. the one for guarded recursion, lead to a previously unknown system that improves upon earlier work. Second, the predictable behavior of our rules allows us to prove metatheoretic results about large classes of instantiations of our system. For example, our canonicity theorem applies irrespective of the chosen mode theory. As a result, we only need to prove such theorems *once*. Returning to the previous examples, careful choices of mode theory yield two systems that closely resemble the calculi of [BCM+20] and MLTT⌾ [GSB19a] respectively, so that our proof of canonicity applies to both.

In fact, we take things one step further: MTT is not just multimodal, but also *multimode*. That is, each judgment of MTT can be construed as existing in a particular *mode*. All modes have some things in common—e.g. there will be dependent sums in each—but some might possess distinguishing features. From a semantic point of view, different modes correspond to different context categories. In this light, modalities intuitively correspond to *functors* between those categories: in fact, they will be structures slightly weaker than *dependent right adjoints* (DRAs) [BCM+20].

Mode theories. At a high level, MTT can be thought of as a machine that converts a concrete description of modes and modalities into a type theory. This description, which is often called a *mode theory*, is given in the form of a *small strict 2-category* [Ree09, LS16,

LSR17]. A mode theory gives rise to the following correspondence:

$$\text{object} \sim \text{mode}$$
$$\text{morphism} \sim \text{modality}$$
$$\text{2-cell} \sim \text{natural map between modalities}$$

The equations between morphisms and between 2-cells in a mode theory can be used to precisely specify the interactions we want between different modalities. We will illustrate this point with an example.

Instantiating MTT. Suppose we have a mode theory $\mathcal{M}$ with a single object m, a single generating morphism $\mu : m \to m$, and no non-trivial 2-cells. Equipping MTT with $\mathcal{M}$ produces a type theory with a single modal type constructor, $\langle \mu \mid - \rangle$. This is the simplest non-trivial setting, and we can prove very little about it without additional 2-cells.

If we add a 2-cell $\epsilon : \mu \Rightarrow 1$ to $\mathcal{M}$, we can define a function

$$\mathsf{extract}_A : \langle \mu \mid A \rangle \to A$$

inside the type theory. If we also add a 2-cell $\delta : \mu \Rightarrow \mu \circ \mu$ then we can also define

$$\mathsf{duplicate}_A : \langle \mu \mid A \rangle \to \langle \mu \mid \langle \mu \mid A \rangle \rangle$$

Furthermore, we can control the precise interaction between $\mathsf{duplicate}_A$ and $\mathsf{extract}_A$ by adding more equations that relate ϵ and δ. For example, we may ask that $\mathcal{M}$ be the *walking comonad* [SS86] which leads to a type theory with a dependent S4-like modality [Pfe01, dR15, Shu18]. We can be even more specific, e.g. by asking that (μ, ϵ, δ) be *idempotent*.

Thus, a morphism $\mu : n \to m$ introduces a modality $\langle \mu \mid - \rangle$, and a 2-cell $\alpha : \mu \Rightarrow \nu$ of $\mathcal{M}$ allows for the definition of a function of type $\langle \mu \mid A \rangle \to \langle \nu \mid A \rangle$ at mode m.

Relation to other modal type theories. Most work on modal type theories still defies classification. However, we can informatively position MTT with respect to two qualitative criteria, viz. usability and generality.

Much of the prior work on modal type theory has focused on bolting a specific modality onto a type theory. The benefit of this approach is that the syntax can be designed to be as convenient as possible for the application at hand. For example, spatial/cohesive type theory [Shu18] features two modalities, $\flat$ and $\sharp$, and is presented in a dual-context style. This judgmental structure, however, is applicable only because of the particular properties of $\flat$ and $\sharp$. Nevertheless, the numerous pen-and-paper proofs in *op. cit.* demonstrate that the resulting system is easy to use.

At the other end of the spectrum, the framework of Licata-Shulman-Riley (LSR) [LSR17] comprises an extremely general toolkit for simply-typed, substructural modal type theory. Its dependent generalization, which is currently under development, is able to handle a very large class of modalities. However, this generality comes at a price: its syntax is complex and unwieldy, even in the simply-typed case.

MTT attempts to strike a delicate balance between those two extremes. By avoiding substructural settings and some kinds of modalities we obtain a noticeably simpler apparatus. Unlike LSR, we need not annotate our term formers with delayed substitutions, and our approach extends to dependent types in a straightforward manner. Most of the pleasant type-theoretic behaviour of MTT is achieved by ensuring that none of its rules 'trim' the context, which would necessitate either delayed substitutions [BGC+16, LSR17] or delicate proofs of the admissibility of substitution [BGM17, BCM+20, GSB19a]. We also show that

MTT can be employed to reason about many models of interest, and that it is simple enough to be used in pen-and-paper calculations.

Contributions. In summary, we make the following contributions:

- We introduce MTT, a general type theory for multiple modes and multiple interacting modalities.
- We present a semantics, which constitute a category of models deriving from the generalized algebraic theory that underlies MTT.
- Using the semantics, we prove that—subject to a technical restriction—MTT satisfies *canonicity*, an important metatheoretic property. This is achieved through a modern *gluing* argument [Shu15, AK16, Coq19, KHS19].
- Finally, we instantiate MTT with various mode theories, and show its use in reasoning about two specific modal situations, viz. guarded recursion [BGC$^+$16], and internal adjunctions [Shu18, LOPS18].

2. The Syntax of MTT

As mentioned in the introduction, the syntax of MTT is parameterized by a small 2-category called a *mode theory*. We will later show how to instatiate MTT with a mode theory in order to reason about particular scenarios, but for now we will work over an arbitrary mode theory. We thus fix a mode theory $\mathcal{M}$, and use m, n, o to stand for modes (the objects of $\mathcal{M}$), μ, ν, τ for modalities (the morphisms), and α, β, γ for 2-cells.

In broad terms, MTT consists of a collection of type theories, one for each mode $m \in \mathcal{M}$. These type theories will eventually appear in one another, but only as spectres under a modality. We thus begin by describing the individual type theories at each mode, and then discuss how modalities are used to relate them.

2.1. **The Type Theory at Each Mode.** Each mode of MTT is inhabited by a standard Martin-Löf Type Theory (MLTT), and accordingly includes the usual judgments. For example, we have the judgment Γ ctx $@\, m$ which states that Γ is a well-formed context *in that particular mode m*. There are likewise judgments for types, terms, and substitutions at each mode.

$$\boxed{\Gamma \vdash A \;\mathsf{type}_\ell @\, m}$$

$$\frac{\Gamma \;\mathsf{ctx} @\, m}{\Gamma \vdash \mathsf{U} \;\mathsf{type}_1 @\, m} \qquad \frac{\Gamma \;\mathsf{ctx} @\, m}{\Gamma \vdash \mathbb{B} \;\mathsf{type}_\ell @\, m} \qquad \frac{\Gamma \;\mathsf{ctx} @\, m \qquad \Gamma \vdash A \;\mathsf{type}_\ell @\, m \qquad \ell \le \ell'}{\Gamma \vdash \Uparrow A \;\mathsf{type}_{\ell'} @\, m}$$

$$\frac{\Gamma \;\mathsf{ctx} @\, m \qquad \Gamma \vdash A \;\mathsf{type}_\ell @\, m \qquad \Gamma \vdash M, N : \Uparrow A @\, m}{\Gamma \vdash \mathsf{Id}_A(M, N) \;\mathsf{type}_\ell @\, m}$$

$$\frac{\Gamma \;\mathsf{ctx} @\, m \qquad \Gamma \vdash A \;\mathsf{type}_\ell @\, m \qquad \Gamma, x : \Uparrow A \vdash B \;\mathsf{type}_\ell @\, m}{\Gamma \vdash (x : A) \to B \;\mathsf{type}_\ell @\, m \qquad \Gamma \vdash (x : A) \times B \;\mathsf{type}_\ell @\, m}$$

Figure 1: Selected mode-local rules.

In lieu of an exhaustive list of rules, which we will present in Section 4, we illustrate this point by only showing the important ones in Figure 1. In brief, each mode comprises an ordinary intensional type theory with dependent sums, dependent products, intensional identity types, booleans, and one universe. Both sums and products satisfy an η-rule.

Universes à la Coquand. There are several ways to introduce universes in type theory [Hof97, §2.1.6] [Pal98, Luo12]. We use the approach of [Coq13], which is close to Tarski-style universes. However, instead of inductively defining *codes* that represent particular types, Coquand-style universes come with an *explicit isomorphism* between types and terms of the universe U. However, we must remember to exercise caution: if this isomorphism were to cover all types then *Girard's paradox* [Coq86] would apply, so we must restrict it to *small types*. This, in turn, forces us to stratify our types into small and large.

The judgment $\Gamma \vdash A \: \mathsf{type}_0 @ m$ states that A is a small type, and $\Gamma \vdash A \: \mathsf{type}_1 @ m$ that it is large. The universe itself must be a large type, but otherwise both levels are closed under all other connectives. Finally, we introduce an operator that *lifts* a small type to a large one:

$$\frac{\ell \leq \ell' \qquad \Gamma \vdash A \: \mathsf{type}_\ell @ m}{\Gamma \vdash \Uparrow A \: \mathsf{type}_{\ell'} @ m}$$

The lifting operation commutes definitionally with all the connectives, e.g. $\Uparrow(A \to B) = \Uparrow A \to \Uparrow B$. We will use large types for the most part: only they will be allowed in contexts, and the judgment $\Gamma \vdash M : A @ m$ will presuppose that A is large. As we will not have terms at small types, we will not need the term lifting operations used by [Coq13] and [Ste19].

Following this stratification, we may introduce operations that exhibit the isomorphism:

$$\frac{\Gamma \vdash M : U @ m}{\Gamma \vdash \mathsf{El}(M) \: \mathsf{type}_0 @ m} \qquad\qquad \frac{\Gamma \vdash A \: \mathsf{type}_0 @ m}{\Gamma \vdash \mathsf{Code}(A) : U @ m}$$

along with the equations $\mathsf{Code}(\mathsf{El}(M)) = M$ and $\mathsf{El}(\mathsf{Code}(A)) = A$.

The advantages of universes à la Coquand are now evident: rather than having to introduce Tarski-style codes, we now find that they are *definable*. For example, assuming $M : U$ and $x : \mathsf{El}(M) \vdash N : U$, we let

$$(x : M) \mathbin{\widehat{\to}} N \triangleq \mathsf{Code}((x : \mathsf{El}(M)) \to \mathsf{El}(N)) : U$$

We can then calculate that

$$\mathsf{El}((x : M) \mathbin{\widehat{\to}} N) = \mathsf{El}(\mathsf{Code}((x : \mathsf{El}(M)) \to \mathsf{El}(N))) = (x : \mathsf{El}(M)) \to \mathsf{El}(N)$$

We will often suppress $\Uparrow-$ as well as the explicit isomorphism.

2.2. Introducing a Modality.

Having sketched the basic type theory inhabiting each mode, we now turn to the interaction between them. This is the domain of the modalities.

Suppose $\mathcal{M}$ contains a modality $\mu : n \to m$. We would like to think of μ as a 'map' from mode n to mode m. Then, for each $\vdash A \: \mathsf{type} @ n$ we would like a type $\vdash \langle \mu \mid A \rangle \: \mathsf{type} @ m$. On the level of terms we would similarly like for each $\vdash M : A @ n$ an induced term $\vdash \mathsf{mod}_\mu(M) : \langle \mu \mid A \rangle @ m$.

These constructs would be entirely satisfactory, were it not for the presence of *open terms*. To illustrate the problem, suppose we have a type $\Gamma \vdash A \: \mathsf{type} @ n$. We would hope that the corresponding modal type lives in the same context, i.e. that $\Gamma \vdash \langle \mu \mid A \rangle \: \mathsf{type} @ m$. However, this is not possible, as Γ is only a context at mode n, and cannot be carried over

verbatim to mode m. Hence, the only pragmatic option is to introduce an operation that allows a context to be mapped to another mode.

Forming a modal type. There are several different proposed solutions to this problem in the literature [PD01, Clo18]. We will use a *Fitch-style* discipline [BGM17, BCM+20, GSB19a]: we will require that a modality μ induce an operation on contexts in the *opposite* direction. We will denote this operation by a *lock*:

$$\frac{\Gamma \text{ ctx} @ m}{\Gamma, \blacksquare_\mu \text{ ctx} @ n} \text{ cx/lock}$$

Intuitively, $\blacksquare_\mu$ will behave somewhat like a left adjoint to $\langle \mu \mid - \rangle$. However, $\langle \mu \mid - \rangle$ acts on types while $-, \blacksquare_\mu$ acts on contexts, so this cannot be an ordinary adjunction. Instead, $\langle \mu \mid - \rangle$ will be what [BCM+20] call a *dependent right adjoint* (DRA). A DRA essentially consists of a type former $\mathbf{R}$ and a context operation $\mathbf{L}$ such that

$$\{N \mid \mathbf{L}(\Gamma) \vdash N : A\} \cong \{M \mid \Gamma \vdash M : \mathbf{R}(A)\} \tag{$\dagger$}$$

See [BCM+20] for a formal definition.

Just as with DRAs, the MTT formation and introduction rules for modal types effectively *transpose* types and terms across this adjunction:

$$\frac{\mu : m \to n \qquad \Gamma, \blacksquare_\mu \vdash A \text{ type}_\ell @ n}{\Gamma \vdash \langle \mu \mid A \rangle \text{ type}_\ell @ m} \text{ tp/modal} \qquad \frac{\mu : m \to n \qquad \Gamma, \blacksquare_\mu \vdash M : A @ n}{\Gamma \vdash \text{mod}_\mu(M) : \langle \mu \mid A \rangle @ m} \text{ tm/modal-intro}$$

It remains to show how to eliminate modal types. Previous work on Fitch-style calculi [BCM+20, GSB19a] has employed elimination rules which essentially invert the introduction rule tm/modal-intro. Such rules *remove* one or more locks from the context during type-checking, and sometimes even trim a part of it. For example, a rule of this sort would be

$$\frac{\blacksquare_\mu \notin \Gamma' \qquad \Gamma \vdash M : \langle \mu \mid A \rangle @ m}{\Gamma, \blacksquare_\mu, \Gamma' \vdash \text{open}(M) : A @ n}$$

This kind of rule tends to be unruly, and delicate work is required to prove even basic results about it. For example, see the technical report [GSB19b] for a particularly laborious proof of the admissibility of substitution. The results in *op. cit.* could not possibly reuse any of the work of [BCM+20], as a small change in the syntax leads to many subtle differences in the metatheory. Consequently, it seems unlikely that one could adapt this approach to a modality-agnostic setting like ours.

We will use a different technique, which is reminiscent of dual-context calculi [Kav20]. First, we will let the variable rule control the use of modal variables. Then, we will take a 'modal cut' rule, which will allow the substitution of modal terms for modal variables, to be our modal elimination rule.

Accessing a modal variable. The behavior of modal types can often be clarified by asking a simple question: when can we use a variable $x : \langle \mu \mid A \rangle$ of modal type to construct a term of type A? In previous Fitch-style calculi we would use the modal elimination rule to reduce the goal to $\langle \mu \mid A \rangle$, and then—*had the modal elimination rule not eliminated x from the context*—we would simply use the variable. We may thus write down a term of type A using

a variable $x : \langle \mu \mid A \rangle$ only when our context is structured in a way that does not obstruct the use of x, and the final arbiter of that is the modal elimination rule.

MTT turns this idea on its head: rather than handing control over to the modal elimination rule, we delegate this decision to the variable rule itself. In order to ascertain whether we can use a variable in our calculus, the variable rule examines *the locks to the right of the variable*. The rule of thumb is this: we should always be able to access $\langle \mu \mid A \rangle$ behind $\blacksquare_\mu$. Carrying the illustrative analogy of an adjunction $-, \blacksquare_\mu \dashv \langle \mu \mid - \rangle$ further, we see that the simplest judgment that fits this, namely $\Gamma, x : \langle \mu \mid A \rangle, \blacksquare_\mu \vdash x : A @ n$, corresponds to the *counit* of the adjunction.

To correctly formulate the variable rule, we will require one more idea: following modal type theories based on *left division* [Pfe01, Abe06, Abe08, NVD17, ND18], every variable in the context will be annotated with a modality, $x : (\mu \mid A)$. Intuitively a variable $x : (\mu \mid A)$ is the same as a variable $x : \langle \mu \mid A \rangle$, but the annotations are part of the structure of a context while $\langle \mu \mid A \rangle$ is a type. This small circumlocution will ensure that the variable rule respects substitution.

The most general form of the variable rule will be able to handle the interaction of modalities, so we present it in stages. A first counit-like approximation is then

$$\text{TM/VAR/COUNIT}$$
$$\frac{\blacksquare \notin \Gamma_1 \qquad \Gamma_0, \blacksquare_\mu \vdash A \; \mathsf{type}_1 @ n}{\Gamma_0, x : (\mu \mid A), \blacksquare_\mu, \Gamma_1 \vdash x : A @ n}$$

The first premise requires that no further locks occur in Γ_1, so that the conclusion remains in the same mode n. The second premise is just enough to derive $\Gamma_0 \vdash \langle \mu \mid A \rangle \; \mathsf{type}_1 @ m$.

Context extension. The switch to modality-annotated declarations $x : (\mu \mid A)$ also requires us to revise the context extension rule. The revised version, CX/EXTEND, appears in Figure 2 and closely follows the formation rule for $\langle \mu \mid - \rangle$: if $\Gamma, \blacksquare_\mu \vdash A \; \mathsf{type}_1 @ n$ is a type in the locked context Γ, then we may extend the context Γ to include a declaration $x : (\mu \mid A)$, so that x stands for a term of type A *under the modality* μ.

The elimination rule. The difference between a modal type $\langle \mu \mid A \rangle$ and an annotated declaration $x : (\mu \mid A)$ in the context is navigated by the modal elimination rule. In brief, its role is to enable the substitution of a term of the former type for a variable with the latter declaration. The full rule is complex, so we first discuss the case of a single modality $\mu : n \to m$. The corresponding rule is

$$\text{TM/MODAL-ELIM/SINGLE-MODALITY}$$
$$\frac{\Gamma \vdash M_0 : \langle \mu \mid A \rangle @ m \qquad \Gamma, x : (1 \mid \langle \mu \mid A \rangle) \vdash B \; \mathsf{type}_1 @ m \qquad \Gamma, y : (\mu \mid A) \vdash M_1 : B[\mathsf{mod}_\mu(y)/x] @ m}{\Gamma \vdash \mathsf{let} \; \mathsf{mod}_\mu(y) \leftarrow M_0 \; \mathsf{in} \; M_1 : B[M_0/x] @ m}$$

Forgetting dependence for a moment, we see that this rule is close to the dual-context style [PD01, Kav20]: if we think of annotations as separating the context into multiple zones, then $y : (\mu \mid A)$ clearly belongs to the 'modal' part.

In the dependent case we also need a motive $\Gamma, x : (1 \mid \langle \mu \mid A \rangle) \vdash B \; \mathsf{type}_1 @ m$, which depends on a variable of modal type, but under the identity modality 1. This premise is then fulfilled by M_0 in the conclusion. In a sense, this rule permits a form of *modal induction*: every variable $x : (1 \mid \langle \mu \mid A \rangle)$ can be assumed to be of the form $\mathsf{mod}_\mu(y)$ for some $y : (\mu \mid A)$. This kind of rule has appeared before in the spatial and cohesive type theories of [Shu18].

$$\boxed{\Gamma \;\mathsf{ctx}\,@\,m}$$

CX/LOCK
$$\frac{\mu : n \to m \qquad \Gamma\;\mathsf{ctx}\,@\,m}{\Gamma, \blacksquare_\mu \;\mathsf{ctx}\,@\,n}$$

CX/EXTEND
$$\frac{\mu : n \to m \qquad \Gamma\;\mathsf{ctx}\,@\,m \qquad \Gamma, \blacksquare_\mu \vdash A \;\mathsf{type}_1\,@\,n}{\Gamma, x : (\mu \mid A) \;\mathsf{ctx}\,@\,m}$$

CX/ID
$$\frac{\Gamma\;\mathsf{ctx}\,@\,m}{\Gamma = \Gamma, \blacksquare_1 \;\mathsf{ctx}\,@\,m}$$

CX/COMPOSE
$$\frac{\nu : o \to n \qquad \mu : n \to m \qquad \Gamma\;\mathsf{ctx}\,@\,m}{\Gamma, \blacksquare_\mu, \blacksquare_\nu = \Gamma, \blacksquare_{\mu\circ\nu} \;\mathsf{ctx}\,@\,o}$$

$$\boxed{\Gamma \vdash A \;\mathsf{type}_\ell\,@\,m \qquad \Gamma \vdash M : A\,@\,m}$$

TP/MODAL
$$\frac{\mu : n \to m \qquad \Gamma, \blacksquare_\mu \vdash A \;\mathsf{type}_\ell\,@\,n}{\Gamma \vdash \langle \mu \mid A \rangle \;\mathsf{type}_\ell\,@\,m}$$

TM/VAR
$$\frac{\nu : m \to n \qquad \alpha : \nu \Rightarrow \mathsf{locks}(\Gamma_1)}{\Gamma_0, x : (\nu \mid A), \Gamma_1 \vdash x^\alpha : A^\alpha\,@\,m}$$

TM/MODAL-INTRO
$$\frac{\mu : n \to m \qquad \Gamma, \blacksquare_\mu \vdash M : A\,@\,n}{\Gamma \vdash \mathsf{mod}_\mu(M) : \langle \mu \mid A \rangle\,@\,m}$$

TM/MODAL-ELIM
$$\frac{\mu : n \to m \qquad \nu : m \to o \qquad \Gamma, x : (\nu \mid \langle \mu \mid A \rangle) \vdash B \;\mathsf{type}_1\,@\,o}{\Gamma, \blacksquare_\nu \vdash M_0 : \langle \mu \mid A \rangle\,@\,m \qquad \Gamma, x : (\nu \circ \mu \mid A) \vdash M_1 : B[\mathsf{mod}_\mu(x)/x]\,@\,o}{\Gamma \vdash \mathsf{let}_\nu \;\mathsf{mod}_\mu(x) \leftarrow M_0 \;\mathsf{in}\; M_1 : B[M_0/x]\,@\,o}$$

TM/MODAL-BETA
$$\frac{\nu : m \to o \qquad \mu : n \to m \qquad \Gamma, x : (\nu \mid \langle \mu \mid A \rangle) \vdash B \;\mathsf{type}_1\,@\,o}{\Gamma, \blacksquare_{\nu\circ\mu} \vdash M_0 : A\,@\,n \qquad \Gamma, x : (\nu \circ \mu \mid A) \vdash M_1 : B[\mathsf{mod}_\mu(x)/x]\,@\,o}{\Gamma \vdash \mathsf{let}_\nu \;\mathsf{mod}_\mu(x) \leftarrow \mathsf{mod}_\mu(M_0) \;\mathsf{in}\; M_1 = M_1[M_0/x] : B[\mathsf{mod}_\mu(M_0)/x]\,@\,o}$$

$$\boxed{\mathsf{locks}(\Gamma)}$$

$$\mathsf{locks}(\cdot) = 1 \qquad \mathsf{locks}(\Gamma, x : (\mu \mid A)) = \mathsf{locks}(\Gamma) \qquad \mathsf{locks}(\Gamma, \blacksquare_\mu) = \mathsf{locks}(\Gamma) \circ \mu$$

Figure 2: Selected modal rules.

In the type theory of [BCM$^+$20] modalities are taken to be dependent right adjoints, with terms witnessing Equation †. This isomorphism can encode TM/MODAL-ELIM/SINGLE-MODALITY, but that rule alone cannot encode Equation †. As a result, modalities in MTT are weaker than DRAs.

2.3. **Multiple Modalities.** So far we have only considered a single modality. In this section we discuss the few additional tweaks that are needed to support multiple interacting modalities. The final version of the modal rules is given in Figure 2.

Multimodal locks. Up to this point the operation $-,\blacksquare_\mu$ on contexts has referred to a single modality $\mu : n \to m$. The rule CX/LOCK generalizes it to work with any modality. The only question then is how the resulting operations should interact. This is where the mode theory comes in: locks should be *functorial*, so that $\nu : o \to n$, $\mu : n \to m$, and Γ ctx $@\, m$ imply $\Gamma, \blacksquare_\mu, \blacksquare_\nu = \Gamma, \blacksquare_{\mu \circ \nu}$ ctx $@\, o$. We additionally ask that the identity modality $1 : m \to m$ at each mode has a trivial, invisible action on contexts, i.e.that $\Gamma, \blacksquare_1 = \Gamma$.

These two actions, which are encoded by CX/COMPOSE and CX/ID, ensure that $\blacksquare$ *is a contravariant functor on* $\mathcal{M}$, mapping each mode m to the category of contexts Γ ctx $@\, m$. The contravariance originates from the fact that $\mathcal{M}$ is a specification of the behavior of the modalities $\langle \mu \mid - \rangle$, so that their left-adjoint-like counterparts $-,\blacksquare_\mu$ act with the opposite variance.

The full variable rule. We have seen that $\blacksquare$ induces a functor from $\mathcal{M}$ to categories of contexts, but we have not yet used the 2-cells of $\mathcal{M}$. In short, a 2-cell $\alpha : \mu \Rightarrow \nu$ contravariantly induces a substitution from $\Gamma, \blacksquare_\nu$ to $\Gamma, \blacksquare_\mu$. We will discuss this further in Section 4, but for now we only mention that this arrangement gives rise to an *admissible operation on types*: for each 2-cell we obtain an operation $(-)^\alpha$ such that $\Gamma, \blacksquare_\mu \vdash A$ type $@\, m$ implies $\Gamma, \blacksquare_\nu \vdash A^\alpha$ type $@\, m$.

In order to prove the admissibility of this operation we need a more expressive variable rule that builds in the action of 2-cells. The first iteration (TM/VAR/COUNIT) required that the lock and the variable annotation were an exact match. We relax this requirement by allowing for a mediating 2-cell:

TM/VAR/COMBINED
$$\frac{\mu, \nu : n \to m \qquad \alpha : \mu \Rightarrow \nu}{\Gamma, x : (\mu \mid A), \blacksquare_\nu \vdash x^\alpha : A^\alpha @\, n}$$

The superscript in x^α is now part of the syntax: each variable must be annotated with the 2-cell that 'unlocks' it and enables its occurrence, though we will still write x to mean x^{1_μ}. The final form of the variable rule, which appears as TM/VAR in Figure 2, is only a slight generalization of this last rule: it allows the variable to occur at positions other than the very front of the context. In fact, TM/VAR can be reduced to TM/VAR/COMBINED by using weakening to remove variables to the right of x, and then invoking functoriality to fuse all the locks to the right of x into a single one with modality $\mathsf{locks}(\Gamma_1)$.

The full elimination rule. Recall that the elimination rule for a single modality allowed us to plug in a term of type $\langle \mu \mid A \rangle$ for an assumption $x : (\mu \mid A)$. Some additional generality is needed to cover the case where the motive $x : (\nu \mid \langle \mu \mid A \rangle) \vdash B$ type $@\, m$ depends on x under a modality $\nu \neq 1$. This is where the composition of modalities in $\mathcal{M}$ comes in handy: our new rule will use it to absorb ν by replacing the assumption $x : (\nu \mid \langle \mu \mid A \rangle)$ with $x : (\nu \circ \mu \mid A)$.

The new rule, TM/MODAL-ELIM, is given in Figure 2. The simpler rule may be recovered by setting $\nu \triangleq 1$. In this simpler case, we will suppress the subscript 1 on let, just as in TM/MODAL-ELIM/SINGLE-MODALITY. However, many natural examples require eliminations where $\nu \neq 1$. For instance, in Section 3 we show that $\langle \nu \circ \mu \mid A \rangle \simeq \langle \nu \mid \langle \mu \mid A \rangle \rangle$. The function from the right-hand side to the left crucially depends on the ability to pattern-match on a variable $x : (\nu \mid \langle \mu \mid A \rangle)$, which requires the stronger TM/MODAL-ELIM.

Definitional equality in MTT. A perennial problem in type theory is that of deciding where the boundary between those equalities that are *provable* in the system (e.g. using various forms of induction), and those that are *definitional*, i.e. hold by fiat. While we have simply followed standard practices in the MLTT connectives at each mode, the situation is somewhat more complicated regarding modal types. On the one hand, we have the expected β-rule TM/MODAL-BETA: see Figure 2. On the other hand, we do not include any definitional η-rules: as the eliminator is a *positive* pattern-matching construct, the proper η-rule would need *commuting conversions*, which would enormously complicate the metatheory.

Notational conventions. In the rest of the paper we shall make use of the following notational conventions.

Notation 2.1. When opening a modal term under the modality 1 we will suppress the 1 in the let_1 part of the term, and write $\mathsf{let}\ \mathsf{mod}_\mu(\_) \leftarrow M$ in N instead.

Notation 2.2. As remarked before, Coquand-style universes do not require the introduction of codes that represent various types in the universe, for they are definable. Nevertheless, in examples we will often suppress both $\mathsf{El}(-)$ and $\mathsf{Code}(-)$, and in some straightfoward cases even elide the coercion $\Uparrow-$. This not only makes our terms more perspicuous, but can also be formally justified by an *elaboration procedure* which inserts the missing isomorphisms and coercions when needed.

3. Programming with Modalities

In this section we show how MTT can be used to program and reason with modalities. We we identify a handful of basic modal combinators which demonstrate the behaviour of our modal types. Then, in Section 3.2 we use them to present a type theory featuring an idempotent comonad with almost no additional effort.

3.1. **Modal Combinators.** We first show how each 2-cell $\alpha : \mu \Rightarrow \nu$ with $\mu, \nu : n \rightarrow m$ induces a natural transformation $\langle \mu \mid - \rangle \rightarrow \langle \nu \mid - \rangle$. We call the components of this natural transformation *coercions*. Given $\Gamma, \blacksquare_\mu \vdash A\ \mathsf{type}_1 @ m$, define

$$\mathbf{coe}[\alpha : \mu \Rightarrow \nu](-)\ :\ \langle \mu \mid A \rangle \rightarrow \langle \nu \mid A^\alpha \rangle$$
$$\mathbf{coe}[\alpha : \mu \Rightarrow \nu](x) \triangleq \mathsf{let}\ \mathsf{mod}_\mu(z) \leftarrow x\ \mathsf{in}\ \mathsf{mod}_\nu(z^\alpha)$$

The heart of this combinator is a use of the rule TM/VAR. This operation completes the correspondence sketched in Section 1: objects of $\mathcal{M}$ correspond to modes, morphisms to modalities, and 2-cells to coercions.

Additionally, the assignment $\mu \mapsto \langle \mu \mid - \rangle$ is *functorial*. Unlike the action of locks, this functoriality is not definitional, but only a type-theoretic *equivalence* [Uni13, §4]. Fixing $\nu : o \rightarrow n$, $\mu : n \rightarrow m$, and $\Gamma, \blacksquare_{\mu \circ \nu} \vdash A\ \mathsf{type}_1 @ m$, we let

$$\mathbf{comp}_{\mu,\nu} : \langle \mu \mid \langle \nu \mid A \rangle \rangle \rightarrow \langle \mu \circ \nu \mid A \rangle$$
$$\mathbf{comp}_{\mu,\nu}(x) \triangleq \mathsf{let}\quad \mathsf{mod}_\mu(x_0) \leftarrow x\ \mathsf{in}$$
$$\mathsf{let}_\mu\ \mathsf{mod}_\nu(x_1) \leftarrow x_0\ \mathsf{in}$$
$$\mathsf{mod}_{\mu \circ \nu}(x_1)$$

and

$$\mathbf{comp}^{-1}_{\mu,\nu} : \langle \mu \circ \nu \mid A \rangle \rightarrow \langle \mu \mid \langle \nu \mid A \rangle \rangle$$
$$\mathbf{comp}^{-1}_{\mu,\nu}(x) \triangleq \mathsf{let}\ \mathsf{mod}_{\mu \circ \nu}(x_0) \leftarrow x\ \mathsf{in}\ \mathsf{mod}_\mu(\mathsf{mod}_\nu(x_0))$$

We elide the 2-cell annotations on variables, as they are all identities (i.e. we only need TM/VAR/COUNIT). Even in this small example the context equations for locks are essential: for $\langle \mu \mid \langle \nu \mid A \rangle \rangle$ to be a valid type we need that $\Gamma, \blacksquare_\mu, \blacksquare_\nu = \Gamma, \blacksquare_{\mu \circ \nu}$, which is ensured by CX/COMPOSE. Furthermore, observe that $\mathbf{comp}_{\mu,\nu}$ crucially relies on the multimodal elimination rule TM/MODAL-ELIM: we must pattern-match on x_0, which is under μ in the context.

Similarly, fixing $\Gamma \vdash A\ \mathsf{type}_1\ @\ m$ we have

$$\mathbf{triv}(-) : \langle 1 \mid A \rangle \rightarrow A \qquad\qquad \mathbf{triv}^{-1}(-) : A \rightarrow \langle 1 \mid A \rangle$$
$$\mathbf{triv}(x) \triangleq \mathsf{let}\ \mathsf{mod}_1(x_0) \leftarrow x\ \mathsf{in}\ x_0 \qquad \mathbf{triv}^{-1}(x) \triangleq \mathsf{mod}_1(x)$$

In both cases, these combinators are only propositionally inverse. For example, the proof for one direction of the composition combinator is

$$\_ : (x : \langle \mu \mid \langle \nu \mid A \rangle \rangle) \rightarrow \mathsf{Id}_{\langle \mu | \langle \nu | A \rangle \rangle}(x, \mathbf{comp}^{-1}_{\mu,\nu}(\mathbf{comp}_{\mu,\nu}(x)))$$
$$\_ \triangleq \lambda x.\ \mathsf{let}\ \mathsf{mod}_\mu(x_0) \leftarrow x\ \mathsf{in}\ \mathsf{let}_\mu\ \mathsf{mod}_\nu(x_1) \leftarrow x_0\ \mathsf{in}\ \mathsf{refl}(\mathsf{mod}_\mu(\mathsf{mod}_\nu(x)))$$

This is in many ways a typical example: we use the modal elimination rule to induct on a modally-typed term, which reduces it to a term of the form $\mathsf{mod}(-)$. This is just enough to make various terms compute, and the result then follows by reflexivity.

