TRANSPENSION: THE RIGHT ADJOINT TO THE PI-TYPE

ANDREAS NUYTS* AND DOMINIQUE DEVRIESE°

*Vrije Universiteit Brussel, Belgium
URL: anuyts.github.io

°Vrije Universiteit Brussel, Belgium

Abstract. Presheaf models of dependent type theory have been successfully applied to model HoTT, parametricity, and directed, guarded and nominal type theory. There has been considerable interest in internalizing aspects of these presheaf models, either to make the resulting language more expressive, or in order to carry out further reasoning internally, allowing greater abstraction and sometimes automated verification. While the constructions of presheaf models largely follow a common pattern, approaches towards internalization do not. Throughout the literature, various internal presheaf operators ($\surd$, Φ/extent, Ψ/Gel, Glue, Weld, mill, the strictness axiom and locally fresh names) can be found and little is known about their relative expressiveness. Moreover, some of these require that variables whose type is a shape (representable presheaf, e.g. an interval) be used affinely.

We propose a novel type former, the transpension type, which is right adjoint to universal quantification over a shape. Its structure resembles a dependent version of the suspension type in HoTT. We give general typing rules and a presheaf semantics in terms of base category functors dubbed multipliers. Structural rules for shape variables and certain aspects of the transpension type depend on characteristics of the multiplier. We demonstrate how the transpension type and the strictness axiom can be combined to implement all and improve some of the aforementioned internalization operators (without formal claim in the case of locally fresh names).

1. Introduction and Related Work

1.1. **The power of presheaves.** Presheaf semantics [Hof97, HS97] are an excellent tool for modelling relational preservation properties of (dependent) type theory. They have been applied to parametricity (which is about preservation of relations) [AGJ14, BCM15, ND18a, NVD17], univalent type theory (preservation of equivalences) [BCH14, CMS20, CCHM17, Hub16, KLV12, Ort18, OP18], directed type theory (preservation of morphisms), guarded type theory (preservation of the stage of advancement of computation) [BM20] and even

Key words and phrases: dependent type theory, presheaf models, modal type theory, homotopy type theory, parametricity, directed type theory, guarded type theory.

*Carried out most of this research holding a PhD Fellowship from the Research Foundation - Flanders (FWO) at imec-DistriNet, KU Leuven, Belgium.

Preprint submitted to
Logical Methods in Computer Science

© A. Nuyts and D. Devriese
Creative Commons

combinations thereof [BBC+19, CH20, RS17, WL20].[1] The presheaf models just cited almost all follow a common pattern: First one chooses a suitable base category $\mathcal{W}$. The presheaf category over $\mathcal{W}$ is automatically a model of dependent type theory with the important basic type formers [Hof97] as well as a tower of universes [HS97]. Next, one identifies a suitable notion of fibrancy and replaces or supplements the existing type judgement $\Gamma \vdash T\ \mathsf{type}$ with one that classifies fibrant types:

HoTT: For homotopy type theory (HoTT, [Uni13]), one considers Kan fibrant types, i.e. presheaves in which edges can be composed and inverted as in an ∞-groupoid. The precise definition may differ in different treatments.

Parametricity: For parametric type theory, one considers discrete types [AGJ14, CH20, ND18a, NVD17]: essentially those that satisfy Reynolds' identity extension property [Rey83] which states that homogeneously related objects are equal. This can be expressed by requiring that any non-dependent function $\mathbb{I} \to A$ from the relational interval, is constant.

Directed: In directed type theory, one may want to consider Segal, covariant, discrete and Rezk types [RS17] and possibly also Conduché types [Gir64, Nuy18b][Nuy20a, ex. 8.1.27].

Guarded: In guarded type theory, one considers clock-irrelevant types [BM20]: types A such that any non-dependent function $\textcircled{\scriptsize 1} \to A$ from the clock type, is constant.

Nominal: Nominal type theory [Che12, PMD15] can be modelled in the Schanuel topos [Pit13b, §6.3]. This is the subcategory of nullary affine cubical sets (see example 6.12 later on) that send pushouts in the base category to pullbacks in Set. This ensures that if a cell depending on names $\{i, j, k\}$ in fact only depends on $\{i, j\}$ and in fact also only depends on $\{i, k\}$, then it only depends on $\{i\}$.

To the extent possible, one subsequently proves that the relevant notions of fibrancy are closed under basic type formers, so that we can restrict to fibrant types and still carry out most of the familiar type-theoretic reasoning and programming. Special care is required for the universe U: it is generally straightforward to adapt the standard Hofmann-Streicher universe to classify only fibrant types, but the universe of fibrant types is in general not automatically fibrant itself.

HoTT: In HoTT, the Hofmann-Streicher universe of Kan types is usually automatically Kan.

Parametricity: In earlier work on parametricity with Vezzosi [NVD17, ND18a], we made the universe of discrete types discrete by modifying its presheaf structure and introduced a parametric modality in order to use that universe. In contrast, Atkey et al. [AGJ14] and Cavallo and Harper [CH20] simply accept that their universes of discrete types are not discrete.

Directed: In directed type theory, one could expect, perhaps via a directed univalence result [WL20], that the universe of covariant types is Segal.

Guarded: In guarded type theory, Bizjak et al. [BGC+16] let the universe depend on a collection of in-scope clock variables lest the clock-indexed later modality $\triangleright : \forall(\kappa : \textcircled{\scriptsize 1}).\mathsf{U}_\Delta \to \mathsf{U}_\Delta$ (where $\kappa \in \Delta$) be non-dependent and therefore constant (not clock-indexed) by clock-irrelevance of $\mathsf{U} \to \mathsf{U}$ [BM20].

[1] We omit models that are not explicitly structured as presheaf models [AHH18, LH11, Nor19].

1.2. **Internalizing the power of presheaves.** Purely metatheoretic results about type theory certainly have their value. Parametricity, for instance, has originated and proven its value as a metatheoretic technique for reasoning about programs. However, with dependent type theory being not only a programming language but also a logic, it is preferable to formulate results about it within the type system, rather than outside it. We highlight two particular motivations for doing so: to enlarge the end user's toolbox, and to be able to prove internally that a type is fibrant.

Enlarging the end user's toolbox. One motivation for internalizing metatheorems is to enlarge the toolbox of the end user of the proof assistant. If this is the only goal, then we can prove the desired results in the model on pen and paper and then internalize them ad hoc with an axiom with or without computation rules.

HoTT: Book HoTT [Uni13] simply postulates the univalence axiom without computational behaviour, as justified e.g. by the model of Kan-fibrant simplicial sets [KLV12].

CCHM cubical type theory [CCHM17] provides the Glue type, which comes with introduction, elimination, β- and η-rules and which turns the univalence axiom into a theorem with computational behaviour. It also contains CCHM-Kan-fibrancy of all types as an axiom, in the form of the CCHM-Kan composition operator, with decreed computational behaviour that is defined by induction on the type.

Parametricity: Bernardy, Coquand and Moulin [BCM15, Mou16] (henceforth: BCM) internalize their (unary, but generalizable to k-ary) cubical set model of parametricity using two combinators Φ and Ψ [Mou16], a.k.a. extent and Gel [CH20]. Φ internalizes the presheaf structure of the function type, and Ψ that of the universe.

The combinator Φ and at first sight also Ψ require that the cubical set model lacks diagonals. Indeed, to construct a value over the primitive interval, Φ and Ψ each take one argument for every endpoint and one argument for the edge as a whole. Nested use of these combinators, e.g. to create a square, will take $(k+1)^2$ arguments for k^2 vertices, $2k$ sides and 1 square as a whole but none for specifying the diagonal. For this reason, BCM's type system enforces a form of *affine* use of interval variables. Similarly, connections as in CCHM [CCHM17] are ruled out. In the current paper, we will see that these requirements are not absolute for Ψ: there is apparently a very natural 'automatic' way to define the behaviour on diagonals and connections where the Ψ-type is not explicitly specified by its arguments.

In earlier work with Vezzosi [NVD17], we have internalized parametricity instead using the Glue type [CCHM17] and its dual Weld. Later on, we added a primitive mill [ND18b] for swapping Weld and $\Pi(i : \mathbb{I})$. These operations are sound in presheaves over any base category where we can multiply with $\mathbb{I}$ – including cube categories with diagonals or connections – and are (therefore) strictly less expressive than Φ which is not. Discreteness of all types was internalized as a non-computing *path degeneracy* axiom.[2]

Directed: Weaver and Licata [WL20] use a bicubical set model to show that directed HoTT [RS17] can be soundly extended with a directed univalence *axiom*.

[2]It is worth noting that it was not possible to use affine interval variables in the setting of [NVD17]: The type system features parametric Π-types which are modelled as ordinary Π-types with non-discrete domain. Discreteness of the Π-type can be proven solely from discreteness of the codomain, simply by swapping interval variable and function argument. This is however not possible in the affine setting, where only variables introduced prior to an interval variable are taken to be fresh for that interval variable and the exchange rule with an interval variable only works one way.

Guarded: In guarded type theory [BM20], one axiomatizes Löb induction and clock-irrelevance.

Nominal: One version of nominal type theory [PMD15] provides the locally fresh name abstraction $\nu(i : \mathbb{I})$ which can be used anywhere (i.e. the goal type remains the same after we abstract over a fresh name). The operation introduces a name but requires a body that is fresh for the name (i.e. we do not get to use it). This would be rather useless, were it not that we are allowed to *capture* the fresh name (see section 10).

Internalizing fibrancy proofs. Another motivation to internalize aspects of presheaf categories, is for building parts of the model inside the type theory, thus abstracting away certain categorical details such as the very definition of presheaves, and for some type systems enabling automatic verification of these constructions. Given the common pattern in models described in the previous section, it is particularly attractive to try and define fibrancy and prove results about it internally.

In the context of HoTT, Orton and Pitts [Ort18, OP18] study CCHM-Kan-fibrancy [CCHM17] in a type theory extended with a set of axioms, of which all but one serve to characterize the interval and the notion of cofibration. One axiom, *strictness*, provides a type former Strict for strictifying partial isomorphisms, which exists in every presheaf category. In order to prove fibrancy of the universe, Licata et al. postulate an "amazing right adjoint" $\mathbb{I} \surd \sqcup$ to the non-dependent path functor $\mathbb{I} \to \sqcup$ [LOPS18, Ort18], which indeed exists in presheaves over cartesian base categories if $\mathbb{I}$ is representable. Since $\mathbb{I} \surd \sqcup$ and its related axioms are global operations (only applicable to closed terms, unless you want to open Pandora's box as we do in the current paper), they keep everything sound by introducing a judgemental comonadic *global* modality $\flat$.

Orton et al.'s formalization [LOPS18, Ort18, OP18] is only what we call *meta-internal*: the argument is internalized to *some* type theory which still only serves as a metatheory of the type system of interest. Ideally, we would define and prove fibrancy of types *within* the type theory of interest, which we call *auto-internal*. Such treatments exist of discrete types in parametricity [CH20], and discrete, fibrewise-Segal and Rezk types in directed type theory [RS17], but not yet for covariant, Segal or Kan fibrant types due to the need to consider paths in the context $\mathbb{I} \to \Gamma$.

1.3. **The transpension type.** What is striking about the previous section is that, while most authors have been able to solve their own problems, a common approach is completely absent. We have encountered Φ and Ψ [Mou16], the amazing right adjoint $\surd$ [LOPS18], Glue [CCHM17, NVD17], Weld [NVD17], mill [ND18b] and the strictness axiom [OP18]. We have also seen that Φ and Ψ presently require an affine base category, and that $\surd$ presently requires the global modality $\flat$.

The goal of the current paper is to develop a smaller collection of internal primitives that impose few restrictions on the choice of base category and allow the internal construction of the aforementioned operators when sound. To this end, we introduce the **transpension** type former $\Diamond i : \mathrm{Ty}(\Gamma) \to \mathrm{Ty}(\Gamma, i : \mathbb{I})$ which in cartesian settings is right adjoint to $\Pi(i : \mathbb{I}) : \mathrm{Ty}(\Gamma, i : \mathbb{I}) \to \mathrm{Ty}(\Gamma)$ and is therefore not a quantifier binding i, but a coquantifier that *depends* on it. This same operation was already considered in topoi by Yetter [Yet87], who named it ∇. Using the transpension and Strict, we can construct Φ (when sound), Ψ, $\surd$ and Glue, and heuristically translate a subsystem of the nominal dependent type system

FreshMLTT [PMD15] featuring variable capture and locally fresh names. Given a type former for certain pushouts, we can also construct Weld and mill.

The transpension coquantifier $\talloblong(u : \mathbb{U}) : \mathrm{Ty}(\Gamma) \to \mathrm{Ty}(\Gamma, u : \mathbb{U})$ is part of a sequence of adjoints $\Sigma\, u \dashv \Omega\, u \dashv \Pi\, u \dashv \talloblong\, u$, preceded by the Σ-type, weakening and the Π-type. Adjointness of the first three is provable from the structural rules of type theory. However, it is not immediately clear how to add typing rules for a further adjoint. Birkedal et al. [BCM$^+$20] explain how to add a single modality that has a left adjoint in the semantics. If we want to have two or more adjoint modalities internally, then we can use a multimodal type system such as MTT [GKNB21, GKNB20]. Each modality in MTT needs a semantic left adjoint, so we can only internalize $\Omega\, u$, $\Pi\, u$ and $\talloblong\, u$. A drawback which we accept (as a challenge for future work), is that $\Omega\, u$ and $\Pi\, u$ become modalities which are a bit more awkward to deal with than ordinary weakening and Π-types.

A further complication is that the aforementioned modalities bind or depend on a variable, a phenomenon which is not supported by MTT. We solve this by grouping shape variables such as $u : \mathbb{U}$ in a **shape context** which is not considered part of the type-theoretic context but instead serves as the *mode* of the judgement.

1.4. **Contributions.** Our central contribution is to reduce the plethora of interal presheaf operators in the literature to only a few operations.

- To this end, we introduce the **transpension type** $\talloblong(u : \mathbb{U})$, right adjoint to $\Pi(u : \mathbb{U})$, with typing rules built on *extensional* MTT [GKNB21, GKNB20]. We explain how it is reminiscent of the suspension type from HoTT [Uni13].
- More generally, the transpension type can be right adjoint to any quantifier-like operation $\forall(u : \mathbb{U})$ which need neither respect the exchange rule, nor weakening or contraction. In this setting, we also introduce the **fresh weakening** coquantifier $\exists(u : \mathbb{U})$, which is left adjoint to $\forall(u : \mathbb{U})$ and therefore coincides with weakening $\Omega(u : \mathbb{U})$ in cartesian settings.
- We provide a categorical semantics for $\talloblong(u : \mathbb{U})$ in almost any presheaf category $\mathrm{Psh}(\mathcal{W})$ over base category $\mathcal{W}$, for almost any representable object $\mathbb{U} = \mathbf{y}U$, $U \in \mathcal{W}$. To accommodate non-cartesian variables, our system is not parametrized by a representable object $\mathbb{U} = \mathbf{y}U$, but by an arbitrary endofunctor $\sqcup \ltimes U$ on $\mathcal{W}$: the **multiplier**.[3] We introduce **criteria** for characterizing the multiplier – viz. semi-cartesian, cartesian, cancellative, affine, connection-free and quantifiable – which we use as requirements for internal type theoretic features. We identify a complication dubbed **spookiness** in certain models (most notably in guarded and nominal type theory), and define dimensionally split morphisms (a generalization of split epimorphisms) in order to include spooky models. We exhibit relevant multipliers in base categories found in the literature (section 6.3).
- We show that **all general presheaf internalization operators** that we are aware of – viz. Φ/extent (when sound), Ψ/Gel [Mou16, BCM15], the amazing right adjoint $\surd$ [LOPS18], Glue [CCHM17, NVD17], Weld [NVD17], mill [ND18b] and (with no formal claim) locally fresh names – can be **recovered** from just the transpension type, the strictness axiom and pushouts along $\mathsf{snd} : \varphi \times A \to A$ where $\varphi : \mathsf{Prop}$. In the process, some of these operators can be **improved**: We generalize Ψ from affine to arbitrary multipliers, including cartesian ones and we justify $\mathbb{U} \surd \sqcup$ without a global modality and get β- and η-rules for it. Moreover, since our system provides an operation $\text{🔒}_{\talloblong\, u}$ for quantifying over contexts, we take a step towards auto-internalizing Orton et al.'s work

[3] In the technical report [Nuy20b], we generalize multipliers beyond endofunctors.

[LOPS18, Ort18, OP18]. When Φ is not sound (e.g. in cartesian or non-connection-free settings), we suggest the internal notion of **transpensive** types to retain some of its power. Finally, a form of higher dimensional pattern matching is enabled by exposing $\forall(u : \mathbb{U})$ internally as a left adjoint.
- In a technical report [Nuy20b], we investigate how the modalities introduced in this paper commute with each other, and with prior modalities (i.e. those already present before adding the transpension type). We also consider composite multipliers, and natural transformations between modalities (called 2-cells in MTT) arising from natural transformations between multipliers.

While MTT [GKNB21, GKNB20] should satisfy canonicity[4], decidable type-checking will at least require a computational understanding of the mode theory, and of some new typing rules that we add to MTT. For this reason, we build on extensional MTT [GKNB20], and defer decidability and canonicity to future work.

1.5. **Overview of the paper.** In section 2, we study in a simple setting how the transpension type resembles the suspension type from HoTT and demonstrate how to put it in action. In section 3, we give a brief overview of the typing rules of MTT and introduce two optional notations: a system of *ticks* (lock variables) for naming the MTT locks in the context, and a *lockless* notation that is one cognitive step closer to the semantics. In section 4, we define the mode theory on which we will instantiate MTT. This mode theory is extremely general and essentially contains all adjoint pairs of a given domain and codomain as modalities between the corresponding modes. In the next two sections, we highlight a number of interesting modalities: in section 5 we consider modalities related to shape substitutions, and in section 6 we define and study multipliers and consider modalities arising from them, including the transpension modality. In section 7, we compare the obtained instance of MTT which features a modal transpension type, with the naïve system in section 2. In section 8, we supplement our MTT instance with a few specialized typing rules. In section 9, we investigate the structure of the transpension type. In section 10, we explain how to recover known internal presheaf operators. We conclude in section 11.

2. A Naïve Transpension Type

In this section, for purposes of demonstration, we present simplified typing rules for the transpension type. Using these, we will already be able to exhibit the transpension type as similar to a dependent version of the suspension type in HoTT [Uni13], and to prove internally that it is right adjoint to universal quantification. Moreover, in order to showcase how the transpension type allows us to internalize the presheaf structure of other types, we will demonstrate a technique which we call higher-dimensional pattern matching and which has already been demonstrated by Pitts [Pit14] in nominal type theory using locally fresh names [PMD15].

[4]The current state of affairs is that MTT extended with a single typing rule has been proven to satisfy canonicity. That rule is not compatible with the model used in this paper, but Gratzer et al. are working on a canonicity theorem for MTT proper.

Affine shape variables:

$$\frac{\Gamma\,\mathsf{ctx}}{\Gamma, u : \mathbb{U}\,\mathsf{ctx}} \qquad \frac{\sigma : \Gamma \to \Gamma'}{(\sigma, u/v) : (\Gamma, u : \mathbb{U}) \to (\Gamma', v : \mathbb{U})} \qquad \frac{\sigma : \Gamma \to \Gamma'}{\sigma : (\Gamma, u : \mathbb{U}) \to \Gamma'}$$

Affine function type:

$$\frac{\Gamma, u : \mathbb{U} \vdash A\,\mathsf{type}}{\Gamma \vdash \forall\, u.A\,\mathsf{type}} \qquad \frac{\Gamma, u : \mathbb{U} \vdash a : A}{\Gamma \vdash \lambda u.a : \forall\, u.A} \qquad \frac{\Gamma \vdash f : \forall\, u.A}{\Gamma, u : \mathbb{U}, \delta : \Delta \vdash f\, u : A}$$

Telescope quantification:

$$\begin{array}{lcl} \forall\, u.() & = & () \\ \forall\, u.(\delta : \Delta, x : A) & = & (\forall\, u.(\delta : \Delta)), {}^u x : \forall\, u.(A[{}^u\delta\, u/\delta]) \\ \forall\, u.(\delta : \Delta, v : \mathbb{U}) & = & (\forall\, u.(\delta : \Delta)), v : \mathbb{U} \end{array}$$

$$\begin{array}{lcl} {}^u() & = & () \\ {}^u(\delta, x) & = & ({}^u\delta, {}^u x) \\ {}^u(\delta, v) & = & ({}^u\delta, v) \end{array} \qquad \begin{array}{lcl} ()\, w & = & () \\ (\delta, x)\, w & = & (\delta\, w, x\, w) \\ (\delta, v)\, w & = & (\delta\, w, v) \end{array}$$

Transpension type:

$$\frac{\Gamma, u : \mathbb{U} \vdash (\delta : \Delta)\,\mathsf{telescope} \quad \Gamma, \forall\, u.(\delta : \Delta) \vdash A\,\mathsf{type}}{\Gamma, u : \mathbb{U}, \delta : \Delta \vdash \talloblong\, u.A\,\mathsf{type}} \qquad \frac{\Gamma, \forall\, u.(\delta : \Delta) \vdash a : A}{\Gamma, u : \mathbb{U}, \delta : \Delta \vdash \mathsf{mer}\, u\, a : \talloblong\, u.A} \qquad \frac{\Gamma, u : \mathbb{U} \vdash t : \talloblong\, u.A}{\Gamma \vdash \mathsf{unmer}(u.t) : A}$$

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash \mathsf{unmer}(u.\mathsf{mer}\, u\, a) = a : A} \qquad \frac{\Gamma, u : \mathbb{U}, \delta : \Delta \vdash t : \talloblong(u : \mathbb{U}).A}{\Gamma, u : \mathbb{U}, \delta : \Delta \vdash t = \mathsf{mer}\, u\, (\mathsf{unmer}(v.t[v/u, {}^u\delta\, v/\delta])) : \talloblong\, u.A}$$

$$\frac{\Gamma, \forall\, u.(\delta : \Delta) \vdash A\,\mathsf{type} \quad \Gamma, v : \mathbb{U}, \theta : \Theta \vdash \tau : \Delta[v/u]}{\Gamma, v : \mathbb{U}, \theta : \Theta \vdash (\talloblong\, u.A)[v/u, \tau/\delta] = \talloblong\, v.(A[\lambda w.\tau[w/v, {}^v\theta\ w/\theta]/{}^u\delta])\,\mathsf{type}}$$

$$\frac{\Gamma, \forall\, u.(\delta : \Delta) \vdash a : A \quad \Gamma, v : \mathbb{U}, \theta : \Theta \vdash \tau : \Delta}{\Gamma, v : \mathbb{U}, \theta : \Theta \vdash (\mathsf{mer}\, u\, a)[v/u, \tau/\delta] = \mathsf{mer}\, v\, (a[\lambda w.\tau[w/v, {}^v\theta\ w/\theta]/{}^u\delta]) : (\talloblong\, u.A)[v/u, \tau/\delta]}$$

Figure 1: Selection of typing rules for a naïve transpension type.

2.1. **Typing rules.** To do this, we first present, in fig. 1, typing rules for the transpension type in a very simple setting: a type system with affine shape variables $u : \mathbb{U}$. Variables to the left of u are understood to be fresh for u; variables introduced after u may be substituted with terms depending on u. In particular, we have no contraction $(w/u, w/v) : (w : \mathbb{U}) \to (u, v : \mathbb{U})$, while exchange $(x : A, u : \mathbb{U}) \to (u : \mathbb{U}, x : A)$ only works in one direction. This is enforced by the special substitution rules for shape variables. Weakening of shape variables is allowed. The examples in which the type system will be put to use, are agnostic as to whether exchange of shape variables $(u : \mathbb{U}, v : \mathbb{U}) \to (v : \mathbb{U}, u : \mathbb{U})$ is possible and models of both situations exist.

The system features an affine function type $\forall\, u.A$ over $\mathbb{U}$, with unsurprising formation and introduction rules. The rule for $f\, u$ requires that the function f be fresh for u, i.e. that f depend only on variables to the left of u [BCM15, Mou16].

Additionally, the system contains a transpension type $\talloblong\, u.A$ over $\mathbb{U}$, with more unusual rules. When checking the type $\talloblong\, u.A$ in context $(\Gamma, u : \mathbb{U}, y : B)$, the part A will be checked

in a modified context [BV17, ND19], were the variable $y : B$ (which potentially depends on u) will change type, becoming a function variable ${}^u y : \forall\, u.B$ that can be applied to $v : \mathbb{U}$ yielding ${}^u y\, v : B[v/u]$.[5] In other words, we get hold of the dependency of y on u. This applies more generally when $u : \mathbb{U}$ in the context is not followed by a single variable $y : B$ but by an entire telescope Δ. In this case, every type in Δ will become universally quantified over u.

The *meridian* constructor $\mathsf{mer}\, u\, a$ is checked in a similar way, and is modelled by transposition for the adjunction $\forall\, u \dashv \flat\, u$.[6] We remark that both $\flat\, u.A$ and $\mathsf{mer}\, u\, a$ depend on u, whereas A and a do not, so in a way the transpension lifts data to a higher dimension, turning points into $\mathbb{U}$-cells. The elimination rule takes data again to a lower dimension: it turns a dependent $\mathbb{U}$-cell in the transpension into a point in A. This is the co-unit of the adjunction. The β- and η-rules internalize the adjunction laws.

These typing rules are sound in certain presheaf categories (those where $\mathbb{U}$ is cancellative and affine, see definition 6.1), but are unsatisfactory in several respects. First, we have no story for substitutions which exist in cubical type systems such as endpoints $(0/i) : \Gamma \to (\Gamma, i : \mathbb{I})$ [BCM15, BCH14, CCHM17] or connections $(j \wedge k/i) : (\Gamma, j, k : \mathbb{I}) \to (\Gamma, i : \mathbb{I})$ [CCHM17], as there is no formation rule for $\flat\, 0.A$ or $\flat\, (j \wedge k).A$. Secondly, in non-affine generalizations, the transpension is not stable under substitution of the shape variables preceding u, so in those settings the way we internalized the transpension type here was too naïve.[7] In order to obtain a type system that does not fail in the presence of endpoints, connections or non-affinity, in the rest of the paper we will rely on MTT, which we briefly summarize in section 3.

2.2. **Poles.** We can still try to get a grasp on $\flat\, 0.A$ in cubical type systems, however. In general we have $T[0/i] \cong (\forall\, i.(i \equiv_{\mathbb{I}} 0) \to T)$. Assuming $T = \flat\, i.A$ and $\mathsf{oneIsNotZero} : (1 \equiv_{\mathbb{I}} 0) \to \mathsf{Empty}$, the latter type is inhabited by

$$\mathsf{pole}_0 := \lambda i.\lambda e.\mathsf{mer}\, i\, \left(\mathsf{case}\, \left(\mathsf{oneIsNotZero}\ \left({}^i e\ 1\right)\right)\, \mathsf{of}\{\}\right) : \forall\, i.(i \equiv_{\mathbb{I}} 0) \to \flat\, i.A$$

where ${}^i e$ has type $\forall\, i.(i \equiv_{\mathbb{I}} 0)$. Moreover, using the η-rules for functions and the transpension type and a (provable propositional) η-rule for Empty, we can show that this is the only element. Thus we see that the transpension type essentially consists of one meridian $(i : \mathbb{I}) \to \flat\, i.T$ for every $t : T$, and that these meridians are all equal to pole_0 at $i = 0$ and analogously to pole_1 at $i = 1$. This makes the transpension type quite reminiscent of a dependent version of the suspension type from HoTT [Uni13], although the quantification of the context in the formation and construction rules is obviously a distinction.

[5] Note that we view ${}^u y$ as a single variable name. As such, it could even be desugared to a de Bruijn index that bears no mention of the context modification whatsoever. The renaming of y to ${}^u y$ that accompanies the modification of the type merely serves as a reminder to human readers.

[6] More accurately, by *dependent* transposition [BCM+20][Nuy18a, §2.1.3][Nuy20a, §5.1.3-5.2].

[7] Indeed, write $\Omega\, u$ for the operation of cartesian weakening over a shape variable $u : \mathbb{U}$, which is an example of a substitution involving shape variables. If in general $\Omega\, u \circ \flat\, v \cong \flat\, v \circ \Omega\, u$, then by uniqueness of the left adjoint we would find that $\Pi\, v \circ \Sigma\, u \cong \Sigma\, u \circ \Pi\, v$. This is clearly false for cartesian shapes such as the interval $\mathbb{I}$ in HoTT. For more information on how the transpension type commutes with other operations, see the technical report [Nuy20b].

2.3. **Internal transposition.** We can internally show that the following types are isomorphic:

$$(\forall(u : \mathbb{U}).A) \to B \quad \cong \quad \forall(u : \mathbb{U}).(A \to \talloblong\, u.B).$$

Indeed, given $f : (\forall(u : \mathbb{U}).A) \to B$, we can define $g : \forall(u : \mathbb{U}).(A \to \talloblong\, u.B)$ by

$$g\, u\, a = \mathsf{mer}\, u\, (f\, ({}^u a)).$$

Conversely, given $g : \forall(u : \mathbb{U}).(A \to \talloblong\, u.B)$, we can define $f : (\forall(u : \mathbb{U}).A) \to B$ by

$$f\, \hat{a} = \mathsf{unmer}(u.g\, u\, (\hat{a}\, u)).$$

These constructions are mutually inverse. Indeed, plugging the definition of f into that of g, we find

$$\mathsf{mer}\, u\, (\mathsf{unmer}(v.g\, v\, ({}^u a\, v))) = \mathsf{mer}\, u\, (\mathsf{unmer}(v.(g\, u\, a)[v/u, {}^u a\, v/a])) = g\, u\, a$$

using the η-rule of the transpension type. Conversely, plugging g into f:

$$\begin{aligned}\mathsf{unmer}(u.(\mathsf{mer}\, u\, (f\, ({}^u a)))[u/u, \hat{a}\, u/a]) &= \mathsf{unmer}(u.\mathsf{mer}\, u\, ((f\, ({}^u a))[\lambda w.(\hat{a}\, u)[w/u]/{}^u a])) \\ &= \mathsf{unmer}(u.\mathsf{mer}\, u\, ((f\, ({}^u a))[\hat{a}/{}^u a])) \\ &= \mathsf{unmer}(u.\mathsf{mer}\, u\, (f\, \hat{a})) = f\, \hat{a}\end{aligned}$$

using substitution and β-rules of the transpension type as well as η-contraction of $\hat{a}$.

2.4. **Higher-dimensional pattern matching.** Now that we know internally that $\forall\, u$ is a left adjoint (with internal right adjoint $\talloblong\, u$), we can proceed to conclude that it preserves colimits, e.g. we can show $i : (\forall\, u.A \uplus B) \cong (\forall\, u.A) \uplus (\forall\, u.B)$. The map to the left is trivially defined by case analysis. The map to the right is equivalent by transposition to a function $\forall\, u.(A \uplus B \to \talloblong\, u.((\forall\, v.A[v/u]) \uplus (\forall\, v.B[v/u])))$. This is in turn constructed by case analysis from the transpositions of the coproduct's constructors inl and inr.

By straightforward application of section 2.3, the transpositions of the constructors are:

$$\begin{aligned}\lambda u.\lambda a.\mathsf{mer}\, u\, (\mathsf{inl}\, {}^u a) &: \forall\, u.(A \to \talloblong\, u.((\forall\, v.A[v/u]) \uplus (\forall\, v.B[v/u]))) \\ \lambda u.\lambda b.\mathsf{mer}\, u\, (\mathsf{inr}\, {}^u b) &: \forall\, u.(B \to \talloblong\, u.((\forall\, v.A[v/u]) \uplus (\forall\, v.B[v/u])))\end{aligned}$$

Pasting these together, we get

$$\lambda u.\lambda c.\mathsf{case}\, c\, \mathsf{of} \left\{ \begin{array}{lcl} \mathsf{inl}\, a & \mapsto & \mathsf{mer}\, u\, (\mathsf{inl}\, {}^u a) \\ \mathsf{inr}\, b & \mapsto & \mathsf{mer}\, u\, (\mathsf{inr}\, {}^u b) \end{array} \right\} : \forall\, u.(A \uplus B \to \talloblong\, u.((\forall\, v.A[v/u]) \uplus (\forall\, v.B[v/u]))) \ .$$

Transposing again as in section 2.3, we find

$$\begin{aligned} i &: (\forall\, u.A \uplus B) \to (\forall\, u.A) \uplus (\forall\, u.B) \\ i\, \hat{c} &= \mathsf{unmer} \left(u.\mathsf{case}\, \hat{c}\, u\, \mathsf{of} \left\{ \begin{array}{lcl} \mathsf{inl}\, a & \mapsto & \mathsf{mer}\, u\, (\mathsf{inl}\, {}^u a) \\ \mathsf{inr}\, b & \mapsto & \mathsf{mer}\, u\, (\mathsf{inr}\, {}^u b) \end{array} \right\} \right). \end{aligned}$$

Let us consider our categorically motivated creation from a more type-theoretical perspective. We obtain an argument $\hat{c} : \forall\, u.A \uplus B$ which we would like to pattern match on, in order to create an element of type $(\forall\, u.A) \uplus (\forall\, u.B)$. Of course we cannot pattern match on a function, so we call the unmer constructor which brings $u : \mathbb{U}$ in scope and changes the goal to $\talloblong\, u.((\forall\, v.A[v/u]) \uplus (\forall\, v.B[v/u]))$. We can then reduce $\hat{c}$ by one dimension by applying it to u, allowing a case analysis. The first case brings in scope $a : A$ (and the second case will be analogous), so we are in context $(\Gamma, \hat{c} : \forall\, u.A \uplus B, u : \mathbb{U}, a : A)$. We then use the meridian constructor, which again removes u from scope, turns $a : A$ into a function

${}^u a : \forall\, u.A$ and again reduces the goal to $(\forall\, u.A) \uplus (\forall\, u.B)$, so that inl completes the proof. We have essentially pattern matched on a higher-dimensional object!

Let us now check that i is indeed inverse to the trivial implementation of i^{-1}. We have

$$\begin{aligned}(i \circ i^{-1})(\mathsf{inl}\,\hat{a}) = i(\lambda u.\mathsf{inl}\,(\hat{a}\,u)) &= \mathsf{unmer}(u.(\mathsf{mer}\,u\,(\mathsf{inl}\,{}^u a))[u/u, \hat{a}\,u/a]) \\ &= \mathsf{unmer}(u.\mathsf{mer}\,u\,((\mathsf{inl}\,{}^u a)[\lambda w.(\hat{a}\,u)[w/u]/{}^u a])) \\ &= \mathsf{unmer}(u.\mathsf{mer}\,u\,(\mathsf{inl}\,\hat{a})) = \mathsf{inl}\,\hat{a}\end{aligned}$$

and similar for $(i \circ i^{-1})(\mathsf{inr}\,\hat{b})$. Using the technique of higher dimensional pattern matching just developed, we can prove the other equation also by pattern matching! We have

$$(i^{-1} \circ i)(\lambda u.\mathsf{inl}\,(\hat{a}\,u)) = \ldots = i^{-1}(\mathsf{inl}\,\hat{a}) = \lambda u.\mathsf{inl}\,(\hat{a}\,u),$$

where the dots replace the reasoning from above, and a similar result for $(i^{-1} \circ i)(\lambda u.\mathsf{inr}\,(\hat{b}\,u))$.

3. Multimode Type Theory

As announced, we will rely on the extensional version of Gratzer et al.'s multimode and multimodal dependent type system MTT [GKNB21, GKNB20] in order to frame the transpension and its left adjoints as modal operators. We refer to the original work for details, but in this section we highlight some important aspects of MTT and introduce our notation using *ticks*. In section 3.5, we discuss an alternative *lockless* notation of MTT that is one cognitive step closer to the semantics.

3.1. **The mode theory.** MTT is parametrized by a *mode theory*, which is a strict 2-category whose objects, morphisms and 2-cells we will refer to as **modes**, **modalities** and, well, **2-cells** respectively. Semantically, every mode p will correspond to an entire model of dependent type theory $[\![p]\!]$. A modality $\mu : p \to q$ will consist of a functor $[\![🔒_\mu]\!] : [\![q]\!] \to [\![p]\!]$ and an operation $[\![\mu]\!]$ that is almost a dependent right adjoint (DRA [BCM+20]) to $[\![🔒_\mu]\!]$; for all our purposes it will be an actual DRA and even one arising from a weak CwF morphism [BCM+20, lemma 17][Nuy18a]. A 2-cell $\alpha : \mu \Rightarrow \nu$ is interpreted as a natural transformation $[\![\alpha \downarrow]\!] : [\![🔒_\nu]\!] \to [\![🔒_\mu]\!]$ and hence also gives rise to an appropriate transformation $[\![\alpha]\!] : [\![\mu]\!] \to [\![\nu]\!]$.

3.2. **Judgement forms.** The judgement forms of MTT are listed in fig. 2. All forms are annotated with a mode p which specifies in what category they are to be interpreted. Every judgement form also has a corresponding equality judgement, which is respected by everything as the typing rules are to be read as a specification of a generalized algebraic theory (GAT [Car86]). The statements $p\,\mathsf{mode}$, $\mu : p \to q$ and $\alpha : \mu \Rightarrow \nu$ are simply requirements about the mode theory. This means we give no syntax or equality rules for modalities and 2-cells: these are fixed by the choice of mode theory.

3.3. **Typing rules.** The typing rules are listed in figs. 3 to 6; the reader is invited to ignore the grey boldface super- and subscript annotations for now. These are tick annotations and will be discussed in section 3.4. We discuss the typing rules in turn.

Judgement forms:

$p \mid \Gamma\,\mathsf{ctx}$	Γ is a context at mode p,
$p \mid \sigma : \Gamma \to \Delta$	σ is a simultaneous substitution from Γ to Δ at mode p,
$p \mid \Gamma \vdash T\,\mathsf{type}$	T is a type in context Γ at mode p,
$p \mid \Gamma \vdash t : T$	t has type T in context Γ at mode p.

Figure 2: Judgement forms of MTT [GKNB21].

The type theory at each mode:

Basic rules of dependent type theory (including all desired types) at each mode q, e.g.:

CTX-EMPTY
$$\frac{q\,\mathsf{mode}}{q \mid \cdot\,\mathsf{ctx}}$$

CTX-EXT
$$\frac{q \mid \Gamma \vdash T\,\mathsf{type}}{q \mid \Gamma, x : T\,\mathsf{ctx}}$$

CTX-EXT:INTRO
$$\frac{q \mid \sigma : \Gamma \to \Delta \quad q \mid \Gamma \vdash t : T[\sigma]}{q \mid (\sigma, t/x) : \Gamma \to (\Delta, x : T)}$$
where $\tau = ((x/\oslash) \circ \tau, x[\tau]/x)$
$(\sigma, t/x) \circ \rho = (\sigma \circ \rho, t[\rho]/x)$

CTX-EXT:WKN
$$\frac{q \mid \Gamma\,\mathsf{ctx} \qquad q \mid \Gamma \vdash T\,\mathsf{type}}{q \mid (x/\oslash) : (\Gamma, x : T) \to \Gamma}$$
where $(x/\oslash) \circ (\sigma, t/x) = \sigma$

CTX-EXT:VAR
$$\frac{q \mid \Gamma\,\mathsf{ctx} \qquad q \mid \Gamma \vdash T\,\mathsf{type}}{q \mid \Gamma, x : T \vdash x : T[(x/\oslash)]}$$
where $x[\sigma, t/x] = t$

SIGMA
$$\frac{q \mid \Gamma \vdash A\,\mathsf{type} \qquad q \mid \Gamma, x : A \vdash B\,\mathsf{type}}{q \mid \Gamma \vdash (x : A) \times B\,\mathsf{type}}$$

UNI
$$\frac{q \mid \Gamma\,\mathsf{ctx} \quad \ell \in \mathbb{N}}{q \mid \Gamma \vdash \mathsf{U}^q_\ell\,\mathsf{type}_{\ell+1}}$$

UNI:ELIM
$$\frac{q \mid \Gamma \vdash t : \mathsf{U}^q_\ell}{q \mid \Gamma \vdash \mathsf{El}(t)\,\mathsf{type}_\ell}$$
where $\mathsf{El}(\ulcorner T \urcorner) = T$

UNI:INTRO
$$\frac{q \mid \Gamma \vdash T\,\mathsf{type}_\ell}{q \mid \Gamma \vdash \ulcorner T \urcorner : \mathsf{U}^q_\ell}$$
where $\ulcorner \mathsf{El}(t) \urcorner = t$

Figure 3: MTT includes all rules of ordinary DTT at each mode.