As a final example, we will show that each modal type satisfies *the K axiom*,[1] a central axiom of Kripke-style modal logics. This combinator will be immediately recognizable to functional programmers: it is the term that witnesses that $\langle \mu \mid - \rangle$ is an *applicative functor* [MP08].

$$- \circledast_\mu - \ :\ \langle \mu \mid A \rightarrow B \rangle \rightarrow \langle \mu \mid A \rangle \rightarrow \langle \mu \mid B \rangle$$
$$f \circledast_\mu a \ \triangleq \mathsf{let}\ \mathsf{mod}_\mu(f_0) \leftarrow f\ \mathsf{in}\ \mathsf{let}\ \mathsf{mod}_\mu(a_0) \leftarrow a\ \mathsf{in}\ \mathsf{mod}_\mu(f_0(a_0))$$

We can also define a stronger combinator which corresponds to a dependent form of the Kripke axiom [BCM$^+$20] along the same lines. As it generalizes $\circledast_\mu$ to dependent products, this operation has precisely the same implementation but a more complex type:

$$\langle \mu \mid (x : A) \rightarrow B \rangle \rightarrow (x_0 : \langle \mu \mid A \rangle) \rightarrow (\mathsf{let}\ \mathsf{mod}_\mu(x) \leftarrow x_0\ \mathsf{in}\ \langle \mu \mid B \rangle)$$

In order to ensure that $\langle \mu \mid B \rangle$ is well-typed, the context must contain $x : (\mu \mid A)$, but instead we have bound $x_0 : (1 \mid \langle \mu \mid A \rangle)$. We correct this mismatch by eliminating x_0 and binding the result to x.

3.2. Idempotent Comonads in MTT.

A great deal of prior work in modal type theory has focused on *comonads* [PD01, dR15, Shu18, GSB19a], and in particular *idempotent* comonads. [Shu18, Theorem 4.1] has shown that such modalities necessitate changes to the judgmental structure, as the only idempotent comonads that are internally definable in type theory are of the form $- \times U$ for some proposition U. In this section we present a mode theory for idempotent comonads, and prove that the resulting type theory internally satisfies the expected equations. In fact, we only use the combinators of the previous section.

We define the mode theory $\mathcal{M}_{\mathsf{ic}}$ to consist of a single mode m, and a single non-trivial morphism $\mu : m \rightarrow m$. We will enforce idempotence by setting $\mu \circ \mu = \mu$. Finally, in order

---

[1]Not to be confused with Streicher's *axiom K*.

to induce a morphism $\langle \mu \mid A \rangle \to A$ we include a unique non-trivial 2-cell $\epsilon : \mu \Rightarrow 1$. In order to ensure that this 2-cell to be unique, we add equations such as $\epsilon \star 1_\mu = 1_\mu \star \epsilon : \mu \circ \mu \Rightarrow \mu$, where $\star$ denotes the horizontal composition of 2-cells. The resulting mode theory is a 2-category, albeit a very simple one: it is in fact only a *poset-enriched* category.

We can show that $\langle \mu \mid A \rangle$ is a comonad by defining the expected operations using the combinators of Section 3.1:

$$\mathsf{dup}_A : \langle \mu \mid A \rangle \to \langle \mu \mid \langle \mu \mid A \rangle \rangle \qquad \mathsf{extract}_A : \langle \mu \mid A \rangle \to A^\epsilon$$
$$\mathsf{dup}_A \triangleq \mathbf{comp}^{-1}_{\mu,\mu} \qquad\qquad\qquad \mathsf{extract}_A \triangleq \mathbf{triv}^{-1}(-) \circ \mathbf{coe}[\epsilon : \mu \Rightarrow 1]$$

We must also show that dup_A and $\mathsf{extract}_A$ satisfy the comonad laws, but that automatically follows from general facts pertaining to $\mathbf{coe}$ and $\mathbf{comp}$.[2] This is indicative of the benefits of using MTT: every general result about it also applies to this instance, including the canonicity theorem of Section 5.

4. Algebraic Syntax

Until this point we have presented a curated, high-level view of MTT, and we have avoided any discussion of its metatheory. Yet, syntactic matters can be quite complex, and have historically proven to be sticking points for modal type theory. While such details are not necessary for the casual reader, it is essential to validate that MTT is syntactically well-behaved, enjoying e.g. a substitution principle. The aim of this section is to provide a setting for this study: we introduce the formal counterpart of MTT, which is given as a *generalized algebraic theory* (GAT) [Car78, KKA19].

Historically, GATs were used in the semantics of type theory, but modern techniques show that they are also useful in the analysis of syntax. For example, recasting MTT as a GAT naturally leads us to include *explicit substitutions* [Cur90, ML92, Gra11] in the syntax. Thus, substitution in MTT is not a metatheoretic operation on raw terms, but a syntactic operation within the theory. This presentation helps us carefully state the equations that govern substitutions and their interaction with type formers. We consequently obtain an elegant *substitution calculus*, which can often be quite complex for modal type theories.

This approach proffers a number of technical advantages. Amongst other things, the theorems proven in the aforementioned works on GATs imply the following points:

(1) We absolve ourselves from having to prove tedious syntactic metatheorems, e.g. admissibility of substitution.
(2) We automatically obtain a notion of *model* of our theory, which is given in entirely algebraic terms.
(3) We obtain a notion of *homomorphism of models*. (NB that this notion is rather *strict* and not fit for every purpose.)
(4) In an equally automatic fashion, we obtain an *initial model* for the algebraic theory, which we consider as our main formal object of study.
(5) The unique morphism of models from this initial model to any other is the *semantic interpretation map*. We then have no need to explicitly describe these semantic maps and prove that they are well-defined on derivations, as done e.g. by [Hof97].

---

[2]In particular, our modal combinators satisfy a variant of the *interchange law* of a 2-category.

While this approach is straightforward and uncluttered, some readers might object to the lack of a more traditional formulation, e.g. a *named syntax* with variables and a metatheoretic substitution operation, like the one we informally presented in Section 2. We believe that it is indeed possible to define such a syntax and systematically show how to *elaborate* its terms to the algebraic syntax.

However, such a named syntax would not be directly suitable for implementation: for that purpose we ought to develop an entirely different *algorithmic syntax*. We believe that such a syntax can be constructed as an extension of existing *bidirectional* presentations of type theory [Coq96, PT00] as has been done for existing modal calculi [GSB19a]. Such a bidirectional presentation would occupy a midpoint between the maximally annotated algebraic syntax we present here, and the more typical named syntax of Section 2: it would contain only a select few annotations to ensure the decidability of typechecking, yet maintain readability. The development of such a syntax is a substantial undertaking that requires a proof of normalization, and is orthogonal to the foundational metatheoretic results that we seek to develop here. We thus refrain from developing it, and instead work directly with the GAT.

4.1. **Sorts.** We begin by defining the different *sorts* (contexts, types, terms, etc.) that constitute our type theory. In order to support multiple modes, our sorts will be parameterized in modes. Thus, rather than having a single sort of types, we will have a sort of types *at mode* $m \in \mathcal{M}$, and likewise for contexts at mode m, terms at mode m, etc.

Moreover, we take care to index our types by *levels*. The reason for doing so was discussed in Section 2.1: we seek to introduce a hierarchy of sizes, which we can then use to introduce universes à la [Coq13]. We stratify our types in two levels, drawn from the set $\mathcal{L} = \{0, 1\}$. There are no technical obstacles on the way to a richer hierarchy, but two levels suffice for our purposes: we aim to divide our types into *small types* (i.e. those that can be reified in a universe) and *large types* (which also include the universe itself). In order to enforce cumulativity we will also include an explicit *coercion* operator, which includes small types into large types.

The levelled approach raises an obvious question: on which level should we admit terms? We could follow the approach of [Ste19] in allowing terms at both, but this requires the introduction of term-level coercions, which then require equations relating term formers at different levels. Thus, for the sake of simplicity we will only allow the formation of terms at large types. Similarly, we will only allow the extension of a context by a large type.

MTT has four families of sorts, which are introduced by the following rules:

$$\frac{m : \mathcal{M}}{\mathsf{ctx}_m \text{ sort}} \qquad \frac{\ell : \mathcal{L} \qquad m : \mathcal{M} \qquad \Gamma : \mathsf{ctx}_m}{\mathsf{type}_m^\ell(\Gamma) \text{ sort}} \qquad \frac{m : \mathcal{M} \qquad \Gamma : \mathsf{ctx}_m \qquad A : \mathsf{type}_m^1(\Gamma)}{\mathsf{tm}_m(\Gamma, A) \text{ sort}}$$

$$\frac{m : \mathcal{M} \qquad \Gamma, \Delta : \mathsf{ctx}_m}{\mathsf{sb}_m(\Gamma, \Delta) \text{ sort}}$$

In the interest of clarity we will use the following shorthands:

$$\Gamma \text{ ctx} @ m \triangleq \Gamma : \mathsf{ctx}_m \qquad\qquad \Gamma \vdash A \text{ type}_\ell @ m \triangleq A : \mathsf{type}_m^\ell(\Gamma)$$
$$\Gamma \vdash M : A @ m \triangleq M : \mathsf{tm}_m(\Gamma, A) \qquad\qquad \Gamma \vdash \delta : \Delta @ m \triangleq \delta : \mathsf{sb}_m(\Gamma, \Delta)$$

$$\boxed{\Gamma \text{ ctx} @ m}$$

$$\frac{}{\cdot \text{ ctx} @ m} \qquad \frac{\Gamma \text{ ctx} @ m \qquad \mu : \text{Hom}_{\mathcal{M}}(n, m)}{\Gamma.\blacksquare_\mu \text{ ctx} @ n}$$

$$\frac{\Gamma \text{ ctx} @ m \qquad \mu : \text{Hom}_{\mathcal{M}}(n, m) \qquad \Gamma.\blacksquare_\mu \vdash A \text{ type}_1 @ n}{\Gamma.(\mu \mid A) \text{ ctx} @ m}$$

$$\frac{\Gamma \text{ ctx} @ m \qquad \nu : \text{Hom}_{\mathcal{M}}(o, n) \qquad \mu : \text{Hom}_{\mathcal{M}}(n, m)}{\Gamma.\blacksquare_\mu.\blacksquare_\nu = \Gamma.\blacksquare_{\mu \circ \nu} \text{ ctx} @ o} \qquad \frac{\Gamma \text{ ctx} @ m}{\Gamma.\blacksquare_1 = \Gamma \text{ ctx} @ m}$$

Figure 3: MTT Contexts

Even though we will use this more familiar notation, we will take no prisoners in terms of rigour: we will carefully avoid overloading and ambiguity, and we will enforce *presupposition*.

4.2. **Judgments.** We shall now introduce the type theory itself by writing down the constructors and equalities of its GAT. In the interest of brevity, we elide a number of standard rules, including

- the congruence rules pushing substitutions inside terms and types;
- the congruence rules pushing explicit lifts inside of type formers;
- the associativity, unit, and weakening laws for the explicit substitutions;
- the β laws for Π, Σ, $\mathbb{B}$ and Id;
- the η laws for Π and Σ;

The specification of the GAT is given in Figures 3–9. As the judgments are defined in a mutually recursive manner, the division of the rules between different figures is merely presentational. Given $\Delta \vdash \gamma : \Gamma @ m$ and $\Gamma.\blacksquare_\mu \vdash A \text{ type}_\ell @ m$ we write

$$\Delta.(\mu \mid A[\gamma.\blacksquare_\mu]) \vdash \gamma^+ \triangleq (\gamma \circ \uparrow).\mathbf{v}_0 : \Gamma.(\mu \mid A) @ m$$

for the 'weakened' substitution.

4.3. **Discussion.** We record some points on the generalized algebraic theory.

Modal dependent products. The algebraic presentation of MTT includes a primitive *modal dependent product* type $(\mu \mid A) \to B$. This is a combination of the modality $\langle \mu \mid - \rangle$ and the ordinary dependent product. Using a named syntax, it may be understood as

$$(x : (\mu \mid A)) \to B \triangleq (x_0 : \langle \mu \mid A \rangle) \to (\text{let mod}_\mu(x) \leftarrow x_0 \text{ in } B)$$

However, the modal types of MTT do not readily support a definitional η-equality, so this definition is not equivalent to the modal dependent product of the GAT. We use the latter because it is convenient for programming, and also has a natural semantics, which we will present in Section 5.2.1.

$$\boxed{\Gamma \vdash A \ \mathsf{type}_\ell \, @ \, m}$$

$$\frac{\Gamma \ \mathsf{ctx} \, @ \, m}{\Gamma \vdash \mathbb{B} \ \mathsf{type}_\ell \, @ \, m} \qquad \frac{\Gamma \ \mathsf{ctx} \, @ \, m}{\Gamma \vdash \mathsf{U} \ \mathsf{type}_1 \, @ \, m} \qquad \frac{\Gamma \ \mathsf{ctx} \, @ \, m \qquad \Gamma \vdash M : \mathsf{U} \, @ \, m}{\Gamma \vdash \mathsf{El}(M) \ \mathsf{type}_0 \, @ \, m}$$

$$\frac{\ell \leq \ell' \qquad \Gamma \ \mathsf{ctx} \, @ \, m \qquad \Gamma \vdash A \ \mathsf{type}_\ell \, @ \, m}{\Gamma \vdash \Uparrow A \ \mathsf{type}_{\ell'} \, @ \, m}$$

$$\frac{\Gamma \ \mathsf{ctx} \, @ \, m \qquad \Gamma \vdash A \ \mathsf{type}_\ell \, @ \, m \qquad \Gamma \vdash M, N : \Uparrow A \, @ \, m}{\Gamma \vdash \mathsf{Id}_A(M, N) \ \mathsf{type}_\ell \, @ \, m}$$

$$\frac{\Gamma \ \mathsf{ctx} \, @ \, m \qquad \mu : \mathrm{Hom}_{\mathcal{M}}(n, m) \qquad \Gamma.\blacksquare_\mu \vdash A \ \mathsf{type}_\ell \, @ \, n}{\Gamma \vdash \langle \mu \mid A \rangle \ \mathsf{type}_\ell \, @ \, m}$$

$$\frac{\mu : \mathrm{Hom}_{\mathcal{M}}(n, m) \qquad \Gamma \ \mathsf{ctx} \, @ \, m \qquad \Gamma.\blacksquare_\mu \vdash A \ \mathsf{type}_\ell \, @ \, n \qquad \Gamma.(\mu \mid \Uparrow A) \vdash B \ \mathsf{type}_\ell \, @ \, m}{\Gamma \vdash (\mu \mid A) \to B \ \mathsf{type}_\ell \, @ \, m}$$

$$\frac{\Gamma \ \mathsf{ctx} \, @ \, m \qquad \Gamma \vdash A \ \mathsf{type}_\ell \, @ \, m \qquad \Gamma.(1 \mid \Uparrow A) \vdash B \ \mathsf{type}_\ell \, @ \, m}{\Gamma \vdash \sum(A, B) \ \mathsf{type}_\ell \, @ \, m}$$

$$\frac{\Gamma, \Delta \ \mathsf{ctx} \, @ \, m \qquad \Delta \vdash A \ \mathsf{type}_\ell \, @ \, m \qquad \Gamma \vdash \delta : \Delta \, @ \, m}{\Gamma \vdash A[\delta] \ \mathsf{type}_\ell \, @ \, m}$$

Figure 4: MTT Types

$$\boxed{\Gamma \vdash \delta : \Delta \, @ \, m}$$

$$\frac{\Gamma \ \mathsf{ctx} \, @ \, m}{\Gamma \vdash \cdot : \cdot \, @ \, m} \qquad \frac{\Gamma \ \mathsf{ctx} \, @ \, n \qquad \mu : \mathrm{Hom}_{\mathcal{M}}(n, m) \qquad \Gamma.\blacksquare_\mu \vdash A \ \mathsf{type}_1 \, @ \, n}{\Gamma.(\mu \mid A) \vdash \uparrow : \Gamma \, @ \, m} \qquad \frac{\Gamma \ \mathsf{ctx} \, @ \, m}{\Gamma \vdash \mathsf{id} : \Gamma \, @ \, m}$$

$$\frac{\Gamma, \Delta, \Xi \ \mathsf{ctx} \, @ \, m \qquad \Gamma \vdash \gamma : \Delta \, @ \, m \qquad \Delta \vdash \delta : \Xi \, @ \, m}{\Gamma \vdash \delta \circ \gamma : \Xi \, @ \, m}$$

$$\frac{\Gamma, \Delta \ \mathsf{ctx} \, @ \, m \qquad \mu : \mathrm{Hom}_{\mathcal{M}}(n, m) \qquad \Gamma \vdash \delta : \Delta \, @ \, m}{\Gamma.\blacksquare_\mu \vdash \delta.\blacksquare_\mu : \Delta.\blacksquare_\mu \, @ \, n}$$

$$\frac{\Gamma \ \mathsf{ctx} \, @ \, m \qquad \mu, \nu : \mathrm{Hom}_{\mathcal{M}}(n, m) \qquad \alpha : \nu \Rightarrow \mu}{\Gamma.\blacksquare_\mu \vdash \mathbf{k}_\Gamma^\alpha : \Gamma.\blacksquare_\nu \, @ \, n}$$

$$\frac{\begin{array}{c} \mu : \mathrm{Hom}_{\mathcal{M}}(n, m) \\ \Gamma, \Delta \ \mathsf{ctx} \, @ \, m \qquad \Gamma \vdash \delta : \Delta \, @ \, m \qquad \Delta.\blacksquare_\mu \vdash A \ \mathsf{type}_1 \, @ \, n \qquad \Gamma.\blacksquare_\mu \vdash M : A[\delta.\blacksquare_\mu] \, @ \, n \end{array}}{\Gamma \vdash \delta.M : \Delta.(\mu \mid A) \, @ \, m}$$

Figure 5: MTT Substitutions

$$\boxed{\Gamma \vdash M : A \,@\, m}$$

$$\frac{\mu : \mathrm{Hom}_{\mathcal{M}}(n,m) \qquad \Gamma \text{ ctx} \,@\, m \qquad \Gamma.\blacksquare_\mu \vdash A \text{ type}_1 \,@\, n}{\Gamma.(\mu \mid A).\blacksquare_\mu \vdash \mathbf{v}_0 : A[\uparrow.\blacksquare_\mu] \,@\, n} \qquad\qquad \frac{\Gamma \text{ ctx} \,@\, m}{\Gamma \vdash \mathsf{tt}, \mathsf{ff} : \mathbb{B} \,@\, m}$$

$$\frac{\Gamma \text{ ctx} \,@\, m \qquad \Gamma.(1 \mid \mathbb{B}) \vdash A \text{ type}_1 \,@\, m \qquad \Gamma \vdash M_t : A[\mathsf{id}.\mathsf{tt}] \,@\, m \qquad \Gamma \vdash M_f : A[\mathsf{id}.\mathsf{ff}] \,@\, m \qquad \Gamma \vdash N : \mathbb{B} \,@\, m}{\Gamma \vdash \mathsf{if}(A; M_t; M_f; N) : A[\mathsf{id}.N] \,@\, m}$$

$$\frac{\Gamma \text{ ctx} \,@\, m \qquad \Gamma \vdash A \text{ type}_0 \,@\, m}{\Gamma \vdash \mathsf{Code}(A) : \mathsf{U} \,@\, m} \qquad\qquad \frac{\Gamma \text{ ctx} \,@\, m \qquad \Gamma \vdash A \text{ type}_1 \,@\, m \qquad \Gamma \vdash M : A \,@\, m}{\Gamma \vdash \mathsf{refl}(M) : \mathsf{Id}_A(M, M) \,@\, m}$$

$$\frac{\Gamma \text{ ctx} \,@\, m \qquad \Gamma \vdash A \text{ type}_1 \,@\, m \qquad \Gamma.(1 \mid A).(1 \mid A[\uparrow]).(1 \mid \mathsf{Id}_{A[\uparrow^2]}(\mathbf{v}_1, \mathbf{v}_0)) \vdash B \text{ type}_1 \,@\, m \qquad \Gamma.(1 \mid A) \vdash M : B[\uparrow.\mathbf{v}_0.\mathbf{v}_0.\mathsf{refl}(\mathbf{v}_0)] \,@\, m \qquad \Gamma \vdash N_0, N_1 : A \,@\, m \qquad \Gamma \vdash P : \mathsf{Id}_A(N_0, N_1) \,@\, m}{\Gamma \vdash \mathsf{J}(B, M, P) : B[\mathsf{id}.N_0.N_1.P] \,@\, m}$$

$$\frac{\Gamma \text{ ctx} \,@\, m \qquad \mu : \mathrm{Hom}_{\mathcal{M}}(n,m) \qquad \Gamma.\blacksquare_\mu \vdash A \text{ type}_1 \,@\, n \qquad \Gamma.\blacksquare_\mu \vdash M : A \,@\, n}{\Gamma \vdash \mathsf{mod}_\mu(M) : \langle \mu \mid A \rangle \,@\, m}$$

$$\frac{\nu : \mathrm{Hom}_{\mathcal{M}}(o,n) \qquad \mu : \mathrm{Hom}_{\mathcal{M}}(n,m) \qquad \Gamma \text{ ctx} \,@\, m \qquad \Gamma.\blacksquare_\mu.\blacksquare_\nu \vdash A \text{ type}_1 \,@\, o \qquad \Gamma.\blacksquare_\mu \vdash M_0 : \langle \nu \mid A \rangle \,@\, n \qquad \Gamma.(\mu \mid \langle \nu \mid A \rangle) \vdash B \text{ type}_1 \,@\, m \qquad \Gamma.(\mu \circ \nu \mid A) \vdash M_1 : B[\uparrow.\mathsf{mod}_\nu(\mathbf{v}_0)] \,@\, m}{\Gamma \vdash \mathsf{let}_\mu \, \mathsf{mod}_\nu(\_) \leftarrow M_0 \text{ in } M_1 : B[\mathsf{id}.M_0] \,@\, m}$$

$$\frac{\mu : \mathrm{Hom}_{\mathcal{M}}(n,m) \qquad \Gamma \text{ ctx} \,@\, m \qquad \Gamma.\blacksquare_\mu \vdash A \text{ type}_1 \,@\, n \qquad \Gamma.(\mu \mid A) \vdash B \text{ type}_1 \,@\, m \qquad \Gamma.(\mu \mid A) \vdash M : B \,@\, m}{\Gamma \vdash \lambda(M) : (\mu \mid A) \to B \,@\, m}$$

$$\frac{\mu : \mathrm{Hom}_{\mathcal{M}}(n,m) \qquad \Gamma \text{ ctx} \,@\, m \qquad \Gamma.\blacksquare_\mu \vdash A \text{ type}_1 \,@\, n \qquad \Gamma.(\mu \mid A) \vdash B \text{ type}_1 \,@\, m \qquad \Gamma \vdash M_0 : (\mu \mid A) \to B \,@\, m \qquad \Gamma.\blacksquare_\mu \vdash M_1 : A \,@\, n}{\Gamma \vdash M_0(M_1) : B[\mathsf{id}.M_1] \,@\, m}$$

$$\frac{\Gamma \text{ ctx} \,@\, m \qquad \Gamma \vdash A \text{ type}_1 \,@\, m \qquad \Gamma.(1 \mid A) \vdash B \text{ type}_1 \,@\, m \qquad \Gamma \vdash M_0 : A \,@\, m \qquad \Gamma \vdash M_1 : B[\mathsf{id}.M_0] \,@\, m}{\Gamma \vdash (M_0, M_1) : \sum(A, B) \,@\, m}$$

$$\frac{\Gamma \text{ ctx} \,@\, m \qquad \Gamma \vdash A \text{ type}_1 \,@\, m \qquad \Gamma.(1 \mid A) \vdash B \text{ type}_1 \,@\, m \qquad \Gamma \vdash M : \sum(A, B) \,@\, m}{\Gamma \vdash \mathsf{pr}_0(M) : A \,@\, m \qquad \Gamma \vdash \mathsf{pr}_1(M) : B[\mathsf{id}.\mathsf{pr}_0(M)] \,@\, m}$$

$$\frac{\Gamma, \Delta \text{ ctx} \,@\, m \qquad \Delta \vdash A \text{ type}_1 \,@\, m \qquad \Gamma \vdash \delta : \Delta \,@\, m \qquad \Delta \vdash M : A \,@\, m}{\Gamma \vdash M[\delta] : A[\delta] \,@\, m}$$

Figure 6: MTT Terms

$$\boxed{\Gamma \vdash A = B \; \mathsf{type}_\ell \, @ \, m}$$

$$\frac{\mu : \mathrm{Hom}_{\mathcal{M}}(n,m) \qquad \Gamma, \Delta \; \mathsf{ctx} \, @ \, m \qquad \Gamma \vdash \delta : \Delta \, @ \, m \qquad \Delta.\blacksquare_\mu \vdash A \; \mathsf{type}_\ell \, @ \, n}{\Gamma \vdash \langle \mu \mid A \rangle [\delta] = \langle \mu \mid A[\delta.\blacksquare_\mu] \rangle \; \mathsf{type}_\ell \, @ \, m}$$

$$\frac{\mu : \mathrm{Hom}_{\mathcal{M}}(n,m)}{\Gamma, \Delta \; \mathsf{ctx} \, @ \, m \qquad \Gamma \vdash \delta : \Delta \, @ \, m \qquad \Gamma.\blacksquare_\mu \vdash A \; \mathsf{type}_\ell \, @ \, n \qquad \Gamma.(\mu \mid \Uparrow A) \vdash B \; \mathsf{type}_\ell \, @ \, m}{\Gamma \vdash ((\mu \mid A) \to B)[\delta] = (\mu \mid A[\delta.\blacksquare_\mu]) \to B[\delta^+] \; \mathsf{type}_\ell \, @ \, m}$$

$$\frac{\Gamma \vdash A \; \mathsf{type}_\ell \, @ \, m}{\Gamma \vdash \Uparrow A = A \; \mathsf{type}_\ell \, @ \, m} \qquad\qquad \frac{\ell_0 \le \ell_1 \le \ell_2 \qquad \Gamma \vdash A \; \mathsf{type}_{\ell_0} \, @ \, m}{\Gamma \vdash \Uparrow\Uparrow A = \Uparrow A \; \mathsf{type}_{\ell_2} \, @ \, m}$$

$$\frac{\ell \le \ell' \qquad \mu \in \mathrm{Hom}_{\mathcal{M}}(n,m) \qquad \Gamma \; \mathsf{ctx} \, @ \, m \qquad \Gamma.\blacksquare_\mu \vdash A \; \mathsf{type}_\ell \, @ \, n \qquad \Gamma.(\mu \mid \Uparrow A) \vdash B \; \mathsf{type}_\ell \, @ \, m}{\Gamma \vdash \Uparrow((\mu \mid A) \to B) = (\mu \mid \Uparrow A) \to \Uparrow B \; \mathsf{type}_{\ell'} \, @ \, m}$$

$$\frac{\Gamma \; \mathsf{ctx} \, @ \, m \qquad \Gamma \vdash A \; \mathsf{type}_0 \, @ \, m}{\Gamma \vdash \mathsf{El}(\mathsf{Code}(A)) = A \; \mathsf{type}_0 \, @ \, m} \qquad\qquad \frac{\Gamma, \Delta \; \mathsf{ctx} \, @ \, m \qquad \Gamma \vdash \delta : \Delta \, @ \, m}{\Gamma \vdash \mathsf{U}[\delta] = \mathsf{U} \; \mathsf{type}_1 \, @ \, m}$$

Figure 7: Equality of Types

$$\boxed{\Gamma \vdash M = N : A \, @ \, m}$$

$$\frac{\mu : \mathrm{Hom}_{\mathcal{M}}(n,m) \qquad \Gamma \vdash \delta : \Delta \, @ \, m \qquad \Delta.\blacksquare_\mu \vdash A \; \mathsf{type}_1 \, @ \, n \qquad \Gamma.\blacksquare_\mu \vdash M : A[\delta.\blacksquare_\mu] \, @ \, n}{\Gamma.\blacksquare_\mu \vdash \mathbf{v}_0[(\delta.M).\blacksquare_\mu] = M : A[\delta.\blacksquare_\mu] \, @ \, n}$$

$$\frac{\Gamma \; \mathsf{ctx} \, @ \, m \qquad \Gamma \vdash M : \mathsf{U} \, @ \, m}{\Gamma \vdash \mathsf{Code}(\mathsf{El}(M)) = M : \mathsf{U} \, @ \, m}$$

$$\frac{\nu : \mathrm{Hom}_{\mathcal{M}}(o,n)}{\begin{array}{c} \mu : \mathrm{Hom}_{\mathcal{M}}(n,m) \qquad \Gamma \; \mathsf{ctx} \, @ \, m \qquad \Gamma.\blacksquare_\mu.\blacksquare_\nu \vdash A \; \mathsf{type}_1 \, @ \, o \qquad \Gamma.\blacksquare_\mu.\blacksquare_\nu \vdash M_0 : A \, @ \, o \\ \Gamma.(\mu \mid \langle \nu \mid A \rangle) \vdash B \; \mathsf{type}_1 \, @ \, m \qquad \Gamma.(\mu \circ \nu \mid A) \vdash M_1 : B[\uparrow.\mathsf{mod}_\mu(\mathbf{v}_0)] \, @ \, m \end{array}}{\Gamma \vdash \mathsf{let}_\mu \, \mathsf{mod}_\nu(\_\,) \leftarrow \mathsf{mod}_\nu(M_0) \; \mathsf{in} \; M_1 = M_1[\mathsf{id}.M_0] : B[\mathsf{id}.\mathsf{mod}_\nu(M_0)] \, @ \, m}$$

$$\frac{\mu : \mathrm{Hom}_{\mathcal{M}}(n,m)}{\begin{array}{c} \Delta, \Gamma \; \mathsf{ctx} \, @ \, m \qquad \Gamma \vdash \delta : \Delta \, @ \, m \qquad \Delta.\blacksquare_\mu \vdash A \; \mathsf{type}_1 \, @ \, n \qquad \Delta.\blacksquare_\mu \vdash M : A \, @ \, n \end{array}}{\Gamma \vdash \mathsf{mod}_\mu(M)[\delta] = \mathsf{mod}_\mu(M[\delta.\blacksquare_\mu]) : \langle \mu \mid A[\delta.\blacksquare_\mu] \rangle \, @ \, m}$$

$$\frac{\begin{array}{c} \nu : \mathrm{Hom}_{\mathcal{M}}(o,n) \qquad \mu : \mathrm{Hom}_{\mathcal{M}}(n,m) \\ \Gamma, \Delta \; \mathsf{ctx} \, @ \, m \qquad \Gamma \vdash \delta : \Delta \, @ \, m \qquad \Delta.\blacksquare_\mu.\blacksquare_\nu \vdash A \; \mathsf{type}_1 \, @ \, o \qquad \Delta.\blacksquare_\mu \vdash M_0 : \langle \nu \mid A \rangle \, @ \, n \\ \Delta.(\mu \mid \langle \nu \mid A \rangle) \vdash B \; \mathsf{type}_1 \, @ \, m \qquad \Delta.(\mu \circ \nu \mid A) \vdash M_1 : B[\uparrow.\mathsf{mod}_\mu(\mathbf{v}_0)] \, @ \, m \end{array}}{\Gamma \vdash (\mathsf{let}_\mu \, \mathsf{mod}_\nu(\_\,) \leftarrow M_0 \; \mathsf{in} \; M_1)[\delta] = \mathsf{let}_\mu \, \mathsf{mod}_\nu(\_\,) \leftarrow M_0[\delta.\blacksquare_\mu] \; \mathsf{in} \; M_1[\delta^+] : B[\delta.M_0[\delta.\blacksquare_\mu]] \, @ \, m}$$

Figure 8: Equality of Terms

$$\boxed{\Gamma \vdash \gamma = \delta : \Delta \,@\, m}$$

$$\frac{\begin{array}{ccccc} \Gamma_0, \Gamma_1 \text{ ctx} \,@\, n & & \Delta \text{ ctx} \,@\, n & & \Delta.\text{\faLock}_\mu \vdash A \text{ type}_1 \,@\, m \\ \mu : \text{Hom}_{\mathcal{M}}(m,n) & \Gamma_0 \vdash \gamma : \Gamma_1 \,@\, n & \Gamma_1 \vdash \delta : \Delta \,@\, n & \Gamma_1.\text{\faLock}_\mu \vdash M : A[\delta.\text{\faLock}_\mu] \,@\, m \end{array}}{\Gamma_0 \vdash (\delta.M) \circ \gamma = (\delta \circ \gamma).M[\gamma.\text{\faLock}_\mu] : \Delta.(\mu \mid A) \,@\, n}$$

$$\frac{\Gamma, \Delta \text{ ctx} \,@\, o \qquad \mu : \text{Hom}_{\mathcal{M}}(m,n) \qquad \nu : \text{Hom}_{\mathcal{M}}(n,o) \qquad \Gamma \vdash \delta : \Delta \,@\, m}{\Gamma.\text{\faLock}_{\nu\circ\mu} \vdash \delta.\text{\faLock}_{\nu\circ\mu} = \delta.\text{\faLock}_\nu.\text{\faLock}_\mu : \Delta.\text{\faLock}_{\nu\circ\mu} \,@\, m}$$

$$\frac{\Gamma, \Delta \text{ ctx} \,@\, m \qquad \Gamma \vdash \delta : \Delta \,@\, m}{\Gamma \vdash \delta.\text{\faLock}_1 = \delta : \Delta \,@\, m} \qquad\qquad \frac{\Gamma \text{ ctx} \,@\, n \qquad \mu : \text{Hom}_{\mathcal{M}}(m,n)}{\Gamma.\text{\faLock}_\mu \vdash \text{id}.\text{\faLock}_\mu = \text{id} : \Gamma.\text{\faLock}_\mu \,@\, m}$$

$$\frac{\Gamma, \Delta, \Xi \text{ ctx} \,@\, n \qquad \mu : \text{Hom}_{\mathcal{M}}(m,n) \qquad \Gamma \vdash \delta : \Delta \,@\, n \qquad \Delta \vdash \xi : \Xi \,@\, n}{\Gamma.\text{\faLock}_\mu \vdash (\xi \circ \delta).\text{\faLock}_\mu = \xi.\text{\faLock}_\mu \circ \delta.\text{\faLock}_\mu : \Xi.\text{\faLock}_\mu \,@\, m}$$

$$\frac{\Gamma \text{ ctx} \,@\, n \qquad \mu : \text{Hom}_{\mathcal{M}}(m,n)}{\Gamma.\text{\faLock}_\mu \vdash \text{id} = \text{\faKey}_\Gamma^{1_\mu} : \Gamma.\text{\faLock}_\mu \,@\, m}$$

$$\frac{\Gamma, \Delta \text{ ctx} \,@\, n \qquad \mu, \nu : \text{Hom}_{\mathcal{M}}(m,n) \qquad \Gamma \vdash \delta : \Delta \,@\, n \qquad \alpha : \nu \Rightarrow \mu}{\Gamma.\text{\faLock}_\mu \vdash \text{\faKey}_\Gamma^\alpha \circ (\delta.\text{\faLock}_\mu) = (\delta.\text{\faLock}_\nu) \circ \text{\faKey}_\Delta^\alpha : \Delta.\text{\faLock}_\nu \,@\, m}$$

$$\frac{\Gamma \text{ ctx} \,@\, m \qquad \mu_0, \mu_1, \mu_2 : \text{Hom}_{\mathcal{M}}(n,m) \qquad \alpha_0 : \mu_0 \Rightarrow \mu_1 \qquad \alpha_1 : \mu_1 \Rightarrow \mu_2}{\Gamma.\text{\faLock}_{\mu_2} \vdash \text{\faKey}_\Gamma^{\alpha_1 \circ \alpha_0} = \text{\faKey}_\Gamma^{\alpha_0} \circ \text{\faKey}_\Gamma^{\alpha_1} : \Gamma.\text{\faLock}_{\mu_0} \,@\, n}$$

$$\frac{\Gamma \text{ ctx} \,@\, m \qquad \nu_0, \nu_1 : \text{Hom}_{\mathcal{M}}(o,n) \qquad \mu_0, \mu_1 : \text{Hom}_{\mathcal{M}}(n,m) \qquad \beta : \nu_0 \Rightarrow \nu_1 \qquad \alpha : \mu_0 \Rightarrow \mu_1}{\Gamma.\text{\faLock}_{\mu_0 \circ \nu_0} \vdash \text{\faKey}_\Gamma^{\alpha \star \beta} = \text{\faKey}_\Gamma^\alpha.\text{\faLock}_{\nu_1} \circ \text{\faKey}_{\Gamma.\text{\faLock}_{\mu_0}}^\beta : \Gamma.\text{\faLock}_{\mu_1 \circ \nu_1} \,@\, o}$$