3.3.1. *The type theory at each mode.* Since every mode corresponds to a model of all of dependent type theory (DTT), we start by importing **all the usual typing rules of DTT**, to be applied in MTT at any given fixed mode. Some examples of such rules are given in fig. 3, where we have consciously included rules for non-modal context extension, even though these will be generalized to modal rules later on. One reason to do so is that other rules of DTT, such as SIGMA, depend on these and therefore cannot be imported without. Another is that this way, we have a warm-up towards the modal rules and in particular we can make a point about de Bruijn indices. Although variables in fig. 3 are named, the rules CTX-EXT:WKN and CTX-EXT:VAR effectively enforce a **de Bruijn** discipline, where we can only name the last variable in the context and have to weaken explicitly if it is deeper down, e.g.

$$x : A, y : B, z : C \vdash x[(y/\oslash)][(z/\oslash)] : A[(x/\oslash)][(y/\oslash)][(z/\oslash)].$$

We take the viewpoint[8] that the official system is unnamed and uses this substitution-based de Bruijn discipline to refer to variables. In order to improve human communication, we will name variables anyway and use the resulting redundancy to leave weakening substitutions

[8] as is done in the MTT technical report [GKNB20]; the paper [GKNB21] speaks from a more implementation-oriented perspective.

WDRA

$$\frac{\mu : p \to q \qquad p \mid \Gamma, \text{🔒}^{\mathrm{m}}_{\mu} \vdash A\,\mathsf{type}_{\ell}}{q \mid \Gamma \vdash \langle \mu \mid^{\mathrm{m}} A \rangle\,\mathsf{type}_{\ell}}$$

WDRA:INTRO

$$\frac{\mu : p \to q \qquad p \mid \Gamma, \text{🔒}^{\mathrm{m}}_{\mu} \vdash a : A}{q \mid \Gamma \vdash \mathsf{mod}^{\mathrm{m}}_{\mu}\, a : \langle \mu \mid^{\mathrm{m}} A \rangle}$$

WDRA:ELIM

$$\frac{\begin{array}{l}\mu : p \to q \qquad \nu : q \to r \\ q \mid \Gamma, \text{🔒}^{\mathrm{n}}_{\nu} \vdash \hat{a} : \langle \mu \mid^{\mathrm{m}} A \rangle \\ r \mid \Gamma, \nu \circ \mu \vartriangleright x :^{\mathrm{nm}} A \vdash c : C[\mathsf{mod}^{\mathrm{m}}_{\mu}\, x\!\downarrow_{\mathrm{nm}} / \hat{x}\!\downarrow_{\mathrm{n}}]\end{array}}{\begin{array}{l} r \mid \Gamma \vdash \mathsf{let}^{\mathrm{n}}_{\nu}\,(\mathsf{mod}^{\mathrm{m}}_{\mu}\, x\!\downarrow_{\mathrm{nm}} = \hat{a})\,\mathsf{in}\, c : C[\hat{a}/\hat{x}\!\downarrow_{\mathrm{n}}] \\ \text{where } \mathsf{let}^{\mathrm{n}}_{\nu}\,(\mathsf{mod}^{\mathrm{m}}_{\mu}\, x\!\downarrow_{\mathrm{nm}} = \mathsf{mod}^{\mathrm{m}}_{\mu}\, a)\,\mathsf{in}\, c = c[a/x\!\downarrow_{\mathrm{nm}}]\end{array}}$$

Figure 4: Typing rules for MTT's modal types (weak DRAs) [GKNB21][Nuy20a, fig. 5.6].

implicit unambiguously. This allows for the following unofficial admissible 'rule'

$$\frac{q \mid \Gamma, x : T, \Delta\,\mathsf{ctx}}{q \mid \Gamma, x : T, \Delta \vdash x : T}\ \text{CTX-EXT:VAR:LOOKUP}.$$

Furthermore, we use other common notational conventions such as writing (t/x) instead of $(\mathrm{id}_{\Gamma}, t/x) : \Gamma \to (\Gamma, x : T)$.

We assume that DTT has a **universe** à la Coquand with mutually inverse encoding and decoding operations (which we will henceforth suppress), and we ignore cumulativity-related hassle, referring to Gratzer et al. [GKNB21] for details.

3.3.2. *Modal types, part 1.* Before proceeding to the MTT-specific structural rules, let us first have a look at the formation and introduction rules WDRA and WDRA:INTRO of **modal types** $\langle \mu \mid A \rangle$ in fig. 4. These are not unlike the formation and introduction rules of the naïve transpension type in fig. 1 and work by transposition: we apply the left adjoint of the modality μ (in the form of a lock) to the premise's context. As such, they behave like DRAs, but their elimination rule WDRA:ELIM (which we consider later) is weaker, so we call them weak DRAs.

3.3.3. *Structural rules.* The structural rules of MTT are listed in fig. 5. **Context formation** starts with the empty context which exists at any mode, and proceeds by adding locks and variables.

Adding **locks** (LOCK) is strictly functorial: it preserves identity and composition of modalities. In fact, it is strictly 2-functorial: it also has an action on 2-cells (LOCK:FMAP, producing substitutions between locked contexts) that preserves identity and composition of 2-cells. It is also strictly bifunctorial: we can combine a substitution and a 2-cell to a substitution between locked contexts. If the 2-cell is the identity, then we write $\downarrow^{\mathrm{m}}_{\mathrm{m}'}$ for $1_{\mu}\!\downarrow^{\mathrm{m}}_{\mathrm{m}'}$.

A **modal variable** $\mu \vartriangleright x :^{\mathrm{m}} T$ introduced by CTX-MODEXT is essentially the same as a non-modal variable $\hat{x} : \langle \mu \mid^{\mathrm{m}} T \rangle$ (which in turn is shorthand for $1 \vartriangleright \hat{x} :^{\bullet} \langle \mu \mid^{\mathrm{m}} T \rangle$), but the judgemental modal annotation allows direct access to a variable of type A through the variable rule. Hence, the type T is checked the same way as it would be in $\langle \mu \mid^{\mathrm{m}} T \rangle$. Terms t substituted for a modal variable x are also checked in the locked context, as if we would be

Context locking:

Note: $\downarrow^{\mathrm{m}}_{\mathrm{m}}$ is shorthand for $1\downarrow^{\mathrm{m}}_{\mathrm{m}}$.

LOCK

$$\frac{q \mid \Gamma\ \mathsf{ctx} \quad \mu : p \to q}{p \mid \Gamma, \mathbb{a}^{\mathrm{m}}_{\mu}\ \mathsf{ctx}}$$

where $\Gamma = (\Gamma, \mathbb{a}^{\bullet}_{1})$

$(\Gamma, \mathbb{a}^{\mathrm{n}}_{\nu}, \mathbb{a}^{\mathrm{m}}_{\mu}) = (\Gamma, \mathbb{a}^{\mathrm{nm}}_{\nu\circ\mu})$

LOCK:FMAP

$$\frac{q \mid \sigma : \Gamma \to \Delta \qquad \mu, \nu : p \to q \qquad \alpha : \mu \Rightarrow \nu}{p \mid (\sigma, \alpha\downarrow^{\mathrm{m}}_{\mathrm{n}}) : (\Gamma, \mathbb{a}^{\mathrm{n}}_{\nu}) \to (\Delta, \mathbb{a}^{\mathrm{m}}_{\mu})}$$

where $\sigma = (\sigma, \downarrow^{\bullet}_{\bullet})$ $\quad (\sigma, \alpha'\downarrow^{\mathrm{m}'}_{\mathrm{n}'}, \alpha\downarrow^{\mathrm{m}}_{\mathrm{n}}) = (\sigma, (\alpha' \star \alpha)\downarrow^{\mathrm{m}'\mathrm{m}}_{\mathrm{n}'\mathrm{n}})$

$1 = (1, \downarrow^{\mathrm{m}}_{\mathrm{m}})$ $\quad (\sigma, \alpha\downarrow^{\mathrm{m}}_{\mathrm{n}}) \circ (\tau, \beta\downarrow^{\mathrm{n}}_{\mathrm{o}}) = (\sigma \circ \tau, (\beta \circ \alpha)\downarrow^{\mathrm{m}}_{\mathrm{o}})$

Modal context extension:

We consider the non-modal rule CTX-EXT and its introduction, elimination and computation rules as a special case of CTX-MODEXT for $p = q$ and $\mu = 1$.

CTX-MODEXT

$$\frac{q \mid \Gamma\ \mathsf{ctx} \quad \mu : p \to q \qquad p \mid \Gamma, \mathbb{a}^{\mathrm{m}}_{\mu} \vdash T\ \mathsf{type}}{q \mid \Gamma, \mu \vartriangleright x :^{\mathrm{m}} T\ \mathsf{ctx}}$$

CTX-MODEXT:INTRO

$$\frac{q \mid \sigma : \Gamma \to \Delta \qquad \mu : p \to q \qquad p \mid \Gamma, \mathbb{a}^{\mathrm{m}}_{\mu} \vdash t : T[\sigma, \downarrow^{\mathrm{m}}_{\mathrm{m}}]}{q \mid (\sigma, t/x\downarrow_{\mathrm{m}}) : \Gamma \to (\Delta, \mu \vartriangleright x :^{\mathrm{m}} T)}$$

where $\tau = ((x/\oslash) \circ \tau, (x\downarrow_{\mathrm{m}})[\tau, \downarrow^{\mathrm{m}}_{\mathrm{m}}]/x\downarrow_{\mathrm{m}})$

$(\sigma, t/x\downarrow_{\mathrm{m}}) \circ \rho = (\sigma \circ \rho, t[\rho, \downarrow^{\mathrm{m}}_{\mathrm{m}}]/x\downarrow_{\mathrm{m}})$

CTX-MODEXT:WKN

$$\frac{q \mid \Gamma\ \mathsf{ctx} \qquad \mu : p \to q \qquad p \mid \Gamma, \mathbb{a}^{\mathrm{m}}_{\mu} \vdash T\ \mathsf{type}}{q \mid (x/\oslash) : (\Gamma, \mu \vartriangleright x :^{\mathrm{m}} T) \to \Gamma}$$

where $(x/\oslash) \circ (\sigma, t/x\downarrow_{\mathrm{m}}) = \sigma$

CTX-MODEXT:VAR

$$\frac{q \mid \Gamma\ \mathsf{ctx} \qquad \mu : p \to q \qquad p \mid \Gamma, \mathbb{a}^{\mathrm{m}}_{\mu} \vdash T\ \mathsf{type}}{q \mid \Gamma, \mu \vartriangleright x :^{\mathrm{m}} T, \mathbb{a}^{\mathrm{m}'}_{\mu} \vdash x\downarrow_{\mathrm{m}'} : T[(x/\oslash), \downarrow^{\mathrm{m}}_{\mathrm{m}'}]}$$

where $x\downarrow_{\mathrm{m}'}[\sigma, t/x\downarrow_{\mathrm{m}'''}, \downarrow^{\mathrm{m}'}_{\mathrm{m}''}] = t[1, \downarrow^{\mathrm{m}'''}_{\mathrm{m}''}]$

Figure 5: Structural rules of MTT [GKNB21][Nuy20a, fig. 5.5].

substituting $\mathsf{mod}^{\mathrm{m}}_{\mu}\, t$ instead. The **variable** rule does not produce $x : \langle \mu \mid^{\mathrm{m}} T \rangle$ but instead uses transposition to move the modality μ to the left in the form of a lock. As such, it can be seen as implicitly involving a co-unit.

By analogy with CTX-EXT:VAR:LOOKUP, we would like a more general unofficial variable 'rule' that allows accessing a variable x that is buried under a general telescope Δ rather than a single lock. By LOCK:FMAP, we can weaken under locks (which, like ordinary weakening, we will leave notationally implicit), so we can easily remove all variables from Δ and then apply strict functoriality of LOCK to fuse the remaining locks, obtaining a single lock $\mathbb{a}^{\mathsf{ticks}(\Delta)}_{\mathsf{locks}(\Delta)}$, where the modality $\mathsf{locks}(\Delta)$ is defined as follows (ignore the definition of $\mathsf{ticks}(\Delta)$ for now):

$$\mathsf{locks}(\cdot) = 1 \quad \mathsf{locks}(\Delta, \mathbb{a}^{\mathrm{m}}_{\mu}) = \mathsf{locks}(\Delta) \circ \mu \quad \mathsf{locks}(\Delta, \mu \vartriangleright x :^{\mathrm{m}} T) = \mathsf{locks}(\Delta)$$
$$\mathsf{ticks}(\cdot) = \bullet \quad \mathsf{ticks}(\Delta, \mathbb{a}^{\mathrm{m}}_{\mu}) = \mathsf{ticks}(\Delta)\mathbf{m} \quad \mathsf{ticks}(\Delta, \mu \vartriangleright x :^{\mathrm{m}} T) = \mathsf{ticks}(\Delta)$$

Then the only remaining thing we need is a 2-cell $\alpha : \mu \Rightarrow \mathsf{locks}(\Delta)$. This leads to the following admissible variable 'rule'

CTX-MODEXT:VAR:LOOKUP

$$\frac{q \mid \Gamma\ \mathsf{ctx} \qquad \mu : p \to q \qquad p \mid \Gamma, \mathbb{a}^{\mathrm{m}}_{\mu} \vdash T\ \mathsf{type} \qquad \alpha : \mu \Rightarrow \mathsf{locks}(\Delta)}{q \mid \Gamma, \mu \vartriangleright x :^{\mathrm{m}} T, \Delta \vdash x\,\alpha\downarrow_{\mathsf{ticks}(\Delta)} : T[\mathrm{id}_{\Gamma}, \alpha\downarrow^{\mathrm{m}}_{\mathsf{ticks}(\Delta)}]}$$

$$\textsc{ModPi}\quad \frac{p \mid \Gamma, \mathbb{m}_{\mu}^{\mathrm{m}} \vdash A\ \mathsf{type}_\ell \quad \mu : p \to q \qquad q \mid \Gamma, \mu \vartriangleright x :^{\mathrm{m}} A \vdash B\ \mathsf{type}_\ell}{q \mid \Gamma \vdash (\mu \vartriangleright x :^{\mathrm{m}} A) \to B\ \mathsf{type}_\ell}$$

$$\textsc{ModPi:Intro}\quad \frac{\mu : p \to q \qquad q \mid \Gamma, \mu \vartriangleright x :^{\mathrm{m}} A \vdash b : B}{q \mid \Gamma \vdash \lambda(\mu \vartriangleright x).b : (\mu \vartriangleright x :^{\mathrm{m}} A) \to B}\ \text{where } \lambda(\mu \vartriangleright x).(f \cdot^{\mathrm{m}} x{\downarrow_{\mathrm{m}}}) = f$$

$$\textsc{ModPi:Elim}\quad \frac{q \mid \Gamma \vdash f : (\mu \vartriangleright x :^{\mathrm{m}} A) \to B \qquad p \mid \Gamma, \mathbb{m}_{\mu}^{\mathrm{m}} \vdash a : A \quad \mu : p \to q}{q \mid \Gamma \vdash f \cdot^{\mathrm{m}} a : B[a/x]}\ \text{where } (\lambda(\mu \vartriangleright x).b) \cdot^{\mathrm{m}} a = b[a/x{\downarrow_{\mathrm{m}}}]$$

Figure 6: Typing rules for MTT's modal Π-types [GKNB21][Nuy20a, fig. 5.7] (of which non-modal Π-types are a special case for $p = q$ and $\mu = 1$).

where $x\,\alpha{\downarrow_{\mathsf{ticks}(\Delta)}}$ is defined as $x{\downarrow_{\mathrm{m}'}}[\mathrm{id}_{(\Gamma,\mu\vartriangleright x:^{\mathrm{m}}T)}, \alpha{\downarrow^{\mathrm{m}'}_{\mathsf{ticks}(\Delta)}}]$, leaving the necessary weakenings under locks implicit. Substitution is then given by

$$\begin{aligned} x\,\alpha{\downarrow_{\mathsf{ticks}(\Delta)}}[\mathrm{id}_\Gamma, a/x{\downarrow_{\mathrm{m}}}, \mathrm{id}_\Delta] &= x{\downarrow_{\mathrm{m}'}}[\mathrm{id}_{(\Gamma,\mu\vartriangleright x:^{\mathrm{m}}T)}, \alpha{\downarrow^{\mathrm{m}'}_{\mathsf{ticks}(\Delta)}}][\mathrm{id}_\Gamma, a/x{\downarrow_{\mathrm{m}}}, \mathrm{id}_\Delta] \\ &= x{\downarrow_{\mathrm{m}'}}[\mathrm{id}_\Gamma, a/x{\downarrow_{\mathrm{m}}}, {\downarrow^{\mathrm{m}'}_{\mathrm{m}'}}][\mathrm{id}_\Gamma, \alpha{\downarrow^{\mathrm{m}'}_{\mathsf{ticks}(\Delta)}}] \\ &= a[\mathrm{id}_\Gamma, {\downarrow^{\mathrm{m}}_{\mathrm{m}'}}][\mathrm{id}_\Gamma, \alpha{\downarrow^{\mathrm{m}'}_{\mathsf{ticks}(\Delta)}}] = a[\mathrm{id}_\Gamma, \alpha{\downarrow^{\mathrm{m}}_{\mathsf{ticks}(\Delta)}}]. \end{aligned}$$

3.3.4. *Modal types, part 2.* **Modal elimination** (WDRA:ELIM, fig. 4) uses a let-syntax to turn the modal type into a judgemental annotation on a variable.

3.3.5. *Modal function types.* Modal **function type** formation and introduction (fig. 6) are by simple abstraction. Modal function application checks the argument in a locked context, just like modal variable substitution does.

3.4. **Ticks: 2-cell covariables.** (Readers may choose to skip this section and ignore ticks altogether, perhaps coming back when in need of a stronger handle on locks and 2-cell substitutions.) Just as de Bruijn indices are a perfectly valid and unambiguous way to deal with variables but unsuitable for human communication, some humans may find that our current notations for locks and 2-cell substitutions are unreadable. This assessment is likely to grow stronger as said humans move from passively reading a text about MTT to actively carrying out their own pen-and-paper type and program manipulations. The manipulations needed in the remainder of the current paper can get relatively complex and for this reason, we choose to unofficially *name* every lock in the context. We shall call these names *ticks* using terminology from Bahr et al. [BGM17] whose 'tick of the clock' is essentially a named lock for the later modality.

Whereas the redundancy introduced by using variable names is heavily relied on when we leave weakenings implicit, we will only ever so slightly rely on the redundancy introduced by using ticks. The main bonus, other than increased readability of the system as it stands, is that we can now denote the obvious substitution $(\Gamma, \mathbb{m}_{\nu}^{\mathrm{n}}, \Delta) \to (\Gamma, \mathbb{m}_{\mu}^{\mathrm{m}}, \Delta[\mathrm{id}_\Gamma, \alpha{\downarrow^{\mathrm{n}}_{\mathrm{m}}}])$ arising from a 2-cell $\alpha : \mu \Rightarrow \nu$ as $\alpha{\downarrow^{\mathrm{n}}_{\mathrm{m}}}$ instead of $(\mathrm{id}_\Gamma, \alpha{\downarrow^{\mathrm{n}}_{\mathrm{m}}}, \mathrm{id}_\Delta)$. This would rarely be ambiguous

in practice in the first place (it is when $\mu = \nu$ and Δ contains another such lock) but is now truly safe.