Figure 9: Equality of Substitutions

Modal substitutions. In addition to the usual rules, MTT features substitutions corresponding to the 1-cells and 2-cells of the mode theory. First, recall that for each modality $\mu : n \to m$ we have the operation $\text{\faLock}_\mu$ on contexts. Its action extends to substitutions:

$$\text{SB/LOCK}$$
$$\frac{\mu : n \to m \qquad \Gamma \vdash \delta : \Delta \,@\, m}{\Gamma.\text{\faLock}_\mu \vdash \delta.\text{\faLock}_\mu : \Delta.\text{\faLock}_\mu \,@\, n}$$

Second, each 2-cell $\alpha : \mu \Rightarrow \nu$ induces a *natural transformation* between $\text{\faLock}_\nu$ and $\text{\faLock}_\mu$, whose component at Γ is the 'key' substitution

$$\text{SB/KEY}$$
$$\frac{\alpha : \mu \Rightarrow \nu}{\Gamma.\text{\faLock}_\nu \vdash \text{\faKey}_\Gamma^\alpha : \Gamma.\text{\faLock}_\mu \,@\, n}$$

Recalling that $\mathcal{M}^{\text{coop}}$ is the 2-category with morphisms and 2-cells opposite from $\mathcal{M}$, we see that these substitutions come with equations postulating that $-.\text{\faLock}_\mu$ is a functor, $\text{\faKey}_\Gamma^\alpha$ is a natural transformation, and that together they form a 2-functor $\mathcal{M}^{\text{coop}} \to \textbf{Cat}$. As a consequence, our type theory is forced to contain a calculus of (strict) 2-categories. Indeed,

the given equations for keys above suffice to derive the two ways of internally stating the *interchange laws*, viz.

$$\frac{\Gamma \; \mathsf{ctx} \, @ \, m \qquad \nu_0, \nu_1, \nu_2 : \mathrm{Hom}_{\mathcal{M}}(o, n) \qquad \mu_0, \mu_1, \mu_2 : \mathrm{Hom}_{\mathcal{M}}(n, m)}{\alpha_0 : \mu_0 \Rightarrow \mu_1 \qquad \alpha_1 : \mu_1 \Rightarrow \mu_2 \qquad \beta_0 : \nu_0 \Rightarrow \nu_1 \qquad \beta_1 : \nu_1 \Rightarrow \nu_2}$$

$$\overline{\Gamma.\blacksquare_{\mu_2 \circ \nu_2} \vdash \mathbf{key}_\Gamma^{\alpha_0 \star \beta_0} \circ \mathbf{key}_\Gamma^{\alpha_1 \star \beta_1} = \mathbf{key}_\Gamma^{\alpha_1 \circ \alpha_0}.\blacksquare_{\nu_0} \circ \mathbf{key}_{\Gamma.\blacksquare_{\mu_2}}^{\beta_1 \circ \beta_0} : \Gamma.\blacksquare_{\mu_0 \circ \nu_0} \, @ \, o}$$

$$\frac{\Gamma \; \mathsf{ctx} \, @ \, m \qquad \nu_0, \nu_1, \nu_2 : \mathrm{Hom}_{\mathcal{M}}(o, n) \qquad \mu_0, \mu_1, \mu_2 : \mathrm{Hom}_{\mathcal{M}}(n, m)}{\alpha_0 : \mu_0 \Rightarrow \mu_1 \qquad \alpha_1 : \mu_1 \Rightarrow \mu_2 \qquad \beta_0 : \nu_0 \Rightarrow \nu_1 \qquad \beta_1 : \nu_1 \Rightarrow \nu_2}$$

$$\overline{\Gamma.\blacksquare_{\mu_2 \circ \nu_2} \vdash \mathbf{key}_\Gamma^{\alpha_0 \star \beta_0} \circ \mathbf{key}_\Gamma^{\alpha_1 \star \beta_1} = \mathbf{key}_{\Gamma.\blacksquare_{\mu_0}}^{\beta_1 \circ \beta_0} \circ \mathbf{key}_\Gamma^{\alpha_1 \circ \alpha_0}.\blacksquare_{\nu_2} : \Gamma.\blacksquare_{\mu_0 \circ \nu_0} \, @ \, o}$$

In fact, the second version of the interchange law follows from the first one and the equation that expresses the naturality of $\mathbf{key}_-$. Conversely, except the two laws for the identity 2-cell and naturality, the given equations follow from either one of the two interchange laws.

While it is no longer necessary to prove that substitution is *admissible* in the setting of the GAT, we would still like to show that explicit substitutions can be eliminated on closed terms. The proof of canonicity implicitly contains such an algorithm, but that is overkill: a simple, direct argument proves that explicit substitutions can be propagated down to variables. Moreover, we may define the admissible operation mentioned in Section 2 by

$$A^\alpha \triangleq A[\mathbf{key}_\Gamma^\alpha] \qquad\qquad M^\alpha \triangleq M[\mathbf{key}_\Gamma^\alpha]$$

We may then use the aforementioned algorithm to eliminate the keys.

Pushing substitutions under modalities. In order for the aforementioned algorithm to work, we must specify how substitutions commute with the modal connectives of MTT. Unlike previous work [GSB19b], the necessary equations are straightforward:

$$\langle \mu \mid A \rangle[\delta] = \langle \mu \mid A[\delta.\blacksquare_\mu] \rangle \qquad\qquad \mathsf{mod}_\mu(M)[\delta] = \mathsf{mod}_\mu(M[\delta.\blacksquare_\mu])$$

This simplicity is not coincidental. Previous modal type theories included rules that, in one way or another, *trimmed* the context during type checking: some removed variables [Pra65, PD01, Shu18], while others erased context formers, e.g. locks [BCM$^+$20, GSB19a]. In either case, it was necessary to show that the trimming operation, which we may write as $\|\Gamma\|$, is functorial: $\Gamma \vdash \delta : \Delta$ should imply $\|\Gamma\| \vdash \|\delta\| : \|\Delta\|$. Unfortunately, the proof of this fact is almost always very complicated. Some type theories avoid it by 'forcing' substitution to be admissible using delayed substitutions [Bd00, LSR17], but this causes serious complications in the equational theory.

MTT circumvents this by avoiding any context trimming. As a result, we need neither delayed substitutions nor a complex proof of admissibility.

5. Models

In the preceding section we presented the formal definition of MTT in the form of a GAT. As a consequence we automatically obtained a *category of models* of MTT, as well as *(strict) homomorphisms* between them [Car78, KKA19]. Moreover, this category of models had an initial object, i.e. the syntax of MTT itself. This category of models is inhabited by algebras for this GAT. Hence, showing that a mathematical structure is a model of MTT becomes a laborious task: one must show that each and every construct can be interpreted

in a manner that makes the postulated equations hold. In this section we shall take on the task of decomposing these algebraic models into more tractable pieces.

A moment's thought reveals that many of the equations of the GAT given in Section 4 are rather close to the familiar notion of *categories with families* (CwFs) [Dyb96], which can be adapted to the present setting.[3] However, we will take things a bit further by opting for a category-theoretic reformulation of CwFs known as *natural models* [Awo18].

Natural models build on the view of CwFs as consisting of a presheaf of types (over the category of contexts), coupled with a presheaf of terms (over the elements of those types). We find this relatively recent technology helpful for two reasons. First, it concisely encodes the many naturality conditions normally required of a CwF. Second, it aids in uncovering the implicit universal properties of type-theoretic connectives, which are not quite so evident in the usual GAT-like formulation of CwFs.

In Section 5.1 we demonstrate how the basic notions of context, type, term, and context extension in MTT can be presented in terms of natural models. Then, in Section 5.2 we show how to interpret the various connectives—including the modality—in the language of natural models; this discussion concludes with a concise definition of a model of MTT in Section 5.2.5. Following that, in Section 5.3 we briefly discuss a strict notion of morphism of models.

5.1. Contexts, Types, and Terms.

5.1.1. *Contexts.* First, we observe that a model of our type theory must contain a set of contexts at each mode $m \in \mathcal{M}$. Equipped with the substitutions at the same mode, which can be composed associatively and have the identity substitution as a unit, these sets are readily seen to form a category—the *context category* at $m \in \mathcal{M}$—for which we write $\mathcal{C}[m]$.

Moreover, recall that for $\Gamma \, \mathsf{ctx} \, @ \, m$ and $\mu : n \to m$ we have a context $\Gamma, \blacksquare_\mu \, \mathsf{ctx} \, @ \, n$, and that this construction extends to substitutions in a functorial fashion. Hence, we will require for each modality $\mu : n \to m$ a functor

$$\llbracket \blacksquare_\mu \rrbracket : \mathcal{C}[m] \to \mathcal{C}[n]$$

Similarly, each $\alpha : \mu \Rightarrow \nu$ induces a natural transformation. Accordingly, a model should come with a natural transformation

$$\llbracket \mathbf{q}^\alpha \rrbracket : \llbracket \blacksquare_\nu \rrbracket \Rightarrow \llbracket \blacksquare_\mu \rrbracket$$

The equalities of the GAT require that the assignments $\mu \mapsto \blacksquare_\mu$ and $\alpha \mapsto \mathbf{q}^\alpha$ be strictly 2-functorial. Thus, this part of the model can be succinctly summarized as follows.

Definition 5.1. A *modal context structure* for a mode theory $\mathcal{M}$ is a (strict) 2-functor

$$\llbracket - \rrbracket : \mathcal{M}^{\mathsf{coop}} \to \mathbf{Cat}_1$$

where $\mathcal{M}^{\mathsf{coop}}$ is the 2-category $\mathcal{M}$ with the direction of *both* 1-cells and 2-cells reversed, and $\mathbf{Cat}_1$ is the full subcategory of (large) categories with a terminal object.

This double contravariance may seem peculiar at first sight. Recall that the 2-category $\mathcal{M}$ specifies the behaviour of the modal types $\langle \mu \mid - \rangle$, which are supposed to have a right-adjoint-like behaviour, with the corresponding left-adjoint-like operators being the lock functors $-.\blacksquare_\mu$. Being left-adjoint-like, the interpretation $\llbracket \blacksquare_- \rrbracket$ of each lock will behave with

---

[3]The conference version of this paper used such a presentation in the interest of brevity.

variance opposite to the specification of $\mathcal{M}$. Of course, this is merely an analogy, as these constructions are not truly adjoint.

5.1.2. *Types and Context Extension.* The following definition plays a central role.

Definition 5.2 (Representable natural transformation). Let $\mathbb{C}$ be a small category, and let $P, Q : \mathbf{PSh}(\mathbb{C})$ be presheaves on $\mathbb{C}$. A natural transformation $\alpha : P \Rightarrow Q$ is *representable* just if for every $\Gamma : \mathbb{C}$ and $x : \mathbf{y}(\Gamma) \Rightarrow P$ there exists a $y : \mathbf{y}(\Delta) \Rightarrow Q$ and a morphism $\gamma : \Delta \to \Gamma$ in $\mathbb{C}$ such that there is a pullback square

$$
\begin{array}{ccc}
\mathbf{y}(\Delta) & \xrightarrow{\ \ y\ \ } & Q \\
{\scriptstyle \mathbf{y}(\gamma)}\big\downarrow & \lrcorner & \big\downarrow{\scriptstyle \alpha} \\
\mathbf{y}(\Gamma) & \xrightarrow[\ \ x\ \ ]{} & P
\end{array}
$$

This enables a very succinct definition of a model of type theory [Awo18].

Definition 5.3. Let $\mathbb{C}$ be a small category with a terminal object $\mathbf{1}$, and let $\widetilde{\mathcal{T}}, \mathcal{T} : \mathbf{PSh}(\mathbb{C})$. A *natural model of type theory* is a representable natural transformation $\tau : \widetilde{\mathcal{T}} \Rightarrow \mathcal{T}$.

It is shown in *op. cit.* that this corresponds to the usual notion of CwF: the representability of $\tau : \widetilde{\mathcal{T}} \Rightarrow \mathcal{T}$ is a clever way to encode context extension and comprehension in a manner that automatically ensures naturality with respect to substitution: see also [Fio12]. Moreover, one can use this economy to write down very concise interpretations of type formers. Our objective here is to adapt this to modes and modalities.

To begin, given a mode $m \in \mathcal{M}$ we define two presheaves on the context category $\mathcal{C}[m]$:

$$
\mathcal{T}_m(\Gamma) \triangleq \mathsf{type}_m^1(\Gamma) \qquad \widetilde{\mathcal{T}}_m(\Gamma) \triangleq \{(A, M) \mid A \in \mathsf{type}_m^1(\Gamma), M \in \mathsf{tm}_m(\Gamma, A)\}
$$

The first one maps a context at mode $m \in \mathcal{M}$ to the set of large types over it. The second one maps a context to the set of pointed types, i.e. to the set of pairs consisting of a type and a term of that type. The presheaf action is given by substitution. We immediately obtain a natural transformation $\tau_m : \widetilde{\mathcal{T}}_m \Rightarrow \mathcal{T}_m$: at each context Γ, $\tau_{m,\Gamma}$ projects a pair (A, M) to the underlying type A. As a result, the fibres of τ_m are the terms of a given type.

Context extension postulates that for any object $\Gamma : \mathcal{C}[m]$, modality $\mu \in \mathrm{Hom}(n, m)$, and large type $A \in \mathsf{type}_n^1(\llbracket \blacksquare_\mu \rrbracket \Gamma)$ there exists an object $\Gamma.(\mu \mid A) : \mathcal{C}[m]$ along with a morphism and a term

$$
\mathbf{p} : \mathrm{Hom}_{\mathcal{C}[m]}(\Gamma.(\mu \mid A), \Gamma) \qquad \mathbf{q} \in \mathsf{tm}_n(\llbracket \blacksquare_\mu \rrbracket(\Gamma.(\mu \mid A)), A[\llbracket \blacksquare_\mu \rrbracket(\mathbf{p})])
$$

The object $\Gamma.(\mu \mid A)$ is universal with respect to $\mathbf{p}$ and $\mathbf{q}$: for any $\gamma \in \mathrm{Hom}_{\mathcal{C}[m]}(\Delta, \Gamma)$ and term $M \in \mathsf{tm}_n(\llbracket \blacksquare_\mu \rrbracket \Delta, A[\gamma.\blacksquare_\mu])$ there is a *unique* $\gamma.M : \Delta \to \Gamma.(\mu \mid A)$ such that

$$
\mathbf{p} \circ (\gamma.M) = \gamma \ : \Delta \to \Gamma \tag{5.1}
$$

$$
\mathbf{q}[(\gamma.M).\blacksquare_\mu] = M : \mathsf{tm}_n(\llbracket \blacksquare_\mu \rrbracket(\Gamma), A[\gamma.\blacksquare_\mu]) \tag{5.2}
$$

As usual, (5.2) is only well-typed because of (5.1). The only difference to the usual context extension of CwFs is that A and Γ are in different modes.

This can be encoded in the style of natural models as follows. We write $\lfloor - \rfloor$ for the Yoneda isomorphism. Given $\mu : \mathrm{Hom}_{\mathcal{M}}(n, m)$, context $\Gamma : \mathcal{C}[m]$, and a type $A : \mathsf{type}_n^1(\llbracket \blacksquare_\mu \rrbracket(\Gamma))$,

there is a chosen context Γ' (cf. $\Gamma.(\mu \mid A)$), a chosen morphism $\mathbf{p} : \Gamma' \to \Gamma$, and a chosen morphism $\lfloor \mathbf{q} \rfloor : \mathbf{y}(\llbracket \blacksquare_\mu \rrbracket \Gamma') \to \widetilde{\mathcal{T}}_n$ that make the following square commute:

$$
\begin{array}{ccc}
\mathbf{y}(\llbracket \blacksquare_\mu \rrbracket \Gamma') & \xrightarrow{\ \lfloor \mathbf{q} \rfloor\ } & \widetilde{\mathcal{T}}_n \\
{\scriptstyle \mathbf{y}(\llbracket \blacksquare_\mu \rrbracket \mathbf{p})} \downarrow & & \downarrow {\scriptstyle \tau_n} \\
\mathbf{y}(\llbracket \blacksquare_\mu \rrbracket \Gamma) & \xrightarrow[\ \lfloor A \rfloor\ ]{} & \mathcal{T}_n
\end{array}
$$

We have surreptitiously 'decoded' the top arrow into a term $\mathbf{q} \in \mathsf{tm}_n(\llbracket \blacksquare_\mu \rrbracket (\Gamma'), A[\llbracket \blacksquare_\mu \rrbracket \mathbf{p}])$.

The universality of these objects is expressed by asking that for a given $\Delta : \mathcal{C}[m]$, $\gamma : \mathrm{Hom}_{\mathcal{C}[m]}(\Delta, \Gamma)$, and $\lfloor M \rfloor : \mathbf{y}(\llbracket \blacksquare_\mu \rrbracket \Delta) \Rightarrow \widetilde{\mathcal{T}}_n$, there must be a unique morphism $\gamma' : \Delta \to \Gamma'$ (which stands for $\gamma.M$) such that the following square commutes:

$$
\begin{array}{ccc}
\mathbf{y}(\llbracket \blacksquare_\mu \rrbracket (\Delta)) & & \\
& \searrow{\scriptstyle \lfloor M \rfloor} & \\
{\scriptstyle \mathbf{y}(\llbracket \blacksquare_\mu \rrbracket \gamma')} \dashdownarrow & & \\
{\scriptstyle \mathbf{y}(\llbracket \blacksquare_\mu \rrbracket \gamma)} \quad \mathbf{y}(\llbracket \blacksquare_\mu \rrbracket \Gamma') & \xrightarrow{\ \lfloor \mathbf{q} \rfloor\ } & \widetilde{\mathcal{T}}_n \\
\downarrow {\scriptstyle \mathbf{y}(\llbracket \blacksquare_\mu \rrbracket \mathbf{p})} & & \downarrow {\scriptstyle \tau_n} \\
\mathbf{y}(\llbracket \blacksquare_\mu \rrbracket \Gamma) & \xrightarrow[\ \lfloor A \rfloor\ ]{} & \mathcal{T}_n
\end{array}
$$

This diagram is not a pullback, but we can make it into one. Recall that for any functor $f : \mathcal{C} \to \mathcal{D}$ we can define the precomposition functor $f^* : \mathbf{PSh}(\mathcal{D}) \to \mathbf{PSh}(\mathcal{C})$ by

$$
f^*(P) \triangleq \mathcal{C}^{\mathsf{op}} \xrightarrow{f^{\mathsf{op}}} \mathcal{D}^{\mathsf{op}} \xrightarrow{P} \mathbf{Set}
$$

Then, for any $c : \mathcal{C}$ and $Q : \mathbf{PSh}(\mathcal{D})$ we can use the Yoneda lemma to establish a series of natural isomorphisms

$$
\mathrm{Hom}_{\mathbf{PSh}(\mathcal{D})}(\mathbf{y}(f(c)), Q) \cong Q(f(c)) = f^*Q(c) \cong \mathrm{Hom}_{\mathbf{PSh}(\mathcal{C})}(\mathbf{y}(c), f^*Q)
$$

We can then *transpose* the diagram in order to obtain

$$
\begin{array}{ccc}
\mathbf{y}(\Delta) & & \\
& \searrow{\scriptstyle \lfloor M \rfloor} & \\
{\scriptstyle \gamma'} \dashdownarrow & & \\
{\scriptstyle \gamma}\quad \mathbf{y}(\Gamma') & \xrightarrow{\ \lfloor \mathbf{q} \rfloor\ } & \llbracket \blacksquare_\mu \rrbracket^* \widetilde{\mathcal{T}}_n \\
\downarrow {\scriptstyle \mathbf{y}(\mathbf{p})} & & \downarrow {\scriptstyle \llbracket \blacksquare_\mu \rrbracket^* \tau_n} \\
\mathbf{y}(\Gamma) & \xrightarrow[\ \lfloor A \rfloor\ ]{} & \llbracket \blacksquare_\mu \rrbracket^* \mathcal{T}_n
\end{array}
\qquad (5.3)
$$

where $\gamma' : \Delta \to \Gamma'$ is the unique arrow that makes the diagram commute. The requirement that this diagram be a pullback leads us to the following definition.

Definition 5.4. A *modal natural model* on a context structure $[\![-]\!] : \mathcal{M}^{\mathsf{coop}} \to \mathbf{Cat}_1$ consists of a family of natural transformations of presheaves

$$\left(\tau_m : \widetilde{\mathcal{T}}_m \Rightarrow \mathcal{T}_m \right)_{m \in \mathcal{M}}$$

where $\widetilde{\mathcal{T}}_m, \mathcal{T}_m : \mathbf{PSh}(\mathcal{C}[m])$ such that for every $\mu : \mathrm{Hom}_{\mathcal{M}}(m, n)$ the natural transformation

$$[\![\blacksquare_\mu]\!]^* \tau_n : [\![\blacksquare_\mu]\!]^* \widetilde{\mathcal{T}}_n \Rightarrow [\![\blacksquare_\mu]\!]^* \mathcal{T}_n$$

is a natural model.

We will write $\Gamma.(\mu \mid A)$ for the object Γ' that makes (5.3) a pullback, as we do in the type theory.

5.2. **Connectives.** We shall only discuss the key cases of Π types, modal types, Boolean types, and universes. The interpretation of the other connectives largely follows the style of [Awo18]. More details can be found in the tech report [GKNB20].

5.2.1. Π *Structure.* Even though MTT Π types are close to traditional Π types they are not quite the same, as they involve a modality in the domain. Thus, we need to construct an appropriate variation of the interpretation given by [Awo18]. To begin, we need some way to represent the *binding* of an additional assumption. This is achieved through the use of *polynomial endofunctors*. Given a 'display map' $\ell : E \to B$ we define a polynomial endofunctor $\mathbf{P}_{\ell : E \to B} : \mathbf{PSh}(\mathcal{C}[m]) \to \mathbf{PSh}(\mathcal{C}[m])$ by[4]

$$\mathbf{P}_{\ell : E \to B}(A) \triangleq \sum_{b : B} A^{\ell^{-1}(b)}$$

When specialized to the 'modalized' natural model $\ell \triangleq [\![\blacksquare_\mu]\!]^*(\tau_n) : [\![\blacksquare_\mu]\!]^* \widetilde{\mathcal{T}}_n \Rightarrow [\![\blacksquare_\mu]\!]^* \mathcal{T}_n$, this functor has a useful property: morphisms $\mathbf{y}(\Gamma) \Rightarrow \mathbf{P}_{[\![\blacksquare_\mu]\!]^* \tau_n}(\mathcal{T}_m)$ are in bijection with tuples

$$(A \in \mathcal{T}_n([\![\blacksquare_\mu]\!](\Gamma)), B \in \mathcal{T}_m(\Gamma.(\mu \mid A))) \tag{5.4}$$

This enables the representation of a pair of types $\Gamma.\blacksquare_\mu \vdash A \ \mathsf{type}_1 @ n$ and $\Gamma.(\mu \mid A) \vdash B \ \mathsf{type}_1 @ m$—i.e. the premises of Π formation—as a single morphism $\mathbf{y}(\Gamma) \Rightarrow \mathbf{P}_{[\![\blacksquare_\mu]\!]^* \tau_n}(\mathcal{T}_n)$. A similar observation applies to the presheaf of terms $\widetilde{\mathcal{T}}_m$. See [Awo18, Lemma 5] for a detailed proof of this property.

A model is equipped with a $\prod$-structure if for $\mu : \mathrm{Hom}_{\mathcal{M}}(n, m)$ we have a pullback

$$
\begin{array}{ccc}
\mathbf{P}_{[\![\blacksquare_\mu]\!]^* \tau_n}(\widetilde{\mathcal{T}}_m) & \xrightarrow{\ \mathbf{lam}\ } & \widetilde{\mathcal{T}}_m \\
\downarrow & & \downarrow {\scriptstyle \tau_m} \\
\mathbf{P}_{[\![\blacksquare_\mu]\!]^* \tau_n}(\mathcal{T}_m) & \xrightarrow[\ \prod\ ]{} & \mathcal{T}_m
\end{array}
$$

---

[4]This is given in the internal language, but may also be written purely categorically, as is done in *op. cit.*

The lower morphism $\prod$ models the formation rule: the premises of the rule constitute a pair of the form (5.4). We may thus combine them into an arrow $\mathbf{y}(\Gamma) \Rightarrow \mathbf{P}_{[\![\spadesuit_\mu]\!]^*\tau_n}(\mathcal{T}_n)$, and then postcompose $\prod$ to obtain a morphism $\mathbf{y}(\Gamma) \Rightarrow \mathcal{T}_m$, i.e. a type at mode m in context $\Gamma \; \mathsf{ctx} @ m$. In a similar fashion, the top morphism **lam** models the introduction rule. The β and η laws then follow from the pullback property of the square: see [Awo18].

5.2.2. *Modal Structure.* The interpretation of the modal types is a bit more involved. Intuitively, the reason is that $\langle \mu \mid A \rangle$ behaves like a *positive type former*, i.e. one with a 'let-style' pattern-matching eliminator, and no η-rule. These features render its behaviour closer to that of intensional identity types.

First, for each $\mu : \mathrm{Hom}_\mathcal{M}(n, m)$ the formation and introduction rules for $\langle \mu \mid - \rangle$ are given by a commuting square

$$
\begin{array}{ccc}
[\![\spadesuit_\mu]\!]^*\widetilde{\mathcal{T}}_n & \xrightarrow{\;\;\mathbf{mod}_\mu\;\;} & \widetilde{\mathcal{T}}_m \\
{\scriptstyle [\![\spadesuit_\mu]\!]^*\tau_n}\big\downarrow & & \big\downarrow{\scriptstyle \tau_m} \\
[\![\spadesuit_\mu]\!]^*\mathcal{T}_n & \xrightarrow[\;\;\mathbf{Mod}_\mu\;\;]{} & \mathcal{T}_m
\end{array}
\qquad (5.5)
$$

By Yoneda, every type $\Gamma.\spadesuit_\mu \vdash A \; \mathsf{type}_1 @ n$ can be seen as a morphism $\mathbf{y}(\Gamma) \Rightarrow [\![\spadesuit_\mu]\!]^*\tau_n$. Postcomposition with $\mathbf{Mod}_\mu$ gives a morphism $\mathbf{y}(\Gamma) \Rightarrow \mathcal{T}_m$, which constitutes the interpretation of the type $\Gamma \vdash \langle \mu \mid A \rangle \; \mathsf{type}_1 @ m$. $\mathbf{mod}_\mu$ interprets the introduction rule in a similar fashion. Nevertheless, asking that this square be a pullback is stronger than the elimination rule. In Section 7 we shall see that states that $\mathbf{Mod}_\mu$ is a *dependent right adjoint*.

Instead, we will model our elimination rule by a *lifting structure*. We phrase this definition in the internal language of the presheaf topos $\mathbf{PSh}(\mathcal{C}[m])$, i.e. extensional type theory.[5] This has a serious technical advantage: as the definition is given in an empty context, the given lifts are automatically natural.

Definition 5.5 (Left lifting structure). Given $\vdash A, I, B \; \mathsf{type}$, a family $b : B \vdash E[b] \; \mathsf{type}$ and a section $a : A \vdash i[a] : I$, we define the type $\vdash i[-] \pitchfork E[-] \; \mathsf{type}$ of *left lifting structures* for i with respect to E to be

$$
i[-] \pitchfork E[-] \triangleq \prod_{C:I \to B} \prod_{c:\prod_{a:A} E[C(i[a])]} \left\{ j : \prod_{p:I} E[C(p)] \mid \forall a : A.\; j(i[a]) = c(a) \right\}
$$

Informally, left lifting structures provide *diagonal fillers* for the diagram

$$
\begin{array}{ccc}
A & \xrightarrow{\;\;\langle C(i[-]), c\rangle\;\;} & \sum_{b:B} E[b] \\
{\scriptstyle i[-]}\big\downarrow & \overset{\displaystyle j}{\nearrow} & \big\downarrow{\scriptstyle \pi_1} \\
I & \xrightarrow[\;\;C\;\;]{} & B
\end{array}
$$

Intuitively, $C : I \to B$ is the *motive* of an elimination: we would like to prove $E[C(p)]$ for all $p : I$. At the same time, $c : \prod_{a:A} E[C(i[a])]$ is a given section that specifies the desired computational behaviour of this elimination at the 'special case' A. The left lifting structure

---

[5]This is derived from unpublished work by Jon Sterling, Daniel Gratzer, Carlo Angiuli, and Lars Birkedal.

then provides a section j of $E[-]$ defined on all of I. This section is above C, and extends c. Note that these fillers are not necessarily unique. Moreover, they are automatically *natural*: as all the types involved in this definition are closed, we are at liberty to weaken the context.

This style of lifting structure is an essential ingredient in recent work on models of intensional identity types. First, they play an important rôle in natural models: [Awo18, Lemma 19] shows that they precisely correspond to enriched left lifting properties in the sense of categorical homotopy theory [Rie14, §13]. In fact, the above definition given above is a word-for-word restatement in the internal language. Second, such lifting structures are also central devices in internal presentations of models of cubical type theory, in particular the recent work of [OP18].

We can now approach this in a manner similar to intensional identity types in *op. cit.* Recall that the elimination rule for $\langle \nu \mid A \rangle$ is

$$\frac{\begin{array}{c} \nu : \operatorname{Hom}_{\mathcal{M}}(o, n) \\ \mu : \operatorname{Hom}_{\mathcal{M}}(n, m) \quad \Gamma \text{ ctx} @ m \quad \Gamma.\blacksquare_\mu.\blacksquare_\nu \vdash A \text{ type}_1 @ o \quad \Gamma.\blacksquare_\mu \vdash M_0 : \langle \nu \mid A \rangle @ n \\ \Gamma.(\mu \mid \langle \nu \mid A \rangle) \vdash B \text{ type}_1 @ m \quad \Gamma.(\mu \circ \nu \mid A) \vdash M_1 : B[\uparrow.\operatorname{mod}_\nu(\mathbf{v}_0)] @ m \end{array}}{\Gamma \vdash \operatorname{let}_\mu \operatorname{mod}_\nu(\_) \leftarrow M_0 \text{ in } M_1 : B[\operatorname{id}.M_0] @ m}$$

First, we must remove the 'implicit cut' with M_0. We construct the substitution

$$\Gamma.(\mu \mid \langle \nu \mid A \rangle).(\mu \circ \nu \mid A[\uparrow.\blacksquare_{\mu \circ \nu}]) \vdash \sigma \triangleq \uparrow^2.\mathbf{v}_0 : \Gamma.(\mu \circ \nu \mid A) @ m$$

It then suffices to construct the elimination rule

$$\frac{\begin{array}{c} \nu : \operatorname{Hom}_{\mathcal{M}}(o, n) \quad \mu : \operatorname{Hom}_{\mathcal{M}}(n, m) \quad \Gamma \text{ ctx} @ m \quad \Gamma.\blacksquare_\mu.\blacksquare_\nu \vdash A \text{ type}_1 @ o \\ \Gamma.(\mu \mid \langle \nu \mid A \rangle) \vdash B \text{ type}_1 @ m \quad \Gamma.(\mu \circ \nu \mid A) \vdash M_1 : B[\uparrow.\operatorname{mod}_\nu(\mathbf{v}_0)] @ m \end{array}}{\Gamma.(\mu \mid \langle \nu \mid A \rangle) \vdash \operatorname{let}_\mu \operatorname{mod}_\nu(\_) \leftarrow \mathbf{v}_0 \text{ in } M_1[\sigma] : B @ m}$$

because we can calculate that

$$\Gamma \vdash (\operatorname{let}_\mu \operatorname{mod}_\nu(\_) \leftarrow \mathbf{v}_0 \text{ in } M_1[\sigma])[\operatorname{id}.M_0] = \operatorname{let}_\mu \operatorname{mod}_\nu(\_) \leftarrow M_0 \text{ in } M_1 : B[\operatorname{id}.M_0] @ m$$

We can rephrase this as the existence of a diagonal filler in the diagram

$$\begin{array}{ccc} \mathbf{y}(\Gamma.(\mu \circ \nu \mid A)) & \xrightarrow{\lfloor M_1[\sigma] \rfloor} & \widetilde{\mathcal{T}}_m \\ {\scriptstyle \mathbf{y}(\uparrow.\operatorname{mod}_\nu(\mathbf{v}_0))} \downarrow \quad {\scriptstyle \lfloor \operatorname{let}_\mu \operatorname{mod}_\nu(\_) \leftarrow \mathbf{v}_0 \text{ in } M_1[\sigma] \rfloor} \nearrow & & \downarrow {\scriptstyle \tau_m} \\ \mathbf{y}(\Gamma.(\mu \mid \langle \nu \mid A \rangle)) & \xrightarrow[\lfloor B \rfloor]{} & \mathcal{T}_m \end{array}$$

We can use a left lifting structure on a carefully chosen slice category to obtain such diagonal fillers. The internal language approach still applies because of the well-known lemma stating that the slice of a presheaf topos is also a presheaf topos, but over the corresponding category of elements. In symbols, for any $P : \mathbf{PSh}(\mathcal{C})$ we have an equivalence $\mathbf{PSh}(\mathcal{C})/P \simeq \mathbf{PSh}(\int_{\mathcal{C}} P)$: see [MLM92, III Ex. 8]

First, given $\nu : \mathrm{Hom}_{\mathcal{M}}(o, n)$ we construct the following pullback:

$$
\begin{array}{ccc}
 & \mathbf{mod}_\nu & \\
[\![\boxvoid_\nu]\!]^* \widetilde{\mathcal{T}}_o & & \\
\quad m \searrow & M \longrightarrow \widetilde{\mathcal{T}}_n & \\
[\![\boxvoid_\nu]\!]^* \tau_o \quad & \llcorner \quad h \Big\downarrow \quad \Big\downarrow \tau_n & \\
 & [\![\boxvoid_\nu]\!]^* \mathcal{T}_o \xrightarrow{\;\mathbf{Mod}_\nu\;} \mathcal{T}_n &
\end{array}
$$

The outer commuting square is that given by the formation and introduction for $\langle \nu \mid - \rangle$, as in (5.5). Intuitively, M is a 'generic ν-modal terms object' that consists of terms $\Gamma \vdash M : \langle \nu \mid A \rangle @ n$, where $\Gamma.\boxvoid_\nu \vdash A \; \mathsf{type}_1 @ o$. We know that $[\![\boxvoid_\mu]\!]^*$ has a left adjoint, so it preserves pullbacks. Applying it to this diagram yields

$$
\begin{array}{ccc}
 & [\![\boxvoid_\mu]\!]^* \mathbf{mod}_\nu & \\
[\![\boxvoid_{\mu \circ \nu}]\!]^* \widetilde{\mathcal{T}}_o \;\; _{[\![\boxvoid_\mu]\!]^* m} & & \\
\searrow & [\![\boxvoid_\mu]\!]^* M \longrightarrow [\![\boxvoid_\mu]\!]^* \widetilde{\mathcal{T}}_n & \\
[\![\boxvoid_{\mu \circ \nu}]\!]^* \tau_o \quad & \llcorner \quad [\![\boxvoid_\mu]\!]^* h \Big\downarrow \quad \Big\downarrow [\![\boxvoid_\mu]\!]^* \tau_n & \\
 & [\![\boxvoid_{\mu \circ \nu}]\!]^* \mathcal{T}_o \xrightarrow{\;[\![\boxvoid_\mu]\!]^* \mathbf{Mod}_\nu\;} [\![\boxvoid_\mu]\!]^* \mathcal{T}_n &
\end{array}
\tag{5.6}
$$

We have also used the fact that $(-)^*$ is functorial to contract the two locks into one. Moreover, we get that the unique mediating morphism is indeed $[\![\boxvoid_\mu]\!]^* m$.