Whereas a variable name is a placeholder for another program to be *pre*composed, a tick is a placeholder for another 2-cell to be *post*composed. In this sense, ticks can be seen as 2-cell covariables. Thus, it may seem silly that our notation cleanly separates ticks from 2-cells, and we could write $\mathrm{n} \circ \alpha / \mathrm{m}$ instead of $\alpha{\downarrow}^{\mathrm{m}}_{\mathrm{n}}$. While mathematically more elegant, we found this to be completely unreadable in practice. Instead, taking inspiration from the Einstein summation convention, we have opted for a discipline where superscript ticks are binders and subscript ticks are usages, although this is inverted in the notation $t/x\downarrow_{\mathrm{m}}$ where $x\downarrow_{\mathrm{m}}$ is thought of as a pattern and thus m is a binder. A substitution $t[\alpha{\downarrow}^{\mathrm{m}}_{\mathrm{n}}]$ uses n and binds m in t.

A complication is functoriality of LOCK, which requires the set of ticks to be a monoid (with unit $\bullet$ and multiplication denoted by juxtaposition). This raises all sorts of interesting questions, but with the argument that our usage of ticks is mainly for ease of communication and almost completely redundant, we choose to shrug these off.[9]

It is worth noting that there are no weakening, exchange and contraction rules for locks. As such, ticks should be used *in order* and *exactly once*, with the understanding that many operations are 'additive' in the linear logic sense, e.g. in a pair (a, b), both components should consume all ticks (in order and exactly once), whereas unit can consume any list of ticks. A variable lookup $x\,\alpha{\downarrow}_{\mathsf{ticks}(\Delta)}$ implicitly consumes the ticks to the left of x in the context.

When rereading section 3.3, the usage and manipulation of ticks should be fairly self-evident.

3.5. **Lockless notation.** (Readers may choose to skip this section and ignore the lockless notation altogether for now or forever. It suffices to know that we will sometimes assign a semantically motivated name κ to the operation $🔒_\mu$, writing $\kappa \dashv \mu$, and similar for 2-cells $\omega \dashv \alpha$.)

In MTT, we can on one hand apply a modality $\mu : p \to q$ to a type T to obtain a modal type $\langle \mu \mid^{\mathrm{m}} T \rangle$, and on the other hand we can apply its left adjoint $🔒_\mu$ to a context Γ to obtain $\Gamma, 🔒^{\mathrm{m}}_\mu$. Type-theoretically, it is only sensible that the lock $🔒_\mu$ mentions μ for μ is a premise of the LOCK rule. From a semantic/categorical viewpoint however, the indirection of mentioning μ when you are actually talking about some left adjoint functor $K = [\![🔒_\mu]\!]$ to $M = [\![\mu]\!]$ can really get in the way of understanding what is going on.

For example, in section 6, we will come up with a modality $\wr\, u$ for the transpension, and then the introduction rule WDRA:INTRO will take the form

$$\frac{p \mid \Delta, 🔒^{\mathrm{t}}_{\wr\, u} \vdash t : T}{q \mid \Delta \vdash \mathsf{mod}^{\mathrm{t}}_{\wr\, u}\, t : \langle \wr\, u \mid^{\mathrm{t}} T \rangle}$$

for certain modes p and q, which is quite reminiscent of the introduction rule of the transpension type in the naïve system from section 2 *if you bear in mind* that $[\![🔒_{\wr\, u}]\!]$ is essentially $\forall\, u$.

Instead of littering this paper with remarks of the form 'recall that $[\![🔒_\mu]\!] = \ldots$', we will *supplement* the MTT notation used hitherto with an alternative notation that is more sensible categorically. Concretely, we will assign to every modality $\mu : p \to q$ a *left name*

[9] An actual implementation would likely feature tick-splitting annotations such as $\mathsf{let}\, 🔒^{\mathrm{m}}_\mu 🔒^{\mathrm{n}}_\nu = 🔒^{\mathrm{o}}_{\mu\circ\nu}\, \mathsf{in}\, \ldots$.

$\kappa : q \to p$ (writing this succinctly as $\kappa \dashv \mu : p \to q$) and to every 2-cell $\alpha : \mu \Rightarrow \mu'$ a left name $\omega : \kappa' \Rightarrow \kappa$ (writing succinctly $\kappa \Leftarrow \kappa' : \omega \dashv \alpha : \mu \Rightarrow \mu'$). Of course if $\kappa' \dashv \mu'$ and $\kappa \dashv \mu$, then the composite will be $\kappa \circ \kappa' \dashv \mu' \circ \mu$. If a modality μ has a left adjoint *modality* ν, then we will always use ν as the left name of μ, and similar for 2-cells.

The lockless notation for MTT is derived from the original notation by changing the notation for:

- locking contexts (LOCK), writing $\kappa\,^{\mathrm{k}}\,(\Gamma)$ instead of $\Gamma, \mathbf{🔒}_{\mu}^{\mathrm{m}}$,
  - (Readers interested in ticks *and* in the lockless notation may still choose to ignore ticks *in* the lockless notation.) Ticks' names are now (as a good practice) inspired by the *left* name of a modality and correspondingly, they compose the other way around: $\Gamma, \mathbf{🔒}_{\mu'}^{\mathrm{m}'}, \mathbf{🔒}_{\mu}^{\mathrm{m}} = \Gamma, \mathbf{🔒}_{\mu' \circ \mu}^{\mathrm{m}' \mathrm{m}}$ becomes $\kappa\,^{\mathrm{k}}\left(\kappa'\,^{\mathrm{k}'}\,(\Gamma)\right) = \kappa \circ \kappa'\,^{\mathrm{k}\mathrm{k}'}\,(\Gamma)$. Of course, we cannot change the order of binders, so the order in which ticks need to be used is, from left to right: first all bound ticks in reverse order, then all ticks from the context in order.
- 2-cell substitutions (LOCK:FMAP), writing $\omega{\uparrow}^{\mathrm{k}}_{\mathrm{k}'}(\sigma)$ instead of $(\sigma, \alpha{\downarrow}^{\mathrm{m}}_{\mathrm{m}'})$, and abbreviating $\omega{\uparrow}^{\mathrm{k}}_{\mathrm{k}'}(\mathrm{id})$ to $\omega{\uparrow}^{\mathrm{k}}_{\mathrm{k}'}$,
  - Note that in lockless notation, ticks are not covariables for 2-cells (i.e. placeholders for 2-cells to be postcomposed), but variables for left names of 2-cells (i.e. placeholders for left names to be precomposed). In symbols, whereas we could read $\alpha{\downarrow}^{\mathrm{m}}_{\mathrm{m}'}$ as $\mathrm{m}' \circ \alpha / \mathrm{m}$, we should read $\omega{\uparrow}^{\mathrm{k}}_{\mathrm{k}'}$ as $\omega \circ \mathrm{k}' / \mathrm{k}$.
- variables (CTX-MODEXT:VAR:LOOKUP), writing $x\,\omega{\uparrow}_{\mathrm{k}}$ instead of $x\,\alpha{\downarrow}_{\mathrm{m}}$,
- variable substitution (CTX-MODEXT:INTRO), writing $(\sigma, t/x{\uparrow}_{\mathrm{k}})$ instead of $(\sigma, t/x{\downarrow}_{\mathrm{m}})$,
- for specific modalities, we may assign a name to the modal constructor (WDRA:INTRO), e.g. the constructor for $◊\, u$ will be called mer_u (for *meridian*) instead of $\mathsf{mod}^{\mathrm{a}}_{◊\, u}$.

Then the modal introduction rule for the transpension modality becomes:

$$\frac{p \mid \forall\, u\,^{\mathrm{a}}\,(\Delta) \vdash t : T}{q \mid \Delta \vdash \mathsf{mer}^{\mathrm{a}}_{u}\, t : \langle ◊\, u \mid^{\mathrm{a}} T \rangle}.$$

3.6. **Results.** We highlight some results about MTT that are relevant in the current paper.

Proposition 3.1. *We have* $\langle 1 \mid^{\bullet} A \rangle \cong A$ *and* $\langle \nu \circ \mu \mid^{\mathrm{nm}} A \rangle \cong \langle \nu \mid^{\mathrm{n}} \langle \mu \mid^{\mathrm{m}} A \rangle \rangle$.

Proposition 3.2. *For any 2-cell* $\alpha : \mu \Rightarrow \nu$*, we have* $\langle \mu \mid^{\mathrm{m}} A \rangle \to \langle \nu \mid^{\mathrm{n}} A[\alpha{\downarrow}^{\mathrm{m}}_{\mathrm{n}}] \rangle$.

Proposition 3.3 (Projection)**.** *If* $\kappa \dashv \mu$ *internal to the mode theory, then there is a function* $\mathsf{prmod}_{\mu} : (\kappa \vdash^{\mathrm{k}} \langle \mu \mid^{\mathrm{m}} A \rangle) \to A[\varepsilon{\downarrow}^{\mathrm{km}}_{\bullet}]$*, satisfying a* β*- and (thanks to extensionality) an* η*-law:*

$$\mathsf{prmod}_{\mu} \cdot^{\mathrm{k}} (\mathsf{mod}^{\mathrm{m}}_{\mu}\, a) = a[\varepsilon{\downarrow}^{\mathrm{km}}_{\bullet}], \qquad \hat{a} = \mathsf{mod}^{\mathrm{m}}_{\mu}\, (\mathsf{prmod}_{\mu} \cdot^{\mathrm{k}} (\hat{a}[\eta{\downarrow}^{\bullet}_{\mathrm{mk}}])).$$

Combined with these rules, prmod_{μ} *is equally expressive as the* let*-eliminator for* $\langle \mu \mid \sqcup \rangle$.

In lockless notation, we will simply write ε instead of prmod_{μ}.

Proposition 3.4 (Internal transposition)**.** *If $\kappa \dashv \mu$ internal to the mode theory, with unit $\eta : 1 \Rightarrow \mu \circ \kappa$, then there is an isomorphism of contexts (in lockful and lockless notation)*

$$\sigma = (x\,\eta{\downarrow_{\mathrm{mk}}}/y{\downarrow_{\mathrm{k}}}) : (\Gamma, x : A, \textstyle\bigcirc_\mu^{\mathrm{m}}) \cong (\Gamma, \bigcirc_\mu^{\mathrm{m}}, \kappa \mathbin{\triangleright} y :^{\mathrm{k}} A[\eta{\downarrow^{\bullet}_{\mathrm{mk}}}])$$
$$\sigma = (x\,\varepsilon'{\uparrow_{\mathrm{zk}}}/y{\uparrow_{\mathrm{z}}}) : \kappa\,{}^{\mathrm{k}}(\Gamma, x : A) \cong (\kappa\,{}^{\mathrm{k}}(\Gamma), \kappa \mathbin{\triangleright} y :^{\mathrm{z}} A[\varepsilon'{\uparrow^{\bullet}_{\mathrm{zk}}}])$$
$$\text{where } \zeta \dashv \kappa, \qquad \kappa \dashv \mu,$$
$$1 \Leftarrow \zeta \circ \kappa : \varepsilon' \dashv \eta : 1 \Rightarrow \mu \circ \kappa$$

with inverse

$$\sigma^{-1} = (\mathrm{id}_\Gamma, \eta{\downarrow^{\bullet}_{\mathrm{mk}}}, y{\downarrow_{\mathrm{k}}}/x{\downarrow_{\bullet}}, {\downarrow^{\mathrm{m}}_{\mathrm{m}'}}) \circ (\mathrm{id}_{(\Gamma, \bigcirc_\mu^{\mathrm{m}}, y)}, \varepsilon{\downarrow^{\mathrm{km}'}_{\bullet}})$$
$$\sigma^{-1} = {\uparrow^{\mathrm{k}}_{\mathrm{k}'}}(\varepsilon'{\uparrow^{\bullet}_{\mathrm{zk}}}(\mathrm{id}_\Gamma), y{\uparrow_{\mathrm{z}}}/x{\uparrow_{\bullet}}) \circ \eta'{\uparrow^{\mathrm{k}'\mathrm{z}}_{\bullet}}(\mathrm{id}_{(\kappa\,{}^{\mathrm{k}}(\Gamma), y)})$$
$$\text{where } \kappa \circ \zeta \Leftarrow 1 : \eta' \dashv \varepsilon : \kappa \circ \mu \Rightarrow 1.$$

Correspondingly, given B in the latter context, there is an isomorphism of types

$$((x : A) \to \langle \mu \mid^{\mathrm{m}} B[\sigma] \rangle) \cong \left\langle \mu \mid^{\mathrm{m}} (\kappa \mathbin{\triangleright} y :^{\mathrm{k}} A[\eta{\downarrow^{\bullet}_{\mathrm{mk}}}]) \to B \right\rangle$$
$$\left((x : A) \to \left\langle \mu \mid^{\mathrm{k}} B[\sigma] \right\rangle\right) \cong \left\langle \mu \mid^{\mathrm{k}} (\kappa \mathbin{\triangleright} y :^{\mathrm{z}} A[\varepsilon'{\uparrow^{\bullet}_{\mathrm{zk}}}]) \to B \right\rangle.$$

4. (Co)quantifiers as Modalities

As mentioned in section 1.3, the transpension $⟆\, u$ and its adjoints weakening $\Omega\, u$ and universal quantification $\Pi\, u$ ($\Omega\, u \dashv \Pi\, u \dashv ⟆\, u$) or, more generally in potentially non-cartesian systems, fresh weakening $⫤ u$ and substructural universal quantification $\forall\, u$ ($⫤ u \dashv \forall\, u \dashv ⟆\, u$) will be internalized as MTT modalities. The further left adjoints $\Sigma\, u$ (cartesian) or $\exists\, u$ (potentially non-cartesian) cannot be internalized because every internal MTT modality needs to have a further semantic left adjoint.

Notably, the aforementioned modalities all bind or depend on a variable, a phenomenon which is not supported by MTT. We shall address this issue in the current section by grouping shape variables such as $u : \mathbb{U}$ in a **shape context** which is not considered part of the type-theoretic context but instead serves as the *mode* of the judgement.

We assume that there are no prior modalities, i.e. that the type system to which we wish to add a transpension type is non-modal in the sense that it has a single mode and only the identity modality. We assume that this single prior mode is modelled by the presheaf category $\mathrm{Psh}(\mathcal{W})$. Prior modalities and in particular their commutation with the modalities mentioned above, are considered in the technical report [Nuy20b].

4.1. **Shape contexts.** Assume we have in the prior system a context Ξ modelled by a presheaf over $\mathcal{W}$. Then the presheaves $\mathrm{Psh}(\mathcal{W}/\Xi)$ over the category of elements $\mathcal{W}/\Xi$ of the presheaf Ξ are also a model of dependent type theory. Denoting the judgements of the latter system with a prefix $\Xi \mid$, it happens to be the case that judgements $\Xi \mid \Gamma \vdash J$ (i.e. $\Gamma \vdash J$ in $\mathrm{Psh}(\mathcal{W}/\Xi)$) have precisely the same meaning as judgements $\Xi.\Gamma \vdash J$ in $\mathrm{Psh}(\mathcal{W})$ (for a suitable but straightforward translation of J). Thus, we will group together all shape variables (variables for which we want a transpension type) in a **shape context** Ξ in front of the typing context. Our judgements will then take the form $\Xi \mid \Gamma \vdash J$. Modal techniques

will be used to signal what part of the context Γ are fresh for a shape variable $u : \mathbb{U}$, as this can then no longer be signalled by the position of $u : \mathbb{U}$ in the context.

All of this allows us to frame the shape context Ξ as the *mode* of the judgement, as it determines the category $\mathrm{Psh}(\mathcal{W}/\Xi)$ in which the judgement is modelled.

4.2. **Mode theory.** For simplicity, we take a highly general mode theory and will then only be able to say interesting things about specific modes, modalities and 2-cells. In practice, and especially in implementations, one will want to select a more syntactic subtheory right away.

As **modes**, we take the set of all small presheaves over $\mathcal{W}$, which we think of as **shape contexts**. The mode Ξ is modelled in $\mathrm{Psh}(\mathcal{W}/\Xi)$. As we will see later on, the available shapes must in some sense already be present in the base category, so that a context consisting purely of shapes will in general be representable. As such, we could alternatively restrict the set of modes to *representable* presheaves over $\mathcal{W}$, which via the Yoneda-embedding are just the objects of $\mathcal{W}$. This is perfectly possible and would (again by inserting the Yoneda-embedding) require virtually no changes to our approach, although a number of intermediate results in the technical report [Nuy20b] would become unnecessary. However, the current approach is strictly more general, allows us to speak about boundaries (definitions 6.5 and 6.22) in the shape context, and did not require any compromise in the strength of our results.

As **modalities** $\mu : \Xi_1 \to \Xi_2$, we take all functors $[\![🔒_\mu]\!] : \mathrm{Psh}(\mathcal{W}/\Xi_2) \to \mathrm{Psh}(\mathcal{W}/\Xi_1)$ which have a right adjoint $[\![\mu]\!]$ that is then automatically a weak CwF morphism [Nuy20b] [Nuy20a, thm. 6.4.1] and gives rise to a DRA [BCM+20, lemma 17][Nuy18a].[10]

As **2-cells** $\alpha : \mu \Rightarrow \nu$, we take all natural transformations.

5. (Co)quantifier Modalities for Substitution

In section 4, we explained that we will group shape variables in the shape context which is the mode of the judgement, and we subsequently defined the set of modes to be the set of presheaves over $\mathcal{W}$. As such, we have neglected to provide an actual syntax for shape contexts and instead simply decided to use semantic shape contexts, i.e. presheaves over $\mathcal{W}$.

Similarly, there shall be no syntax for shape variable substitutions $\Xi_1 \vdash \sigma : \Xi_2$ and we shall instead consider all presheaf morphisms $\sigma : \Xi_1 \to \Xi_2$. Whereas shape contexts are internalized as modes, substitutions will be internalized as modalities. However, whereas we defined modes to *be* presheaves over $\mathcal{W}$, modalities $\mu : \Xi_2 \to \Xi_1$ were not defined to be presheaf morphisms $\sigma : \Xi_1 \to \Xi_2$ but rather functors $\mathrm{Psh}(\mathcal{W}/\Xi_2) \to \mathrm{Psh}(\mathcal{W}/\Xi_1)$. As such, a presheaf morphism *is* not a modality but it gives rise to a pair of modalities:[11]

Theorem 5.1. *Any presheaf morphism $\sigma : \Xi_1 \to \Xi_2$ gives rise to a triple of adjoint functors*

$$\Sigma^{\sigma|} \dashv \Omega^{\sigma|} \dashv \Pi^{\sigma|},$$

[10] Note that a designated right adjoint can be retrieved from the left adjoint without the axiom of choice [Nuy20b, §2.3.6].

[11] The usage of presheaves as modes has the unfortunate consequence that the notation for the set of presheaf morphisms is the same as the one for the set of modalities. However modalities shall be typeset in colour and, as we just explained, a presheaf morphism is *never* a modality. Note in particular that $\Omega\,\sqcup$ turns the arrow around: a presheaf morphism (shape substitution) $\sigma : \Xi_1 \to \Xi_2$ gives rise to a substitution modality $\Omega\,\sigma : \Xi_2 \to \Xi_1$ sending types T in shape context Ξ_2 to types $\langle \Omega\,\sigma \mid^\circ T \rangle$ in shape context Ξ_1.

$$\Sigma^{\sigma|}, \Pi^{\sigma|} : \mathrm{Psh}(\mathcal{W}/\Xi_1) \to \mathrm{Psh}(\mathcal{W}/\Xi_2) \qquad \Omega^{\sigma|} : \mathrm{Psh}(\mathcal{W}/\Xi_2) \to \mathrm{Psh}(\mathcal{W}/\Xi_1)$$

the latter two of which can be internalized as modalities $\Omega\,\sigma \dashv \Pi\,\sigma$ *with*

$$[\![🔒_{\Omega\,\sigma}]\!] = \Sigma^{\sigma|}, \qquad [\![\Omega\,\sigma]\!] = \Omega^{\sigma|}, \qquad [\![🔒_{\Pi\,\sigma}]\!] = \Omega^{\sigma|}, \qquad [\![\Pi\,\sigma]\!] = \Pi^{\sigma|}.$$

We denote the units and co-units as

$$\mathsf{copy}^{\sigma|} : 1 \to \Omega^{\sigma|} \circ \Sigma^{\sigma|} \qquad \mathsf{drop}^{\sigma|} : \Sigma^{\sigma|} \circ \Omega^{\sigma|} \to 1$$

$$\mathsf{const}^{\sigma|} : 1 \to \Pi^{\sigma|} \circ \Omega^{\sigma|} \qquad \mathsf{app}^{\sigma|} : \Omega^{\sigma|} \circ \Pi^{\sigma|} \to 1$$

$$\mathsf{const}_{\sigma} : 1 \Rightarrow \Pi\,\sigma \circ \Omega\,\sigma \qquad \mathsf{app}_{\sigma} : \Omega\,\sigma \circ \Pi\,\sigma \Rightarrow 1$$

Under the correspondence of semantic contexts $\Xi \mid \Gamma\,\mathsf{ctx}$ *(i.e. presheaves over* $\mathcal{W}/\Xi$*) with semantic types* $\Xi \vdash \Gamma\,\mathsf{type}$*, the functor* $\Omega^{\sigma|}$ *is exactly the semantics of ordinary type substitution in the standard presheaf model* [Hof97] *and hence, if* σ *is a weakening substitution, then* $\Sigma^{\sigma|}$ *and* $\Pi^{\sigma|}$ *are naturally isomorphic to the semantics of ordinary* Σ*- and* Π*-types.*

The functor $\Omega^{\sqcup|}$ *and the modality* $\Pi\,\sqcup$ *are strictly functorial (they respect identity and composition of presheaf morphisms on the nose) whereas the functors* $\Sigma^{\sqcup|}$, $\Pi^{\sqcup|}$ *and the modality* $\Omega\,\sqcup$ *are pseudofunctorial*[12].