From this point onwards we will also work in the slice $\mathbf{PSh}(\mathcal{C}[m])/Z$, where $Z \triangleq [\![\boxvoid_{\mu \circ \nu}]\!]^* \mathcal{T}_o$. In order to model the elimination rule we will ask for a left lifting structure *in the slice category*, of type

$$
\vdash \mathbf{open}_\nu^\mu : [\![\boxvoid_\mu]\!]^* m \pitchfork Z^*(\tau_m)
\tag{5.7}
$$

where both of these are considered as morphisms in the slice $\mathbf{PSh}(\mathcal{C}[m])/Z$, respectively of type

$$
[\![\boxvoid_\mu]\!]^* m : [\![\boxvoid_{\mu \circ \nu}]\!]^* \tau_o \to [\![\boxvoid_\mu]\!]^* h
$$

$$
Z^*(\tau_m) : Z^*(\widetilde{\mathcal{T}}_m) \to Z^*(\mathcal{T}_m)
$$

Following [Awo18] we may calculate that this models the rule. We suppose its premises, and construct the diagram of Figure 10. The right (both top and bottom) part of the diagram is just (5.6). The bottom composite is easily seen to correspond to the application of the introduction rule of $\langle \nu \mid - \rangle$ to the type $\Gamma.\boxvoid_\mu.\boxvoid_\nu \vdash A \; \mathsf{type}_1 @ o$, and hence to the type $\Gamma.\boxvoid_\mu \vdash \langle \nu \mid A \rangle \; \mathsf{type}_1 @ n$. The outer bottom square is the natural model pullback square that defines the object $\Gamma.(\mu \mid \langle \nu \mid A \rangle)$, and we thus get a mediating morphism to $[\![\boxvoid_\mu]\!]^* M$, and that the bottom-left square is also a pullback. The left (both top and bottom) part of

Figure 10: Modelling the elimination rule

the diagram is the natural model pullback square that defines the object $\Gamma.(\mu \circ \nu \mid A)$. We hence get a mediating morphism $\mathbf{p.mod}_\nu(\mathbf{q}) : \Gamma.(\mu \circ \nu \mid A) \to \Gamma.(\mu \mid \langle \nu \mid A \rangle)$. Finally, for the same reasons as the bottom composite, the top composite is easily seen to correspond to the term $\mathsf{mod}_\nu(\mathbf{q})$.

We write $\sum_Z : \mathbf{PSh}(\mathcal{C}[m])/Z \to \mathbf{PSh}(\mathcal{C}[m])$ for the usual domain projection functor, so that $\sum_Z \dashv Z^*$. Now, using the usual approach to slice categories—where the cartesian product $\times_Z$ is the pullback—we see from the diagram that

$$\sum_Z (\lfloor A \rfloor \times_Z [\![\blacksquare_{\mu \circ \nu}]\!]^* \widetilde{\mathcal{T}}_o) \cong \mathbf{y}(\Gamma.(\mu \circ \nu \mid A))$$

$$\sum_Z (\lfloor A \rfloor \times_Z [\![\blacksquare_\mu]\!]^* h) \cong \mathbf{y}(\Gamma.(\mu \mid \langle \nu \mid A \rangle)) \tag{5.8}$$

$$\sum_Z (\mathsf{id}_{\lfloor A \rfloor} \times_Z [\![\blacksquare_\mu]\!]^* m) \cong \mathbf{y}(\mathbf{p.mod}_\nu(\mathbf{q}))$$

Recall that we are trying to find a diagonal filler to the $\mathbf{PSh}(\mathcal{C}[m])$ diagram

$$
\begin{array}{ccc}
\mathbf{y}(\Gamma.(\mu \circ \nu \mid A)) & \xrightarrow{\ \lfloor M_1 \rfloor\ } & \widetilde{\mathcal{T}}_m \\
{\scriptstyle \mathbf{y}(\mathbf{p}.\mathbf{mod}_\nu(\mathbf{q}))} \downarrow & & \downarrow {\scriptstyle \tau_m} \\
\mathbf{y}(\Gamma.(\mu \mid \mathbf{Mod}_\nu(A))) & \xrightarrow[\ \lfloor B \rfloor\ ]{} & \mathcal{T}_m
\end{array}
\tag{5.9}
$$

We use the adjunction $\sum_Z \dashv Z^*$ to transpose this diagram, and we compose with the isomorphisms (5.8) to obtain the following diagram in $\mathbf{PSh}(\mathcal{C}[m])/Z$:

$$
\begin{array}{ccc}
\lfloor A \rfloor \times_Z \llbracket \blacksquare_{\mu \circ \nu} \rrbracket^* \widetilde{\mathcal{T}}_o & \xrightarrow{\ \widehat{\lfloor M_1 \rfloor}\ } & Z^*(\widetilde{\mathcal{T}}_m) \\
{\scriptstyle \mathsf{id} \times_Z \llbracket \blacksquare_\mu \rrbracket^* m} \downarrow & {\scriptstyle \mathbf{open}_\nu^\mu} & \downarrow {\scriptstyle Z^*(\tau_m)} \\
\lfloor A \rfloor \times_Z \llbracket \blacksquare_\mu \rrbracket^* h & \xrightarrow[\ \widehat{\lfloor B \rfloor}\ ]{} & Z^*(\mathcal{T}_m)
\end{array}
$$

We may then use the lifting structure to prove a diagonal filler, and transpose backwards along the adjunction to obtain a filler for (5.9). The naturality of all these steps (composing isomorphisms, transposition, and lifting structure) ensure that the choice is natural.

5.2.3. *Boolean Structure.* A boolean structure is defined similarly to the structure for modal types. First, we require two constants, as well as naturally given diagonal fillers for the appropriate squares:

$$
\begin{array}{ccc}
1 & \overset{\mathbf{tt}}{\underset{\mathbf{ff}}{\rightrightarrows}} & \widetilde{\mathcal{T}}_m \\
\Vert & & \downarrow {\scriptstyle \tau_m} \\
1 & \xrightarrow[\mathbf{Bool}]{} & \mathcal{T}_m
\end{array}
\qquad\qquad
\begin{array}{ccc}
1 + 1 & \longrightarrow & \widetilde{\mathcal{T}}_m \\
{\scriptstyle [\mathbf{tt}, \mathbf{ff}]} \downarrow & \nearrow & \downarrow {\scriptstyle \tau_m} \\
\tau_m^{-1}(\mathbf{Bool}) & \longrightarrow & \mathcal{T}_m
\end{array}
$$

$\tau_m^{-1}(\mathbf{Bool})$ is the fibre of τ_m over $\mathbf{Bool}$, and the map $[\mathbf{tt}, \mathbf{ff}]$ is obtained as the cotuple of the maps obtained by factoring $\mathbf{tt}$ and $\mathbf{ff}$ through the fibre. Requiring a left lifting structure

$$
\mathbf{if} : [\mathbf{tt}, \mathbf{ff}] \pitchfork \tau_m[-]
$$

in the internal language provides enough naturality to yield diagonal fillers for all squares

$$
\begin{array}{ccc}
\mathbf{y}(\Gamma) + \mathbf{y}(\Gamma) & \longrightarrow & \widetilde{\mathcal{T}}_m \\
{\scriptstyle [\mathsf{id}.\mathbf{tt}, \mathsf{id}.\mathbf{ff}]} \downarrow & \nearrow & \downarrow {\scriptstyle \tau_m} \\
\mathbf{y}(\Gamma.\mathbf{Bool}) & \longrightarrow & \mathcal{T}_m
\end{array}
$$

5.2.4. *Universe.* The universe itself is given by a presheaf $\mathcal{S}_m$ at each mode m. The Coquand-style isomorphism is implemented by a natural transformation $\mathbf{Uni} : 1 \Rightarrow \tau_m$, which stands for the universe type at mode m, as well as a natural isomorphism

$$\tau_m^{-1}(\mathbf{Uni}) \cong \mathcal{S}_m$$

As the pullback of τ_m along $\mathbf{Uni}$ is $\mathbf{y}(1.\mathbf{Uni})$, this exactly postulates an isomorphism between terms of the universe and small types. The coercion from small to large type is interpreted by a natural transformation $\mathbf{lift} : \mathcal{S}_m \Rightarrow \mathcal{T}_m$ that maps each small type to its associated large type. Moreover, we ask that the formation rules *factor* through small types: we require a mediating morphism in each of the following diagrams:[6]

$$
\begin{array}{ccc}
\mathbf{P}_{[\![\mathbf{a}_\mu]\!]^* \tau_n}(\mathcal{S}_m) \dashrightarrow \mathcal{S}_m
& [\![\mathbf{a}_\mu]\!]^* \mathcal{S}_n \dashrightarrow \mathcal{S}_m
& 1 \dashrightarrow \mathcal{S}_m \\
\downarrow \qquad \quad \downarrow \mathbf{lift}
& \downarrow \qquad \quad \downarrow \mathbf{lift}
& \downarrow \qquad \quad \downarrow \mathbf{lift} \\
\mathbf{P}_{[\![\mathbf{a}_\mu]\!]^* \tau_n}(\mathcal{T}_m) \xrightarrow{\prod} \mathcal{T}_m
& [\![\mathbf{a}_\mu]\!]^* \mathcal{T}_n \xrightarrow{\mathbf{Mod}_\mu} \mathcal{T}_m
& 1 \xrightarrow{\mathbf{Bool}} \mathcal{T}_m
\end{array}
$$

These factorisations ensure that type formation is closed under small types, and commutation ensures that the coercions commute with the type formers definitionally.

5.2.5. *The full definition.* We have shown how to interpret each rule of MTT through natural models. In fact, every step of our working is reversible: each contraption we have introduced precisely corresponds to the portion of the generalized algebraic theory it was used to interpret. In summary, we can make the following definition.

Definition 5.6. A model of MTT over $\mathcal{M}$ consists of
- a modal context structure for $\mathcal{M}$ (as in Definition 5.1), and a
- a modal natural model on that context structure (as in Definition 5.4)

such that the modal natural model supports

- dependent product types
- dependent sum types (at each mode)
- intensional identity types (at each mode)
- modal types
- a boolean type (at each mode), and
- a universe of small types.

5.3. **Morphisms of Models.** The generalized algebraic theory (GAT) of MTT also induces a notion of morphism between models. Traditionally neglected, morphisms are of paramount importance when one produces semantic proofs of metatheoretic properties, such as canonicity, a proof of which we will present in Section 6.

The last decade has seen much use of relatively weak morphisms of CwFs, i.e. morphisms which preserve structures only up to isomorphism: see e.g. [CD14, BCM+20]. However, our proof of canonicity will require the strictest notion of CwF morphism, i.e. a GAT homomorphism. Such morphisms preserve *all* structure on-the-nose, including context

---

[6]There are also similar diagrams for Σ and intensional identity types.

extension, and were introduced alongside CwFs by [Dyb96]. As we are using natural models, we will use an adaptation due to [New18, §2.3]. We believe that one may construct a biequivalence or biadjunction between a category based on strict morphisms and one based on weaker ones, as done by e.g. [Uem19], but we will leave that to future work.

Definition 5.7 (Strict morphism of natural models). A morphism of natural models $(\mathcal{C}, \widetilde{\mathcal{T}}_c \xrightarrow{\tau_c} \mathcal{T}_c) \to (\mathcal{D}, \widetilde{\mathcal{T}}_d \xrightarrow{\tau_d} \mathcal{T}_d)$ comprises a functor $F : \mathcal{C} \to \mathcal{D}$ and a commuting diagram

$$
\begin{array}{ccc}
\widetilde{\mathcal{T}}_c & \xrightarrow{\ \widetilde{\varphi}\ } & F^*\widetilde{\mathcal{T}}_d \\
{\scriptstyle \tau_c}\big\downarrow & & \big\downarrow{\scriptstyle F^*\tau_d} \\
\mathcal{T}_c & \xrightarrow{\ \varphi\ } & F^*\mathcal{T}_d
\end{array}
\tag{5.10}
$$

such that $F(\mathbf{1}) = \mathbf{1}$ and the canonical morphism $F(\Gamma.A) \to F(\Gamma).\varphi(A)$ is an identity.

The type $\varphi(A)$ in the last line is defined as follows. Given $\lfloor A \rfloor : \mathbf{y}(\Gamma) \Rightarrow \mathcal{T}_c$ we let

$$
k \triangleq \mathbf{y}(\Gamma) \xrightarrow{\lfloor A \rfloor} \mathcal{T}_c \xrightarrow{\varphi} F^*\widetilde{\mathcal{T}}_d
$$

By Yoneda this induces a natural isomorphism

$$
\mathrm{Hom}_{\mathbf{PSh}(\mathcal{C})}(\mathbf{y}(\Gamma), F^*\mathcal{T}_d) \cong F^*\mathcal{T}_d(\Gamma) = \mathcal{T}_d(F(\Gamma)) \cong \mathrm{Hom}_{\mathbf{PSh}(\mathcal{D})}(\mathbf{y}(F\Gamma), \mathcal{T}_d)
\tag{5.11}
$$

We define $\lfloor \phi(A) \rfloor : \mathbf{y}(F\Gamma) \Rightarrow \mathcal{T}_d$ to be k transported under this isomorphism. Also, let

$$
\lfloor M \rfloor : \mathbf{y}(\Gamma) \Rightarrow \mathcal{T}_c \quad \longmapsto \quad \lfloor \widetilde{\varphi}(M) \rfloor : \mathbf{y}(F(\Gamma)) \Rightarrow \mathcal{T}_d
$$

which maps a term $\Gamma \vdash M : A$ to a term $F\Gamma \vdash \widetilde{\varphi}(M) : \varphi(A)$ in a similar manner.

Returning to the last condition in the definition, we may now form the diagram

where the outer square is the diagram composed by pasting together the context extension diagram for $\Gamma.A$ and (5.10), followed by transposing along the natural isomorphism (5.11). We then ask that the unique induced arrow be the identity.

We can lift these natural transformations to the formation data of the connectives (making special use of the final equality for the polynomial functors). For instance, we can define a morphism

$$
\mathbf{P}_{\tau_c}(\mathcal{T}_c) \xrightarrow{(\varphi, \widetilde{\varphi})} \mathbf{P}_{F^*\tau_d}(F^*\mathcal{T}_d) \triangleq \mathbf{P}_{\tau_c}(\mathcal{T}_c) \to \mathbf{P}_{F^*\tau_d}(\mathcal{T}_c) \xrightarrow{\mathbf{P}_{F^*\tau_d}(\varphi)} \mathbf{P}_{F^*\tau_d}(F^*\mathcal{T}_d)
$$

The first component comes from a natural transformation $\mathbf{P}_{\tau_c}(-) \Rightarrow \mathbf{P}_{F^*\tau_d}(-)$, which exists because (5.10) not only commutes, but is a pullback square. That is a nontrivial fact proven

laboriously by [New18, §2.3.14]. A more conceptual proof is given by [Uem19, Cor. 3.14] in the language of discrete fibrations.

We then require that all connectives ($\prod, \sum, \mathbf{refl}$) strictly commute with these morphisms. Finally, we can extend this to a model of MTT by requiring not just a functor, but a natural transformation $\mathcal{C} \Rightarrow \mathcal{D}$, where $\mathcal{C}, \mathcal{D} : \mathcal{M}^{\mathsf{coop}} \to \mathbf{Cat}$ satisfy the obvious generalizations of the conditions written above. Specifying this formally:

Definition 5.8. A morphism between two models of MTT, $\mathcal{C}, \mathcal{D}$, is given by a 2-natural transformation $\alpha : \mathcal{C} \Rightarrow \mathcal{D}$. Moreover, we require a choice of commuting squares:

$$
\begin{array}{ccc}
\widetilde{\mathcal{U}}_{\mathcal{C}[m]} & \xrightarrow{\widetilde{\varphi}_m} & \alpha_m^* \widetilde{\mathcal{U}}_{\mathcal{D}[m]} \\
{\scriptstyle \tau_{\mathcal{C}[m]}} \downarrow & & \downarrow {\scriptstyle \alpha^* \tau_{\mathcal{D}[m]}} \\
\mathcal{U}_{\mathcal{C}[m]} & \xrightarrow{\varphi_m} & \alpha^* \mathcal{U}_{\mathcal{D}[m]}
\end{array}
$$

Moreover, we require that $(\varphi, \widetilde{\varphi})$ strictly commutes with all operations.

$$\alpha_m(\Gamma.(\mu \mid A)) = \alpha_m(\Gamma).(\mu \mid \varphi(A))$$

$$\prod \circ (\varphi, \varphi) = \varphi \circ \prod \qquad\qquad \mathbf{lam} \circ (\varphi, \widetilde{\varphi}) = \widetilde{\varphi} \circ \mathbf{lam}$$

$$\sum \circ (\varphi, \varphi) = \varphi \circ \sum \qquad\qquad \mathbf{pair} \circ (\varphi, \widetilde{\varphi}) = \widetilde{\varphi} \circ \mathbf{pair}$$

$$\mathbf{Mod}_\mu \circ [\![\blacksquare_\mu]\!]^* \varphi = \varphi \circ \mathbf{Mod}_\mu \qquad\qquad \mathbf{mod}_\mu \circ [\![\blacksquare_\mu]\!]^* \widetilde{\varphi} = \widetilde{\varphi} \circ \mathbf{mod}_\mu$$

$$\mathbf{open}_\mu^\nu \circ (\widetilde{\varphi}, [\![\blacksquare_\mu]\!]^* \widetilde{\varphi}) = \widetilde{\varphi} \circ \mathbf{open}_\mu^\nu$$

$$\mathbf{Bool} = \varphi \circ \mathbf{Bool} \qquad\qquad \mathbf{tt} = \widetilde{\varphi} \circ \mathbf{tt} \quad \mathbf{ff} = \widetilde{\varphi} \circ \mathbf{ff}$$

$$\mathbf{if} \circ (\widetilde{\varphi}, \widetilde{\varphi}, \widetilde{\varphi}) = \widetilde{\varphi} \circ \mathbf{if}$$

$$\mathbf{Id} \circ (\varphi, \widetilde{\varphi}, \widetilde{\varphi}) = \varphi \circ \mathbf{Id} \qquad\qquad \mathbf{refl} \circ \widetilde{\varphi} = \widetilde{\varphi} \circ \mathbf{refl}$$

$$\mathbf{J} \circ (\widetilde{\varphi}, \widetilde{\varphi}) = \widetilde{\varphi} \circ \mathbf{J}$$

Remark 5.9 (The Initiality of Syntax)**.** Under this definition of homomorphism, we immediately have an initial model [Car78, KKA19]. We will *define* this model to be our syntax and designate it $(\mathbb{S}[m])_{m \in \mathcal{M}}$.

6. Canonicity

Equipped with the generalized algebraic theory of Section 4 and its reformulation through natural models in Section 5, we are ready to show that the syntax of MTT is well-behaved. In this section we will sketch the main parts of a proof of *canonicity* for MTT. This is a basic well-behavedness property which guarantees that terms of ground type, e.g. $\mathbb{B}$, can be normalized. As expected, the proof is independent of the mode theory.

Proposition 6.1. *If* $\vdash M : \mathbb{B} @ m$, *then either* $\vdash M = \mathsf{tt} : \mathbb{B} @ m$ *or* $\vdash M = \mathsf{ff} : \mathbb{B} @ m$.

This kind of result would traditionally be established by producing a rewriting system along with a lengthy PER model construction. We will instead opt for a proof given by constructing a *glued* model [Coq19, KHS19]. The contexts, types, and terms of this model

will contain a syntactic component (resp. a context, type, or term), along with a *proof-relevant predicate* that is appropriately fibred over it. The base types of this model are carefully chosen so that a normal form can be extracted from proofs of the predicate. By interpreting a term of ground type in the glued model we automatically obtain a proof of the predicate, from which we extract a normal form.

Such proofs involve two steps: defining the glued construction, and proving that it is a model. While the first step is often straightforward, the second usually involves checking innumerable equations. In order to shorten the proof sketched here we will make a simplifying assumption (effectively adding an equation to the algebraic syntax): we will assume that locks preserve the empty context, i.e. that

$$\cdot.\blacksquare_\mu = \cdot \; \mathsf{ctx} \, @ \, m$$

for $\mu : \mathrm{Hom}_{\mathcal{M}}(m, n)$. Using the universal property of the terminal context, this implies

$$\cdot.\blacksquare_\mu \vdash \mathbf{q}_{\cdot}^{\alpha} = \cdot = \mathsf{id} : \cdot.\blacksquare_\nu \, @ \, m \tag{6.1}$$

Requiring this equation unfortunately limits our class of models to those where the left adjoint strictly preserves the terminal product. Despite this simplification the proof remains rather long, so we will only sketch the construction of the modal natural models. The missing details may be found in an accompanying technical report.

Remark 6.2. In what follows we will assume the existence of two Grothendieck universes $\mathcal{V}' \subset \mathcal{V} : \mathbf{Set}$. We could make do with just one but at the price of some contortions, which are both unnecessary and tiresome. We will assume that the sets of contexts, substitutions, types, and terms of the syntactic model are $\mathcal{V}'$-small.

6.1. **The Glued Model.** We begin by defining the context structure.

Definition 6.3 (Glued Contexts)**.** A glued context Γ at mode m consists of a context $\Gamma^{\triangleleft} \in \mathsf{ctx}_m$, a predicate $\Gamma^{\blacktriangleright} \in \mathcal{V}$, and a function

$$\phi_\Gamma : \Gamma^{\blacktriangleright} \to \mathsf{sb}_m(\cdot, \Gamma^{\triangleleft})$$

A glued context $\Gamma = (\Gamma^{\triangleleft}, \Gamma^{\blacktriangleright})$ can be thought of as a proof-relevant predicate over substitutions into the syntactic context $\Gamma^{\triangleleft}$. An element $x \in \Gamma^{\blacktriangleright}$ can be thought of as a proof that the predicate holds of the substitution $\phi_\Gamma(x) : \cdot \to \Gamma^{\triangleleft}$. We will henceforth use the metavariable Γ to range over *glued contexts*, and denote contexts of the syntax by $\Gamma^{\triangleleft}$.

Definition 6.4 (Glued Substitutions)**.** A glued substitution from Δ to Γ at mode m is a pair of a substitution $\gamma^{\triangleleft} \in \mathsf{sb}_m(\Delta^{\triangleleft}, \Gamma^{\triangleleft})$ and a function $\gamma^{\blacktriangleright} : \Delta^{\blacktriangleright} \to \Gamma^{\blacktriangleright}$ such that

$$\forall x \in \Delta^{\blacktriangleright}. \; \phi_\Gamma(\gamma^{\blacktriangleright}(x)) = \gamma^{\triangleleft} \circ \phi_\Delta(x) : \cdot \to \Gamma^{\triangleleft}$$

Glued contexts and glued substitutions form a category, viz. the comma category

$$\mathcal{C}[m] \triangleq (1_{\mathcal{V}} \downarrow \mathsf{sb}_m(\cdot, -))$$

which we take as the category of contexts at mode m. Next, we define a 2-functor from $\mathcal{M}$ sending each m to $\mathcal{C}[m]$. For each $\mu : \mathrm{Hom}_{\mathcal{M}}(m, n)$ a functor $[\![\blacksquare_\mu]\!] : \mathcal{C}[n] \to \mathcal{C}[m]$ as by

taking the $\mathcal{C}[n]$ morphism

$$
\begin{array}{ccc}
\Delta^\blacktriangleright & \xrightarrow{\quad\gamma^\blacktriangleright\quad} & \Gamma^\blacktriangleright \\
\Big| & & \Big| \\
\phi_\Delta \Big\downarrow & & \phi_\Gamma \Big\downarrow \\
\mathsf{sb}_n(\cdot, \Delta^\triangleleft) & \xrightarrow[\gamma^\triangleleft \circ -]{} & \mathsf{sb}_n(\cdot, \Gamma^\triangleleft)
\end{array}
$$

to the $\mathcal{C}[m]$ morphism:

$$
\begin{array}{ccc}
\Delta^\blacktriangleright & \xrightarrow{\quad\gamma^\blacktriangleright\quad} & \Gamma^\blacktriangleright \\
\Big| & & \Big| \\
\phi_{\Delta.\blacksquare_\mu} \Big\downarrow & & \phi_{\Gamma.\blacksquare_\mu} \Big\downarrow \\
\mathsf{sb}_m(\cdot, \Delta^\triangleleft.\blacksquare_\mu) & \xrightarrow[\gamma^\triangleleft.\blacksquare_\mu \circ -]{} & \mathsf{sb}_m(\cdot, \Gamma^\triangleleft.\blacksquare_\mu)
\end{array}
$$

where the function $\phi_{\Delta.\blacksquare_\mu}$ is defined by

$$
\phi_{\Delta.\blacksquare_\mu}(x) \triangleq \phi_\Delta(x).\blacksquare_\mu \; : \; \cdot \to \Delta^\triangleleft.\blacksquare_\mu
$$

Notice that the equation $\cdot.\blacksquare_\mu = \cdot$ is necessary to ensure that this definition is well-typed. The diagram commutes because locks act functorially on substitutions. It is also functorial in μ, because $\Gamma.\blacksquare_\mu.\blacksquare_\nu = \Gamma.\blacksquare_{\mu\circ\nu}$, and $\Gamma.\blacksquare_1 = \Gamma$.

We define a 2-cell $[\![\blacksquare_\mu]\!] \Rightarrow [\![\blacksquare_\nu]\!]$ for each $\alpha : \nu \Rightarrow \mu$. The component at $(\Gamma^\blacktriangleright, \Gamma^\triangleleft, \phi_\Gamma)$ is

$$
\begin{array}{ccc}
\Gamma^\blacktriangleright & =\!=\!=\!=\!=\!=\!= & \Gamma^\blacktriangleright \\
\Big| & & \Big| \\
\phi_{\Gamma.\blacksquare_\mu} \Big\downarrow & & \phi_{\Gamma.\blacksquare_\nu} \Big\downarrow \\
\mathsf{sb}_m(\cdot, \Gamma^\triangleleft.\blacksquare_\mu) & \xrightarrow[\;\mathbf{\kappa}^\alpha_{\Gamma^\triangleleft} \circ -\;]{} & \mathsf{sb}_m(\cdot, \Gamma^\triangleleft.\blacksquare_\nu)
\end{array}
$$

This diagram commutes because of (6.1), so it is a morphism in comma category. Naturality follows from the numerous equations pertaining to keys and their composition.

This completes the definition of a strict 2-functor $\mathcal{M}^{\mathsf{coop}} \to \mathbf{Cat}_1$ as per Section 5.1. Next, we must define the modal natural model structure for each category of contexts.

Remark 6.5. For the rest of this section we will freely use type-theoretic notation, viewing the predicate $\Gamma^\blacktriangleright \to \mathsf{sb}_m(\cdot, \Gamma^\triangleleft)$ as a family fibred over $\mathsf{sb}_m(\cdot, \Gamma^\triangleleft)$, i.e. a map $\mathsf{sb}_m(\cdot, \Gamma^\triangleleft) \to \mathcal{V}$.

We will follow the convention that symbols annotated with $(-)^\blacktriangleright$ correspond to proof-relevant constructions—i.e. members of the predicate, or maps between predicates—whereas symbols annotated with $(-)^\triangleleft$ correspond to pieces of syntax (e.g. terms, contexts, substitutions). In particular, $\gamma^\blacktriangleright$ will not necessarily refer to a fibred map between proof-relevant predicates, but also to a generalized element of $\Gamma^\blacktriangleright$.

In other words, when $\gamma^\blacktriangleright \in \Gamma^\blacktriangleright$ and $\phi_\Gamma(\gamma^\blacktriangleright) = \gamma^\triangleleft : \cdot \to \Gamma^\triangleleft$, we will abusively write $\gamma^\blacktriangleright : \Gamma^\blacktriangleright(\gamma^\triangleleft)$. That is, we will view $\gamma^\blacktriangleright$ as living in the fibre of ϕ_Γ over $\gamma^\triangleleft$. This amounts to

considering $\gamma^{\blacktriangleright}$ as a *proof* that the predicate $\Gamma^{\blacktriangleright}$ holds at the substitution $\gamma^{\triangleleft}$. Observe that if $\gamma^{\blacktriangleright} : \Gamma.\blacklozenge_{\mu}^{\blacktriangleright}(\gamma^{\triangleleft})$ then $\gamma^{\triangleleft}$ must be of the form $\theta^{\triangleleft}.\blacklozenge_{\mu}$ for some $\theta^{\triangleleft} : \cdot \to \Gamma^{\triangleleft}$ with $\gamma^{\blacktriangleright} : \Gamma^{\blacktriangleright}(\theta^{\triangleleft})$.

Types over $\mathcal{C}[m]$ are given by the presheaf

$$\mathcal{T}_m(\Gamma) \triangleq \{$$
$$A^{\triangleleft} \in \mathsf{type}^1_m(\Gamma^{\triangleleft});$$
$$A^{\blacktriangleright} : (\gamma^{\triangleleft} : \mathsf{sb}_m(\cdot, \Gamma^{\triangleleft})) \to (\gamma^{\blacktriangleright} : \Gamma^{\blacktriangleright}(\gamma^{\triangleleft})) \to \mathsf{tm}_m(\cdot, A^{\triangleleft}[\gamma^{\triangleleft}]) \to \mathcal{V}$$
$$\}$$

Extending this presheaf with some additional data gives a presheaf of terms over $\mathcal{C}[m]$:

$$\widetilde{\mathcal{T}}_m(\Gamma) \triangleq \{$$
$$A^{\triangleleft} \in \mathsf{type}^1_m(\Gamma^{\triangleleft});$$
$$A^{\blacktriangleright} : (\gamma^{\triangleleft} : \mathsf{sb}_m(\cdot, \Gamma^{\triangleleft})) \to (\gamma^{\blacktriangleright} : \Gamma^{\blacktriangleright}(\gamma^{\triangleleft})) \to \mathsf{tm}_m(\cdot, A^{\triangleleft}[\gamma^{\triangleleft}]) \to \mathcal{V}$$
$$M^{\triangleleft} \in \mathsf{tm}_m(\Gamma^{\triangleleft}, A^{\triangleleft});$$
$$M^{\blacktriangleright} : (\gamma^{\triangleleft} : \mathsf{sb}_m(\cdot, \Gamma^{\triangleleft})) \to (\gamma^{\blacktriangleright} : \Gamma^{\blacktriangleright}(\gamma^{\triangleleft})) \to A^{\blacktriangleright}(\gamma^{\triangleleft}, \gamma^{\blacktriangleright}, M^{\triangleleft}[\gamma^{\triangleleft}])$$
$$\}$$

Thus, a *type* over $\Gamma = (\Gamma^{\triangleleft}, \varphi_\Gamma)$ in the glued model consists of a type $\Gamma^{\triangleleft} \vdash A^{\triangleleft} \mathsf{type}_1 @ m$ of MTT, along with another predicate, a family of $\mathcal{V}$-small sets, indexed over both closing substitutions $\gamma^{\triangleleft}$ that satisfy the predicate $\Gamma^{\blacktriangleright}$ and terms of type $A^{\triangleleft}[\Gamma^{\triangleleft}]$.

A *term* over Γ in the glued model adds to the above a term $\Gamma^{\triangleleft} \vdash M^{\triangleleft} : A^{\triangleleft} @ m$ of that type, and a section $M^{\blacktriangleright}$ of the aforementioned predicate. This section produces a proof that the predicate holds at that term after we close it by applying any substitution $\gamma^{\triangleleft}$ of which the $\Gamma^{\blacktriangleright}$ holds. The reindexing action of these presheaves is defined by the action of substitution on contexts, types, and terms of MTT.