Proof. The morphism σ gives rise to a functor $\Sigma^{/\sigma} : \mathcal{W}/\Xi_1 \to \mathcal{W}/\Xi_2 : (W, \psi) \to (W, \sigma\psi)$ and hence via left Kan extension, precomposition and right Kan extension [Sta19] to a triple of adjoint functors $\Sigma^{\sigma|} \dashv \Omega^{\sigma|} \dashv \Pi^{\sigma|}$ between the presheaf categories. The claim about type substitution follows from unfolding the definitions and the claims about Σ- and Π-types then follow from uniqueness of adjoints.

Strict functoriality of $\Omega^{\sqcup|}$ follows immediately from the construction. Strict functoriality of $\Pi\,\sqcup$ then follows from the fact that a modality μ is fully defined by the semantic left adjoint $[\![🔒_{\mu}]\!]$. Pseudofunctoriality of the others follows by uniqueness of the adjoint. □

Remark 5.2. It should be noted that, since $🔒_{\Omega\,\sigma} = \Sigma\,\sigma$ does not preserve the empty context, the operation $🔒_{\Pi\,\sigma} = \Omega\,\sigma$ applicable to contexts has a different meaning than the modality $\Omega\,\sigma$ applicable to types. When applied to a context, $🔒_{\Pi\,\sigma} = \Omega\,\sigma$ is an actual substitution of the shape context. When applied to a type, $\Omega\,\sigma$ is semantically still a substitution, but between two isomorphic semantic contexts $\Xi_1.\Gamma$ and $\Xi_2.(\Sigma^{\sigma|}\Gamma)$. This is especially clear if $\sigma : \Xi.A \to \Xi$ is a weakening substitution, in which case we are dealing with $\Xi.A.\Gamma$ and $\Xi.(\Sigma A \Gamma)$.

Notation 5.3. We will often denote **shape contexts** as a list of shape variables, e.g. $(u : \mathbb{U}, v : \mathbb{V}, w : \mathbb{W})$. The precise meaning of this will be explained later on; for now bear in mind that this is not syntax but sugar. In principle, there are no shape variables and no syntactic shape contexts; there are only presheaves over $\mathcal{W}$.

We will use a slightly unconventional notation for **substitutions** in order to have them make maximal sense both as a substitution (as in $\Omega\,\sigma$) and as a function domain (as in $\Pi\,\sigma$):

- Every weakened variable (if weakening is available for the given shape) will be declared, e.g. we get a presheaf morphism $(u : \mathbb{U}) : (\Xi, u : \mathbb{U}) \to \Xi$ and hence $\Omega(u : \mathbb{U}) : \Xi \to (\Xi, u : \mathbb{U})$ and $\Pi(u : \mathbb{U}) : (\Xi, u : \mathbb{U}) \to \Xi$. Furthermore, we may omit the shape, writing just $\Omega\,u$ and

[12] However, Gratzer et al. [GKNB20] have a strictification theorem for models of MTT which could be used to strictify $\Omega\,\sqcup$.

$\Pi\, u$. Thus, this is what weakening and shape abstraction look like (in lockful and lockless notation):

$$\frac{\Xi \mid \Gamma, \blacklozenge^{\mathrm{o}}_{\Omega\, u} \vdash t : T}{\Xi, u : \mathbb{U} \mid \Gamma \vdash \mathsf{mod}^{\mathrm{o}}_{\Omega\, u}\, t : \langle \Omega\, u \mid^{\mathrm{o}} T \rangle}, \qquad \frac{\Xi, u : \mathbb{U} \mid \Gamma, \blacklozenge^{\mathrm{p}}_{\Pi\, u} \vdash t : T}{\Xi \mid \Gamma \vdash \mathsf{mod}^{\mathrm{p}}_{\Pi\, u}\, t : \langle \Pi\, u \mid^{\mathrm{p}} T \rangle}.$$

$$\frac{\Xi \mid \Sigma\, u\, {}^{\mathrm{s}}\, (\Gamma) \vdash t : T}{\Xi, u : \mathbb{U} \mid \Gamma \vdash \mathsf{mod}^{\mathrm{s}}_{\Omega\, u}\, t : \langle \Omega\, u \mid^{\mathrm{s}} T \rangle}, \qquad \frac{\Xi, u : \mathbb{U} \mid \Omega\, u\, {}^{\mathrm{o}}\, (\Gamma) \vdash t : T}{\Xi \mid \Gamma \vdash \lambda^{\mathrm{o}}_{u}\, t : \langle \Pi\, u \mid^{\mathrm{o}} T \rangle}.$$

The projection function (proposition 3.3) for $\Pi\, u$ is function application:

$$\mathsf{prmod}_{\Pi\, u} : (\Omega\, u \shortmid^{\mathrm{o}} \langle \Pi\, u \mid^{\mathrm{p}} A \rangle) \to A[\mathsf{app}_u{\downarrow^{\mathrm{op}}_{\bullet}}]$$
$$\mathsf{app}_u : (\Omega\, u \shortmid^{\mathrm{s}} \langle \Pi\, u \mid^{\mathrm{o}} A \rangle) \to A[\mathsf{copy}_u{\uparrow^{\mathrm{os}}_{\bullet}}].$$

The 2-cell $\Omega\, u \circ \Sigma\, u \Leftarrow 1 : \mathsf{copy}_u \dashv \mathsf{app}_u : \Omega\, u \circ \Pi\, u \Rightarrow 1$ signals a contraction of shape variables, namely of the one bound by the Π-modality and the one to which the function is applied.

- When a variable is substituted, we denote this as $u := t$ instead of t/u, e.g. in a cubical type theory we get a presheaf morphism $(i := 0) : \Xi \to (\Xi, i : \mathbb{I})$ and hence $\Omega(i := 0) : (\Xi, i : \mathbb{I}) \to \Xi$ and $\Pi(i := 0) : \Xi \to (\Xi, i : \mathbb{I})$, so we may substitute 0 for i but we may also abstract over the assumption that i is 0:

$$\frac{\Xi, i : \mathbb{I} \mid \Gamma, \blacklozenge^{\mathrm{o}}_{\Omega(i:=0)} \vdash t : T}{\Xi \mid \Gamma \vdash \mathsf{mod}^{\mathrm{o}}_{\Omega(i:=0)}\, t : \langle \Omega(i := 0) \mid^{\mathrm{o}} T \rangle}, \qquad \frac{\Xi \mid \Gamma, \blacklozenge^{\mathrm{p}}_{\Pi(i:=0)} \vdash t : T}{\Xi, i : \mathbb{I} \mid \Gamma \vdash \mathsf{mod}^{\mathrm{p}}_{\Pi(i:=0)}\, t : \langle \Pi(i := 0) \mid^{\mathrm{p}} T \rangle}.$$

$$\frac{\Xi, i : \mathbb{I} \mid \Sigma(i := 0)\, {}^{\mathrm{s}}\, (\Gamma) \vdash t : T}{\Xi \mid \Gamma \vdash \mathsf{mod}^{\mathrm{s}}_{\Omega(i:=0)}\, t : \langle \Omega(i := 0) \mid^{\mathrm{s}} T \rangle}, \qquad \frac{\Xi \mid \Omega(i := 0)\, {}^{\mathrm{o}}\, (\Gamma) \vdash t : T}{\Xi, i : \mathbb{I} \mid \Gamma \vdash \lambda^{\mathrm{o}}_{(i:=0)}\, t : \langle \Pi(i := 0) \mid^{\mathrm{o}} T \rangle}.$$

And apply:

$$\mathsf{prmod}_{\Pi(i:=0)} : (\Omega(i := 0) \shortmid^{\mathrm{o}} \langle \Pi(i := 0) \mid^{\mathrm{p}} A \rangle) \to A[\mathsf{app}_{(i:=0)}{\downarrow^{\mathrm{op}}_{\bullet}}]$$
$$\mathsf{app}_{(i:=0)} : (\Omega(i := 0) \shortmid^{\mathrm{s}} \langle \Pi(i := 0) \mid^{\mathrm{o}} A \rangle) \to A[\mathsf{copy}_{(i:=0)}{\uparrow^{\mathrm{os}}_{\bullet}}].$$

- Finally, if σ involves weakening, then the codomain of the co-unit may be a **variable renaming** that is sugar for the identity, e.g.

$$\mathsf{app}_{(v/u:\mathbb{U})} : \Omega(v : \mathbb{U}) \circ \Pi(u : \mathbb{U}) \Rightarrow \Omega(v : \mathbb{U}, u := v)$$

is exactly the same thing as

$$\mathsf{app}_{(u:\mathbb{U})} : \Omega(u : \mathbb{U}) \circ \Pi(u : \mathbb{U}) \Rightarrow 1.$$

This way, we may adjust $\mathsf{prmod}_{\Pi\, u}$ in order to be able to apply to a *different* variable:

$$\lambda(\Omega\, v \shortmid f :^{\mathrm{o}} \langle \Pi\, u \mid^{\mathrm{p}} A \rangle).\mathsf{mod}^{\bullet}_{\Omega(v,u:=v)}\, (\mathsf{prmod}_{\Pi\, u} \cdot^{\mathrm{o}}_{\Omega\, v} f{\downarrow_{\mathrm{o}}})$$
$$: (\Omega\, v \shortmid^{\mathrm{o}} \langle \Pi\, u \mid^{\mathrm{p}} A \rangle) \to \left\langle \Omega(v, u := v) \mid^{\bullet} A[\mathsf{app}_{(v/u)}{\downarrow^{\mathrm{op}}_{\bullet}}] \right\rangle$$
$$\lambda(\Omega\, v \shortmid f :^{\mathrm{s}} \langle \Pi\, u \mid^{\mathrm{p}} A \rangle).\mathsf{mod}^{\bullet}_{\Omega(v,u:=v)}\, (\mathsf{app}_{(v/u)} \cdot^{\mathrm{s}}_{\Omega\, v} f{\uparrow_{\mathrm{s}}})$$
$$: (\Omega\, v \shortmid^{\mathrm{s}} \langle \Pi\, u \mid^{\mathrm{o}} A \rangle) \to \left\langle \Omega(v, u := v) \mid^{\bullet} A[\mathsf{copy}_{(v/u)}{\uparrow^{\mathrm{os}}_{\bullet}}] \right\rangle$$

We will refer to these operations by the same name $\mathsf{prmod}_{\Pi\, u}$ / $\mathsf{app}_{(v/u)}$.

Again, bear in mind that even the very idea to view presheaf morphisms as shape substitutions is frivolity; the corresponding notations are sugar at best.

Example 5.4. If $\mathbb{U}$ is cartesian, i.e. $(\Xi, u : \mathbb{U})$ is sugar for $\Xi \times \mathbb{U}$, then there is a diagonal substitution $(w : \mathbb{U}, u := w, v := w) : (\Xi, w : \mathbb{U}) \to (\Xi, u : \mathbb{U}, v : \mathbb{U})$. Writing

$$\begin{aligned}\alpha = 1_{\Pi\, u} \star 1_{\Pi\, v} \star \mathsf{const}_{(w,u:=w,v:=w)} &: \Pi\, u \circ \Pi\, v \Rightarrow \\ &\Pi\, u \circ \Pi\, v \circ \Pi(w, u := w, v := w) \circ \Omega(w, u := w, v := w) \\ &= \Pi((u : \mathbb{U}) \circ (v : \mathbb{U}) \circ (w, u := w, v := w)) \circ \Omega(w, u := w, v := w) \\ &= \Pi((u : \mathbb{U}) \circ (w, u := w)) \circ \Omega(w, u := w, v := w) \\ &= \Pi\, w \circ \Omega(w, u := w, v := w),\end{aligned}$$

where the equations use strict functoriality of $\Pi\, \sqcup$ and ordinary calculation of composition of substitutions, this allows us to type the naïvely typed function $\lambda f.\lambda w.f\, w\, w : (\Pi\, u.\Pi\, v.A) \to \Pi\, w.A[w/u, w/v]$ as

$$\langle \Pi(u : \mathbb{U}) \mid^{\mathbb{P}_u} \langle \Pi(v : \mathbb{U}) \mid^{\mathbb{P}_v} A \rangle \rangle \to \left\langle \Pi(w : \mathbb{U}) \mid^{\mathbb{P}_w} \left\langle \Omega(w : \mathbb{U}, u := w, v := w) \mid^{\mathbb{O}} A[\alpha\downarrow^{\mathbb{P}_u \mathbb{P}_v}_{\mathbb{P}_w \mathbb{O}}] \right\rangle \right\rangle.$$

Remark 5.5. The reframing of shape substitutions as a modality, has the annoying consequence that substitution no longer reduces. However, both $\langle \Omega\, \sigma \mid \sqcup \rangle$ and $\mathsf{mod}_{\Omega\, \sigma}$ are semantically an ordinary substitution.[13] Thus, we could add computation rules such as:

$$\begin{aligned} \langle \Omega\, \sigma \mid^{\mathbb{O}} A \times B \rangle &= \langle \Omega\, \sigma \mid^{\mathbb{O}} A \rangle \times \langle \Omega\, \sigma \mid^{\mathbb{O}} B \rangle, & \langle \Omega\, \sigma \mid^{\mathbb{O}} \mathsf{U} \rangle &= \mathsf{U}, \\ \mathsf{mod}^{\mathbb{O}}_{\Omega\, \sigma}\, (a, b) &= (\mathsf{mod}^{\mathbb{O}}_{\Omega\, \sigma}\, a, \mathsf{mod}^{\mathbb{O}}_{\Omega\, \sigma}\, b), & \mathsf{mod}^{\mathbb{O}}_{\Omega\, \sigma}\, \ulcorner A \urcorner &= \ulcorner \langle \Omega\, \sigma \mid^{\mathbb{O}} A \rangle \urcorner. \end{aligned}$$

This is fine in an extensional type system, but would not play well with the β-rule for modal types in an intensional system. Indeed, β-reduction for $\langle \Omega\, \sigma \mid^{\mathbb{O}} A \rangle$ requires a solution to the following problem: when $\hat{a} = \mathsf{mod}^{\mathbb{O}}_{\Omega\, \sigma}\, a$ definitionally, then we need to be able to infer a up to definitional equality from $\hat{a}$. Alternatively, the eliminator for $\langle \Omega\, \sigma \mid^{\mathbb{O}} A \rangle$ should somehow proceed by induction on A, e.g. an element of $\langle \Omega\, \sigma \mid^{\mathbb{O}} A \times B \rangle$ could be eliminated as an element of $\langle \Omega\, \sigma \mid^{\mathbb{O}} A \rangle \times \langle \Omega\, \sigma \mid^{\mathbb{O}} B \rangle$.

Remark 5.6. In type theory, we generally expect admissibility of substitution: given a derivable judgement $\Gamma \vdash J$ and a substitution $\sigma : \Delta \to \Gamma$, we expect derivability of $\Delta \vdash J[\sigma]$, where the operation $\sqcup[\sigma]$ can be applied to any term, type or other object in context and traverses its structure, leaving everything untouched except variables. A good way to guarantee admissibility of substitution is by making sure that every inference rule has a conclusion in a general context Γ and that the context of any premise is obtained by applying a functorial operation to Γ.

There is no such result for shape substitutions. The conclusion of modal inference rules often has a non-general shape context, and the transpension type is in general not even respected by shape substitution [Nuy20b]. However, until we extend our instantiation of MTT with additional rules in sections 8 to 10, we do have a form of the usual result: given a derivable judgement $\Xi \mid \Gamma \vdash J$ and a substitution $\Xi \mid \sigma : \Delta \to \Gamma$, we can derive $\Xi \mid \Delta \vdash J[\sigma]$.

6. Multipliers

In this section, we introduce multipliers as a semantics for shapes, as well as the associated modalities $\lrcorner\, u \dashv \forall\, u \dashv \Diamond\!\!\!\!-\, u$. In section 6.1, we define multipliers and a number of criteria by which we can classify them. In section 6.2, we deal with a technical complication that we dubbed *spookiness*, which shows up especially in models of guarded and nominal type theory.

[13] Not along $\sigma : \Xi_1 \to \Xi_2$, but along $\sigma.\mathsf{copy}^{\sigma|} : \Xi_1.\Gamma \cong \Xi_2.\Sigma^{\sigma|}\Gamma$, which happens to be an isomorphism.

In section 6.3, we discuss an extensive number of examples. In section 6.4, we consider how *semicartesian* multipliers give rise to shape *weakening* modalities that are a special instance of the modalities in section 5. In section 6.5, we discuss how a multiplier and its associated operations lift from acting on base and slice categories to acting on categories of elements of shape contexts. Then in section 6.6, we are finally ready to define the transpension modality and its adjoints and to state the quantification theorem 6.27 that helps to understand them. In section 6.7, we say a bit more on cartesian multipliers. In section 6.8, we briefly list the matters that are not discussed in the current paper but can be found in the technical report [Nuy20b].

6.1. **Shapes and multipliers.** In notation 5.3, we already mentioned that we will denote shape contexts as lists of shape variables $u : \mathbb{U}$, and we hinted at the fact that these shape variables need not satisfy all the usual structural rules (weakening, exchange and contraction). In this section, we will precisely define what is meant by this notation.

We associate to each shape $\mathbb{U}$ a functor $\sqcup \ltimes U : \mathcal{W} \to \mathcal{W}$ which extends by left Kan extension to a functor $\sqcup \ltimes \mathbf{y}U : \mathrm{Psh}(\mathcal{W}) \to \mathrm{Psh}(\mathcal{W})$.[14] Internally, we will use shape variables to increase human readability and to reduce the heaviness of notation for shape substitutions, writing $(\Xi, u : \mathbb{U})$ for the shape context $\Xi \ltimes \mathbf{y}U$. However, since we have defined shape contexts as presheaves over $\mathcal{W}$, shape variables are just syntactic sugar and shapes are just symbols that we use to refer to their associated multipliers.

Of course, if we model shape context extension with $u : \mathbb{U}$ by an *arbitrary* functor, then we will not be able to prove many results. Depending on the properties of the functor, the variable u will behave like an affine one or like a cartesian one (or perhaps neither) and the Φ-combinator [Mou16, BCM15] will or will not be sound for $\mathbb{U}$. For this reason, we introduce some criteria that help us classify shapes:

Definition 6.1. Assume $\mathcal{W}$ has a terminal object $\top$. A **multiplier** for an object U is an endofunctor[15] $\sqcup \ltimes U : \mathcal{W} \to \mathcal{W}$ such that $\top \ltimes U \cong U$. This gives us a natural second projection $\pi_2 : (\sqcup \ltimes U) \to U$. We define the **fresh weakening functor** to the slice category as $\exists_U : \mathcal{W} \to \mathcal{W}/U : W \mapsto (W \ltimes U, \pi_2)$. We say that a multiplier (as well as its shape) is:

- **Semicartesian** if it is copointed, i.e. has a first projection $\pi_1 : (\sqcup \ltimes U) \to \mathrm{Id}$,
- **Cartesian** if it is naturally isomorphic to the cartesian product with U,
- **Cancellative** if $\exists_U$ is faithful,
- **Affine** if $\exists_U$ is full,
- **Connection-free** if $\exists_U$ is essentially surjective on objects (V, ψ) such that ψ is dimensionally split (definition 6.5),
- **Quantifiable** if $\exists_U$ has a left adjoint $\exists_U : \mathcal{W}/U \to \mathcal{W}$.[16]

Variables of shape $\mathbb{U}$ admit weakening if and only if the multiplier is semicartesian, and exchange and contraction if (but not only if) it is cartesian. Weakening, exchange and contraction are all shape substitutions, which will be internalized as modalities.

A non-trivial multiplier cannot be both affine and cartesian:

Proposition 6.2. *If a multiplier is affine and cartesian, then it is naturally isomorphic to the identity functor.* [Nuy20b]

[14] Both $\sqcup \ltimes U$ and $\sqcup \ltimes \mathbf{y}U$ are to be regarded as single-character symbols, i.e. $\ltimes$ in itself is meaningless.

[15] In the technical report [Nuy20b], we generalize beyond endofunctors.

[16] A functor $\sqcup \ltimes U$ with this property is called a *parametric* or *local right adjoint*.

Proof. Consider the following diagram:

$$\begin{array}{ccc} \top \times U & \xrightarrow{(\mathrm{id},\pi_2)} & (\top \times U) \times U \\ & {\scriptstyle \pi_2} \searrow \quad \swarrow {\scriptstyle \pi_2} & \\ & U & \end{array} \tag{6.1}$$

This is a morphism of slices $(\mathrm{id}, \pi_2) : \exists_U \top \to \exists_U(\top \times U)$ and thus, by affinity, of the form $\exists_U \upsilon$ for some $\upsilon : \top \to \top \times U$. This means in particular that

$$\mathrm{id}_{\top \times U} = \pi_1 \circ (\mathrm{id}_{\top \times U}, \pi_2) = \pi_1 \circ (\upsilon \times U) = \upsilon \circ \pi_1 : \top \times U \to \top \times U, \tag{6.2}$$

so the identity on $\top \times U$ factors over $\top$. Then $\top \times U \cong U$ is a terminal object and hence $\sqcup \times U \cong \mathrm{id}$. □

6.2. **Dimensional splitness, spookiness and boundaries.** Before we move on to a list of examples, we owe the reader a definition for dimensional splitness (although the impatient reader may first read the example section 6.3, ignoring connection-freedom, spookiness and boundaries).

In most popular base categories, namely all but the *spooky* ones, we could have gotten away with saying 'split epi' instead of 'dimensionally split':

Definition 6.3. Let $\mathcal{W}$ be a category with terminal object $\top$. An object W is **spooky** if $() : W \to \top$ is not split epi. A category is **spooky** if it has a spooky object. Therefore, a category is non-spooky if and only if all morphisms to the terminal object are split epi.

Proposition 6.4. *Let $\sqcup \ltimes U$ be a multiplier on a non-spooky category $\mathcal{W}$. Then for any object W, the slice $\exists_U W$ is split epi.*

Proof. Any functor preserves split epimorphisms. We have $\exists_U W = (W \ltimes U, \pi_2)$ and $\pi_2 : W \ltimes U \to U \cong \top \ltimes U$ is essentially the image of $W \to \top$. □

When dealing with a spooky category, the above theorem does not hold and the definition of connection-freedom w.r.t. split epi slices would not make sense, so we need a somewhat more general notion:

Definition 6.5. Given a multiplier $\sqcup \ltimes U : \mathcal{W} \to \mathcal{W}$, we say that a morphism $\varphi : V \to U$ is **dimensionally split** if there is some $W \in \mathcal{W}$ such that $\pi_2 : W \ltimes U \to U$ factors over φ. The other factor $\chi : W \ltimes U \to V$ such that $\pi_2 = \varphi \circ \chi$ will be called a **(dimensional) section** of φ. We write $\mathcal{W}//U$ for the full subcategory of $\mathcal{W}/U$ of dimensionally split slices.

We define the **boundary** ∂U as the subpresheaf of the Yoneda-embedding $\mathbf{y}U$ consisting of those morphisms that are *not* dimensionally split.

Thus, a multiplier is connection-free if and only if every dimensionally split slice has an invertible dimensional section.

Proposition 6.6. *Let $\sqcup \ltimes U$ be a multiplier on $\mathcal{W}$. Then for any object W, the slice $\exists_U W$ is dimensionally split with section* $\mathrm{id}_{W \ltimes U}$. □

Proposition 6.7. *If $\mathcal{W}$ is non-spooky, then a morphism $\varphi : V \to U$ is split epi if and only if it is dimensionally split.* [Nuy20b]

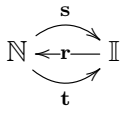
The notion of dimensionally split morphisms lets us consider the boundary and connection-freedom (a requirement for modelling Φ) also in spooky base categories, where the output of $\exists_U$ may not be split epi.

6.3. **Examples.** Let us look at some examples of multipliers. Their properties are listed in fig. 7. Most properties are easy to verify, so we omit the proofs.

Example 6.8 (Identity)**.** The identity functor on an arbitrary category $\mathcal{W}$ is an endomultiplier for $\top$. All slices $(W, ())$ are dimensionally split with section id_V; hence the boundary is empty. It is quantifiable, with $\exists_\top : \mathcal{W}/\top \to \mathcal{W} : (W, ()) \mapsto W$.

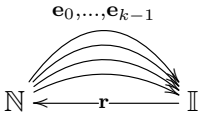
Example 6.9 (Cartesian product)**.** Let $\mathcal{W}$ be a category with finite products and $U \in \mathcal{W}$. Then $\sqcup \times U$ is an endomultiplier for U, which is affine if and only if it is the identity (i.e. U is terminal, cf. proposition 6.2 and example 6.8). It is quantifiable with $\exists_U : \mathcal{W}/U \to \mathcal{W} : (W, \psi) \mapsto W$. Hence, we have $\exists_U \exists_U = \sqcup \times U$.

Definition 6.10. Let RG be the category generated by the following diagram and equations:



$$\begin{aligned} \mathbf{r} \circ \mathbf{s} &= \mathrm{id}_{\mathbb{N}}, \\ \mathbf{r} \circ \mathbf{t} &= \mathrm{id}_{\mathbb{N}}. \end{aligned}$$

A presheaf over RG is a reflexive graph. More generally, let ${}^k\mathsf{RG}$ be the category generated by the following:



$$\mathbf{r} \circ \mathbf{e}_i = \mathrm{id}_{\mathbb{N}}.$$

Example 6.11 (Cartesian cubes)**.** Let ${}^a\mathsf{Cube}$ be the category of cartesian a-ary cubes. It is the free cartesian monoidal category with same terminal object over ${}^a\mathsf{RG}$. Concretely:

- Its objects take the form $(i_1 : \mathbb{I}, \ldots, i_n : \mathbb{I})$ (the names are desugared to de Bruijn indices, i.e. the objects are really just natural numbers),
- Its morphisms $(i_1 : \mathbb{I}, \ldots, i_n : \mathbb{I}) \to (j_1 : \mathbb{I}, \ldots, j_m : \mathbb{I})$ are arbitrary functions
$$\varphi : \{j_1, \ldots, j_m\} \to \{i_1, \ldots, i_n\} \cup \{0, \ldots, a-1\} : j \mapsto j\langle\varphi\rangle.$$
We also write $\varphi = (j_1\langle\varphi\rangle / j_1, \ldots, j_m\langle\varphi\rangle / j_m)$. If a variable i is not used, we may write $i/\oslash$ to emphasize this.

This category is spooky if and only if $a = 0$. On this category, we consider the multiplier $\sqcup \times (i : \mathbb{I})$, which is an instance of example 6.9. A slice (V, φ) is dimensionally split if $i\langle\varphi\rangle$ is not an endpoint, i.e. $i\langle\varphi\rangle \notin \{0, \ldots, a-1\}$, and in that case it is isomorphic to $\exists_{(i:\mathbb{I})} V'$ where V' is V with $i\langle\varphi\rangle$ removed. Clearly then, any morphism on the boundary factors through one of a morphisms $(\varepsilon / i) : \top \to (i : \mathbb{I})$ where $\varepsilon \in \{0, \ldots, a-1\}$, so $\partial\mathbb{I} = \biguplus_{k=0}^{a-1} \top$.

Example 6.12 (Affine cubes)**.** Let ${}^a\mathsf{Cube}_\square$ be the category of affine a-ary cubes as used in [BCH14] (binary) or [BCM15] (unary). It is the free semicartesian monoidal category with (same) terminal unit over ${}^a\mathsf{RG}$. Concretely:

- Objects are as in ${}^a\mathsf{Cube}$,
- Morphisms are as in ${}^a\mathsf{Cube}$ such that if $j\langle\varphi\rangle = k\langle\varphi\rangle \notin \{0, \ldots, a-1\}$, then $j = k$. This rules out diagonal maps.

Example	Base category	Multiplier	Spooky category	Weakening/ Semicartesian	Exchange	Contraction	Cartesian	Cancellative	Affine	Connection-free	Quantifiable
6.8	$\mathcal{W}$	Id	?	✔	✔	✔	✔	✔	✔	✔	✔
6.9	$\mathcal{W}$	$(\sqcup \times U) \ncong \mathrm{Id}$	?	✔	✔	✔	✔	?	✖	?	✔
6.11	${}^a\mathsf{Cube}$	$\sqcup \times (i : \mathbb{I})$	$a = 0$	✔	✔	✔	✔	✔	✖	✔	✔
6.12	${}^a\mathsf{Cube}_\square$	$\sqcup * (i : \mathbb{I})$	$a = 0$	✔	✔	✖	✖	✔	✔	✔	✔
6.13	CCHM	$\sqcup \times (i : \mathbb{I})$	✖	✔	✔	✔	✔	✔	✖	✖	✔
6.14	DCube_d	$\sqcup \times (i : (\!|k|\!))$	✖	✔	✔	✔	✔	✔	✖	✔	✔
6.15	Clock	$\sqcup \times (i : \circlearrowright_k)$	✔	✔	✔	✔	✔	✔	✖	✔	✔
6.16	TwCube	$\sqcup \ltimes \mathbb{I}$	✖	✖	✖	✖	✖	✔	✔	✔	✔
6.17	n	$\min(\sqcup, i)$	✔	✔	✔	✔	✔	✔	✖	✔	✔
6.18	$\mathsf{Simplex}$	$\sqcup \uplus_< [0]$	✖	✔	✖	✔	✖	✔	✔	✖	✔
6.19	${}^2\mathsf{Cube}_\bot$	$\sqcup \times \bot$	✖	✔	✔	✔	✔	✖	✖	✔	✔

Figure 7: Some interesting multipliers and their properties.

This category is spooky if and only if $a = 0$. On this category, we consider the functor $\sqcup * (i : \mathbb{I}) : W \mapsto (W, i : \mathbb{I})$, which is a multiplier for $(i : \mathbb{I})$. Dimensional splitness and the boundary are as in ${}^a\mathsf{Cube}$. This functor is quantifiable with $\exists_{(i:\mathbb{I})}((W, j : \mathbb{I}), (j/i)) = W$ and $\exists_{(i:\mathbb{I})}(W, (\varepsilon/i)) = W$ for each of the a endpoints ε.

In the nullary case, ${}^0\mathsf{Cube}_\square$ is the base category of the Schanuel topos, which is equivalent to the category of nominal sets [Pit13a]. In that case, $\exists_{(i:\mathbb{I})}$ is not just left adjoint to $\lrcorner_{(i:\mathbb{I})}$, but in fact an inverse and hence also right adjoint. This is in line with the fact that in nominal type theory [PMD15], there is a single name quantifier which can be read as either existential or universal quantification.

Example 6.13 (CCHM cubes)**.** Let CCHM be the category of CCHM cubes [CCHM17]. Its objects are as in ${}^2\mathsf{Cube}$ and its morphisms $(i_1 : \mathbb{I}, \dots, i_n : \mathbb{I}) \to (j_1 : \mathbb{I}, \dots, j_m : \mathbb{I})$ are functions from $\{j_1, \dots, j_m\}$ to the free de Morgan algebra over $\{i_1, \dots, i_n\}$. We again consider $\sqcup \times (i : \mathbb{I})$, another instance of example 6.9. A slice (V, φ) is now dimensionally split if $i\langle\varphi\rangle$ is not an endpoint, so the boundary is again $\top \uplus \top$. Connection-freedom is now violated by $(j \vee k/i), (j \wedge k/i) : (j : \mathbb{I}, k : \mathbb{I}) \to (i : \mathbb{I})$, which have dimensional sections $(i/j, 0/k) : (i : \mathbb{I}) \to (j : \mathbb{I}, k : \mathbb{I})$ and $(i/j, 1/k) : (i : \mathbb{I}) \to (j : \mathbb{I}, k : \mathbb{I})$ respectively but are not in the image of $\lrcorner_{(i:\mathbb{I})}$.

Example 6.14 (Depth d cubes)**.** Let DCube_d with $d \geq -1$ be the category of depth d cubes, used as a base category in degrees of relatedness [ND18a, Nuy18a]. This is a generalization of the category of binary cartesian cubes Cube, where instead of typing every dimension with *the* interval $\mathbb{I}$, we type them with the k-interval $(\!|k|\!)$, where $k \in \{0, \dots, d\}$ is called the degree of relatedness of the edge. Its objects take the form $(i_1 : (\!|k_1|\!), \dots, i_n : (\!|k_n|\!))$. Conceptually, we have a map $(\!|k|\!) \to (\!|\ell|\!)$ if $k \geq \ell$. Thus, morphisms $\varphi : (i_1 : (\!|k_1|\!), \dots, i_n : (\!|k_n|\!)) \to (j_1 : (\!|\ell_1|\!), \dots, j_m : (\!|\ell_m|\!))$ send every variable $j : (\!|\ell|\!)$ of the codomain to a value $j\langle\varphi\rangle$, which is either 0, 1 or a variable $i : (\!|k|\!)$ of the domain such that $k \geq \ell$.

- If $d = -1$, then there is only one object $()$ and only the identity morphism, i.e. we have the point category.

- If $d = 0$, we just get Cube.
- If $d = 1$, we obtain the category of bridge/path cubes $\mathsf{BPCube} := \mathsf{DCube}_1$. We write $\mathbb{P}$ for $(\!|0|\!)$ (the path interval) and $\mathbb{B}$ for $(\!|1|\!)$ (the bridge interval). Bridge/path cubical sets are used as a model for parametric quantifiers [NVD17, Nuy18a].

On this category, we consider $\sqcup \times (i : (\!|k|\!))$, which is another instance of example 6.9. A slice (V, φ) is dimensionally split w.r.t. this multiplier if $i\langle\varphi\rangle$ is a variable of type $(\!|k|\!)$ (and not $k' > k$), in which case a preimage is obtained as in Cube. Hence, a slice is on the boundary if it factors over $(0/i) : () \to (i : (\!|k|\!))$ or over $(i/i) : (i : (\!|k'|\!)) \to (i : (\!|k|\!))$ for $k' > k$. These morphisms correspond to the cells of $\mathbf{y}(i : (\!|k+1|\!))$ for $k < d$, so $\partial(i : (\!|k|\!)) \cong \mathbf{y}(i : (\!|k+1|\!))$ for $k < d$ and $\partial(i : (\!|d|\!)) \cong \top \uplus \top$.

Example 6.15 (Clocks)**.** Let Clock be the category of clocks, used as a base category in guarded type theory [BM20]. It is the free cartesian category over ω. Concretely:

- Its objects take the form $(i_1 : \clubsuit_{k_1}, \ldots, i_n : \clubsuit_{k_n})$ where all $k_j \geq 0$. We can think of a variable of type $\clubsuit_k$ as representing a clock (i.e. a time dimension) paired up with a certificate that we do not care what happens after the time on this clock exceeds k.
- Correspondingly, we should have a map $\clubsuit_k \to \clubsuit_\ell$ if $k \leq \ell$, because if the time exceeds ℓ, then it certainly exceeds k so our certificate can be legitimately adjusted. Then morphisms $\varphi : (i_1 : \clubsuit_{k_1}, \ldots, i_n : \clubsuit_{k_n}) \to (j_1 : \clubsuit_{\ell_1}, \ldots, j_m : \clubsuit_{\ell_m})$ are functions that send (de Bruijn) variables $j : \clubsuit_\ell$ of the codomain to a variable $j\langle\varphi\rangle : \clubsuit_k$ of the domain such that $k \leq \ell$.

This category is spooky. On this category we consider $\sqcup \times (i : \clubsuit_k)$, which is another instance of example 6.9. A slice (V, φ) is dimensionally split w.r.t. this multiplier if $i\langle\varphi\rangle$ is a variable of type $\clubsuit_k$ (and not $k' < k$), in which case a preimage is obtained as in Cube. Thus, $\partial(i : \clubsuit_k) \cong \mathbf{y}(i : \clubsuit_{k-1})$ for $k > 0$ and $\partial(i : \clubsuit_0) \cong \bot$.

Example 6.16 (Twisted cubes)**.** Pinyo and Kraus's category of twisted cubes TwCube [PK20] can be described as a subcategory of the category of non-empty finite linear orders (or, if you want, of its skeletalization: the category of simplices $\mathsf{Simplex}$). On $\mathsf{Simplex}$, we can define a functor $\sqcup \ltimes \mathbb{I}$ such that $W \ltimes \mathbb{I} = W^{\mathsf{op}} \uplus_< W$, where we consider elements from the left smaller than those from the right. Now TwCube is the subcategory of $\mathsf{Simplex}$ whose objects are generated by $\top$ and $\sqcup \ltimes \mathbb{I}$ (note that every object then also has an opposite since $\top^{\mathsf{op}} = \top$ and $(V \ltimes \mathbb{I})^{\mathsf{op}} \cong V \ltimes \mathbb{I}$), and whose morphisms are given by

- $(\varphi, 0) : \mathrm{Hom}_{\mathsf{TwCube}}(V, W \ltimes \mathbb{I})$ for all $\varphi : \mathrm{Hom}_{\mathsf{TwCube}}(V, W^{\mathsf{op}})$,
- $(\varphi, 1) : \mathrm{Hom}_{\mathsf{TwCube}}(V, W \ltimes \mathbb{I})$ for all $\varphi : \mathrm{Hom}_{\mathsf{TwCube}}(V, W)$,
- $\varphi \ltimes \mathbb{I} : \mathrm{Hom}_{\mathsf{TwCube}}(V \ltimes \mathbb{I}, W \ltimes \mathbb{I})$ for all $\varphi : \mathrm{Hom}_{\mathsf{TwCube}}(V, W)$,
- $() : \mathrm{Hom}_{\mathsf{TwCube}}(V, \top)$.

Note that this collection automatically contains all identities, composites, and opposites. Isomorphism to Pinyo and Kraus's category of twisted cubes can be seen from their ternary representation [PK20, def. 34]. We now consider the multiplier $\sqcup \ltimes \mathbb{I} : \mathsf{TwCube} \to \mathsf{TwCube}$, which Pinyo and Kraus call the twisted prism functor. A slice (V, φ) is dimensionally split if and only if it is of the form $\varphi = \psi \ltimes I$. Hence, all slices on the boundary factor over $((), 0) : () \to () \ltimes \mathbb{I}$ or $((), 1) : () \to () \ltimes \mathbb{I}$, so that $\partial\mathbb{I} \cong \top \uplus \top$. The multiplier is quantifiable with

$$\exists_{\mathbb{I}} : \begin{cases} (W, ((), 0)) & \mapsto & W^{\mathsf{op}} \\ (W, ((), 1)) & \mapsto & W \\ (W \ltimes \mathbb{I}, () \ltimes \mathbb{I}) & \mapsto & W, \end{cases} \tag{6.3}$$

with the obvious action on morphisms.

Example 6.17 (Finite ordinals)**.** In the base category ω of the topos of trees, used in guarded type theory [BMSS12], where $\mathrm{Hom}(i,j) = \{* \mid i \leq j\}$, a cartesian product is given by $i \times j = \min(i,j)$. However, this category lacks a terminal object. Instead, on the subcategory n with terminal object $n-1$, which is endowed with the same cartesian product, we consider the multiplier $\sqcup \times i$, which is again an instance of example 6.9. Any slice $(j,*)$ (where necessarily $j \leq i$) is dimensionally split with section $* : \min(i,j) \to j$; hence $\partial i = \bot$.