It can be shown that the projection $\tau_m(\Gamma) \triangleq (A^{\triangleleft}, A^{\blacktriangleright}, M^{\triangleleft}, M^{\blacktriangleright}) \mapsto (A^{\triangleleft}, A^{\blacktriangleright})$ that maps terms to types by forgetting the additional data defines a representable natural transformation in the sense of Section 5.1.2; the full proof can be found in the technical report. With respect to the connectives, we only show how to interpret the base type $\mathbb{B}$, as per Section 5.2.3. For the formation and introduction rules we define:

$$\mathbf{Bool}^{\triangleleft} = \mathbb{B} \qquad \mathbf{Bool}^{\blacktriangleright} = \lambda \gamma^{\triangleleft}, \gamma^{\blacktriangleright}, M^{\triangleleft}. \ (M^{\triangleleft}[\gamma^{\triangleleft}] = \mathsf{tt}) + (M^{\triangleleft}[\gamma^{\triangleleft}] = \mathsf{ff})$$
$$\mathbf{tt}^{\triangleleft} = \mathsf{tt} \qquad \mathbf{tt}^{\blacktriangleright} = \lambda_-. \ \iota_0(\star)$$
$$\mathbf{ff}^{\triangleleft} = \mathsf{ff} \qquad \mathbf{ff}^{\blacktriangleright} = \lambda_-. \ \iota_1(\star)$$

We must now define the left lifting structure $\mathbf{if} : [\mathbf{tt}, \mathbf{ff}] \pitchfork \tau_m$. In type-theoretic notation:

$$\mathbf{if}(C, [c_0, c_1], M)^{\triangleleft} = \mathsf{if}(C^{\triangleleft}; c_0^{\triangleleft}; c_1^{\triangleleft}; M^{\triangleleft})$$

$$\mathbf{if}(C, [c_0, c_1], M)^{\blacktriangleright} = \lambda \gamma^{\triangleleft}, \gamma^{\blacktriangleright}. \begin{cases} c_0^{\blacktriangleright}(\gamma^{\triangleleft}, \gamma^{\blacktriangleright}) & \text{if } M^{\blacktriangleright}(\gamma^{\triangleleft}, \gamma^{\blacktriangleright}) = \iota_0(\star) \\ c_1^{\blacktriangleright}(\gamma^{\triangleleft}, \gamma^{\blacktriangleright}) & \text{if } M^{\blacktriangleright}(\gamma^{\triangleleft}, \gamma^{\blacktriangleright}) = \iota_1(\star) \end{cases}$$

6.2. **Deriving Canonicity.** With the gluing model constructed, the rest of the proof is surprisingly easy and boils down to one fact, which is immediate by inspection:

Theorem 6.6. *The 2-natural transformation $\pi : \mathcal{C}[-] \Rightarrow \mathbb{S}[-]$ from the glued model to the syntactic model which forgets the predicates extends to a morphism of models.*

Thus, assuming $\cdot . \blacksquare_\mu = \cdot$ ctx $@\, m$ it follows that

Corollary 6.7. *For any closed term $\vdash M : A\, @\, m$, there is a witness for $[\![A]\!]^{\blacktriangleright}(M)$.*

Proof. By Theorem 6.6 and initiality we must have $\pi([\![M]\!]) = M$, and so $[\![M]\!]^{\blacktriangleright}$ is a witness. $\qquad\square$

Theorem 6.8 (Closed Term Canonicity). *If $\cdot \vdash M : A\, @\, m$ is a closed term, then*
- *If $A = \mathbb{B}$ then $\cdot \vdash M = \mathsf{tt} : \mathbb{B}\, @\, m$ or $\cdot \vdash M = \mathsf{ff} : \mathbb{B}\, @\, m$.*
- *If $A = \mathsf{Id}_{A_0}(N_0, N_1)$ then $\cdot \vdash N_0 = N_1 : A_0\, @\, m$ and $\cdot \vdash M = \mathsf{refl}(N_0) : \mathsf{Id}_{A_0}(N_0, N_1)\, @\, m$.*
- *If $A = \langle \nu \mid A_0 \rangle$ then there is an $\cdot \vdash N : A_0\, @\, n$ such that $\cdot \vdash M = \mathsf{mod}_\nu(N) : \langle \nu \mid A_0 \rangle\, @\, m$.*

Proof. Immediate by Corollary 6.7 and the definition of the semantic predicates at $\mathbb{B}$, $\mathsf{Id}_{A_0}(N_0, N_1)$, and $\langle \nu \mid A_0 \rangle$ respectively. $\qquad\square$

7. Dependent Right Adjoints

Over the past couple of years the structure of a *dependent right adjoint (DRA)* has arisen as a natural notion of dependent modality in Martin-Löf type theory. In this section we will study the relationship between MTT modalities and DRAs in detail. After reviewing the definition of a DRA, we will prove that a suitably functorial collection of DRAs induces a model of MTT. As mentioned before, this implies that MTT modalities are weaker than DRAs. Following that, we will investigate sufficient conditions for extending an ordinary right adjoint to a DRA.

7.1. **Dependent right adjoints in natural models.** A dependent right adjoint[7] is an adaptation of the notion of adjunction to the dependent setting: instead of acting on objects of the context category, the 'right adjoint' only acts on types and terms.

Given a pair of natural models $(\mathcal{D}, \tau_{\mathcal{D}})$ and $(\mathcal{C}, \tau_{\mathcal{C}})$, a DRA from the second to the first comprises a functor $L : \mathcal{D} \to \mathcal{C}$ between the underlying context categories, as well as a pullback diagram of the following shape in $\mathbf{PSh}(\mathcal{D})$:

$$
\begin{array}{ccc}
L^* \widetilde{\mathcal{T}_{\mathcal{C}}} & \xrightarrow{\quad r \quad} & \widetilde{\mathcal{T}_{\mathcal{D}}} \\
{\scriptstyle L^* \tau_{\mathcal{C}}} \downarrow\ \lrcorner & & \downarrow {\scriptstyle \tau_{\mathcal{D}}} \\
L^* \mathcal{T}_{\mathcal{C}} & \xrightarrow[\quad R \quad]{} & \mathcal{T}_{\mathcal{D}}
\end{array}
$$

R is the action on types, and r is the action on terms. Note that, while the 'left adjoint' L acts on context categories, the 'right adjoint' (R, r) only acts on types and terms. The fact

---

[7]DRAs were introduced by [BCM+20] as endomodalities, but we generalise them to multiple modes.

the square is a pullback amounts to requiring a multimode generalization of the definition given by [BCM$^+$20]. Intuitively, for each term $\Gamma \vdash M : R(A)$, the pullback square gives a unique term $L(\Gamma) \vdash N : A$ such that $\Gamma \vdash M = r(N) : R(A)$. If we wish the modality to preserve the size of types, we must also require a R' such that

$$
\begin{array}{ccc}
L^*\mathcal{S}_\mathcal{C} & \xrightarrow{\quad R' \quad} & \mathcal{S}_\mathcal{D} \\
\downarrow & & \downarrow \mathbf{lift} \\
L^*\mathcal{T}_\mathcal{C} & \xrightarrow{\quad R \quad} & \mathcal{T}_\mathcal{D}
\end{array}
$$

7.2. DRAs as Models of MTT.

We will now show that DRAs can be used to construct models of MTT. As a consequence, MTT modalities are slightly weaker than DRAs.

Theorem 7.1. *Suppose that we have*
- *for each $m \in \mathcal{M}$ a natural model $(\mathcal{C}[m], \widetilde{\mathcal{T}}_m \xrightarrow{\tau_m} \mathcal{T}_m)_{m \in \mathcal{M}}$ of MLTT;*
- *for each modality $\mu : \mathrm{Hom}_\mathcal{M}(m, n)$ a size-preserving DRA $(\llbracket \blacksquare_\mu \rrbracket, \mathbf{Mod}_\mu, \mathbf{mod}_\mu)$ from $(\mathcal{C}[m], \widetilde{\mathcal{T}}_m \xrightarrow{\tau_m} \mathcal{T}_m)$ to $(\mathcal{C}[n], \widetilde{\mathcal{T}}_n \xrightarrow{\tau_n} \mathcal{T}_n)$;*
- *for each 2-cell $\alpha : \mu \Rightarrow \nu$ in $\mathcal{M}$ a natural transformation $\llbracket \maltese^\alpha \rrbracket : \llbracket \blacksquare_\nu \rrbracket \Rightarrow \llbracket \blacksquare_\mu \rrbracket$.*

Moreover, suppose that the above choices are 2-functorial. Then this data can be assembled into a model of MTT, where each each mode m is interpreted by $(\mathcal{C}[m], \widetilde{\mathcal{T}}_m \xrightarrow{\tau_m} \mathcal{T}_m)$.

Proof. Define a 2-functor $\mathcal{M}^{\mathsf{coop}} \to \mathbf{Cat}$ by $m \mapsto \mathcal{C}[m]$, $\mu \mapsto \llbracket \blacksquare_\mu \rrbracket$, and $\alpha \mapsto \llbracket \maltese^\alpha \rrbracket$. We must show how to define context extension, and how to interpret the connectives. As before, we only show the modal cases, the others being straightforward.

Modal Context Extension: For each type $\mathbf{y}(\Gamma) \xrightarrow{\lfloor A \rfloor} \llbracket \blacksquare_\mu \rrbracket^* \mathcal{T}_n$ we need a pullback

$$
\begin{array}{ccc}
\mathbf{y}(\Gamma') & \xrightarrow{\lfloor q' \rfloor} & \llbracket \blacksquare_\mu \rrbracket^* \widetilde{\mathcal{T}}_n \\
\mathbf{y}(p') \downarrow \;\; \lrcorner & & \downarrow \llbracket \blacksquare_\mu \rrbracket^* \tau_n \\
\mathbf{y}(\Gamma) & \xrightarrow{\lfloor A \rfloor} & \llbracket \blacksquare_\mu \rrbracket^* \mathcal{T}_n
\end{array}
$$

Write $\lfloor \mathbf{Mod}_\mu(A) \rfloor \triangleq \mathbf{Mod}_\mu \circ \lfloor A \rfloor$. τ_m is a natural model, so form the pullback square for $\Gamma' \triangleq \Gamma.\mathbf{Mod}_\mu(A)$. Pasting this with the DRA pullback for $\mathbf{Mod}_\mu$ forms

$$
\begin{array}{ccccc}
\mathbf{y}(\Gamma.\mathbf{Mod}_\mu(A)) & \xrightarrow{\widehat{\lfloor \mathbf{q} \rfloor}} & \llbracket \blacksquare_\mu \rrbracket^* \widetilde{\mathcal{T}}_n & \xrightarrow{\mathbf{mod}_\mu} & \widetilde{\mathcal{T}}_m \\
\mathbf{y}(\mathbf{p}) \downarrow \;\; \lrcorner & & \downarrow \llbracket \blacksquare_\mu \rrbracket^* \tau_n \;\; \lrcorner & & \downarrow \tau_m \\
\mathbf{y}(\Gamma) & \xrightarrow{\lfloor A \rfloor} & \llbracket \blacksquare_\mu \rrbracket^* \mathcal{T}_n & \xrightarrow{\mathbf{Mod}_\mu} & \mathcal{T}_m
\end{array}
\qquad (7.1)
$$

As the outer square commutes, we can fill in the dotted arrow. By the pullback lemma, the square on the left is a pullback too. Letting $\Gamma.(\mu \mid A) \triangleq \Gamma.\mathbf{Mod}_\mu(A)$ proves that $(\tau_m)_{m \in \mathcal{M}}$ is a modal natural model.

Modal Types: This is the heart of the proof. First, we need a commuting square

$$
\begin{array}{ccc}
[\![\blacksquare_\mu]\!]^* \widetilde{\mathcal{T}}_n & \xrightarrow{\ \mathbf{mod}_\mu\ } & \widetilde{\mathcal{T}}_m \\
\downarrow & & \downarrow{\scriptstyle \tau_m} \\
[\![\blacksquare_\mu]\!]^* \mathcal{T}_n & \xrightarrow[\ \mathbf{Mod}_\mu\ ]{} & \mathcal{T}_m
\end{array}
\tag{7.2}
$$

Such a square is given as part of a DRA by definition, and is in fact a pullback!

To model the elimination rule, recall the definition of the object M used in Section 5.2.2:

As $[\![\blacksquare_\mu]\!]^*$ preserves pullbacks, the outer square is a pullback too. Hence $[\![\blacksquare_\mu]\!]^* m$ must be an isomorphism. The elimination rule requires a left-lifting structure:

$$
\vdash \mathbf{open}_\nu^\mu : ([\![\blacksquare_\mu]\!]^* m) \pitchfork ([\![\blacksquare_{\mu \circ \nu}]\!]^* \mathcal{T}_o)^*(\tau_m[-])
$$

Using the inverse of $[\![\blacksquare_\mu]\!]^* m$ we can construct this by

$$
\mathbf{open}_\nu^\mu \triangleq \lambda C.\ \lambda c.\ c \circ [\![\blacksquare_\mu]\!]^*(m^{-1})
$$

$\prod$ Structure:

Equipping each $\widetilde{\mathcal{T}}_m \xrightarrow{\tau_m} \mathcal{T}_m$ with a modal $\prod$ structure is relatively straightforward to do in the internal language; intuitively, the reason is the isomorphism

$$
([\![\blacksquare_\mu]\!]^* \tau_n)^{-1}(A) \cong \tau_m^{-1}(\mathbf{Mod}_\mu A)
$$

which is derived from the fact $\Gamma.(\mu \mid A) \triangleq \Gamma.\mathbf{Mod}_\mu A$ (where the first dot is the defined context extension, and the second dot is given by the natural model). However, we

can also prove it in a more abstract way: we paste together the two pullback squares

$$\begin{array}{ccccc}
\mathbf{P}_{[\![\blacksquare_\mu]\!]^*\tau_n}(\widetilde{\mathcal{T}}_m) & \xrightarrow{\phi_{\widetilde{\mathcal{T}}_m}} & \mathbf{P}_{\tau_m}(\widetilde{\mathcal{T}}_m) & \xrightarrow{\;\mathbf{lam}\;} & \widetilde{\mathcal{T}}_m \\
\mathbf{P}_{[\![\blacksquare_\mu]\!]^*\tau_n}(\tau_m) \downarrow & & \mathbf{P}_{\tau_m}(\tau_m) \downarrow & & \downarrow \tau_m \\
\mathbf{P}_{[\![\blacksquare_\mu]\!]^*\tau_n}(\mathcal{T}_m) & \xrightarrow[\phi_{\mathcal{T}_m}]{} & \mathbf{P}_{\tau_m}(\mathcal{T}_m) & \xrightarrow[\prod]{} & \mathcal{T}_m
\end{array}$$

The square on the right is the pullback that interprets $\prod$ in the natural model τ_m. The square on the left is a naturality square of the natural transformation

$$\phi : \mathbf{P}_{[\![\blacksquare_\mu]\!]^*\tau_n}(-) \Rightarrow \mathbf{P}_{\tau_m}(-)$$

which exists because the pullback square (7.2) defines a morphism of polynomials. Moreover, the naturality squares of ϕ are cartesian: see [New18, §§1.2.16–1.2.18]. $\square$

This theorem is a particularly flexible tool, as many modalities naturally form DRAs, and it is easier to check the DRA conditions than MTT model conditions as summarized in Definition 5.6. As a first example of this flexibility we show that it leads to an almost immediate proof of consistency.

Corollary 7.2. *No matter what the mode theory is, there is no term $\cdot \vdash M : \mathsf{Id}_{\mathbb{B}}(\mathsf{tt}, \mathsf{ff}) @ m$. In other words, MTT is consistent.*

Proof. Suppose that we have a model of MLTT with one universe in some category $\mathcal{C}$. We may construct a functor $\mathcal{M}^{\mathsf{coop}} \to \mathbf{Cat}$ by sending every mode to $\mathcal{C}$, and everything else to the identity. This is stricty 2-functorial, and each identity functor is a DRA. Hence, by Theorem 7.1 there is a model of MTT in which each mode is interpreted by $\mathcal{C}$. Therefore, if a term $M : \mathsf{Id}_{\mathbb{B}}(\mathsf{tt}, \mathsf{ff})$ were definable in MTT, we would have a term of that type in every model of MLTT. But MLTT itself is consistent: see [Coq19] for a short proof. $\square$

7.3. **DRAs from right adjoints.** Having established that a series of models of MLTT related by DRAs can be used to interpret MTT, we now turn to the problem of constructing those DRAs themselves. We shall prove a lemma that allows us to lift any well-behaved right adjoint to a DRA. Versions of this result have appeared before, both in the paper on DRAs [BCM$^+$20, Lemma 17], and in a technical report By the third author [Nuy18, Prop. 2.1.4].

In Section 5.3 we discussed the notion of a *strict* morphism of natural models. Using the same notation we define the following weaker notion.

Definition 7.3. A *weak morphism of natural models* $(\mathcal{C}, \mathcal{T}_c) \to (\mathcal{D}, \mathcal{T}_d)$ consists of a functor $F : \mathcal{C} \to \mathcal{D}$, and a commuting square

$$\begin{array}{ccc}
\widetilde{\mathcal{T}}_c & \xrightarrow{\;\widetilde{\varphi}\;} & F^*\widetilde{\mathcal{T}}_d \\
\tau_c \downarrow & & \downarrow F^*\tau_d \\
\mathcal{T}_c & \xrightarrow[\;\varphi\;]{} & F^*\mathcal{T}_d
\end{array}$$

such that $F(1) = 1$ and the canonical morphism $F(\Gamma.A) \to F\Gamma.\varphi(A)$ is an isomorphism. We say that this morphism *preserves size* whenever there is a commuting square

$$
\begin{array}{ccc}
\mathcal{S}_c & \xrightarrow{\ \mathcal{S}_\varphi\ } & F^*\mathcal{S}_d \\
{\scriptstyle \mathbf{lift}}\Big\downarrow & & \Big\downarrow {\scriptstyle F^*\mathbf{lift}} \\
\mathcal{T}_c & \xrightarrow[\ \varphi\ ]{} & F^*\mathcal{T}_d
\end{array}
$$

This kind of morphism can also be found in the thesis of [New18, §§2.3.9]. We are interested in it because it captures exactly the necessary good behaviour which is required to extend a right adjoint to act on types and terms.

Lemma 7.4. *Suppose that $(\mathcal{C}, \tau_\mathcal{C})$ and $(\mathcal{D}, \tau_\mathcal{D})$ are natural models, and that $L \dashv R$ is an adjunction between $\mathcal{C}$ and $\mathcal{D}$. If the right adjoint $R : \mathcal{C} \to \mathcal{D}$ extends to a weak morphism of natural models then it gives rise to a dependent right adjoint. Moreover, the resulting DRA is size-preserving whenever R is.*

Proof. We first fix some notation: we write $\eta : \mathsf{Id} \Rightarrow RL$ for the unit of the adjunction $L \dashv R$. Moreover, we assume a commuting square

$$
\begin{array}{ccc}
\widetilde{\mathcal{T}_\mathcal{C}} & \xrightarrow{\ \mathsf{r}\ } & R^*\widetilde{\mathcal{T}_\mathcal{D}} \\
{\scriptstyle \tau_\mathcal{C}}\Big\downarrow & & \Big\downarrow {\scriptstyle R^*\tau_\mathcal{D}} \\
\mathcal{T}_\mathcal{C} & \xrightarrow[\ \mathsf{R}\ ]{} & R^*\mathcal{T}_\mathcal{D}
\end{array}
\tag{7.3}
$$

that witnesses the weak natural model morphism structure of R, and write

$$
\nu_{\Gamma,A} : R\Gamma.\mathsf{R}(A) \xrightarrow{\cong} R(\Gamma.A)
$$

for the canonical isomorphism corresponding to $\lfloor A \rfloor : \mathbf{y}(\Gamma) \Rightarrow \mathcal{T}_\mathcal{C}$.

We construct the DRA by first applying the weak morphism R to a dependent type over a context of the form $L(\Delta)$, and then pulling that back along the unit of the adjunction. Diagrammatically, we define the square

$$
\begin{array}{ccccc}
L^*\widetilde{\mathcal{T}_\mathcal{C}} & \xrightarrow{\ L^*\mathsf{r}\ } & L^*R^*\widetilde{\mathcal{T}_\mathcal{D}} & \xrightarrow{\ \eta^*_{\widetilde{\mathcal{T}_\mathcal{D}}}\ } & \widetilde{\mathcal{T}_\mathcal{D}} \\
{\scriptstyle L^*\tau_\mathcal{C}}\Big\downarrow & & {\scriptstyle L^*R^*\tau_\mathcal{D}}\Big\downarrow & & \Big\downarrow {\scriptstyle \tau_\mathcal{D}} \\
L^*\mathcal{T}_\mathcal{C} & \xrightarrow[\ L^*\mathsf{R}\ ]{} & L^*R^*\mathcal{T}_\mathcal{D} & \xrightarrow[\ \eta^*_{\mathcal{T}_\mathcal{D}}\ ]{} & \mathcal{T}_\mathcal{D}
\end{array}
$$

The left part is the image of (7.3) under the L^* functor, and the right part is a naturality square for the natural transformation $\eta^* : L^*R^* = (RL)^* \Rightarrow \mathsf{Id}$ induced by the unit.

To show that this is a DRA we must show that this is a pullback, and it suffices to do so on the representables. Assume we have $\lfloor A \rfloor : \mathbf{y}(\Delta) \Rightarrow L^*\mathcal{T}_\mathcal{C}$ and a $\lfloor M \rfloor : \mathbf{y}(\Delta) \Rightarrow \widetilde{\mathcal{T}_\mathcal{D}}$ such

that the diagram commutes. Switching to type-theoretic notation, this amounts to a type $L(\Delta) \vdash A$ type—which gives rise to a type $R(L(\Delta)) \vdash \mathsf{R}(A)$ type by applying R—and a term $\Delta \vdash M : \mathsf{R}(A)[\eta_\Delta]$. The universal property of the pullback dictates that we must show the existence of a unique term $L(\Delta) \vdash N : A$ such that

$$RL(\Delta) \vdash \mathsf{r}(N)[\eta_\Delta] = M : \mathsf{R}(A)[\eta_\Delta] \tag{7.4}$$

First, observe that we can form the substitution $\eta_\Delta.M : \Delta \to RL\Delta.\mathsf{R}(A)$. We can then postcompose the isomorphism $\nu_{\Delta,A}$ to obtain a morphism of type $\Delta \to R(L\Delta.A)$. To this we can apply L and postcompose the counit $\epsilon_{L\Delta.A}$ to obtain a substitution

$$k \triangleq \epsilon_{L\Delta.A} \circ L(\nu_{\Delta,A} \circ \eta_\Delta.M) : L\Delta \to L\Delta.A$$

Using naturality of the counit and the equations satisfied by the canonical isomorphism $\nu_{\Delta,A}$, it is easy to show that $\mathbf{p} \circ k = \mathsf{id} : L\Delta \to L\Delta$, and hence that we can extract a term

$$L(\Delta) \vdash N \triangleq \mathbf{q}[k] : A$$

Using naturality of $\mathsf{r}(-)$, naturality of the unit, and one of the triangle identities, we can calculate that this term satisfies equation (7.4). Finally, we can prove this choice is unique by calculating that any such N necessarily satisfies $k = \mathsf{id}.N$, and hence that $\mathbf{q}[k] = N$.

It is routine to show that this is size-preserving, using the fact that R preserves size. $\square$

The converse is not in general true: a dependent right adjoint need not extend to a functor on the category of contexts. Nevertheless, it does whenever the category of contexts is *democratic* [CD14], i.e. if every context is isomorphic to extending the empty context by some type: see [BCM+20, §4.1] for a proof.

8. Presheaf Models

It is well-known that the category $\mathbf{PSh}(C)$ of presheaves over any small category $\mathcal{C}$ is a model of Martin-Löf type theory. A functor $\mu : \mathcal{C} \to \mathcal{D}$ induces by *precomposition* a functor

$$\mu^* : \mathbf{PSh}(\mathcal{D}) \to \mathbf{PSh}(\mathcal{C})$$

between categories of presheaves. This functor has a right adjoint

$$\mu_* : \mathbf{PSh}(\mathcal{C}) \to \mathbf{PSh}(\mathcal{D})$$

given by *right Kan extension* [ML78, §X.3] [Awo10, §9.6] [Rie16, §6]. We show that an appropriately functorial version of this structure can be bootstrapped into a model of MTT, where the modalities are right adjoints to this precomposition functor. More concretely, starting with a small 2-category $\mathcal{I}$, and a functor

$$J : \mathcal{I} \to \mathbf{Cat}$$

we will construct a model of MTT where each mode corresponds to the category $\mathbf{PSh}(J(i))$, and the modalities are the functors $J(f)_*$, for each $f \in \mathrm{Hom}_\mathcal{I}(i_0, i_1)$.

Context structure. We define a strict 2-functor $[\![-]\!] : \mathcal{I}^{\mathsf{coop}} \to \mathbf{Cat}$ by

$$
\begin{aligned}
i &\longmapsto \mathcal{C}[i] \triangleq \mathbf{PSh}(J(i)) \\
f : i \to j &\longmapsto [\![\blacksquare_f]\!] \triangleq J(f)^* : \mathbf{PSh}(J(j)) \to \mathbf{PSh}(J(i)) \\
\alpha : f \Rightarrow g &\longmapsto [\![\mathbf{\&}^\alpha]\!] \triangleq J(\alpha)^* : J(g)^* \Rightarrow J(f)^*
\end{aligned}
$$

The variance is correct: recall that precomposition is a strict 2-functor

$$(-)^* : \mathbf{Cat}^{\mathsf{coop}} \to \mathbf{Cat}$$

which maps a functor $f : \mathcal{C} \to \mathcal{D}$ to $f^* : \mathbf{PSh}(\mathcal{D}) \to \mathbf{PSh}(\mathcal{C})$, and a natural transformation $\alpha : f \Rightarrow g$ to $\alpha^* : g^* \Rightarrow f^*$, given by $\alpha^*_{P,c} \triangleq P(\alpha_c) : P(g(c)) \to P(f(c))$. 2-functoriality is immediate, as for example $J(f)^* \circ J(g)^* = (J(g) \circ J(f))^* = J(g \circ f)^*$.

Modal natural models. To interpret the 'mode-local' structure we must construct a modal natural model in each $\mathcal{C}[i]$. It is well-known that every presheaf topos $\mathbf{PSh}(\mathcal{C})$ gives rise to a rich model of MLTT: see e.g. [Hof97, §4.1] or [Coq13].

Contexts are interpreted as objects of the presheaf category $\mathbf{PSh}(\mathcal{C})$. *Types* are presheaves $\mathbf{PSh}(\int \Gamma)$ over the category of elements $\int \Gamma$ of a context $\Gamma : \mathbf{PSh}(\mathcal{C})$.[8] We define the action of a substitution $\sigma : \Delta \Rightarrow \Gamma$ on a type $A : \mathbf{PSh}(\int \Gamma)$ by

$$
A[\sigma] \triangleq (\textstyle\int \Delta)^{\mathsf{op}} \xrightarrow{(\int \sigma)^{\mathsf{op}}} (\textstyle\int \Gamma)^{\mathsf{op}} \xrightarrow{A} \mathbf{Set}
$$

This is functorial because $\int - : \mathbf{PSh}(\mathcal{C}) \to \mathbf{Cat}$ and $-^{\mathsf{op}} : \mathbf{Cat} \to \mathbf{Cat}$ are.

A *term* of type A is a global section of A, i.e. a morphism $\mathrm{Hom}_{\mathbf{PSh}(\int \Gamma)}(1, A)$. We define the action of a substitution $\sigma : \Delta \Rightarrow \Gamma$ on a term $M : \mathrm{Hom}(1, A)$ by whiskering:

$$
M[\sigma] \triangleq M * (\textstyle\int \sigma)^{\mathsf{op}} : 1 \circ (\textstyle\int \sigma)^{\mathsf{op}} \Rightarrow A \circ (\textstyle\int \sigma)^{\mathsf{op}} = A[\sigma]
$$

As $1 \circ \int \sigma^{\mathsf{op}} = 1$, this has the right type. It is functorial because whiskering is.

Remark 8.1 (Size Issues). One cannot be too careful with size issues when considering presheaf models. In Section 5.2 we demanded that the category of contexts be *small*, so that we can then formulate a large category of models. $\mathbf{PSh}(\mathcal{C})$ is certainly not small. We can mend this by assuming a Grothendieck universe $\mathcal{V}$ large enough to contain $\mathcal{C}$ in the ambient set theory, and re-defining $\mathbf{PSh}(\mathcal{C})$ to consist of the presheaves $P : \mathcal{C}^{\mathsf{op}} \to \mathcal{V}$ with *small fibers*. As $\mathcal{V}$ is closed under all set-theoretic operations, this is still a model, and $\mathbf{PSh}(\mathcal{C})$ is small.

To interpret universes we need to know that the fibers of types in $\mathbf{PSh}(\int \mathcal{C})$ are even *smaller*. Thus, we further assume a second, inner Grothendieck universe $\mathcal{V}' \subset \mathcal{V}$. To a type theorist, this is just the standard technique of 'bumping' a universe level.

Connectives. Presheaf models support dependent sums and products, and extensional identity types (and therefore intensional identity types): see [Hof97, §4.2]. On the premise that the underlying set theory has a set-theoretic universe, they also support a universe, through a construction of [HS97]. See also [Coq13].

---

[8]There is an equivalence $\mathbf{PSh}(\int \Gamma) \simeq \mathbf{PSh}(\mathcal{C})/\Gamma$ which shows that types are families $P \Rightarrow \Gamma$ in the slice category. However, using the latter definition would lead to strictness issues.

Modalities. It remains to show that each $\llbracket \blacksquare_f \rrbracket \triangleq J(f)^* : \mathbf{PSh}(J(j)) \to \mathbf{PSh}(J(i))$ has a corresponding modality acting on terms and types. We will do so by using previous results. By Theorem 7.1, it suffices to construct a *dependent right adjoint* to $J(f)^*$. But recall that each lock functor $J(f)^*$ already has an (ordinary) right adjoint, viz. $J(f)_*$. Thus, by Lemma 7.4 it will suffice to show that the action of this right adjoint extends to types and terms. We then have DRAs, and hence a model of MTT.

The following result has previously been shown in a tech report by the third author [Nuy18, Prop. 2.2.9]. We reproduce it here for the sake of completeness.

Lemma 8.2. *The right adjoint to precomposition* $\mu_* : \mathbf{PSh}(\mathcal{C}) \to \mathbf{PSh}(\mathcal{D})$ *induces a DRA for any* $\mu : \mathcal{C} \to \mathcal{D}$. *Moreover,* μ_* *is size-preserving for any Grothendieck universe.*

Proof. We use Lemma 7.4 once more. The action of μ_* on types and terms is given by

$$\mu_* A \in \mathbf{PSh}(\textstyle\int \mu_* \Gamma) = (D \in \mathcal{D}, a \in \mu_* \Gamma(D)) \mapsto \mathrm{Hom}_{\mathbf{PSh}(\int \mu^*(\mathbf{y}(D)))}(1, A\left[\widehat{\lfloor a \rfloor}\right])$$

$$\mu_* M \in \mathrm{Hom}(1, \mu_* A) = (D \in \mathcal{D}, a \in \mu_* \Gamma(D)) \mapsto (\textstyle\int \widehat{\lfloor a \rfloor})^* M$$

Both of these actions are well-typed. For types, as $a \in \mu_* \Gamma(D)$ we have $\lfloor a \rfloor : \mathbf{y}(D) \Rightarrow \mu_* \Gamma$, so by transposition $\widehat{\lfloor a \rfloor} : \mu^*(\mathbf{y}(D)) \Rightarrow \Gamma$. For terms, notice that $\int \widehat{\lfloor a \rfloor} : \int \mu^*(\mathbf{y}(D)) \to \int \Gamma$, recall that $A[\sigma] \triangleq A \circ \int \sigma$, and that precomposition preserves the terminal object on-the-nose.

The presheaf action. The action of $\mu_* A$ is subtle: it is given by the functor

$$(\textstyle\int \mu^* \mathbf{y}(f))^* : \mathbf{PSh}(\textstyle\int \mu^* \mathbf{y}(D)) \to \mathbf{PSh}((\textstyle\int \mu^* \mathbf{y}(D'))$$

for each $f : \mathrm{Hom}_{\mathcal{D}}(D', D)$. In more detail, given $f : \mathrm{Hom}_{\mathcal{D}}(D', D)$, $a \in \mu_* \Gamma(D)$, $A \in \mathbf{PSh}(\int \Gamma)$, and $x \in \mu_* A(D, a) \triangleq \mathrm{Hom}(1, A\left[\widehat{\lfloor a \rfloor}\right])$, we define $x \cdot f \in \mu_* A(D', a \cdot f)$ by

$$x \cdot f \triangleq (\textstyle\int \mu^* \mathbf{y}(f))^*(x) : \mathrm{Hom}_{\mathbf{PSh}(\int \mu^*(\mathbf{y}(D')))}((\textstyle\int \mu^* \mathbf{y}(f))^* 1, (\textstyle\int \mu^* \mathbf{y}(f))^*(A\left[\widehat{\lfloor a \rfloor}\right]))$$

This is of the right type; reindexing preserves the terminal, $(\int \mu^* \mathbf{y}(f))^* 1 = 1$. Moreover,

$$\widehat{\lfloor a \rfloor} \circ \mu^* \mathbf{y}(f) = \widehat{\lfloor a \rfloor \circ \mathbf{y}(f)} = \widehat{\lfloor a \cdot f \rfloor}$$

by naturality of the adjunction and of Yoneda. Using this calculation, we see that

$$(\textstyle\int \mu^* \mathbf{y}(f))^*(A\left[\widehat{\lfloor a \rfloor}\right]) \triangleq A \circ \textstyle\int \widehat{\lfloor a \rfloor} \circ \textstyle\int \mu^* \mathbf{y}(f) = A \circ \textstyle\int \widehat{\lfloor a \cdot f \rfloor} \triangleq A\left[\widehat{\lfloor a \cdot f \rfloor}\right]$$

Hence $x \cdot f \in \mu_* A(D', a \cdot f)$. This assignment is functorial because $\int -$, $(-)^*$ and $\mathbf{y}(-)$ are.

Naturality. We must show that both of these definitions are natural with respect to substitution, i.e. that $(\mu_* A)[\mu_* \gamma] = \mu_*(A[\gamma])$, and similarly for terms.

For types, suppose we are given $\gamma : \Delta \to \Gamma$ and $A \in \mathbf{PSh}(\int \Gamma)$. Carefully unfolding both sides of the desired equation, for any $D \in \mathcal{D}$ and $a \in \mu_* \Delta(D)$ we must show that

$$\mathrm{Hom}_{\mathbf{PSh}(\int \mu^* \mathbf{y}(D))}(1, A\left[\widehat{\lfloor \mu_* \gamma_D(a) \rfloor}\right]) = \mathrm{Hom}_{\mathbf{PSh}(\int \mu^* \mathbf{y}(D))}(1, A[\gamma]\left[\widehat{\lfloor a \rfloor}\right])$$

But, by naturality of both the adjunction and Yoneda:

$$\gamma \circ \widehat{\lfloor a \rfloor} = \widehat{\mu_* \gamma \circ \lfloor a \rfloor} = \widehat{\lfloor \mu_* \gamma_D(a) \rfloor}$$

Hence the two sets are the same. The calculation for terms is of a similar ilk.

Preservation of context extension. We would like to show that the canonical morphism

$$\mu_*(\Gamma.A) \xrightarrow{\langle \mu_* \mathbf{p}, \mu_* \mathbf{q} \rangle} \mu_* \Gamma.\mu_* A$$

is invertible. Consider an element $e : \mathbf{y}(D) \Rightarrow \mu_*(\Gamma.A)$. We can transpose along the adjunction $\mu^* \dashv \mu_*$ and decompose it to obtain a substitution and a term

$$e_0 : \mu^* \mathbf{y}(D) \Rightarrow \Gamma \qquad\qquad e_1 : \mathrm{Hom}_{\mathbf{PSh}(\int \mu^* \mathbf{y}(D))}(1, A[e_0])$$

We can thus write $e = \widehat{\langle e_0, e_1 \rangle}$. Thus, we can use naturality of the adjunction and of substitution to compute the action of $\langle \mu_* \mathbf{p}, \mu_* \mathbf{q} \rangle$ on this e:

$$\langle \mu_* \mathbf{p}, \mu_* \mathbf{q} \rangle \circ e = \langle \mu_* \mathbf{p} \circ e, (\mu_* \mathbf{q})[e] \rangle = \langle \mathbf{p} \circ \widehat{\langle e_0, e_1 \rangle}, \mathbf{q}[\langle e_0, e_1 \rangle] \rangle = \langle \widehat{e_0}, e_1 \rangle$$

We can then specify an inverse on generalized elements by $\langle \gamma, M \rangle \mapsto \widehat{\langle \widehat{\gamma}, M \rangle}$.