Example 6.18 (Simplices)**.** In the category of simplices $\mathsf{Simplex}$, we consider the functor $\sqcup \ltimes [1] := \sqcup \uplus_< [0]$, which freely adds a maximal element to a linear order. This is a multiplier for $[1]$. A slice (V,φ) is dimensionally split if $\varphi^{-1}(1) \neq \varnothing$; in that case it has a dimensional section $v^{-1}(0) \ltimes [1] \to V$. Hence, any morphism on the boundary factors through $\lambda 0.0 : [0] \to [1]$ so that $\partial[1] \cong \mathbf{y}[0]$. The multiplier is not connection-free as any slice where $\varphi^{-1}(1)$ has multiple elements, is a connection. It is quantifiable with $\exists_{[1]}(V,\varphi) = \varphi^{-1}(0)$.

Example 6.19 (Non-cancellativity)**.** Let ${}^2\mathsf{Cube}_\bot$ be the category of binary cubes extended with an initial object. We consider the cartesian product $\sqcup \times \bot$ which sends everything to $\bot$. This is not cancellative, as $\exists_\bot$ sends both $(0/i)$ and $(1/i) : () \to (i : \mathbb{I})$ to $[] : (\bot, []) \to (\bot, [])$. It is not affine, as there is no $\psi : () \to \bot$ such that $\psi \times \bot = [] : \exists_\bot() \to \exists_\bot \bot$.

6.4. **(Co)quantifier modalities for weakening.** Recall that we write $\sqcup \ltimes \mathbf{y}U$ for the left Kan extension of a multiplier $\sqcup \ltimes U$. For any **semicartesian** multiplier $\sqcup \ltimes U : \mathcal{W} \to \mathcal{W}$ and any presheaf $\Xi \in \mathrm{Psh}(\mathcal{W})$, we get a presheaf morphism $\pi_1 : \Xi \ltimes \mathbf{y}U \to \Xi$. In this situation, the notations in theorem 5.1 are not very illuminating as they would only mention π_1 and not Ξ or U. Instead, we use the following notations:

Notation 6.20. A functor acting on elements:

- $\Sigma_U^{/\Xi} := \Sigma^{/\pi_1} : \mathcal{W}/\Xi \ltimes \mathbf{y}U \to \mathcal{W}/\Xi$

Functors acting on presheaves:

- $\Sigma_{\mathbf{y}U}^{\Xi|} := \Sigma^{\pi_1|} : \mathrm{Psh}(\mathcal{W}/\Xi \ltimes \mathbf{y}U) \to \mathrm{Psh}(\mathcal{W}/\Xi)$
- $\Omega_{\mathbf{y}U}^{\Xi|} := \Omega^{\pi_1|} : \mathrm{Psh}(\mathcal{W}/\Xi) \to \mathrm{Psh}(\mathcal{W}/\Xi \ltimes \mathbf{y}U)$
- $\Pi_{\mathbf{y}U}^{\Xi|} := \Pi^{\pi_1|} : \mathrm{Psh}(\mathcal{W}/\Xi \ltimes \mathbf{y}U) \to \mathrm{Psh}(\mathcal{W}/\Xi)$

Natural transformations:

$$\mathsf{copy}_{\mathbf{y}U}^{\Xi|} := \mathsf{copy}^{\pi_1|} : 1 \to \Omega_{\mathbf{y}U}^{\Xi|} \circ \Sigma_{\mathbf{y}U}^{\Xi|} \qquad \mathsf{drop}_{\mathbf{y}U}^{\Xi|} := \mathsf{drop}^{\pi_1|} : \Sigma_{\mathbf{y}U}^{\Xi|} \circ \Omega_{\mathbf{y}U}^{\Xi|} \to 1$$
$$\mathsf{const}_{\mathbf{y}U}^{\Xi|} := \mathsf{const}^{\pi_1|} : 1 \to \Pi_{\mathbf{y}U}^{\Xi|} \circ \Omega_{\mathbf{y}U}^{\Xi|} \qquad \mathsf{app}_{\mathbf{y}U}^{\Xi|} := \mathsf{app}^{\pi_1|} : \Omega_{\mathbf{y}U}^{\Xi|} \circ \Pi_{\mathbf{y}U}^{\Xi|} \to 1$$

For modalities, we use the weakening notations already introduced in notation 5.3: we internalize the above functors as $\Omega(u : \mathbb{U}) : \Xi \to (\Xi, u : \mathbb{U})$ and $\Pi(u : \mathbb{U}) : (\Xi, u : \mathbb{U}) \to \Xi$, sometimes abbreviating to $\Omega\, u$ and $\Pi\, u$, and the above natural transformations as const_u and app_u.

6.5. **Acting on elements.** A quantifiable multiplier $\sqcup \ltimes U : \mathcal{W} \to \mathcal{W}$ as defined in definition 6.1 gives rise to a pair of adjoint functors $\exists_U \dashv \exists U$ between $\mathcal{W}$ and the category of slices $\mathcal{W}/U$, and hence a pair of adjoint functors $\exists_U^{/\top} \dashv \exists_U^{/\top}$ between the categories of elements $\mathcal{W}/\top$ and $\mathcal{W}/(\top \ltimes \mathbf{y}U)$ of the empty shape context $() := \top$ and the single variable shape context $(u : \mathbb{U}) := \top \ltimes \mathbf{y}U$ respectively. As any functor between base categories gives rise to a triple of adjoint functors between presheaf categories, the adjoint pair $\exists_U \dashv \exists U$ gives rise to an adjoint quadruple $\exists_{\mathbf{y}U}^{\top|} \dashv \exists_{\mathbf{y}U}^{\top|} \dashv \forall_{\mathbf{y}U}^{\top|} \dashv \Pi_{\mathbf{y}U}^{\top|}$ between the categories $\mathrm{Psh}(\mathcal{W}/\top)$ and $\mathrm{Psh}(\mathcal{W}/\mathbf{y}U)$ that model the modes $()$ and $(u : \mathbb{U})$ respectively. Thus, we are presently well-equipped to study the transpension type in a setting with *at most one shape variable*. Deeming this unsatisfactory, in the current section we intend to generalize the above functors, so that everywhere we mentioned $\top$ above we can instead have an arbitrary presheaf Ξ.

Definition 6.21. Given a multiplier $\sqcup \ltimes U : \mathcal{W} \to \mathcal{W}$, we define:

$$\exists_U^{/\Xi} : \mathcal{W}/\Xi \to \mathcal{W}/(\Xi \ltimes \mathbf{y}U) : (W, \xi) \mapsto (W \ltimes U, \xi \ltimes \mathbf{y}U),$$

where $\xi \ltimes \mathbf{y}U$ denotes the $(W \ltimes U)$-shaped cell of $\Xi \ltimes \mathbf{y}U$ obtained from ξ.

We say that $\sqcup \ltimes U$ is:

- **Providently cancellative** if for all Ξ, the functor $\exists_U^{/\Xi}$ is faithful,
- **Providently affine** if for all Ξ, the functor $\exists_U^{/\Xi}$ is full,
- **Providently connection-free**[17] if for all Ξ, the functor $\exists_U^{/\Xi}$ is essentially surjective on elements $(V, \varphi) \in \mathcal{W}/(\Xi \ltimes \mathbf{y}U)$ such that $\varphi : V \to \Xi \ltimes \mathbf{y}U$ is directly dimensionally split (definition 6.22). A directly dimensionally split element $(V, \xi) \in \mathcal{W}/\Xi \ltimes \mathbf{y}U$ that is not in the image of $\exists_U^{/\Xi}$ even up to isomorphism, will be called a **direct connection** of the multiplier.
- **Providently quantifiable** if for all Ξ, the functor $\exists_U^{/\Xi}$ has a left adjoint $\exists_U^{/\Xi} : \mathcal{W}/(\Xi \ltimes \mathbf{y}U) \to \mathcal{W}/\Xi$. We denote the unit as $\mathsf{copy}_U^{/\Xi} : \mathrm{Id} \to \exists_U^{/\Xi} \exists_U^{/\Xi}$ and the co-unit as $\mathsf{drop}_U^{/\Xi} : \exists_U^{/\Xi} \exists_U^{/\Xi} \to \mathrm{Id}$.

Definition 6.22. Given a multiplier $\sqcup \ltimes U : \mathcal{W} \to \mathcal{W}$, we say that a V-shaped presheaf cell φ of $\Xi \ltimes \mathbf{y}U$ is **directly dimensionally split** with direct dimensional section $\chi : W \ltimes U \to V$ if $\varphi\chi$ is of the form $\xi \ltimes \mathbf{y}U$. The section can alternatively be presented as a morphism of elements $\chi : \exists_U^{/\Xi}(W, \xi) \to (V, \varphi)$. We write $\mathcal{W}/\!/(\Xi \ltimes \mathbf{y}U)$ for the full subcategory of $\mathcal{W}/(\Xi \ltimes \mathbf{y}U)$ of dimensionally split slices.

We define the **(direct) boundary** $\Xi \ltimes \partial U$ as the subpresheaf of the Yoneda-embedding $\mathbf{y}U$ consisting of those morphisms that are *not* dimensionally split.[18]

Since we can instantiate Ξ with $\top$, we see that each of the provident criteria implies the original one from definition 6.1. Below we give *sufficient* conditions for a multiplier to satisfy the provident criteria:

Proposition 6.23. *The multiplier $\sqcup \ltimes U : \mathcal{W} \to \mathcal{W}$ is:*

[17] In the technical report [Nuy20b], we generalize connection-freedom to categories of elements in two ways, an indirect and a direct one. The indirect one is obsolete and is mentioned solely for consistency with [Nuy20a], whereas the direct one is the one used in the current paper.

[18] It is tempting to define $\Xi \ltimes \partial U$ instead as a pullback of $\Xi \ltimes \mathbf{y}U \to \mathbf{y}U \supseteq \partial U$. This is the indirect boundary that leads to the less interesting notion of indirect provident connection-freedom discussed in the technical report [Nuy20b].

- *providently cancellative if it is cancellative,*
- *providently affine if it is cancellative and affine,*
- *providently connection-free if it is affine and connection-free,*
- *providently quantifiable if it is quantifiable.*

Proof. See [Nuy20b]. □

The following (fairly obvious) theorem is paramount to the semantics of transpension elimination (section 9.3) and the Φ-rule (section 10.2):

Theorem 6.24 (Kernel theorem)**.** *If a multiplier* $\sqcup \ltimes U : \mathcal{W} \to \mathcal{W}$ *is cancellative, affine and connection-free, then* $\exists_U^{/\Xi} : \mathcal{W}/\Xi \to \mathcal{W}//(\Xi \ltimes \mathbf{y}U)$ *is an equivalence of categories.* □

6.6. **(Co)quantifier modalities for multipliers.** We are now well-equipped to study the transpension type in a setting with multiple shape variables.

Theorem 6.25. *Any quantifiable*[19] *multiplier* $\sqcup \ltimes U : \mathcal{W} \to \mathcal{W}$ *and any presheaf* $\Xi \in \mathrm{Psh}(\mathcal{W})$ *give rise to a quadruple of adjoint functors*

$$\exists_{\mathbf{y}U}^{\Xi|} \dashv \fint_{\mathbf{y}U}^{\Xi|} \dashv \forall_{\mathbf{y}U}^{\Xi|} \dashv \Qoppa_{\mathbf{y}U}^{\Xi|},$$

$$\exists_{\mathbf{y}U}^{\Xi|}, \forall_{\mathbf{y}U}^{\Xi|} : \mathrm{Psh}(\mathcal{W}/\Xi \ltimes \mathbf{y}U) \to \mathrm{Psh}(\mathcal{W}/\Xi) \qquad \fint_{\mathbf{y}U}^{\Xi|}, \Qoppa_{\mathbf{y}U}^{\Xi|} : \mathrm{Psh}(\mathcal{W}/\Xi) \to \mathrm{Psh}(\mathcal{W}/\Xi \ltimes \mathbf{y}U)$$

the latter three of which can be internalized as modalities $\fint(u : \mathbb{U}) \dashv \forall(u : \mathbb{U}) \dashv \Qoppa(u : \mathbb{U})$ *with*

$$[\![\blacksquare_{\fint u}]\!] = \exists_{\mathbf{y}U}^{\Xi|}, \qquad [\![\fint u]\!] = [\![\blacksquare_{\forall u}]\!] = \fint_{\mathbf{y}U}^{\Xi|}, \qquad [\![\forall u]\!] = [\![\blacksquare_{\Qoppa u}]\!] = \forall_{\mathbf{y}U}^{\Xi|}, \qquad [\![\Qoppa u]\!] = \Qoppa_{\mathbf{y}U}^{\Xi|}.$$

Overloading some notations from theorem 5.1 we denote the units and co-units as

$$\mathsf{copy}_{\mathbf{y}U}^{\Xi|} : 1 \to \fint_{\mathbf{y}U}^{\Xi|} \circ \exists_{\mathbf{y}U}^{\Xi|} \qquad \mathsf{drop}_{\mathbf{y}U}^{\Xi|} : \exists_{\mathbf{y}U}^{\Xi|} \circ \fint_{\mathbf{y}U}^{\Xi|} \to 1$$

$$\mathsf{const}_{\mathbf{y}U}^{\Xi|} : 1 \to \forall_{\mathbf{y}U}^{\Xi|} \circ \fint_{\mathbf{y}U}^{\Xi|} \qquad \mathsf{app}_{\mathbf{y}U}^{\Xi|} : \fint_{\mathbf{y}U}^{\Xi|} \circ \forall_{\mathbf{y}U}^{\Xi|} \to 1$$

$$\mathsf{reidx}_{\mathbf{y}U}^{\Xi|} : 1 \to \Qoppa_{\mathbf{y}U}^{\Xi|} \circ \forall_{\mathbf{y}U}^{\Xi|} \qquad \mathsf{unmer}_{\mathbf{y}U}^{\Xi|} : \forall_{\mathbf{y}U}^{\Xi|} \circ \Qoppa_{\mathbf{y}U}^{\Xi|} \to 1$$

$$\mathsf{const}_u : 1 \Rightarrow \forall u \circ \fint u \qquad \mathsf{app}_u : \fint u \circ \forall u \Rightarrow 1$$

$$\mathsf{reidx}_u : 1 \Rightarrow \Qoppa u \circ \forall u \qquad \mathsf{unmer}_u : \forall u \circ \Qoppa u \Rightarrow 1$$

where reidx *stands for* reindex *and* unmer *is the negation of* mer *which stands for* meridian.

Proof. Via left Kan extension, precomposition and right Kan extension [Sta19], the pair of adjoint functors $\exists_U^{/\Xi} \dashv \fint_U^{/\Xi}$ gives rise to a quadruple of adjoint functors $\exists_{\mathbf{y}U}^{\Xi|} \dashv \fint_{\mathbf{y}U}^{\Xi|} \dashv \forall_{\mathbf{y}U}^{\Xi|} \dashv \Qoppa_{\mathbf{y}U}^{\Xi|}$ between the presheaf categories. (For the middle two, we can choose whether we derive them from $\exists_U^{/\Xi}$ or from $\fint_U^{/\Xi}$; the resulting functors are naturally isomorphic. We will specify our choice when relevant.) □

Notation 6.26. Again, due to the (purely sugarous) usage of shape variables, we may end up with variable renamings that are sugar for the identity, e.g.

$$\mathsf{app}_{(v/u:\mathbb{U})} : \fint(v : \mathbb{U}) \circ \forall(u : \mathbb{U}) \Rightarrow \Omega(v : \mathbb{U}, u := v)$$

$$\mathsf{reidx}_{(v/u:\mathbb{U})} : \Omega(v : \mathbb{U}, u := v) \Rightarrow \Qoppa(v : \mathbb{U}) \circ \forall(u : \mathbb{U})$$

[19] Without quantifiability, we lose the leftmost adjoint functor $\exists_{\mathbf{y}U}^{\Xi|}$ and the leftmost adjoint modality $\fint u$.

are exactly the same 2-cells as

$$\mathsf{app}_{(u:\mathbb{U})} : \exists(u:\mathbb{U}) \circ \forall(u:\mathbb{U}) \Rightarrow 1$$
$$\mathsf{reidx}_{(u:\mathbb{U})} : 1 \Rightarrow \between(u:\mathbb{U}) \circ \forall(u:\mathbb{U}).$$

Whereas theorem 5.1 clearly states the meaning of the functors introduced there, little can be said about the meaning of the functors introduced in theorem 6.25 without knowing more about the multiplier involved. The following theorem clarifies the leftmost three functors:

Theorem 6.27 (Quantification)**.** *If $\sqcup \ltimes U$ is*

(1) *cancellative and affine, then* $\mathsf{drop}^{\Xi|}_{\mathbf{y}U}$, $\mathsf{const}^{\Xi|}_{\mathbf{y}U}$ *and* $\mathsf{unmer}^{\Xi|}_{\mathbf{y}U}$ *are natural isomorphisms.*
(2) *semi-cartesian, then we have*
   (a) $\mathsf{hide}^{\Xi|}_{\mathbf{y}U} : \Sigma^{\Xi|}_{\mathbf{y}U} \to \exists^{\Xi|}_{\mathbf{y}U}$ *(if quantifiable),*
   (b) $\mathsf{spoil}^{\Xi|}_{\mathbf{y}U} : \between^{\Xi|}_{\mathbf{y}U} \to \Omega^{\Xi|}_{\mathbf{y}U}$*, which can be internalized (if quant.) as* $\mathsf{spoil}_u : \between u \Rightarrow \Omega u$*,*
   (c) $\mathsf{cospoil}^{\Xi|}_{\mathbf{y}U} : \Pi^{\Xi|}_{\mathbf{y}U} \to \forall^{\Xi|}_{\mathbf{y}U}$*, which can be internalized as* $\mathsf{cospoil}_u : \Pi u \Rightarrow \forall u$*.*
(3) *cartesian, then we have:*

$$\exists^{\Xi|}_{\mathbf{y}U} = \Sigma^{\Xi|}_{\mathbf{y}U}, \qquad \between^{\Xi|}_{\mathbf{y}U} = \Omega^{\Xi|}_{\mathbf{y}U}, \qquad \forall^{\Xi|}_{\mathbf{y}U} = \Pi^{\Xi|}_{\mathbf{y}U}.$$

The equalities assume that $\exists_U : \mathcal{W}/U \to \mathcal{W}$ is defined on the nose by $\exists_U(W,\psi) = W$ and that $\between^{\Xi|}_{\mathbf{y}U}$ and $\forall^{\Xi|}_{\mathbf{y}U}$ is constructed from $\exists^{/\Xi}_U$ by precomposition and right Kan extension, respectively. Failing this, we only get natural isomorphisms.

Let us try to interpret this on a more intuitive level:

(1) For cancellative and affine $\mathbb{U}$, invertibility of const_u means that any 'function' of type $\langle \forall u \mid^{\mathrm{a}} \langle \between u \mid^{\mathrm{f}} T\rangle\rangle$ is necessarily constant, i.e. elements of $\langle \between u \mid^{\mathrm{f}} T\rangle$ cannot depend on the shape variable u and are instead *fresh* for u. With that in mind, invertibility of $\mathsf{drop}^{\Xi|}_{\mathbf{y}U}$ indicates that the Σ-like operation $\exists^{\Xi|}_{\mathbf{y}U}$ hides its first component $u : \mathbb{U}$. Indeed, knowing that the second component of $\exists^{\Xi|}_{\mathbf{y}U} \between^{\Xi|}_{\mathbf{y}U} \Gamma$ is fresh for u, one might expect that $\exists^{\Xi|}_{\mathbf{y}U} \between^{\Xi|}_{\mathbf{y}U} \Gamma$ behaves somewhat like a non-dependent product (which is exactly what happens in the cartesian setting). Instead, $\exists^{\Xi|}_{\mathbf{y}U}$ cancels out $\between^{\Xi|}_{\mathbf{y}U}$, so apparently if the second component does not depend on u, then u is lost altogether. Finally function application, which is basically the projection operation from proposition 3.3,

$$\mathsf{prmod}_{\forall u} : (\between u \mid^{\mathrm{f}} \langle \forall u \mid^{\mathrm{a}} T\rangle) \to T[\mathsf{app}_u{\downarrow}^{\mathrm{fa}}_{\bullet}]. \tag{6.4}$$

requires that the applied function of type $\langle \forall u \mid^{\mathrm{a}} T\rangle$ be fresh for u, i.e. $\langle \forall u \mid^{\mathrm{a}} T\rangle$ can be thought of as the type of affine functions $\forall u.T$.