Size preservation is immediate: if A is small then so are its reindexings and the collections of points at each slice. $\qquad\square$

8.1. **The other adjunction.** We have so far concentrated on the adjunction $\mu^* \dashv \mu_*$ that arises through right Kan extension. Nevertheless, precomposition also has a left adjoint $\mu_!$ arising from *left Kan extension*. Might we also be able to interpret the lock functors by this left adjoint $\mu_!$, and lift precomposition μ^* to a modality instead?

It is in fact relatively easy to show that μ^* extends to a dependent right adjoint. However, the left Kan extensions $\mu_!$ cannot be assembled into a modal context structure. The reason is that context structures are strict 2-functors, but left Kan extensions do not compose strictly: we only have an isomorphism $F_! \circ G_! \cong (G \circ F)_!$. We have proven a strictification theorem that straightens these issues, but that is beyond the scope of this paper.

9. Guarded Recursion

We now show how MTT can be applied to a well-known modal situation: guarded recursion. By instantiating MTT with a carefully chosen mode theory and axiomitizing certain operations specific to guarded recursion (i.e. Löb induction), we obtain a calculus for guarded recursion simpler than prior hand-crafted calculi. We demonstrate the practicality of this guarded variant of MTT by reproducing some examples from prior work on guarded recursion [BGC+16].

The key idea of guarded recursion [Nak00] is to use a modality $\blacktriangleright$, usually called *later*, to mark the types of data that may be used only if some 'computational progress' (e.g. a tick of a clock) has taken place, thereby enforcing productivity at the level of types. The later modality is usually equipped with three basic operations:

$$\mathsf{next} : A \to \blacktriangleright A \qquad (\circledast) : \blacktriangleright(A \to B) \to \blacktriangleright A \to \blacktriangleright B \qquad \mathrm{l\ddot{o}b} : (\blacktriangleright A \to A) \to A$$

The first two make $\blacktriangleright$ into an *applicative functor* [MP08]. The third, which is commonly known as Löb induction, is a *guarded fixed point operator* [ML13]: it enables us to make definitions by *provably productive* recursion.

$\blacktriangleright$ also applies to the universe, so one can define data types by guarded recursion. The classic example is the *guarded stream type* $\mathsf{Str}_A \cong A \times \blacktriangleright \mathsf{Str}_A$, with constructor

$$\mathsf{cons}_A : A \times \blacktriangleright \mathsf{Str}_A \cong \mathsf{Str}_A$$

$$\delta \circ \gamma \leq 1 \quad 1 = \gamma \circ \delta$$
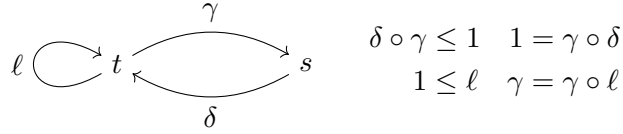$$1 \leq \ell \quad \gamma = \gamma \circ \ell$$

Figure 11: The 'adjoint bowling pin' $\mathcal{M}_g$: a mode theory for guarded recursion.

The presence of the modality enforces the requirement that the head of the stream is available immediately, but the tail may only be accessed after some productive work has taken place. This allows us to e.g. construct an infinite stream of ones:

$$\mathsf{inf\_stream\_of\_ones} \triangleq \mathrm{l\ddot{o}b}(s.\ \mathsf{cons}(1, s))$$

Unlike the ordinary type of streams, Str_A does not behave like a coinductive type: we may only define *causal* operations, which excludes useful functions (e.g. the tail function). In order to regain coinductive behaviour, [CBGB15] introduced the *always* modality $\Box$, an idempotent comonad for which

$$\Box \blacktriangleright A \simeq \Box A. \tag{$*$}$$

Combining $\Box$ and $\blacktriangleright$ in the same system has proven tricky. Previous work has used *delayed substitutions* [BGC$^+$16], or replaced $\Box$ with *clock quantification* [AM13, Møg14, BM15, BGM17]. Neither solution is entirely satisfactory: the former poses serious implementation and usability issues, and the latter does not enjoy the conceptual simplicity of a single modality. We will show that MTT enables us to effortlessly combine the two modalities whilst satisfying ($*$).

To encode guarded recursion inside MTT, we must

(1) construct a mode theory that induces an applicative functor $\blacktriangleright$ and an idempotent comonad $\Box$ satisfying ($*$),
(2) construct the *intended model* of MTT with this mode theory, i.e. a model where these modalities are interpreted in the standard way [BMSS12], and
(3) include Löb induction as an axiom, and use it to reason about guarded streams.

9.1. **A guarded mode theory.** We define $\mathcal{M}_g$ to be the mode theory generated by the graph and equations of Fig. 11. We require that $\mathcal{M}_g$ be poset-enriched, i.e. that there be at most one 2-cell between a pair of modalities μ, ν, which we denote by $\mu \leq \nu$ when it exists. Consequently, we need not state any coherence equations between 2-cells.

Unlike prior guarded type theories, $\mathcal{M}_g$ has *two modes*. We will think of elements of s as being *constant types and terms*, while types in t may *vary over time*. Observe that we can thence construct a composite modality $b \triangleq \delta \circ \gamma$. Moreover, this modality is idempotent, for $b \circ b = \delta \circ \gamma \circ \delta \circ \gamma = \delta \circ \gamma = b$. We will prove in Section 9.3 that

Lemma 9.1. $\langle b \mid - \rangle$ *is an idempotent comonad and* $\langle \ell \mid - \rangle$ *is an applicative functor.*

9.2. **Decomposing the standard model.** The above mode theory arises from a careful and informative decomposition of the standard model of guarded recursion, namely the *topos of trees* $\mathbf{PSh}(\omega)$, along with the later and always endomodalities.

The topos of trees consists of presheaves over the natural numbers, seen as a poset with the usual order. An element $x_n \in X(n)$ of a presheaf $X : \mathbf{PSh}(\omega)$ represents an element computed after n steps of computation. The restriction maps $r_n : X(n+1) \to X(n)$ trim an element computed after $n+1$ steps to its form at the preceding moment in time. The canonical example is given by $X(n) \triangleq \{\text{streams of length } n\}$, where r_n deletes the last element of a stream of length $n+1$. The later and always endomodalities are given by delaying the computation by one step, and by taking global sections (total elements):

$$(\blacktriangleright X)(n) \triangleq \begin{cases} \{*\} & \text{if } n = 0 \\ X(n-1) & \text{if } n > 0 \end{cases} \qquad (\square X)(n) \triangleq \mathrm{Hom}_{\mathbf{PSh}(\omega)}(1, X)$$

To arrive at the mode theory above, one must notice that the comonad $\square$ results in a *constant* presheaf, namely one which consists of the same set at each time. We can thus decompose it into the adjunction

$$\tag{9.1}$$

Γ maps $X : \mathbf{PSh}(\omega)$ to the set of its global sections $\mathrm{Hom}_{\mathbf{PSh}(\omega)}(1, X)$, and Δ maps a set S to the constant presheaf $(\Delta S)(n) \triangleq S$. It is well-known that $\Delta \dashv \Gamma$, and 'always' is given by the induced comonad $\square \triangleq \Delta \circ \Gamma$. This explains the provenance of the two modes in Figure 11: s stands for *sets*, and t for *timed sets*, i.e. presheaves over ω.

We want to bootstrap (9.1) into a model of MTT. We will do so by leveraging an impressive sequence of facts:

- Both categories in (9.1) are presheaf categories, and hence models of MLTT: see Section 8.
- Every functor in (9.1) is a right adjoint.
- The corresponding left adjoints are introduced by precomposition, and hence can easily be arranged into a modal context structure for the mode theory $\mathcal{M}_g$ as per Section 5.1.
- Hence, by uniqueness of adjoints the functors in (9.1) are induced by right Kan extension. Consequently, they can be bootstrapped into dependent right adjoints, by Lemma 8.2.
- Therefore, by Theorem 7.1, this data yields a model of MTT with mode theory $\mathcal{M}_g$.

Let us elaborate on this chain of reasoning. First, we identify the category $\mathbf{Set}$ and the category $\mathbf{PSh}(1)$ of presheaves over the terminal category. Second, we construct the two left adjoints. As ω has an initial object 0, we obtain a left adjoint to the discrete functor Δ, given by

$$\Pi_0(X) \triangleq X(0)$$

It is easy to see that $\Pi_0 \dashv \Delta$: by naturality at the unique morphism $0 \leq n$ we see that any $\alpha : X \Rightarrow \Delta S$ is fully determined by the component $\alpha_0 : X(0) \to S$. Furthermore, recall from the work of [BMSS12] that the later modality $\blacktriangleright$ has a left adjoint $\blacktriangleleft : \mathbf{PSh}(\omega) \to \mathbf{PSh}(\omega)$

(pronounced 'earlier'), given by

$$(\blacktriangleleft X)(n) \triangleq X(n+1)$$

It remains to show that the three left adjoints—Π_0, Δ, and $\blacktriangleleft$—are given by precomposition. We define three monotone functions between the posets $1 \triangleq \{*\}$ and ω:

$$
\begin{array}{ccc}
K_0 : 1 \rightarrow \omega & \quad !_\omega : \omega \rightarrow 1 & \quad l : \omega \rightarrow \omega \\
* \mapsto 0 & \quad n \mapsto * & \quad n \mapsto n+1
\end{array}
$$

Identifying $\mathbf{Set}$ with $\mathbf{PSh}(1)$, we see that

$$\Pi_0 = K_0^* : \mathbf{PSh}(\omega) \rightarrow \mathbf{Set} \quad\quad \Delta = !_\omega^* : \mathbf{Set} \rightarrow \mathbf{PSh}(\omega) \quad\quad \blacktriangleleft = l^* : \mathbf{PSh}(\omega) \rightarrow \mathbf{PSh}(\omega)$$

Moreover, we trivially have the following pointwise equations and inequalities:

$$
\begin{array}{cccc}
\mathrm{id}_\omega \leq l & \quad K_0 \circ !_\omega \leq \mathrm{id}_\omega & \quad \mathrm{id}_1 = !_\omega \circ K_0 & \quad !_\omega = !_\omega \circ l
\end{array}
$$

Seeing posets as categories, pointwise inequalities are simply natural transformations between monotone maps. By feeding them into the strict 2-functor $(-)^* : \mathbf{Cat}^{\mathsf{coop}} \rightarrow \mathbf{Cat}$, we are able to define a strict 2-functor $[\![\blacksquare_-]\!] : \mathcal{M}_g{}^{\mathsf{coop}} \rightarrow \mathbf{Cat}$ which maps

$$
\begin{array}{ccccc}
\gamma & : t \rightarrow s & \longmapsto & [\![\blacksquare_\gamma]\!] = \Delta & : \mathbf{Set} \rightarrow \mathbf{PSh}(\omega) \\
\delta & : s \rightarrow t & \longmapsto & [\![\blacksquare_\delta]\!] = \Pi_0 & : \mathbf{PSh}(\omega) \rightarrow \mathbf{Set} \\
\ell & : t \rightarrow t & \longmapsto & [\![\blacksquare_\ell]\!] = \blacktriangleleft & : \mathbf{PSh}(\omega) \rightarrow \mathbf{PSh}(\omega)
\end{array}
$$

This fully specifies the modal context structure, which consists of left adjoints. Each of these left adjoints is given by precomposition. Thus, the unique corresponding right adjoint is given by right Kan extension (see Section 8). Hence, by Lemma 8.2 and Theorem 7.1,

Theorem 9.2. *There is a model of MTT with mode theory $\mathcal{M}_g$, interpreting s as $\mathbf{Set}$ and t as $\mathbf{PSh}(\omega)$. Furthermore, this model interprets δ by the dependent right adjoint arising from $\Pi_0 \dashv \Delta$, γ by $\Delta \dashv \Gamma$, and ℓ by $\blacktriangleleft \dashv \blacktriangleright$.*

Remark 9.3. This mode theory is a poset-enriched category. As a result, the key substitutions are unique: for any μ, ν there is at most one substitution $\Gamma.\blacksquare_\mu \vdash \mathbf{q}_\Gamma^{\nu \leq \mu} : \Gamma.\blacksquare_\nu @ m$. This property means that we can elide them without ambiguity. However, this may sometimes make type-checking on pen-and-paper difficult, so we employ a simplified notation: we will write $A^{\nu \leq \mu}$ or $M^{\nu \leq \mu}$ for the application of the unique key substitution $\nu \leq \mu$ in context $\Gamma.\blacksquare_\mu$. For instance, given a type $\Gamma.\blacksquare_1 = \Gamma \vdash A \; \mathsf{type}_1 @ t$ we can form the type $\Gamma.\blacksquare_\ell \vdash A^{1 \leq \ell} \; \mathsf{type}_1 @ t$, and hence the type $\Gamma \vdash \langle \ell \mid A^{1 \leq \ell} \rangle \; \mathsf{type}_1 @ t$.

9.3. Guarded recursion, internally.
Given the model that we constructed above, we feel perfectly justified in defining the following shorthands within MTT:

$$
\square A \triangleq \langle b \mid A \rangle \quad\quad \blacktriangleright A \triangleq \langle \ell \mid A \rangle \quad\quad \Gamma A \triangleq \langle \gamma \mid A \rangle \quad\quad \Delta A \triangleq \langle \delta \mid A \rangle
$$

where $b \triangleq \delta \circ \gamma$. The aim of this section is to show that MTT equipped with $\mathcal{M}_g$ and these shorthands can be used to reason about guarded recursion. In particular, we will show that this is strict improvement on previous solutions, by establishing that

(1) When restricted to mode s, the type theory is simply standard Martin-Löf Type Theory.
(2) The modalities on mode t give rise to the standard modalities and operations of Guarded Type Theory [BGC+16] inside the type theory.

First, we wish to show that if we restrict ourselves to endomodalities $\mu \in \mathrm{Hom}(s, s)$ from sets to sets, the type theory is just MLTT. Looking at Fig. 11 as a finite state machine, we see that all loops on s are of the form $\gamma \circ \ell^n \circ \delta$, and the equations of $\mathcal{M}_g$ allow us to prove that they are all equal to the identity 1_s. It follows that $\langle \mu \mid A \rangle \simeq A$. Finally, as there is no non-trivial 2-cell $1_s \Rightarrow 1_s$ the variable rule reduces to

$$\frac{\mu \in \mathrm{Hom}(s, s) \qquad \Gamma \ \mathsf{ctx} @ s \qquad \Gamma \vdash A \ \mathsf{type}_\ell @ s \qquad (x : (\mu \mid A)) \in \Gamma}{\Gamma \vdash x : A @ s}$$

which is essentially the usual variable rule of MLTT.

Second, we use the combinators of Section 3.2 to prove that $\square$ is an idempotent comonad.

$$\begin{aligned}
\mathsf{dup}_A &\quad : \square A \xrightarrow{\simeq} \square\square A \qquad & \mathsf{extract}_A &\quad : \square A \to A^{b \leq 1} \\
\mathsf{dup}_A(x) &\triangleq \mathbf{comp}^{-1}_{b,b}(x) & \mathsf{extract}_A(x) &\triangleq \mathbf{triv}^{-1}(\mathbf{coe}[b \leq 1](x))
\end{aligned}$$

Recall the K operator $- \circledast_b - : \square(A \to B) \to \square A \to \square B$ for the modality b, which was defined in Section 3.1. Writing $\mathsf{box}(M) \triangleq \mathsf{mod}_b(M)$, the claim that $\square$ is an internal idempotent comonad amounts to defining terms of the following types.

$$(x : \square A) \to \mathsf{Id}_{\square A}(x, \mathsf{box}(\mathsf{extract}) \circledast_b \mathsf{dup}(x)) \tag{9.2}$$

$$(x : \square A) \to \mathsf{Id}_{\square A}(x, \mathsf{extract}(\mathsf{dup}(x))) \tag{9.3}$$

$$(x : \square A) \to \mathsf{Id}_{\square\square\square A}(\mathsf{dup}(\mathsf{dup}(x)), \mathsf{box}(\mathsf{dup}) \circledast_b \mathsf{dup}(x)) \tag{9.4}$$

These can be constructed by unfolding and modal induction on $x : \square A$.

The K operator $- \circledast_\ell - : \blacktriangleright(A \to B) \to \blacktriangleright A \to \blacktriangleright B$ for the modality ℓ almost proves that $\blacktriangleright$ is an applicative functor. It remains to show that $\blacktriangleright$ is pointed:

$$\begin{aligned}
\mathsf{next}_A &\quad : A \to \blacktriangleright A \\
\mathsf{next}_A(x) &\triangleq \mathbf{coe}[1 \leq \ell](\mathbf{triv}(x))
\end{aligned}$$

Next, we show the defining equivalence $(*)$. We calculate that $b \circ \ell \triangleq \delta \circ \gamma \circ \ell = \delta \circ \gamma \triangleq b$, and hence that the equivalence is a corollary of a combinator given in Section 3.1:

$$\begin{aligned}
\mathsf{now}_A(x) &\ : \ \square\blacktriangleright A \xrightarrow{\simeq} \square A \\
\mathsf{now}_A(x) &\triangleq \mathbf{comp}^{-1}_{b,\ell}(x)
\end{aligned}$$

As a sanity check, we can compute that the following composite is the identity:

$$\square A \xrightarrow{\ \mathsf{box}(\mathsf{next}) \circledast -\ } \square\blacktriangleright A \xrightarrow{\ \mathsf{now}\ } \square A$$

The calculation is as follows:

$$\begin{aligned}
&\mathbf{comp}_{b,\ell}(\mathsf{mod}_b(\mathbf{coe}[1 \leq \ell](\mathbf{triv}(-))) \circledast x) && \text{by induction, suppose } x = \mathsf{mod}_b(y) \\
&= \mathbf{comp}_{b,\ell}(\mathsf{mod}_b(\mathbf{coe}[1 \leq \ell](\mathbf{triv}(-))) \circledast \mathsf{mod}_b(y)) \\
&= \mathbf{comp}_{b,\ell}(\mathsf{mod}_b(\mathbf{coe}[1 \leq \ell](\mathbf{triv}(y)))) \\
&= \mathbf{comp}_{b,\ell}(\mathsf{mod}_b(\mathsf{mod}_\ell(y))) \\
&= \mathsf{mod}_b(y) && \text{as } b \circ \ell = b \\
&= x
\end{aligned}$$

The only thing that remains is to add *Löb induction*. This is a *modality-specific operation* that cannot be expressed in the mode theory, so we must add it as an axiom:

$$\frac{\Gamma \; \mathsf{ctx} \, @\, t \qquad \Gamma \vdash A \; \mathsf{type}_1 \, @\, t}{\Gamma \vdash \mathrm{l\ddot{o}b} : (\blacktriangleright A^{1 \leq \ell} \to A) \to A \, @\, t}$$

$$\frac{\Gamma \; \mathsf{ctx} \, @\, t \qquad \Gamma \vdash A \; \mathsf{type}_1 \, @\, t \qquad \Gamma \vdash M : \blacktriangleright A^{1 \leq \ell} \to A \, @\, t}{\Gamma \vdash \mathrm{l\ddot{o}b}(M) = M(\mathsf{next}(\mathrm{l\ddot{o}b}(M))) : (\blacktriangleright A^{1 \leq \ell} \to A) \to A \, @\, t}$$

Notice that these rules are only added in mode t, as they only admit an interpretation in $\mathbf{PSh}(\omega)$ [BMSS12, §2]. Unfortunately, these ad-hoc additions mean that the canonicity theorem of Section 6 no longer applies.

9.4. **Reasoning about Streams.** We now put MTT to work: we will use it to reason about infinite streams defined by guarded recursion. We will demonstrate that the rules and axioms given in Section 9.3 suffice to carry out coinductive constructions. In particular, we will reproduce an example of [BGC$^+$16]: we will show that $\mathsf{zipWith}(f)$ on a coinductive stream is commutative whenever f itself is.

In order to simplify our working, we will swap the intensional equality type $\mathsf{Id}_A(M, N)$ with an *extensional identity type* $\mathsf{Eq}_A(M, N)$. This has the same introduction rule, but its elimination is replaced by the usual *equality reflection rule*

$$\frac{\Gamma \vdash P : \mathsf{Eq}_A(M_0, M_1) \, @\, m}{\Gamma \vdash M_0 = M_1 : A \, @\, m}$$

This is straightforwardly interpreted in the model, as both modes are mapped to presheaf toposes. The switch to extensional equality is not strictly necessary: we could carry out the following calculations with intensional identity, at the price of significantly more verbose terms. Moreover, the need for the function extensionality axiom would arise. However, adding Löb induction has already ensured that type-checking is undecidable, so nothing of value is lost by making the switch to extensional type theory for these examples.

We begin with a simple reasoning principle. Eliding $(-)^{1 \leq \ell}$ annotations:

Lemma 9.4. $(A : \mathsf{U})(x, y : \mathsf{El}(A)) \to \blacktriangleright\mathsf{Eq}_{\mathsf{El}(A)}(x, y) \to \mathsf{Eq}_{\blacktriangleright\mathsf{El}(A)}(\mathsf{next}(x), \mathsf{next}(y))$

Proof. Suppose $x, y : \mathsf{El}(A)$ and $p : \blacktriangleright\mathsf{Eq}_{\mathsf{El}(A)}(x, y)$; to show $\mathsf{next}(x) = \mathsf{next}(y) : \blacktriangleright\mathsf{El}(A)$. By congruence and the elimination rule for the modality, it suffices to prove $x = y : \mathsf{El}(A)$ in the locked context $A : \mathsf{U}, x : \mathsf{El}(A), y : \mathsf{El}(A), p : (\ell \mid \mathsf{Eq}_{\mathsf{El}(A)}(x, y)), \blacksquare_\ell$. But by the variable rule we have $p : \mathsf{Eq}_{\mathsf{El}(A)}(x, y)$ in this context, and hence $x = y : \mathsf{El}(A)$. $\qquad\square$

This can be used to prove *internally* that guarded fixed points are unique.

Theorem 9.5. $\mathrm{l\ddot{o}b}(M)$ *is the unique guarded fixed point of* $M : \blacktriangleright\mathsf{El}(A) \to \mathsf{El}(A)$, *i.e.*

$$(A : \mathsf{U})(x : \mathsf{El}(A)) \to \mathsf{Eq}_{\mathsf{El}(A)}(M(\mathsf{next}(x)), x) \to \mathsf{Eq}_{\mathsf{El}(A)}(\mathrm{l\ddot{o}b}(M), x)$$

Proof. Suppose $A : \mathsf{U}$; to show $(x : \mathsf{El}(A)) \to \mathsf{Eq}_{\mathsf{El}(A)}(M(\mathsf{next}(x)), x) \to \mathsf{Eq}_{\mathsf{El}(A)}(\mathrm{l\ddot{o}b}(M), x)$ by Löb induction. Thus, assume that

$$f : \blacktriangleright((x : \mathsf{El}(A)) \to \mathsf{Eq}_{\mathsf{El}(A)}(M(\mathsf{next}(x)), x) \to \mathsf{Eq}_{\mathsf{El}(A)}(\mathrm{l\ddot{o}b}(M), x))$$

If $x : A$ and $p : \mathsf{Eq}_{\mathsf{El}(A)}(M(\mathsf{next}(x)), x)$, we calculate that

$$\text{löb}(M) = M(\mathsf{next}(\text{löb}(M))) \quad \text{unfolding rule for löb}$$
$$= M(\mathsf{next}(x)) \qquad\qquad \text{by Lemma 9.4 on } f \circledast \mathsf{next}(x) \circledast \mathsf{next}(p) : \blacktriangleright\mathsf{Eq}_{\mathsf{El}(A)}(\text{löb}(M), x)$$
$$= x \qquad\qquad\qquad\quad \text{by } p$$

Thus, this type is inhabited by the term $\lambda A.\ \text{löb}(\lambda f.\ \lambda x.\ \lambda p.\ \mathsf{refl}(x))$. $\square$

We can also use the Löb operator on the universe to form *guarded recursive types*. For example, streams can be defined by[9]

$$\mathsf{Str} \quad : \quad \mathsf{U} \to \mathsf{U} \, @ \, s$$
$$\mathsf{Str}(A) \triangleq \Gamma(\text{löb}(\lambda X.\ \Delta A \times \mathsf{let}\ \mathsf{mod}_\ell(Y) \leftarrow X\ \mathsf{in}\ \blacktriangleright Y))$$

Str maps a constant set, i.e. a type $A \, @ \, s$, to the type of streams over A, which is again a constant set. This is done by first defining a timed set

$$A : (1 \mid \mathsf{U}), \blacksquare_\gamma \vdash \mathsf{Str}'(A) \triangleq \text{löb}(\lambda X.\ \Delta A \times \mathsf{let}\ \mathsf{mod}_\ell(Y) \leftarrow X\ \mathsf{in}\ \blacktriangleright Y) : \mathsf{U} \, @ \, t$$

$\mathsf{Str}'(A)$ is defined by Löb induction: assuming $X : (1 \mid \blacktriangleright\mathsf{U}^{1 \leq \ell})$ we must define an element of the timed universe. This is given as the product of

- the set $A \, @ \, s$, considered as a constant-everywhere timed set $\Delta A \, @ \, t$;
- a guarded recursive call, which represents the rest of the stream.

Recalling that $\mathsf{U}^{1 \leq \ell} = \mathsf{U}$, the second component is given by modal elimination. Nevertheless, it is not immediate that the first component type-checks: we must show that

$$A : (1 \mid \mathsf{U}), \blacksquare_\gamma, \blacksquare_\delta \vdash A : \mathsf{U} \, @ \, s$$

But $\gamma \circ \delta = 1$, so the context is equal to $A : (1 \mid \mathsf{U}), \blacksquare_1$ and we can use A. Unfolding the guarded fixed point, we have that

$$\mathsf{Str}'(A) = \Delta A \times \blacktriangleright\mathsf{Str}'(A) : U \, @ \, t$$

We apply Γ to 'totalize' this into the constant set $\mathsf{Str}(A) \, @ \, s$ of guarded streams.

Even though not immediately obvious, there is a serious advantage in expressing this definition in a way that spans two modes. In previous work [BGC$^+$16] the stream type $\mathsf{Str}(A)$ was coinductive only if A was provably a 'constant set,' i.e. if $A \simeq \square A$. Theorems about streams had to carry around a proof of this equivalence. In our case, defining $\mathsf{Str}(A)$ at the mode s of constant sets automatically ensures that. Hence, $\mathsf{Str}(A)$ is equivalent to the familiar definition, but we no longer need to propagate proofs of constancy.

$\mathsf{Str}(A)$ supports the following operations:

$$\mathsf{cons} \qquad\qquad : (A : \mathsf{U}) \to \mathsf{El}(A) \to \mathsf{El}(\mathsf{Str}(A)) \to \mathsf{El}(\mathsf{Str}(A)) \, @ \, s$$
$$\mathsf{cons}_A(h, t) \triangleq \mathsf{let}\ \mathsf{mod}_\gamma(t') \leftarrow t\ \mathsf{in}\ \mathsf{mod}_\gamma((\mathsf{mod}_\delta(h), \mathsf{next}(t')))$$

$$\mathsf{head} \qquad\qquad : (A : \mathsf{U}) \to \mathsf{El}(\mathsf{Str}(A)) \to \mathsf{El}(A) \, @ \, s$$
$$\mathsf{head}_A(s) \quad \triangleq \mathsf{let}\ \mathsf{mod}_\gamma(s') \leftarrow s\ \mathsf{in}\ \mathbf{triv}^{-1}(\mathbf{comp}_{\gamma,\delta}(\mathsf{mod}_\gamma(\mathsf{pr}_0(s'))))$$

$$\mathsf{tail} \qquad\qquad : (A : \mathsf{U}) \to \mathsf{El}(\mathsf{Str}(A)) \to \mathsf{El}(\mathsf{Str}(A)) \, @ \, s$$
$$\mathsf{tail}_A(s) \quad \triangleq \mathsf{let}\ \mathsf{mod}_\gamma(s') \leftarrow s\ \mathsf{in}\ \mathbf{comp}_{\gamma,\ell}(\mathsf{mod}_\gamma(\mathsf{pr}_1(s')))$$

---

[9] We denote modalities and their counterparts on the universe by the same notation. For example, we may write ΔA to mean the type $\Gamma \vdash \langle \delta \mid A \rangle\ \mathsf{type}_1 \, @ \, m$ whenever $\Gamma, \blacksquare_\delta \vdash A\ \mathsf{type}_1 \, @ \, m$, but also to mean the term $\Gamma \vdash \mathsf{Code}(\langle \delta \mid \mathsf{El}(A) \rangle) : \mathsf{U} \, @ \, m$ whenever $\Gamma, \blacksquare_\delta \vdash A : \mathsf{U} \, @ \, m$.

Those familiar with prior work on guarded streams may be surprised by the type of tail. The expected definition would be

$$\mathsf{tail}_A(s) \stackrel{?}{=} \mathsf{let}\ \mathsf{mod}_\gamma(s') \leftarrow s\ \mathsf{in}\ \mathsf{mod}_\gamma(\mathsf{pr}_1(s'))$$

This term has type $\mathsf{El}(\mathsf{Str}(A)) \to \Gamma(\blacktriangleright\mathsf{El}(\mathsf{Str}'(A)))$. However, in our case the Γ modality is sufficiently strong to "absorb" this extra $\blacktriangleright$: the equality $\gamma \circ \ell = \gamma$ induces an equivalence $\Gamma \circ \blacktriangleright \simeq \Gamma$, which we use to obtain the version given above. This small difference is crucial: it will internally make $\mathsf{Str}(A)$ into a final coalgebra!

Lemma 9.6. *These operations satisfy the expected β and η laws, i.e.*

(1) $(h : \mathsf{El}(A))(t : \mathsf{El}(\mathsf{Str}(A))) \to \mathsf{Eq}_{\mathsf{El}(A)}(\mathsf{head}_A(\mathsf{cons}_A(h, t)), h) @ s$
(2) $(h : \mathsf{El}(A))(t : \mathsf{El}(\mathsf{Str}(A))) \to \mathsf{Eq}_{\mathsf{El}(\mathsf{Str}(A))}(\mathsf{tail}_A(\mathsf{cons}_A(h, t)), t) @ s$
(3) $(h : \mathsf{El}(A))(t : \mathsf{El}(\mathsf{Str}(A))) \to \mathsf{Eq}_{\mathsf{El}(\mathsf{Str}(A))}(s, \mathsf{cons}_A(\mathsf{head}_A(s), \mathsf{tail}_A(s))) @ s$

Proof. We prove (2), the other two being similar. If $h : \mathsf{El}(A)$ and $t : \mathsf{El}(\mathsf{Str}(A))$, note that $\mathsf{El}(\mathsf{Str}(A))$ is a type of the form $\Gamma(-)$, and calculate that

$$
\begin{aligned}
&\mathsf{tail}_A(\mathsf{cons}_A(h, t)) \\
&= \mathsf{tail}_A(\mathsf{cons}_A(h, \mathsf{mod}_\gamma(t'))) && \text{write } t = \mathsf{mod}_\gamma(t') \text{ by modal induction} \\
&= \mathsf{tail}_A(\mathsf{mod}_\gamma((\mathsf{mod}_\delta(h), \mathsf{next}(t')))) \\
&= \mathbf{comp}_{\gamma,\ell}(\mathsf{mod}_\gamma(\mathsf{pr}_1((\mathsf{mod}_\delta(h), \mathsf{next}(t'))))) \\
&= \mathbf{comp}_{\gamma,\ell}(\mathsf{mod}_\gamma(\mathsf{next}(t'))) \\
&= \mathsf{mod}_\gamma(t') && \text{as } \gamma \circ \ell = \gamma \\
&= t
\end{aligned}
$$
$\square$

Theorem 9.7. $\mathsf{Str}(A)$ *is the final coalgebra for* $\lambda X.\ \mathsf{El}(A) \times X : \mathsf{U} \to \mathsf{U} @ s$.

Proof. Given $A : \mathsf{U}$ we define a coalgebra $\mathsf{uncons} : \mathsf{Str}(A) \to (\mathsf{El}(A) \times \mathsf{Str}(A)) @ s$ by

$$\mathsf{uncons}(s) \triangleq (\mathsf{head}_A(s), \mathsf{tail}_A(s))$$

To show finality, suppose $c : B \to \mathsf{El}(A) \times B @ s$ is another coalgebra. We define a function $f : B \to \mathsf{Str}(A) @ s$ by

$$
\begin{aligned}
f' \quad &: \Delta B \to \mathsf{El}(\mathsf{Str}'(A)) @ t \\
f' \quad &\triangleq \mathsf{l\ddot{o}b}(\lambda f'', x.\ \mathsf{let}\ \mathsf{mod}_\delta(x') \leftarrow x\ \mathsf{in}\ (h, t)) \\
& \quad \text{where } h = \mathsf{mod}_\delta(\mathsf{pr}_0(c(x'))) \\
& \quad \text{and} \quad t = f'' \circledast_\ell \mathsf{next}(\mathsf{mod}_\delta(\mathsf{pr}_1(c(x')))) \\[6pt]
f \quad &: B \to \mathsf{El}(\mathsf{Str}(A)) @ s \\
f(x) \quad &\triangleq \mathsf{mod}_\gamma(f'(\mathsf{mod}_\delta(x)))
\end{aligned}
$$

This is a morphism of coalgebras: for any $x : B$ we have

$$
\begin{aligned}
\mathsf{uncons}(f(x)) &= (\mathsf{head}_A(f(x)), \mathsf{tail}_A(f(x))) \\
&= (\mathsf{pr}_0(c(x)), \mathsf{tail}_A(f(x))) \\
&= (\mathsf{pr}_0(c(x)), \mathbf{comp}_{\gamma,\ell}(\mathsf{mod}_\gamma(\mathsf{pr}_1(f'(x))))) \\
&= (\mathsf{pr}_0(c(x)), \mathbf{comp}_{\gamma,\ell}(\mathsf{mod}_\gamma(\mathsf{next}(f') \circledast_\ell \mathsf{next}(\mathsf{mod}_\delta(\mathsf{pr}_1(c(x))))))) \\
&= (\mathsf{pr}_0(c(x)), f(\mathsf{pr}_1(c(x))))
\end{aligned}
$$

Finally, we must show that f is the unique coalgebra morphism. Suppose we are given $g : B \to \mathsf{El}(\mathsf{Str}(A)) @ s$ which also satisfies $\mathsf{uncons}(g(x)) = (\mathsf{pr}_0(c(x)), g(\mathsf{pr}_1(c(x))))$. We 'shift' this definition to timed sets, by defining

$$\hat{g} \quad : \Delta B \to \mathsf{El}(\mathsf{Str}'(A)) @ t$$
$$\hat{g}(x) \triangleq \mathbf{coe}[\delta \circ \gamma \leq 1](\mathsf{mod}_\delta(g) \circledast_\delta x)$$

It suffices to show that $\hat{g} = f' @ t$, and we do so by Löb induction and function extensionality. Assume $p : \blacktriangleright \mathsf{Eq}(\hat{g}, f')$, and $x : \Delta B$. To prove $\hat{g}(x) = f'(x) : \Delta B \times \blacktriangleright \mathsf{Str}'(A)$ it suffices to show componentwise equality. By modal induction write $x = \mathsf{mod}_\delta(y)$ for $y : B$.

First, we have that $\mathsf{pr}_0(f'(\mathsf{mod}_\delta(y))) = \mathsf{mod}_\delta(\mathsf{pr}_0(c(y)))$ by the definition of f'. On the other hand, we have that

$$\mathsf{pr}_0(\hat{g}(\mathsf{mod}_\delta(y))) = \mathsf{pr}_0(\mathbf{coe}[\delta \circ \gamma \leq 1](\mathsf{mod}_\delta(g(y)))) = \mathsf{pr}_0(g_x) : \Delta B @ t$$

where we have used modal induction to write $g(y) = \mathsf{mod}_\gamma(g_x)$. That g is a coalgebra morphism implies that $\mathsf{head}(g(y)) = \mathsf{pr}_0(c(y))$. If we now use modal induction to write $\mathsf{pr}_0(g_x) = \mathsf{mod}_\delta(b)$ for $b : B$ and unfold the definition of head, we obtain $b = \mathsf{pr}_0(c(y))$, so $\mathsf{pr}_0(g_x) = \mathsf{mod}_\delta(b) = \mathsf{mod}_\delta(\mathsf{pr}_0(c(y)))$, which shows that the two first components are equal.

For the second component, we compute that

$$\mathsf{pr}_1(f'(\mathsf{mod}_\delta(y)))$$
$$= \mathsf{next}(f') \circledast_\ell \mathsf{next}(\mathsf{mod}_\delta(\mathsf{pr}_1(c(y))))$$
$$= \mathsf{next}(f'(\mathsf{mod}_\delta(\mathsf{pr}_1(c(y)))))$$
$$= \mathsf{next}(\hat{g}(\mathsf{mod}_\delta(\mathsf{pr}_1(c(y))))) \qquad\qquad \text{using } p \text{ through Lemma 9.4}$$
$$= \mathsf{next}(\mathbf{coe}[\delta \circ \gamma \leq 1](\mathsf{mod}_\delta(g(\mathsf{pr}_1(c(x))))))$$
$$= \mathsf{next}(\mathbf{coe}[\delta \circ \gamma \leq 1](\mathsf{mod}_\delta(\mathsf{tail}(g(y))))) \qquad \text{as } g \text{ is a coalgebra morphism}$$
$$= \mathsf{pr}_1(\mathbf{coe}[\delta \circ \gamma \leq 1](\mathsf{mod}_\delta(g(y)))) \qquad\qquad\qquad \text{lemma}$$
$$= \mathsf{pr}_1(\hat{g}(x))$$

The lemma referred to above is the fact that for any $s : \mathsf{Str}(A)$ it is the case that

$$\mathsf{next}(\mathbf{coe}[\delta \circ \gamma \leq 1](\mathsf{mod}_\delta(\mathsf{tail}(s)))) = \mathsf{pr}_1(\mathbf{coe}[\delta \circ \gamma \leq 1](\mathsf{mod}_\delta(s)))$$

which can be shown by a series of modal inductions. $\qquad\qquad\qquad\qquad\qquad\square$

We conclude this section by showing how to use these mechanisms in order to prove properties of coinductive programs. Specifically, we will replicate a proof from [BGC$^+$16] which shows that the `zipWith` operator on streams preserves commutativity. Let

$$\mathsf{zipWith}' \quad : \Delta(\mathsf{El}(A) \to \mathsf{El}(B) \to \mathsf{El}(C)) \to \mathsf{El}(\mathsf{Str}'(A)) \to \mathsf{El}(\mathsf{Str}'(B)) \to \mathsf{El}(\mathsf{Str}'(C))$$
$$\mathsf{zipWith}'(f) \triangleq \mathsf{l\ddot{o}b}(\lambda r.\, \lambda x, y.\, (f \circledast_\delta \mathsf{pr}_0(x) \circledast_\delta \mathsf{pr}_0(y), r \circledast_\ell \mathsf{pr}_1(x) \circledast_\ell \mathsf{pr}_1(y)))$$

$$\mathsf{zipWith} \quad : (\mathsf{El}(A) \to \mathsf{El}(B) \to \mathsf{El}(C)) \to \mathsf{El}(\mathsf{Str}(A)) \to \mathsf{El}(\mathsf{Str}(B)) \to \mathsf{El}(\mathsf{Str}(C))$$
$$\mathsf{zipWith}(f) \triangleq \lambda x, y.\, \mathsf{mod}_\gamma(\mathsf{zipWith}'(\mathsf{mod}_\delta(f))) \circledast_\gamma x \circledast_\gamma y$$

Remark 9.8. Take note of a useful pattern for programming with guarded recursion, which is visible both here and in the proof of Theorem 9.7. We first define an auxiliary function in mode t, which uses Löb induction. The main function itself is then just a thin wrapper which 'corrects' that with the appropriate modalities and modal combinators.

Theorem 9.9. *If f is commutative then* $\mathsf{zipWith}(f)$ *is commutative. That is, given $A, B : \mathsf{U}$ and $f : \mathsf{El}(A) \to \mathsf{El}(A) \to \mathsf{El}(B)$ there is a term of the following type:*

$$((x, y : \mathsf{El}(A)) \to \mathsf{Eq}_{\mathsf{El}(B)}(f(x, y), f(y, x))) \to$$
$$(s, t : \mathsf{El}(\mathsf{Str}(A))) \to \mathsf{Eq}_{\mathsf{El}(\mathsf{Str}(B))}(\mathsf{zipWith}(f, s, t), \mathsf{zipWith}(f, t, s))$$

Proof. Suppose $e : (x, y : \mathsf{El}(A)) \to \mathsf{Eq}_{\mathsf{El}(B)}(f(x, y), f(y, x))$ and $s, t : \mathsf{El}(\mathsf{Str}(A))$. We wish to show that $\mathsf{zipWith}(f, s, t) = \mathsf{zipWith}(f, t, s)$. By the definition of $\mathsf{zipWith}$, it is sufficient to prove that for any $u, v : \mathsf{El}(\mathsf{Str}'(A))$ we have

$$\mathsf{zipWith}'(\mathsf{mod}_\delta(f), u, v) = \mathsf{zipWith}'(\mathsf{mod}_\delta(f), v, u)$$

In turn, it suffices to show that

$$\mathrm{l\ddot{o}b}(F_0) = \mathrm{l\ddot{o}b}(F_1)$$

where

$$F_0 \triangleq \lambda r. \, \lambda x, y. \, (\mathsf{mod}_\delta(f) \circledast_\delta \mathsf{pr}_0(x) \circledast_\delta \mathsf{pr}_0(y), r \circledast_\ell \mathsf{pr}_1({\color{red}x}) \circledast_\ell \mathsf{pr}_1({\color{red}y}))$$
$$F_1 \triangleq \lambda r. \, \lambda x, y. \, (\mathsf{mod}_\delta(f) \circledast_\delta \mathsf{pr}_0(y) \circledast_\delta \mathsf{pr}_0(x), r \circledast_\ell \mathsf{pr}_1({\color{red}y}) \circledast_\ell \mathsf{pr}_1({\color{red}x}))$$

because then

$$\mathsf{zipWith}'(\mathsf{mod}_\delta(f), v, u) \triangleq \mathrm{l\ddot{o}b}(F_0)(u, v) = \mathrm{l\ddot{o}b}(F_1)(u, v) = \mathsf{zipWith}'(\mathsf{mod}_\delta(f), u, v)$$

By Theorem 9.5 we know guarded fixed points are unique, so it suffices to show that

$$\mathrm{l\ddot{o}b}(F_1) = F_0(\mathsf{next}(\mathrm{l\ddot{o}b}(F_1))) \tag{9.5}$$

We use Löb induction to construct a term of type $\mathsf{Eq}(\mathrm{l\ddot{o}b}(F_1), F_0(\mathsf{next}(\mathrm{l\ddot{o}b}(F_1))))$.

$$F_0(\mathsf{next}(\mathrm{l\ddot{o}b}(F_1)))$$
$$= \lambda x, y. \, (\mathsf{mod}_\delta(f) \circledast_\delta \mathsf{pr}_0(x) \circledast_\delta \mathsf{pr}_0(y), \mathsf{next}(\mathrm{l\ddot{o}b}(F_1)) \circledast_\ell \mathsf{pr}_1(x) \circledast_\ell \mathsf{pr}_1(y))$$

by induction let $\mathsf{mod}_\delta(a) \triangleq \mathsf{pr}_0(x)$ and $\mathsf{mod}_\delta(b) \triangleq \mathsf{pr}_0(y)$

$$= \lambda x, y. \, (\mathsf{mod}_\delta(f(a, b)), \mathsf{next}(\mathrm{l\ddot{o}b}(F_1)) \circledast_\ell \mathsf{pr}_1(x) \circledast_\ell \mathsf{pr}_1(y))$$
$$= \lambda x, y. \, (\mathsf{mod}_\delta(f(b, a)), \mathsf{next}(\mathrm{l\ddot{o}b}(F_1)) \circledast_\ell \mathsf{pr}_1(x) \circledast_\ell \mathsf{pr}_1(y))$$
$$= \lambda x, y. \, (\mathsf{mod}_\delta(f(b, a)), \mathsf{next}(F_1(\mathsf{next}(\mathrm{l\ddot{o}b}(F_1)))) \circledast_\ell \mathsf{pr}_1(x) \circledast_\ell \mathsf{pr}_1(y))$$

by induction let $\mathsf{mod}_\ell(s) \triangleq \mathsf{pr}_1(x)$ and $\mathsf{mod}_\ell(t) \triangleq \mathsf{pr}_1(y)$

$$= \lambda x, y. \, (\mathsf{mod}_\delta(f(b, a)), \mathsf{next}(F_1(\mathsf{next}(\mathrm{l\ddot{o}b}(F_1)))(s, t)))$$
$$= \lambda x, y. \, (\mathsf{mod}_\delta(f(b, a)), \mathsf{next}(F_0(\mathsf{next}(\mathrm{l\ddot{o}b}(F_1)))(t, s)))$$
$$= \lambda x, y. \, (\mathsf{mod}_\delta(f(b, a)), \mathsf{next}(F_0(\mathsf{next}(\mathrm{l\ddot{o}b}(F_1)))) \circledast_\ell \mathsf{pr}_1(y) \circledast_\ell \mathsf{pr}_1(x))$$

using the IH through Lemma 9.4

$$= \lambda x, y. \, (\mathsf{mod}_\delta(f(b, a)), \mathsf{next}(\mathrm{l\ddot{o}b}(F_1)) \circledast_\ell \mathsf{pr}_1(y) \circledast_\ell \mathsf{pr}_1(x))$$
$$= \lambda x, y. \, (\mathsf{mod}_\delta(f) \circledast_\delta \mathsf{pr}_0(y) \circledast_\delta \mathsf{pr}_0(x), \mathsf{next}(\mathrm{l\ddot{o}b}(F_1)) \circledast_\ell \mathsf{pr}_1(y) \circledast_\ell \mathsf{pr}_1(x))$$
$$= \mathrm{l\ddot{o}b}(F_1) \qquad \qquad \square$$

Remark 9.10 (Previous approaches)**.** Using dependent type theories to reason about guarded recursion and coinductive types has been a problem for some time [Møg14]. The technical device of *clocks*, due to [AM13], was introduced to deal with productivity in a simply-typed setting. Clocks were then introduced to dependent types [Møg14], and later refined into the extensional guarded type theory **gDTT** of [BGC$^+$16].

In **gDTT** the problem of 'totalising' a type—which corresponds to reasoning by coinduction—was not handled through the 'always' modality, but through clocks. In essence, **gDTT** does not come with a single $\blacktriangleright$ modality, but rather with a collection of them, each one indexed by a clock name. There is a quantifier which allows clock names to be bound inside a particular type, and a crucial isomorphism:

$$\forall \kappa.\ A \cong \forall \kappa.\ \blacktriangleright^\kappa A \tag{$*$}$$

gDTT presents several technical complications. The syntactic problems pertaining to delayed substitutions were resolved by the introduction of Clocked Type Theory (CloTT) [BGM17], which uses additional judgmental structure. It is conjectured that type-checking is decidable for CloTT. The complexity of using clocks also appears in the semantics of clocked type theory. CloTT is modelled in a collection of presheaf categories, with multiple functors navigating between them [MM18].

It was hoped that some of the complexity could be circumvented by replacing clocks with a modality. This led Clouston et al. to introduce the comonadic 'always' modality $\Box$, which replaced the isomorphism $(*)$ with $\Box \blacktriangleright A \cong \Box A$ [CBGB15]. The main advantage of using $\Box$ is that it can be interpreted in $\mathbf{PSh}(\omega)$, which is a much simpler model. On the other hand, the interactions between $\Box$ and $\blacktriangleright$ have proven difficult to capture in the syntax. In fact, the mere addition of $\Box$ to a dependent type theory poses a significant technical challenge: see [BGM17, BCM$^+$20, Shu18, GSB19a]. Despite this concentrated effort, there are still serious technical obstacles to adding $\blacktriangleright$ to a type theory for $\Box$. MTT is the first syntax to accomodate both $\Box, \blacktriangleright$, and validate $\Box \blacktriangleright A \cong \Box A$.

10. Internal Adjoints

In many cases of interest, the need for a pair of *adjoint modalities* arises: we would like a pair of modalities $\mu : n \to m$ and $\nu : m \to n$ so that, in some sense,

$$\langle \nu \mid - \rangle \dashv \langle \mu \mid - \rangle$$

But what does it mean to have an adjunction between two modalities *within* MTT? Does it correspond to an external adjunction? And do all known results from category theory apply? The only thing that is certain is that this scenario is fundamental to modal type theory, as a number of intended models can be elegantly presented through adjunctions [SS12, ND18, Shu18].

In this section we show that when MTT is equipped with the *walking adjunction* as a mode theory, it becomes a useful syntax for reasoning about adjoint modalities. Of course, the adjoint modalities themselves are not exactly adjoint functors: they are something slightly weaker than DRAs, whose 'left adjoints' constitute an adjunction. Nevertheless, we prove that the induced modalities largely behave as expected: the unit and counit are internally definable; some limited forms of internal transposition can be recovered; and left adjoints preserve colimits, as expressed through *crisp induction principles*.

10.1. **The walking adjunction.** As ever, we begin by freely defining a mode theory $\mathcal{M}_{\mathsf{adj}}$. Its generators are two 1-cells $\nu : m \to n$ and $\mu : n \to m$, and two 2-cells

$$\eta : 1_m \Rightarrow \mu \circ \nu \qquad\qquad \epsilon : \nu \circ \mu \Rightarrow 1_n$$

subject to the triangle equations

$\mathcal{M}_{\mathsf{adj}}$ is sometimes called the *walking adjunction* [LS16, §5.1]. It is the *classifying 2-category* for an adjunction: 2-functors $\mathcal{M}_{\mathsf{adj}} \longrightarrow \mathcal{C}$ correspond precisely to (2-categorical) adjunctions in $\mathcal{C}$. The mode theory $\mathcal{M}_{\mathsf{adj}}$ has a very curious property: it is *self-dual*, i.e. there is an equivalence $\mathcal{M}_{\mathsf{adj}}^{\mathsf{coop}} \simeq \mathcal{M}_{\mathsf{adj}}$. This equivalence sends the modes to each other, the adjoints to themselves and the 2-cells η and ϵ again to each other.

10.2. **Models of adjoint modalities.** Recall that a modal context structure of a model of MTT with mode theory $\mathcal{M}_{\mathsf{adj}}$ is a strict 2-functor $[\![-]\!] : \mathcal{M}_{\mathsf{adj}}^{\mathsf{coop}} \to \mathbf{Cat}$. The self-duality of $\mathcal{M}_{\mathsf{adj}}$ implies that such a context structure consists of two categories and an adjunction between them. We immediately obtain the following result.

Corollary 10.1. *If $\mathcal{C}$ and $\mathcal{D}$ carry models of MLTT, and there is a pair of dependent right adjoints between them whose 'left adjoints' are themselves adjoint, then we can construct a model of MTT with mode theory $\mathcal{M}_{\mathsf{adj}}$.*

Proof. Write $[\![\blacksquare_\nu]\!] : \mathcal{C} \to \mathcal{D}$ and $[\![\blacksquare_\mu]\!] : \mathcal{D} \to \mathcal{C}$ for the functors given as part of the DRAs. The notation is then suggestive: $[\![\blacksquare_\nu]\!] \dashv [\![\blacksquare_\mu]\!]$, and Theorem 7.1 applies. $\square$

Conversely,

Theorem 10.2. *Any model of $\mathcal{M}_{\mathsf{adj}}$ must interpret $[\![\blacksquare_\nu]\!]$ and $[\![\blacksquare_\mu]\!]$ as adjoint functors. Moreover, if $\mathbf{Mod}_\mu$ and $\mathbf{Mod}_\nu$ are induced by lifting the adjunctions $[\![\blacksquare_\mu]\!] \dashv R_\mu$ and $[\![\blacksquare_\nu]\!] \dashv R_\nu$ to a dependent right adjoints (by Lemma 7.4), then $R_\nu \dashv R_\mu$.*

Proof. Adjoint functors are precisely adjoint morphisms in the 2-category $\mathbf{Cat}$. As $\mathcal{M}_{\mathsf{adj}}$ is the walking adjunction, and 2-functors preserve adjunctions, we have that $[\![\blacksquare_\nu]\!] \dashv [\![\blacksquare_\mu]\!]$.

If $[\![\blacksquare_\nu]\!] \dashv R_\nu$, then by the uniqueness of adjoint pairs we must have that $R_\nu \cong [\![\blacksquare_\mu]\!]$. If moreover $[\![\blacksquare_\mu]\!] \dashv R_\mu$, then the previous isomorphism yields $R_\nu \dashv R_\mu$. $\square$

The last situation in this lemma is sometimes known as an 'adjunction of adjunctions' [LS16, §5.1]. In particular, the action of the right adjoint modality μ on contexts, viz. $[\![\blacksquare_\mu]\!]$, is in some sense internalized on types and terms by the action of the left adjoint modality ν on types and terms, viz. $\langle \nu \mid - \rangle$.

10.3. **Recovering the adjunction internally.** The foregoing construction of a model interpreted the lock functors required by $\mathcal{M}_{\mathsf{adj}}$ by an adjunction. Consequently, substitutions $\Delta \to \Gamma.\blacksquare_\mu$ are in natural bijection with substitutions $\Delta.\blacksquare_\nu \to \Gamma$. We would like to strengthen this setting by bootstrapping this adjunction into an *internal adjunction.*

It is not immediately clear what an internal adjunction should be. However, we can construct an appropriate definition by internalizing the unit and counit as functions. But that is not immediate either: if $\Gamma \vdash A\ \mathsf{type}_1 @ m$, the construction $\langle \mu \mid \langle \nu \mid A \rangle \rangle$ that we would naïvely try as the codomain of the unit is ill-typed. This can be mended through key substitutions. Recall that $\eta : 1_m \Rightarrow \mu \circ \nu$. The corresponding key substitution at $\Gamma @ m$ is $\mathbf{key}_\Gamma^\eta : \Gamma, \blacksquare_{\mu \circ \nu} \to \Gamma @ m$. We can use this to formally define the notation of Section 2.3 by

$$A^\eta \triangleq A[\mathbf{key}_\Gamma^\eta]$$

As substitutions can be eliminated (e.g. through a subset of the canonicity algorithm), this defines an admissible operation from type $\Gamma \vdash A\ \mathsf{type}_1 @ m$ to type $\Gamma, \blacksquare_{\mu \circ \nu} \vdash A^\eta\ \mathsf{type}_1 @ m$. We can thus define the unit component at $\Gamma \vdash A\ \mathsf{type}_1 @ m$ by

$$
\begin{aligned}
\mathsf{unit} \quad &: A \to \langle \mu \mid \langle \nu \mid A^\eta \rangle \rangle @ m \\
\mathsf{unit}(x) &\triangleq \mathsf{mod}_\mu(\mathsf{mod}_\nu(x^\eta))
\end{aligned}
$$

Dually, for any type $\Gamma, \blacksquare_{\nu \circ \mu} \vdash A\ \mathsf{type}_1 @ n$ we can define the counit component by

$$
\begin{aligned}
\mathsf{counit} \quad &: \langle \nu \mid \langle \mu \mid A \rangle \rangle \to A^\epsilon @ n \\
\mathsf{counit}(x) &\triangleq \mathsf{let}\ \mathsf{mod}_\nu(y_0) \leftarrow x\ \mathsf{in}\ \mathsf{let}_\nu\ \mathsf{mod}_\mu(y_1) \leftarrow y_0\ \mathsf{in}\ y_1^\epsilon
\end{aligned}
$$

We thus obtain the unit and counit internally, but the types of the components have to be adjusted in the presence of dependence. Moreover, we can prove internal versions of the triangle equations; they are given by modal induction:

$$
\begin{aligned}
\_ \quad &: (x : \langle \nu \mid A \rangle) \to \mathsf{Id}_{\langle \nu | A \rangle}(x, \mathsf{counit}(\mathsf{mod}_\nu(\mathsf{unit}) \circledast_\nu x)) \\
\_ &\triangleq \lambda x.\ \mathsf{let}\ \mathsf{mod}_\nu(y) \leftarrow x\ \mathsf{in}\ \mathsf{refl}(\mathsf{mod}_\nu(y))
\end{aligned}
$$

$$
\begin{aligned}
\_ \quad &: (x : \langle \mu \mid A \rangle) \to \mathsf{Id}_{\langle \mu | A \rangle}(x, \mathsf{mod}_\mu(\mathsf{counit}) \circledast_\mu \mathsf{unit}(x)) \\
\_ &\triangleq \lambda x.\ \mathsf{let}\ \mathsf{mod}_\mu(y) \leftarrow x\ \mathsf{in}\ \mathsf{refl}(\mathsf{mod}_\mu(y))
\end{aligned}
$$

The most difficult part is proving that these terms are well-typed. For example, in the first instance we must show that $\mathsf{mod}_\nu(y) = \mathsf{counit}(\mathsf{mod}_\nu(\mathsf{unit}) \circledast_\nu \mathsf{mod}_\nu(y))$ definitionally:

$$
\begin{aligned}
\mathsf{counit}(\mathsf{mod}_\nu(\mathsf{unit}) \circledast_\nu \mathsf{mod}_\nu(y)) &= \mathsf{counit}(\mathsf{mod}_\nu(\mathsf{unit}(y))) \\
&= \mathsf{counit}(\mathsf{mod}_\nu(\mathsf{mod}_\mu(\mathsf{mod}_\nu(y^\eta)))) \\
&= \mathsf{mod}_\nu(y^\eta)^\epsilon \\
&= \mathsf{mod}_\nu((y^\eta)^{\epsilon \star 1_\nu}) \\
&= \mathsf{mod}_\nu(y^{(\epsilon \star 1_\nu) \circ (1_\nu \star \eta)}) \\
&= \mathsf{mod}_\nu(y)
\end{aligned}
$$

Because we are using slightly informal syntax here, it is difficult to see that the steps that introduce whiskering are correct. They become much more perspicuous if we expand $\mathsf{mod}_\nu(y^\eta)^\epsilon$ into algebraic syntax, and use the last equation of Fig. 9 twice to absorb locks:

$$\mathsf{mod}_\nu(y[\mathbf{key}_{\Gamma.\blacksquare_\nu}^\eta])[\mathbf{key}_\Gamma^\epsilon] = \mathsf{mod}_\nu(y[\mathbf{key}_{\Gamma.\blacksquare_\nu}^\eta \circ \mathbf{key}_\Gamma^\epsilon.\blacksquare_\nu]) = \mathsf{mod}_\nu(y[\mathbf{key}_\Gamma^{1_\nu \star \eta} \circ \mathbf{key}_\Gamma^{\epsilon \star 1_\nu}]) = \mathsf{mod}_\nu(y[\mathbf{key}_\Gamma^{1_\mu}])$$

10.4. **Internal transposition.** The previous section offers a perfectly good internal representation of an external adjunction. However, it is usually much more economical to present an adjunction by a natural isomorphism

$$\mathrm{Hom}(L(A), B) \cong \mathrm{Hom}(A, R(B))$$

Unfortunately, this is not achievable in MTT for a multitude of reasons. First, notice that $\langle \nu \mid A \rangle \to B \,@\, n$ and $A \to \langle \mu \mid B \rangle \,@\, m$ are types in different modes, so the putative type $(\langle \nu \mid A \rangle \to B) \simeq (A \to \langle \mu \mid B \rangle)$ that would represent the isomorphism is ill-typed.

Second, even if the two modes coincide—so that ν, μ are endomodalities—the aforementioned type is a bit too strong for our purpose: it is inhabited by *internal equivalences*, which are stronger than bijections of hom-sets. Such equivalences correspond to isomorphisms $B^{L(A)} \cong R(B)^A$ of exponential objects. In turn, these are equivalent to hom-set bijections only if the involved functors are *internal*, which is to say that we have functions

$$(A \to B) \to (\langle \nu \mid A \rangle \to \langle \nu \mid B \rangle) \qquad\qquad (A \to B) \to (\langle \mu \mid A \rangle \to \langle \mu \mid B \rangle)$$

that compute the action of the modality $\langle \mu \mid - \rangle$ on morphisms *within* MTT.[10]

Third, even if we could internalize our modalities, we would be flying too close to the sun. As we have $(1 \to A) \simeq A$ for any A and $\langle \xi \mid 1 \rangle \simeq 1$ for any ξ, we may calculate that

$$A \;\simeq\; 1 \to A \;\simeq\; \langle \nu \mid 1 \rangle \to A \;\simeq\; 1 \to \langle \mu \mid A \rangle \;\simeq\; \langle \mu \mid A \rangle$$

Hence, $\langle \mu \mid - \rangle$ must be the identity functor up to equivalence. This short argument, which is due to [LOPS18, Theorem 5.1], is a *no-go theorem* that obstructs the internalization of an adjunction whose left adjoint preserves terminal objects.

[LOPS18] overcame this barrier by introducing the *global sections modality* $\flat$. Terms of $\flat A$ represent *global* elements of A: terms of $\flat(A \to B)$ are in bijection with morphisms in $\mathrm{Hom}(A, B)$. Thus, the previously problematic equivalence holds under $\flat$.

We can rephrase this argument in our syntax. The key thing to notice is that the functor $\flat : \mathbf{PSh}(\mathcal{C}) \to \mathbf{PSh}(\mathcal{C})$ which maps a presheaf to the constant presheaf $\_ \mapsto \mathrm{Hom}_{\mathbf{PSh}(\mathcal{C})}(1, P)$ of global sections is initial amongst functors that preserve the terminal object. Thus, we postulate an initial modality: suppose that $n = m$, and that $\mathrm{Hom}(m, m)$ is equipped with an initial object, i.e. a 1-cell $\flat : m \to m$ and a unique 2-cell $! : \flat \Rightarrow \xi$ for all ξ. As a consequence, we are able to use variables $x : (\flat \mid A)$ in any context. Assuming function extensionality, we have that

Theorem 10.3. *There is an equivalence* $\langle \flat \mid \langle \nu \mid A^! \rangle \to B \rangle \simeq \langle \flat \mid A \to \langle \mu \mid B^! \rangle \rangle$.

Proof. The equivalence is given by the functions

$$
\begin{aligned}
F &\quad : \langle \flat \mid \langle \nu \mid A^! \rangle \to B \rangle \to \langle \flat \mid A \to \langle \mu \mid B^! \rangle \rangle \\
F(f) &\triangleq \mathsf{let}\ \mathsf{mod}_\flat(g) \leftarrow f\ \mathsf{in}\ \mathsf{mod}_\flat(\lambda x.\ \mathsf{mod}_\mu(g^!) \circledast_\mu \mathsf{unit}(x))
\end{aligned}
$$

$$
\begin{aligned}
G &\quad : \langle \flat \mid A \to \langle \mu \mid B^! \rangle \rangle \to \langle \nu \mid A \rangle \to \langle \flat \mid \langle \nu \mid A^! \rangle \to B \rangle \\
G(g) &\triangleq \mathsf{let}\ \mathsf{mod}_\flat(f) \leftarrow g\ \mathsf{in}\ \mathsf{mod}_\flat(\lambda x.\ \mathsf{counit}(\mathsf{mod}_\nu(f^!) \circledast_\nu x))
\end{aligned}
$$

These are well-typed because, by initiality of $\flat$, $A^\eta = (A^!)^\eta = A^!$, $(B^!)^\epsilon = B^!$. By function extensionality and η for modalities, they are mutually inverse. $\square$

---

[10]Such functors are usually called *enriched* (recall that cartesian closure is a self-enrichment).

The closest we can get to defining internal transposition (without using an initial modality) amounts to the following two functions.

$$\mathbf{transp}^{\rightarrow}_{\nu \dashv \mu} \ : \ \langle \mu \mid \langle \nu \mid A^{\eta} \rangle \rightarrow B \rangle \rightarrow A \rightarrow \langle \mu \mid B \rangle$$
$$\mathbf{transp}^{\rightarrow}_{\nu \dashv \mu} \triangleq \lambda f. \ \lambda x. \ f \circledast_{\mu} \mathsf{unit}(x)$$

$$\mathbf{transp}^{\leftarrow}_{\nu \dashv \mu} \ : \ \langle \nu \mid A \rightarrow \langle \mu \mid B \rangle \rangle \rightarrow \langle \nu \mid A \rangle \rightarrow B^{\epsilon}$$
$$\mathbf{transp}^{\leftarrow}_{\nu \dashv \mu} \triangleq \lambda f. \ \lambda x. \ \mathsf{counit}(f \circledast_{\nu} x)$$

The first is an equivalence (again up to function extensionality), but neither have the expected type. The first transposition $\mathbf{transp}^{\rightarrow}_{\nu \dashv \mu}$ is not without precedent: it is the internal formulation of transposition for adjunctions between monoidal closed categories when the left adjoint preserves monoidal products.

10.5. **Crisp induction.** Having internalized the definition of an adjunction, it is natural to ask whether standard facts about adjoint functors carry over. In this section we prove an internal version of the fact that left adjoints preserve colimits. Within type theory this result takes the form of *crisp induction principles* for various types that arise from colimits.

As a first approximation to the notion of crisp induction, recall the rule for modal induction, i.e. the elimination rule for modal types from Section 2:

$$\frac{\mu : n \rightarrow m \qquad \nu : m \rightarrow o \qquad \Gamma, x : (\nu \mid \langle \mu \mid A \rangle) \vdash B \ \mathsf{type}_1 @ o}{\Gamma, \blacksquare_{\nu} \vdash M_0 : \langle \mu \mid A \rangle @ m \qquad \Gamma, x : (\nu \circ \mu \mid A) \vdash M_1 : B[\mathsf{mod}_{\mu}(x)/x] @ o}{\Gamma \vdash \mathsf{let}_{\nu} \ \mathsf{mod}_{\mu}(x) \leftarrow M_0 \ \mathsf{in} \ M_1 : B[M_0/x] @ o}$$

Notice that there is an "extra" modality parameterizing this rule, ν, which modifies M_0 as well as the data supplied to M_1. This extra generality is not frivolous: we can only define the equivalence $\mathbf{comp}_{\mu,\nu}$ of Section 3 because we can eliminate one modality 'under' another.

One might hope for a similar level of flexibility in all positive eliminators. However, the elimination rule for booleans—stated here in its algebraic form of Section 4—does not allow it:

$$\frac{\Gamma \ \mathsf{ctx} @ m \qquad \Gamma.(1 \mid \mathbb{B}) \vdash A \ \mathsf{type}_1 @ m}{\Gamma \vdash M_t : A[\mathsf{id}.\mathsf{tt}] @ m \qquad \Gamma \vdash M_f : A[\mathsf{id}.\mathsf{ff}] @ m \qquad \Gamma.\blacksquare_1 \vdash N : \mathbb{B} @ m}{\Gamma \vdash \mathsf{if}(A; M_t; M_f; N) : A[\mathsf{id}.N] @ m}$$

Were we to replace 1 with an arbitrary modality, then this rule would state something considerably stronger: not only would we have the expected elimination principle for $\mathbb{B}$, but all of our modalities would *preserve* $\mathbb{B}$. Semantically, this is nonsense: modalities intuitively correspond to right adjoints, and therefore do not necessarily preserve colimits. For example, the later $\blacktriangleright$ modality of Section 9 does not preserve booleans.

Yet, in some circumstances—e.g. when a modality is a left adjoint—the stronger rule is valid. This is the idea behind Shulman's crisp induction principles [Shu18, §5]: cohesive type theory enables the proof of elimination principles for the coproducts and the identity type under the left adjoint in the adjunction $\flat \dashv \sharp$. We will demonstrate that similar principles are derivable within MTT with mode theory $\mathcal{M}_{\mathsf{adj}}$.

Fix a motive $\Gamma, \blacksquare_{\nu \circ \mu}, b : (\nu \mid \mathbb{B}) \vdash C \ \mathsf{type}_1 @ n$. Crisp induction is given by a term

$$\Gamma \vdash \mathsf{crisp\_if}_C : (b : (\nu \mid \mathbb{B})) \rightarrow \langle \nu \circ \mu \mid C(\mathsf{tt}) \rangle \rightarrow \langle \nu \circ \mu \mid C(\mathsf{ff}) \rangle \rightarrow C^{\epsilon}(b) @ n$$

This is a well-formed type, as $\Gamma, b : (\nu \mid \mathbb{B}) = \Gamma, \blacksquare_1, b : (\nu \mid \mathbb{B}) \vdash C^{\epsilon} \ \mathsf{type}_1 @ m$.

To obtain the crisp induction principle we first use the ordinary one at mode n, and apply a number of modal combinators to bring it to mode m.

$$\Gamma, \blacksquare_\nu \vdash h \qquad\qquad : (b : \mathbb{B}) \to \langle \mu \mid C(\mathsf{tt}) \rangle \to \langle \mu \mid C(\mathsf{ff}) \rangle \to \langle \mu \mid C(b^\eta) \rangle @ m$$
$$h(b, t, f) \qquad\qquad \triangleq \mathsf{if}(b.\ \langle \mu \mid C(b^\eta) \rangle; t; f; b)$$

$$\Gamma \vdash \mathsf{crisp\_if}_C \qquad : (b : (\nu \mid \mathbb{B})) \to \langle \nu \circ \mu \mid C(\mathsf{tt}) \rangle \to \langle \nu \circ \mu \mid C(\mathsf{ff}) \rangle \to C^\epsilon(b) @ m$$
$$\mathsf{crisp\_if}_C(b, t, f) \triangleq \mathsf{counit}(\mathsf{mod}_\nu(h(b)) \circledast_\nu \mathbf{comp}_{\nu,\mu}^{-1}(t) \circledast_\nu \mathbf{comp}_{\nu,\mu}^{-1}(f))$$

The reasons why this term is well-typed is subtle. We have that $\Gamma, \blacksquare_\nu, b : (1 \mid \mathbb{B}), \blacksquare_{\mu \circ \nu} \vdash b^\eta : \mathbb{B} @ m$, so $\Gamma, \blacksquare_\nu, b : (1 \mid \mathbb{B}), \blacksquare_\mu \vdash C(b^\eta) \ \mathsf{type}_1 @ n$ by the application rule. Thus, h is well-typed. It remains to show that $C(b^\eta)^\epsilon = C^\epsilon(b)$, which intuitively follows from the triangle identities. We may show it by precisely specifying what these operations mean in the algebraic syntax. First, we construct the substitutions

$$\sigma_0 \triangleq \uparrow.\blacksquare_\mu.\mathbf{v}_0[\mathbf{key}_{\Gamma.\blacksquare_\nu.(1 \mid \mathbb{B})}^\eta] \quad : \Gamma.\blacksquare_\nu.(1 \mid \mathbb{B}).\blacksquare_\mu \to \Gamma.\blacksquare_{\nu \circ \mu}.(\nu \mid \mathbb{B}) @ n$$

$$\sigma_1 \triangleq \uparrow.\blacksquare_\nu.\mathbf{v}_0.\blacksquare_\mu \qquad\qquad : \Gamma.(\nu \mid \mathbb{B}).\blacksquare_{\nu \circ \mu} \to \Gamma.\blacksquare_\nu.(1 \mid \mathbb{B}).\blacksquare_\mu @ n$$

$$\sigma_2 \triangleq (\mathbf{key}_\Gamma^\epsilon \circ \uparrow).\mathbf{v}_0 \qquad : \Gamma.(\nu \mid \mathbb{B}) \to \Gamma.\blacksquare_{\nu \circ \mu}.(\nu \mid \mathbb{B}) @ n$$

We can then interpret $C(b^\eta)$ as the type $\Gamma.\blacksquare_\nu.(1 \mid \mathbb{B}).\blacksquare_\mu \vdash C[\sigma_0] \ \mathsf{type}_1 @ n$. Similarly, $C(b^\eta)^\epsilon$ is the type $\Gamma.(\nu \mid \mathbb{B}) \vdash C[\sigma_0][\sigma_1][\mathbf{key}_{\Gamma.(\nu \mid \mathbb{B})}^\epsilon] \ \mathsf{type}_1 @ n$. Finally, $C^\epsilon(b)$ is the type $\Gamma.(\nu \mid \mathbb{B}) \vdash C[\sigma_2] \ \mathsf{type}_1 @ n$, so it suffices to show that $\sigma_0 \circ \sigma_1 \circ \mathbf{key}_{\Gamma.(\nu \mid \mathbb{B})}^\epsilon = \sigma_2$. This is a monstrous equation which is primarily structural. Moreover, η occurs in σ_0, and ϵ in the key that follows it, so one of the triangle equations must somehow be implicated. Indeed, we can use one of the two equations along with the rules of Section 4 to prove the desired result.

We can now prove that

Theorem 10.4. $\langle \nu \mid \mathbb{B} \rangle \simeq \mathbb{B}$

Proof. We define the two functions

$$\mathsf{b} \qquad : \langle \nu \mid \mathbb{B} \rangle \to \mathbb{B} @ m$$
$$\mathsf{b}(x) \triangleq \mathsf{let}\ \mathsf{mod}_\nu(y) \leftarrow x\ \mathsf{in}\ \mathsf{crisp\_if}_\mathbb{B}(y, \mathsf{mod}_{\nu \circ \mu}(\mathsf{tt}), \mathsf{mod}_{\nu \circ \mu}(\mathsf{ff}))$$

$$\mathsf{b}^{-1} \quad : \mathbb{B} \to \langle \nu \mid \mathbb{B} \rangle @ m$$
$$\mathsf{b}^{-1} \quad \triangleq \lambda x.\ \mathsf{if}(\_.\ \langle \nu \mid \mathbb{B} \rangle; \mathsf{mod}_\nu(\mathsf{tt}); \mathsf{mod}_\nu(\mathsf{ff}); x)$$

We now use full crisp induction to construct for every $x : \langle \nu \mid \mathbb{B} \rangle$ a proof of $\mathsf{Id}_{\langle \nu \mid \mathbb{B} \rangle}(x, \mathsf{b}^{-1}(\mathsf{b}(x)))$. First, use modal induction to write $x = \mathsf{mod}_\nu(y)$ for some $y : (\nu \mid \mathbb{B})$. We then have to prove that $\mathsf{Id}_{\langle \nu \mid \mathbb{B} \rangle}(\mathsf{mod}_\nu(y), \mathsf{b}^{-1}(\mathsf{b}(\mathsf{mod}_\nu(y))))$, so we perform crisp induction on y. If $y \triangleq \mathsf{tt}$, we have that $\mathsf{b}^{-1}(\mathsf{b}(\mathsf{mod}_\nu(\mathsf{tt}))) = \mathsf{mod}_\nu(\mathsf{tt})$, so $\mathsf{mod}_{\nu \circ \mu}(\mathsf{refl}(\mathsf{mod}_\nu(\mathsf{tt})))$ has the right type. The case for $y \triangleq \mathsf{ff}$ is similar. The other direction is simpler, and follows by induction on $\mathbb{B}$. $\square$

Similar results hold for other types with 'positive,' 'pattern-matching,' or 'closed-scope' elimination rules. For example, we can also formulate a crisp induction principle for identity types, which can be used to prove that

Theorem 10.5. $\langle \nu \mid \mathsf{Id}_A(M_0, M_1) \rangle \simeq \mathsf{Id}_{\langle \nu \mid A \rangle}(\mathsf{mod}_\nu(M_0), \mathsf{mod}_\nu(M_1))$

11. Related Work

Modal type theory has been an active area of research for two decades and, as with any active field, a precise taxonomy of modal type theories would be a paper in and of itself. Accordingly, we have not attempted such a task here, and have instead focussed on separating modal type theories into distinct strands based on their judgmental structure. Some of our characterizations are slightly artificial, in that these lines of work are not nearly so separate as we seem to suggest. We feel, however, that this is the simplest way to position MTT in relation to current work.

11.1. **Dual-context modal calculi.** One of the first papers on (non-linear)[11] modal type theory was by [PD01], who constructed a proof theory for S4, i.e. a comonadic modality. The central idea of this approach was to reflect the distinction between modal and non-modal assumptions (referred to as 'truth vs. validity' in *op. cit.*) in the judgmental structure of the system itself. The judgments for this calculus then contained not just a context of true propositions, but rather two contexts: one for intuitionistic propositions, and one for modal ones. Following this methodology, Davies and Pfenning internalized previously known patterns of sequent calculus in a natural deduction style [Kav20].

This kind of judgment straightforwardly allows the incorporation of a product-preserving comonad. The type $\Box A$ merely internalizes a restriction to modal contexts only:

$$\frac{\Delta; \cdot \vdash A \text{ true}}{\Delta; \Gamma \vdash \Box A \text{ true}} \qquad \frac{A \in \Delta \cup \Gamma}{\Delta; \Gamma \vdash A \text{ true}} \qquad \frac{\Delta; \Gamma \vdash \Box A \text{ true} \qquad \Delta, A; \Gamma \vdash B \text{ true}}{\Delta; \Gamma \vdash B \text{ true}}$$

The second author showed that this pattern adapts well to the necessity fragment of a number of normal modal logics [Kav20]. The dual-context style has been succesfully adapted to dependent types: see e.g. the work of [dR15], and the spatial and cohesive type theories of [Shu18]. Similarly, contextual modal type theory [NPP08, BP11, BBS15, PAF+19] has used a dual-context-like structure in order to give a systematic account of higher-order abstract syntax.

[Zwa19] continues this program by formulating a precise categorical semantics based on Awodey's natural models for a dependent type theory with either an adjunction (AdjTT) or comonad (CoTT) [Zwa19]. The categorical semantics of MTT and AdjTT are closely related, though with minor differences in the precise definition of the modality. For instance, in MTT only the 🔒 operator is required to act upon the context, while in AdjTT the modalities themselves must extend to contexts.[12] These differences arise because Zwanziger characterizes only a certain, semantically well-behaved subclass of models, while in Section 5 we describe more general models, which also support the syntactic model and the gluing model of Section 6. Syntactically, AdjTT is a multimode type theory that includes a mode for both ends of the adjunction.

Despite these stories of success, the dual-context style is difficult to generalize: as the complexity of the modal situation increases, so must the complexity of the context structure. For instance, the structure of a dependent dual-context type theory enforces that a 'modal' type (one belonging to Δ) may not depend on an 'intuistionistic' type (one belonging to Γ). This is a reasonable restriction in the case of $\Box$, but it is already somewhat limiting. For

---

[11]The idea of dual contexts arose in linear logic: see [And92, Gir93, Plo93].

[12]This is similar to the relation between a CwF+A and a CwDRA from [BCM+20], and we expect a similar relation to exist between the semantics of MTT with a single modality and AdjTT.

instance, it should be allowed for a valid type to depend on a merely true $\Box A$. Making such an adjustment would not only present a typographical problem (with a type occurring to the left of one of its dependencies), it would render the introduction rule for $\Box A$ nonsensical.

This restriction proves even more difficult to manage once there is not merely one modality, but two distinct modalities ones, say μ and ν. Questions such as "should the μ-modified types be allowed to depend on ν-modified types?" defy general answers. These questions can be addressed for each specific modal situation. For example, both [Shu18] and [Zwa19] hand-craft a system for two modalities. However, these constructions strongly depend on the structure of the underlying model, encouraging the proliferation of tiresome metatheoretic work as we discussed in Section 1.

What is lacking with the dual-context style is the ability to work systematically with a large class of modal situations without reconsidering the properties of the system in each case. Some of the rules of MTT can be directly traced to rules in dual-context calculi (in particular, the elimination rule for modal types), but the structure of our contexts is radically different, in a way which is far more accommodating.

11.2. **Modal type theories based on left division.** A separate strand of modal type theories builds its syntax around a structure that is termed *left division* by [ND18]. Rather than having a fixed number of distinct modal and intuitionistic contexts, there is a single context consisting of variables with *modal annotations*. The earliest appearance of this pattern is in the work of [Pfe01], where the annotations described a variable as having various degrees of proof (ir)relevance.

In a non-dependent type system, the distinction between annotations and different contexts is artificial: we could simply sort variables by their annotation, and separate them into different context zones. However, once generalized to a dependent type theory have a distinct advantage: they do not impose a fixed dependence schedule between different contexts. Instead, a type may depend on anything preceding it in the context, but the nature of that dependence is moderated by the modal annotations.

The term 'left division' is chosen to describe this structure because of the behavior of the introduction rules for modal types. For instance, in [Pfe01], there is a rule for introducing a term in an irrelevant context:

$$\frac{\Gamma^{\oplus} \vdash M : A}{\Gamma \vdash M :_{\mathsf{irr}} A}$$

Here $-^{\oplus}$ is a metatheoretic operation, which traverses the context and removes irrelevance annotations. The effect of this is that all the variables in $\Gamma^{\oplus}$ can be used freely when type-checking M. This is acceptable, because M itself is irrelevant. Viewed properly this is a division operation which 'divides' all the annotations in Γ by irr. The metatheory of a full dependent type theory based on this idea was considered by [AS12], who prove that modelling irrelevance in this way is sound and decidable.

More recent work by the third author [NVD17, ND18] has carried this idea to its natural conclusion by incorporating an entire hierarchy of modalities. In a related but distinct line of work, the Granule Project [GKO$^+$16, OLEI19] has exploited a similar structure to give a systematic account of substructurality. There is ongoing work to extend this to a full dependent type theory.

The modal annotations of MTT are very similar to the modal annotations of variables in calculi with left division. Contrasting MTT with [Pfe01] in particular, we find that there

are three classes of variables in *op. cit.*: normal variables (written $x : A$), irrelevant variables ($x \div A$), and valid variables ($x :: A$). Such a situation would be modeled in MTT by a single mode that has three endomodalities: irrelevance, extensionality (the identity modality), and validity. A composition table for these modalities can be built from the relations in [Pfe01]'s calculus.

The rules for interacting with the modalities in *op. cit.* traverse the context and modify the binding used for each variable. This bulk operation is very different to MTT-style locks, but amounts to similar constraints on variable use. By tagging the context with a lock, every time we use a variable we must ensure that the annotated modality sufficiently strong to overcome the lock. When we bulk-update the context, the same restrictions occur but they are performed 'eagerly.'

The use of 'lazy' locks has several advantages over 'eager' bulk updates. For instance, we do not have to explain what it means to divide one modality by another, and non-trivial 2-cells are possible. Furthermore, when interpreting the calculus in a model, it is unnecessary to describe variable by variable how modality update affects the interpretation of the entire context (which can be challenging: see e.g. [Nuy18]).

11.3. **Fitch-style modal type theories.** A recent series of papers has used a judgmental structure that is similar to MTT in order to manage a variety of modalities [BGM17, BCM$^+$20, GSB19a]. This kind of structure, informally often referred to as the *Fitch-style* [Clo18], divides the context into regions of variables separated by locks, but does not use annotations on individual variables. Locks are dynamically included or removed by the typing rules.

The central advantage of the Fitch-style is the impressively simple introduction rule for modalities: whenever we wish to introduce a modality we simply append a lock to the context—which tags the modal shift—and continue typechecking. In particular, we never need to remove variables from the context during the introduction of a modal term. Of course, like in MTT this style is only sound for a modality which comes equipped with some sort of left-adjoint-like operation.

Another desirable property of the Fitch-style calculi is their support for strong elimination rules for modalities. Instead of the pattern matching-style rules of other systems, Fitch-style calculi have had an *open scope* elimination rule for their modalities, which often permits a definitional η-rule for $\Box A$. It is generally of the following shape:

$$\frac{\mathfrak{F}(\Gamma) \vdash M : \Box A}{\Gamma \vdash \mathsf{open}(M) : A}$$

$\mathfrak{F}$ is a meta-theoretic operation on contexts which removes some number of locks and variables from Γ. For instance, in [BCM$^+$20] the operation $\mathfrak{F}(\Gamma)$ was defined by

$$\mathfrak{F}(\Gamma, \blacksquare, \Gamma') = \Gamma \text{ where } \blacksquare \notin \Gamma'.$$

This rule is convenient, and strictly more powerful than that of MTT (see Section 7). However, it is metatheoretically less than ideal. The source of the trouble in this case is that we must show that substitutions can be pushed under the open construct. For instance, suppose we have some substitution $\gamma : \Delta \to \Gamma, \blacksquare, \Gamma'$. It is necessary to ensure that this substitution uniquely gives rise to a substitution $\mathfrak{F}(\gamma) : \mathfrak{F}(\Delta) \to \Gamma$, which will then be applied to the body M of the term. This property can only be shown by lengthy induction on syntax. Such a property is proven laboriously in [GSB19a] for the MLTT$_\blacksquare$ type theory, and several complex and seemingly artificial typing rules are necessary to show it. The situation

is in some ways similar to dual-context calculi, where meticulous expert attention is needed to show the admissibility of substitution in each modal setting.

The final and most serious issue with the Fitch-style is the difficulty of accounting for multiple distinct modalities. Each modality should give rise to a different lock, but the structural rules governing their interactions are complex. It is well-understood how to model the ▶ modality in a Fitch-style type theory, and [GSB19a] developed an extensive account of the □ modality. However, it is an open problem whether the two may be combined. There is work to this effect in a simple type theory [BGM19], but even in this case there are restrictions on □ and ▶ which prevent the recovery of the MLTT▪ type theory of [GSB19a] as a subsystem.

These issues seem to converge to one cause: rules that 'remove' elements from the context during type-checking appear difficult to manage when combining modalities. As they operate on a syntactic level, they also seem to prohibit the formulation of internal languages. Drawing on this intuition, MTT has adopted the simple introduction rules of Fitch-style calculi, but not the elimination rules. The result is a less powerful type theory, with a weaker definitional equality, and no definitional η-principle. In return, MTT scales to any mode theory, including any number of interacting modalities.

11.4. **Other work.** The question of a multimodal framework for type theory has also been tackled by other recent work [LS16, LSR17]. This line of research is commonly referred to as the *LSR* framework, after the initials of the authors. LSR is designed to handle a wide variety of modal situations in combination with a variety of different *substructural* settings. There has been ongoing work on extending this system to a full dependent type theory, but as of late 2020 this work remains unpublished.

The impetus for the LSR framework is mainly derived from a long-standing wish to address the interaction between dependent types and substructural logics. This is an axis of generalization which is entirely outside the scope of MTT. However, we may compare LSR to MTT along the modal axis.

The idea of parametrizing a type theory by a mode theory, as we have done with MTT, originates in a paper preceding the LSR framework [LS16]. In fact, the modal situations that can be handled by MTT are a strict subset of those which can be handled by pre-LSR/LSR, which also includes a modality representing the *left adjoint* as an operation on types (and not just contexts). By contrast, MTT has a simpler syntax which is amenable to current proof and implementation techniques. This is reflected in our proof of canonicity, and our experimental implementation efforts [Nuy19]. We therefore believe that MTT is a natural halfway point between current modal type theories (which are custom-fitted for each modal situation) and the full generality of LSR: it is a simpler theory which accounts for many situations of interest.

12. Conclusions

We introduced and studied MTT, a dependent type theory parametrized by a mode theory that describes interacting modalities. We have demonstrated that MTT may be used to reason about several important modal settings, and proven basic metatheorems about its syntax, including canonicity.

Several distinct directions of future work present themselves.

Towards an Implementation of MTT. A major point of future work is the development of an implementation of MTT. Substantial preliminary implementation efforts are already underway with `Menkar` [Nuy19]. In addition to the engineering effort, a systematic account for an algorithmic syntax of MTT as well as proof of normalization is needed. We believe that the general ideas of [GSB19a] are applicable to this situation and there is ongoing work to apply them to MTT through more modern *gluing* techniques [Coq19]. Eventually, this work should prove that $\Gamma \vdash M = N : A \,@\, m$ and $\Gamma \vdash A = B \,\mathsf{type}_\ell \,@\, m$ are decidable relative to a decision procedure for equality in the underlying mode theory.

Left Adjoints. As discussed in Section 11.4, MTT trades a measure of generality for a degree of simplicity, as compared to LSR. One might hope, however, that it would be possible to include a connective for *left adjoints*, as well as the current connective which models right adjoints without losing all of this simplicity. It is not obvious that this can be done without significantly changing MTT: the introduction rule for modalities is exceptionally specific to a right adjoint. This additionally flexibility would allow us to model several modalities which are currently out of reach. For instance, when modeling a string of adjoints, we always fail to model the final left adjoint. Concretely speaking, the inclusion of left adjoints would allow MTT to model computational effects [Mog91, Lev12], as we will be able to internally recover the corresponding monad as the composite of the two parts of an adjoint pair.

References

[Abe06]   Andreas Abel. *A Polymorphic Lambda-Calculus with Sized Higher-Order Types*. PhD thesis, Ludwig-Maximilians-Universität München, 2006.

[Abe08]   Andreas Abel. Polarised subtyping for sized types. *Mathematical Structures in Computer Science*, 18(5):797–822, 2008.

[AK16]   Thorsten Altenkirch and Ambrus Kaposi. Normalisation by Evaluation for Dependent Types. In Delia Kesner and Brigitte Pientka, editors, *1st International Conference on Formal Structures for Computation and Deduction (FSCD 2016)*, volume 52 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:16, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[AM13]   Robert Atkey and Conor McBride. Productive Coprogramming with Guarded Recursion. In *Proceedings of the 18th ACM SIGPLAN International Conference on Functional Programming*, ICFP '13, pages 197–208. Association for Computing Machinery, 2013.

[And92]   Jean-Marc Andreoli. Logic Programming with Focusing Proofs in Linear Logic. *Journal of Logic and Computation*, 2(3):297–347, 06 1992.

[AS12]   Andreas Abel and Gabriel Scherer. On Irrelevance and Algorithmic Equality in Predicative Type Theory. *Logical Methods in Computer Science*, 8(1), 2012.

[Awo10]   Steve Awodey. *Category Theory*. Oxford Logic Guides. Oxford University Press, 2010.

[Awo18]   Steve Awodey. Natural models of homotopy type theory. *Mathematical Structures in Computer Science*, 28(2):241–286, 2018.

[BBS15] Peter Brottveit Bock and Carsten Schürmann. A contextual logical framework. volume 9450, pages 402–417, November 2015.

[BCM⁺20] Lars Birkedal, Ranald Clouston, Bassel Mannaa, Rasmus Ejlers Møgelberg, Andrew M. Pitts, and Bas Spitters. Modal dependent type theory and dependent right adjoints. *Mathematical Structures in Computer Science*, 30(2):118–138, 2020.

[Bd00] G. M. Bierman and V. C. V. de Paiva. On an intuitionistic modal logic. *Studia Logica*, 65(3), 2000.

[BGC⁺16] Aleš Bizjak, Hans Bugge Grathwohl, Ranald Clouston, Rasmus E. Møgelberg, and Lars Birkedal. Guarded Dependent Type Theory with Coinductive Types. In Bart Jacobs and Christof Löding, editors, *Foundations of Software Science and Computation Structures*, pages 20–35. Springer Berlin Heidelberg, 2016.

[BGM17] Patrick Bahr, Hans Bugge Grathwohl, and Rasmus Ejlers Møgelberg. The clocks are ticking: No more delays! In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. IEEE, 2017.

[BGM19] Patrick Bahr, Christian Uldal Graulund, and Rasmus Ejlers Møgelberg. Simply ratt: A fitch-style modal calculus for reactive programming without space leaks. *Proc. ACM Program. Lang.*, 3:109:1–109:27, 2019.

[BM15] Aleš Bizjak and Rasmus Ejlers Møgelberg. A model of guarded recursion with clock synchronisation. *Electronic Notes in Theoretical Computer Science*, 319:83 – 101, 2015. The 31st Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXI).

[BMSS12] Lars Birkedal, Rasmus Møgelberg, Jan Schwinghammer, and Kristian Støvring. First steps in synthetic guarded domain theory: step-indexing in the topos of trees. *Logical Methods in Computer Science*, 8(4), 2012.

[BP11] Mathieu Boespflug and Brigitte Pientka. Multi-level contextual type theory. *Electronic Proceedings in Theoretical Computer Science*, 71, October 2011.

[Car78] John Cartmell. *Generalised Algebraic Theories and Contextual Categories*. PhD thesis, University of Oxford, 1978.

[CBGB15] Ranald Clouston, Aleš Bizjak, Hans Bugge Grathwohl, and Lars Birkedal. Programming and reasoning with guarded recursion for coinductive types. In Andrew Pitts, editor, *Foundations of Software Science and Computation Structures*, pages 407–421. Springer Berlin Heidelberg, 2015.

[CD14] Pierre Clairambault and Peter Dybjer. The biequivalence of locally cartesian closed categories and martin-löf type theories. *Mathematical Structures in Computer Science*, 24(6), 2014.

[Clo18] Ranald Clouston. Fitch-Style Modal Lambda Calculi. In Christel Baier and Ugo Dal Lago, editors, *Foundations of Software Science and Computation Structures*, pages 258–275. Springer International Publishing, 2018.

[Coq86] Thierry Coquand. An analysis of girard's paradox. In *Proceedings of the First Annual IEEE Symposium on Logic in Computer Science (LICS 1986)*, pages 227–236. IEEE Computer Society Press, 1986.

[Coq96] Thierry Coquand. An algorithm for type-checking dependent types. *Science of Computer Programming*, 26(1):167–177, 1996.

[Coq13] Thierry Coquand. Presheaf model of type theory, 2013.

[Coq19] Thierry Coquand. Canonicity and normalization for dependent type theory. *Theoretical Computer Science*, 777:184–191, 2019.

[Cur90] P.-L. Curien. Substitution up to isomorphism. *Diagrammes*, 23:43–66, 1990.

[dR15] Valeria de Paiva and Eike Ritter. Fibrational modal type theory. In *Proceedings of the Tenth Workshop on Logical and Semantic Frameworks, with Applications (LSFA 2015)*, 2015.

[Dyb96] Peter Dybjer. Internal type theory. In Stefano Berardi and Mario Coppo, editors, *Types for Proofs and Programs*, pages 120–134, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[Fio12] Marcelo Fiore. Discrete generalised polynomial functors, 2012. Slides from talk given at ICALP 2012.

[Gir93] Jean-Yves Girard. On the unity of logic. *Annals of Pure and Applied Logic*, 59(3):201 – 217, 1993.

[GKNB20] Daniel Gratzer, G. A. Kavvos, Andreas Nuyts, and Lars Birkedal. Multimodal Dependent Type Theory. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 492–506, Saarbrücken Germany, July 2020. ACM.

[GKO+16] Marco Gaboardi, Shin-ya Katsumata, Dominic Orchard, Flavien Breuvart, and Tarmo Uustalu. Combining effects and coeffects via grading. In *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming*, ICFP 2016, 2016.

[GLN+17] Jacob A Gross, Daniel R Licata, Max S New, Jennifer Paykin, Mitchell Riley, Michael Shulman, and Felix Wellen. Differential Cohesive Type Theory (Extended Abstract). In *Extended abstracts for the Workshop "Homotopy Type Theory and Univalent Foundations"*, 2017.

[Gra11] Johan Georg Granström. *Treatise on Intuitionistic Type Theory.* Springer Netherlands, 2011.

[GSB19a] Daniel Gratzer, Jonathan Sterling, and Lars Birkedal. Implementing a Modal Dependent Type Theory. *Proc. ACM Program. Lang.*, 3, 2019.

[GSB19b] Daniel Gratzer, Jonathan Sterling, and Lars Birkedal. Normalization-by-evaluation for modal dependent type theory, 2019. Technical Report for the ICFP paper by the same name.

[Gua18] Adrien Guatto. A generalized modality for recursion. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18. ACM, 2018.

[Hof97] Martin Hofmann. Syntax and Semantics of Dependent Types. In Andrew M. Pitts and P. Dybjer, editors, *Semantics and Logics of Computation*, pages 79–130. Cambridge University Press, 1997.

[HS97] Martin Hofmann and Thomas Streicher. Lifting Grothendieck universes. Unpublished note, 1997.

[Kav19] G. A. Kavvos. Modalities, cohesion, and information flow. *Proceedings of the ACM on Programming Languages*, 3:20:1–20:29, 2019.

[Kav20] G. A. Kavvos. Dual-Context Calculi for Modal Logic. *Logical Methods in Computer Science*, 16(3), 2020.

[KHS19] Ambrus Kaposi, Simon Huber, and Christian Sattler. Gluing for type theory. In Herman Geuvers, editor, *Proceedings of the 4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019)*, volume 131, 2019.

[KKA19] Ambrus Kaposi, András Kovács, and Thorsten Altenkirch. Constructing quotient inductive-inductive types. *Proc. ACM Program. Lang.*, 3(POPL):2:1–2:24, January 2019.

[Law07] F. William Lawvere. Axiomatic cohesion. *Theory and Applications of Categories*, 19(3):41–49, 2007.

[Lev12] P.B. Levy. *Call-By-Push-Value: A Functional/Imperative Synthesis.* Semantics Structures in Computation. Springer Netherlands, 2012.

[LOPS18] Daniel R. Licata, Ian Orton, Andrew M. Pitts, and Bas Spitters. Internal Universes in Models of Homotopy Type Theory. In H. Kirchner, editor, *3rd International Conference on Formal Structures for Computation and Deduction (FSCD 2018)*, Leibniz International Proceedings in Informatics (LIPIcs), pages 22:1–22:17. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[LS16] Daniel R. Licata and Michael Shulman. Adjoint Logic with a 2-Category of Modes. In Sergei Artemov and Anil Nerode, editors, *Logical Foundations of Computer Science*, pages 219–235. Springer International Publishing, 2016.

[LSR17] Daniel R. Licata, Michael Shulman, and Mitchell Riley. A Fibrational Framework for Substructural and Modal Logics. In Dale Miller, editor, *2nd International Conference on Formal Structures for Computation and Deduction (FSCD 2017)*, volume 84 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 25:1–25:22. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017.

[Luo12] Zhaohui Luo. Notes on Universes in Type Theory, 2012. Notes for a talk at Institute for Advanced Study in Princeton in Oct 2012.

[ML78] Saunders Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics.* Springer New York, New York, NY, 1978.

[ML92] Per Martin-Löf. Substitution calculus, 1992. Notes from a lecture given in Göteborg.

[ML13] Stefan Milius and Tadeusz Litak. Guard Your Daggers and Traces: On The Equational Properties of Guarded (Co-)recursion. *Electronic Proceedings in Theoretical Computer Science*, 126(Informatik 8):72–86, August 2013.

[MLM92] Saunders Mac Lane and Ieke Moerdijk. *Sheaves in geometry and logic : a first introduction to topos theory.* Universitext. Springer, 1992.

[MM18] Bassel Mannaa and Rasmus Ejlers Møgelberg. The Clocks They Are Adjunctions Denotational Semantics for Clocked Type Theory. In Hélène Kirchner, editor, *3rd International Conference on Formal Structures for Computation and Deduction (FSCD 2018)*, volume 108 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 23:1–23:17, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Mog91]   Eugenio Moggi. Notions of computation and monads. *Information and Computation*, 93(1):55 – 92, 1991. Selections from 1989 IEEE Symposium on Logic in Computer Science.

[Møg14]   Rasmus Ejlers Møgelberg. A type theory for productive coprogramming via guarded recursion. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, CSL-LICS '14, 2014.

[MP08]   Conor McBride and Ross Paterson. Applicative programming with effects. *Journal of Functional Programming*, 18(1), 2008.

[Nak00]   H. Nakano. A modality for recursion. In *Proceedings Fifteenth Annual IEEE Symposium on Logic in Computer Science (Cat. No.99CB36332)*, pages 255–266. IEEE Computer Society, 2000.

[ND18]   Andreas Nuyts and Dominique Devriese. Degrees of relatedness: A unified framework for parametricity, irrelevance, ad hoc polymorphism, intersections, unions and algebra in dependent type theory. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18. ACM, 2018.

[New18]   Clive Newstead. *Algebraic models of dependent type theory.* Phd thesis, Carnegie Mellon University, 2018.

[NPP08]   Aleksandar Nanevski, Frank Pfenning, and Brigitte Pientka. Contextual Modal Type Theory. *ACM Transactions on Computational Logic*, 9(3), 2008.

[Nuy18]   Andreas Nuyts. Presheaf models of relational modalities in dependent type theory, 2018.

[Nuy19]   Andreas Nuyts. Menkar. `https://github.com/anuyts/menkar`, 2019.

[NVD17]   Andreas Nuyts, Andrea Vezzosi, and Dominique Devriese. Parametric quantifiers for dependent type theory. *Proc. ACM Program. Lang.*, 1(ICFP), 2017.

[OLEI19] Dominic Orchard, Vilem-Benjamin Liepelt, and Harley Eades III. Quantitative program reasoning with graded modal types. *Proc. ACM Program. Lang.*, 2019.

[OP18]   Ian Orton and Andrew M. Pitts. Axioms for Modelling Cubical Type Theory in a Topos. *Logical Methods in Computer Science*, 14(4), 2018.

[PAF⁺19] Brigitte Pientka, Andreas Abel, Francisco Ferreira, David Thibodeau, and Rébecca Zucchini. A type theory for defining logics and proofs. In *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2019.

[Pal98]   Erik Palmgren. On universes in type theory. In G. Sambin and J. Smith, editors, *Twenty Five Years of Constructive Type Theory*, Oxford Logic Guides, pages 191–204. Oxford University Press, 1998.

[PD01]   Frank Pfenning and Rowan Davies. A Judgmental Reconstruction of Modal Logic. *Mathematical Structures in Computer Science*, 11(4):511–540, 2001.

[Pfe01]   F. Pfenning. Intensionality, extensionality, and proof irrelevance in modal type theory. In *Proceedings 16th Annual IEEE Symposium on Logic in Computer Science*, pages 221–230. IEEE, 2001.

[Plo93]   G. D. Plotkin. Type theory and recursion. In *Proceedings Eighth Annual IEEE Symposium on Logic in Computer Science*, pages 374–, 1993.

[Pra65]   Dag Prawitz. *Natural Deduction: a proof-theoretical study.* Almquist and Wiksell, 1965.

[PT00]   Benjamin C. Pierce and David N. Turner. Local type inference. *ACM Transactions Programming Language and Systems*, 22(1):1–44, 2000.

[Ree09]   Jason Reed. A Judgmental Deconstruction of Modal Logic. 2009.

[Rie14]   Emily Riehl. *Categorical homotopy theory*, volume 24 of *New Mathematical Monographs*. Cambridge University Press, 2014.

[Rie16]   Emily Riehl. *Category Theory in Context.* Dover Publications, 2016.

[RSS20]   Egbert Rijke, Michael Shulman, and Bas Spitters. Modalities in homotopy type theory. *Logical Methods in Computer Science*, 16(1), 2020.

[Sch13]   Urs Schreiber. Differential cohomology in a cohesive infinity-topos. 2013.

[Shu15]   Michael Shulman. Univalence for inverse diagrams and homotopy canonicity. *Mathematical Structures in Computer Science*, 25(5):1203–1277, 2015.

[Shu18]   Michael Shulman. Brouwer's fixed-point theorem in real-cohesive homotopy type theory. *Mathematical Structures in Computer Science*, 28(6):856–941, 2018.

[SS86]   Stephen Schanuel and Ross Street. The free adjunction. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 27(1):81–83, 1986.

[SS12]   Urs Schreiber and Michael Shulman. Quantum gauge field theory in cohesive homotopy type theory. In Ross Duncan and Prakash Panangaden, editors, *Proceedings 9th Workshop on Quantum Physics and Logic, QPL 2012, Brussels, Belgium, 10-12 October 2012*, volume 158 of *EPTCS*, pages 109–126, 2012.

[Ste19]   Jonathan Sterling. Algebraic type theory and universe hierarchies. 2019.

[Uem19]   Taichi Uemura. A general framework for the semantics of type theory. 04 2019.

[Uni13]   The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013.

[Zwa19]   Colin Zwanziger. Natural model semantics for comonadic and adjoint type theory: Extended abstract. In *Preproceedings of Applied Category Theory Conference 2019*, 2019.