(2) If $\mathbb{U}$ is semicartesian (which is perfectly combinable with being affine and cancellative), i.e. if weakening is allowed for $u : \mathbb{U}$, then we get three additional operations. The 2-cell spoil_u allows us to forget that something is fresh for u. This would not be possible without weakening because not being fresh for u would require consumption of u. The 2-cell $\mathsf{cospoil}_u$ allows us to unnecessarily restrict a function's usage to affine usages. This is again not possible without weakening because then the function application operation

$$\mathsf{prmod}_{\Pi u} : (\Omega u \mid^{\mathrm{o}} \langle \Pi u \mid^{\mathrm{p}} T\rangle) \to T[\mathsf{app}_u{\downarrow}^{\mathrm{op}}_{\bullet}]. \tag{6.5}$$

would behave 'additively' in the linear logic sense and *require* that the applied function of type $\langle \Pi\, u \mid^{\mathrm{p}} T \rangle$ again consumes u on top of taking it as an argument. The semantic substitution $\mathsf{hide}^{\Xi|}_{\mathbf{y}U} : \Sigma^{\Xi|}_{\mathbf{y}U}\Gamma \to \exists^{\Xi|}_{\mathbf{y}U}\Gamma$ at mode Ξ allows us to hide the first component $u : \mathbb{U}$. The effect when applying this substitution is effectively a weakening over $u : \mathbb{U}$, in the sense that the context $\exists^{\Xi|}_{\mathbf{y}U}\Gamma$ can be regarded as not containing the variable u except in the form of a hidden mention necessary to make the dependencies of Γ work out.

(3) If $\mathbb{U}$ is cartesian, then the leftmost three functors become identical to the ones in section 6.4. In particular, fresh weakening and weakening coincide and the word 'fresh' becomes meaningless.

In order to have some general terminology, we will speak of the **hiding** exponential, **fresh** weakening and **substructural** functions even when the specifics of the multiplier are unclear and it is potentially cartesian.

6.7. **Cartesian multipliers.** The cartesian case of the quantification theorem 6.27 may look like all our efforts with multipliers are useless, but let's not forget that there is now a further right adjoint to these well-known functors:

$$\Sigma^{\Xi|}_{\mathbf{y}U} \dashv \Omega^{\Xi|}_{\mathbf{y}U} \dashv \Pi^{\Xi|}_{\mathbf{y}U} \dashv \between^{\Xi|}_{\mathbf{y}U}.$$

What we have proven here is that, for any representable presheaf $\mathbf{y}U \in \mathrm{Psh}(\mathcal{W})$ such that cartesian products with U exist in the base category $\mathcal{W}$ (yielding a cartesian multiplier $\sqcup \times U : \mathcal{W} \to \mathcal{W}$), there is a right adjoint to the Π-type! Licata et al. [LOPS18] have used a right adjoint to the non-dependent function type functor $(\mathbf{y}U \to \sqcup) = \Pi^{\Xi|}_{\mathbf{y}U} \circ \Omega^{\Xi|}_{\mathbf{y}U}$, called the *amazing right adjoint* and necessarily given by $(\mathbf{y}U \between \sqcup) = \Pi^{\Xi|}_{\mathbf{y}U} \circ \between^{\Xi|}_{\mathbf{y}U}$, but the current result appears stronger.

Remarkably, it is not, and therefore it is not novel either. As conjectured by Lawvere, proven by Freyd and published by Yetter [Yet87], for an arbitrary object $\mathbb{U}$ in an arbitrary topos, the transpension functor (there unnamed and denoted ∇) over $\mathbb{U}$ exists if the amazing right adjoint exists. Indeed, in that case it can be constructed by the following pullback:

$$\begin{array}{ccc} \Sigma(u : \mathbb{U}).\between u.T & \longrightarrow & \mathbb{U} \between (\mathsf{Maybe}\, T) \\ {\scriptstyle \mathsf{fst}}\downarrow & \lrcorner & \downarrow{\scriptstyle \mathbb{U} \between \mathsf{IsJust}} \\ \mathbb{U} & \xrightarrow[(\lambda f. f \equiv \mathrm{id}_{\mathbb{U}})^{\top}]{} & \mathbb{U} \between \mathsf{Prop} \end{array} \qquad \begin{array}{lcl} \mathsf{IsJust}\,(\mathsf{nothing}) & = & \bot \\ \mathsf{IsJust}\,(\mathsf{just}\, t) & = & \top \end{array}$$

where $g^{\top}$ denotes the transpose of g under $(\mathbb{U} \to \sqcup) \dashv (\mathbb{U} \between \sqcup)$. In other words, $\between u.T \cong (x : \mathbb{U} \between (\mathsf{Maybe}\, T)) \times ((\mathbb{U} \between \mathsf{IsJust})\, x \equiv (\lambda f. f \equiv \mathrm{id}_{\mathbb{U}})^{\top}\, u)$.

6.8. **Further reading.** We refer to the technical report [Nuy20b] for more information on

- the contraction rule $\delta : \sqcup \ltimes U \to (\sqcup \ltimes U) \ltimes U$,
- composite multipliers $\sqcup \ltimes (U \ltimes U') := (\sqcup \ltimes U) \ltimes U'$,
- morphisms of multipliers $\sqcup \ltimes v : \sqcup \ltimes U \to \sqcup \ltimes U'$ (together with the previous point one could formalize the exchange rule),
- acting on slices as opposed to acting on elements (section 6.5),
- properties of $\sqcup \ltimes \mathbf{y}U : \mathrm{Psh}(\mathcal{W}) \to \mathrm{Psh}(\mathcal{W})$, the left Kan extension of $\sqcup \ltimes U$, again viewed as a multiplier for $\mathbf{y}U$,

- non-endo multipliers $\sqcup \ltimes U : \mathcal{W} \to \mathcal{V}$ and in particular *embargoes* which may be of interest for *contextual fibrancy* [BT17][Nuy20a, ch. 8],
- rules for commuting (co)quantifiers for multipliers, (co)quantifiers for substitution, and (when adding the transpension type to an already modal type system) prior modalities.

7. Comparison to the naïve system

In this section, we compare the formation, introduction and projection (proposition 3.3) rules for the modal types $\langle \forall\, u \mid^{\mathrm{a}} A \rangle$ and $\langle \between u \mid^{\mathrm{t}} A \rangle$ with the corresponding rules of the naïve type system in section 2. Each time, we will put together: the rule from MTT in lockful and lockless notation and the naïve rule from fig. 1, and then discuss. The constructors in lockless notation will be called Λ_u and mer_u respectively. Finally, we will revisit the applications from section 2: internal transposition and higher-dimensional pattern matching (HDPM).

7.1. **The substructural function type.** The formation and introduction rules are quite parallel:

$$\frac{\Xi, u : \mathbb{U} \mid \Gamma, \mathbin{🔒}^{\mathrm{a}}_{\forall\, u} \vdash A\,\mathsf{type}}{\Xi \mid \Gamma \vdash \langle \forall\, u \mid^{\mathrm{a}} A \rangle\,\mathsf{type}} \qquad \frac{\Xi, u : \mathbb{U} \mid \exists\, u^{\,\mathrm{f}}\,(\Gamma) \vdash A\,\mathsf{type}}{\Xi \mid \Gamma \vdash \langle \forall\, u \mid^{\mathrm{f}} A \rangle\,\mathsf{type}} \qquad \frac{\Gamma, u : \mathbb{U} \vdash A\,\mathsf{type}}{\Gamma \vdash \forall\, u.A\,\mathsf{type}}$$

$$\frac{\Xi, u : \mathbb{U} \mid \Gamma, \mathbin{🔒}^{\mathrm{a}}_{\forall\, u} \vdash a : A}{\Xi \mid \Gamma \vdash \mathsf{mod}^{\mathrm{a}}_{\forall\, u}\, a : \langle \forall\, u \mid^{\mathrm{a}} A \rangle} \qquad \frac{\Xi, u : \mathbb{U} \mid \exists\, u^{\,\mathrm{f}}\,(\Gamma) \vdash a : A}{\Xi \mid \Gamma \vdash \Lambda^{\mathrm{f}}_{u}\, a : \langle \forall\, u \mid^{\mathrm{f}} A \rangle} \qquad \frac{\Gamma, u : \mathbb{U} \vdash a : A}{\Gamma \vdash \lambda u.a : \forall\, u.A}$$

What we see in the naïve system is that the body of a function (as well as, of course, its type) is type-checked in a context with the shape variable $u : \mathbb{U}$ added to the right of Γ. Recall that in the naïve system, variables to the left of $u : \mathbb{U}$ were considered to be fresh for u, whereas those to the right of $u : \mathbb{U}$ can depend on u.

In our instantiation of MTT, the shape variable $u : \mathbb{U}$ is of course added to the shape context, so we cannot use its position to signal which variables in Γ are and are not fresh for u. Instead, we have modalities to do so, and we see that all of Γ is marked fresh using $\mathbin{🔒}_{\forall\, u} = \exists\, u$.

Note that, while shape variables in the naïve system were considered affine, in MTT we are dealing with an arbitrary multiplier and the meaning of the 'freshness marker' $\mathbin{🔒}_{\forall\, u} = \exists\, u$ varies accordingly. For example if $\mathbb{U}$ is cartesian, then $\mathbin{🔒}_{\forall\, u} = \exists\, u$ just means $\mathbin{🔒}_{\Pi\, u} = \Omega\, u$, i.e. weakening.

We proceed to the projection rule, i.e. function application:

$$\frac{\Xi \mid \Gamma, \mathbin{🔒}^{\mathrm{f}}_{\exists\, u} \vdash f : \langle \forall\, u \mid^{\mathrm{a}} A \rangle}{\Xi, u : \mathbb{U} \mid \Gamma \vdash \mathsf{prmod}_{\forall\, u} \cdot^{\mathrm{f}} f : A[\mathsf{app}_u \downarrow^{\mathrm{fa}}_{\bullet}]} \qquad \frac{\Xi \mid \exists\, u^{\,\mathrm{e}}\,(\Gamma) \vdash f : \langle \forall\, u \mid^{\mathrm{f}} A \rangle}{\Xi, u : \mathbb{U} \mid \Gamma \vdash \mathsf{app}_u \cdot^{\mathrm{e}} f : A[\mathsf{copy}_u \uparrow^{\mathrm{fe}}_{\bullet}]}$$

$$\frac{\Gamma \vdash f : \forall\, u.A}{\Gamma, u : \mathbb{U}, \delta : \Delta \vdash f\, u : A}$$

In the naïve system, the rule's conclusion has a context $\Gamma, u : \mathbb{U}, \Delta$ and all variables to the right of u, i.e. all of Δ, is removed from the context of the function f so as to ensure that f be fresh for u.

In MTT, the shape variable $u : \mathbb{U}$ goes to the shape context and $\mathbin{🔒}_{\forall\, u} = \exists\, u$ is used to signal freshness. So the naïve context translates to MTT as $(\Gamma, \mathbin{🔒}^{\mathrm{a}}_{\forall\, u}, \Delta) = (\exists\, u^{\,\mathrm{f}}\,(\Gamma), \Delta)$ –

where Δ is assumed to contain no locks. When we substitute this translated context for Γ in the MTT rule's conclusion, we get a context $(\Gamma, \mathbin{🔒}^{\mathrm{a}}_{\forall\, u}, \Delta, \mathbin{🔒}^{\mathrm{f}}_{\exists\, u}) = \exists\, u^{\,\mathrm{e}}\left(\exists\, u^{\,\mathrm{f}}\,(\Gamma), \Delta\right)$ for f. This is a bit different from what we had in the naïve system, but actually it is more general, because if in MTT we get f in context Γ, then we can move it to the required context by applying the substitution

$$\mathsf{const}_u{\downarrow}^{\bullet}_{\mathrm{af}} : (\Gamma, \mathbin{🔒}^{\mathrm{a}}_{\forall\, u}, \mathbin{🔒}^{\mathrm{f}}_{\exists\, u}) \to \Gamma \qquad\qquad \mathsf{drop}_u{\uparrow}^{\bullet}_{\mathrm{ef}} : \exists\, u^{\,\mathrm{e}}\left(\exists\, u^{\,\mathrm{f}}\,(\Gamma)\right) \to \Gamma$$

and then weakening over Δ. Note that, if $\mathbb{U}$ is cancellative and affine, then by the quantification theorem 6.27 the above substitution is actually an isomorphism. The variables from Δ can in principle not be used in the definition of f as they are guarded by $\mathbin{🔒}_{\exists\, u} = \exists\, u$, unless they are annotated by a modality μ whence there is a 2-cell $\mu \Rightarrow \exists\, u$, i.e. when they are fresh for u.

If on the contrary $\mathbb{U}$ is cartesian, then $\mathbin{🔒}_{\exists\, u} = \exists\, u$ is really just a Σ-type, so that the shape context and type-theoretic context in the premise and the conclusion are semantically isomorphic: they are both essentially $(\Xi, u : \mathbb{U}, \Gamma)$.

The β-law states:

$$\begin{aligned} \mathsf{prmod}_{\forall\, u} \cdot^{\mathrm{f}} (\mathsf{mod}^{\mathrm{a}}_{\forall\, u}\, a) &= a[\mathsf{app}_u{\downarrow}^{\mathrm{fa}}_{\bullet}], \\ \mathsf{app}_u \cdot^{\mathrm{e}} (\Lambda^{\mathrm{f}}_u\, a) &= a[\mathsf{copy}_u{\uparrow}^{\mathrm{fe}}_{\bullet}], \\ (\lambda u.a)v &= a[v/u]. \end{aligned}$$

In the naïve system, the substitution (v/u) could be written as $(v/v, v/u)$ and is really a contraction. The unit $\mathsf{copy}_u : 1 \Rightarrow \exists\, u \circ \exists\, u$ can also be seen as a form of contraction, contracting the first component of the existential with the variable u available in the shape context.

The η-law states:

$$\begin{aligned} f &= \mathsf{mod}^{\mathrm{a}}_{\forall\, u}\,(\mathsf{prmod}_{\forall\, u} \cdot^{\mathrm{f}} (f[\mathsf{const}_u{\downarrow}^{\bullet}_{\mathrm{af}}])), \\ f &= \Lambda^{\mathrm{f}}_u\,(\mathsf{app}_u \cdot^{\mathrm{e}} (f[\mathsf{drop}_u{\uparrow}^{\bullet}_{\mathrm{ef}}])), \\ f &= \lambda u.f\, u. \end{aligned}$$

In the naïve system, no substitution or weakening is necessary because the affine application again removes u from the context. Were we working with non-affine functions, however, then f appearing in the function body would have to be weakened over u. The co-unit $\mathsf{drop}_u : \exists\, u \circ \exists\, u \Rightarrow 1$ can also be seen as a form of weakening over the first component of the existential, and is an isomorphism if $\mathbb{U}$ is cancellative and affine.

7.2. **The transpension type.** The formation and introduction rules are again parallel:

$$\frac{\Xi \mid \Gamma, \mathbin{🔒}^{\mathrm{t}}_{\langle\!\langle\, u} \vdash A\,\mathsf{type}}{\Xi, u : \mathbb{U} \mid \Gamma \vdash \left\langle\!\langle\, u \mid^{\mathrm{t}} A\right\rangle\mathsf{type}} \qquad \frac{\Xi \mid \Gamma, \mathbin{🔒}^{\mathrm{t}}_{\langle\!\langle\, u} \vdash a : A}{\Xi, u : \mathbb{U} \mid \Gamma \vdash \mathsf{mod}^{\mathrm{t}}_{\langle\!\langle\, u}\, a : \left\langle\!\langle\, u \mid^{\mathrm{t}} A\right\rangle\mathsf{type}}$$

$$\frac{\Xi \mid \forall\, u^{\,\mathrm{a}}\,(\Gamma) \vdash A\,\mathsf{type}}{\Xi, u : \mathbb{U} \mid \Gamma \vdash \left\langle\!\langle\, u \mid^{\mathrm{a}} A\right\rangle\mathsf{type}} \qquad \frac{\Xi \mid \forall\, u^{\,\mathrm{a}}\,(\Gamma) \vdash a : A}{\Xi, u : \mathbb{U} \mid \Gamma \vdash \mathsf{mer}^{\mathrm{a}}_u\, a : \left\langle\!\langle\, u \mid^{\mathrm{a}} A\right\rangle}$$

$$\frac{\Gamma, \forall\, u.(\delta : \Delta) \vdash A\,\mathsf{type}}{\Gamma, u : \mathbb{U}, \delta : \Delta \vdash \langle\!\langle\, u.A\,\mathsf{type}} \qquad \frac{\Gamma, \forall\, u.(\delta : \Delta) \vdash a : A}{\Gamma, u : \mathbb{U}, \delta : \Delta \vdash \mathsf{mer}\, u\, a : \langle\!\langle(u : \mathbb{U}).A}$$

Again in the naïve system we have a context $\Gamma, u : \mathbb{U}, \Delta$ in the conclusion, which still translates to MTT as $(\Gamma, \mathbin{🔒}^{\mathrm{a}}_{\forall\, u}, \Delta) = (\lrcorner\, u^{\,\mathrm{f}}\, (\Gamma), \Delta)$ – where Δ is assumed to contain no locks. When we substitute this translated context for Γ in the MTT rule's conclusion, we get a context $(\Gamma, \mathbin{🔒}^{\mathrm{a}}_{\forall\, u}, \Delta, \mathbin{🔒}^{\mathrm{t}}_{\between\, u}) = \forall\, u^{\,\mathrm{a}}\, \big(\lrcorner\, u^{\,\mathrm{f}}\, (\Gamma), \Delta\big)$ for $a : A$. By a straightforward generalization of proposition 3.4, the latter context is isomorphic to $(\Gamma, \mathbin{🔒}^{\mathrm{a}}_{\forall\, u}, \mathbin{🔒}^{\mathrm{t}}_{\between\, u}, \forall\, u\, \Delta) = \big(\forall\, u^{\,\mathrm{a}}\, \big(\lrcorner\, u^{\,\mathrm{f}}\, (\Gamma)\big), \forall\, u\, \Delta\big)$ where $\forall\, u\, \Delta$ denotes the telescope Δ with every variable's modal annotation postcomposed with $\forall\, u \circ \sqcup$ (and appropriately substituted with an isomorphism). Then by the quantification theorem 6.27, if $\mathbb{U}$ is cancellative and affine, this is furthermore isomorphic to $(\Gamma, (\forall\, u\, \Delta)[\mathsf{unmer}_u{\downarrow}^{\mathrm{at}}_{\bullet}]) = (\Gamma, (\forall\, u\, \Delta)[\mathsf{const}_u{\uparrow}^{\mathrm{af}}_{\bullet}])$, so the correspondence with the naïve system is clear.

We proceed to the projection rule:

$$\frac{\Xi, u : \mathbb{U} \mid \Gamma, \mathbin{🔒}^{\mathrm{a}}_{\forall\, u} \vdash t : \big\langle \between\, u \mid^{\mathrm{t}} A \big\rangle}{\Xi \mid \Gamma \vdash \mathsf{prmod}_{\between\, u} \cdot^{\mathrm{a}} t : A[\mathsf{unmer}_u{\downarrow}^{\mathrm{at}}_{\bullet}]} \qquad \frac{\Xi, u : \mathbb{U} \mid \lrcorner\, u^{\,\mathrm{f}}\, (\Gamma) \vdash t : \langle \between\, u \mid^{\mathrm{a}} A \rangle}{\Xi \mid \Gamma \vdash \mathsf{unmer}_u \cdot^{\mathrm{f}} t : A[\mathsf{const}_u{\uparrow}^{\mathrm{af}}_{\bullet}]}$$

$$\frac{\Gamma, u : \mathbb{U} \vdash t : \between\, u.A}{\Gamma \vdash \mathsf{unmer}(u.t) : A}$$

In the naïve system, the shape variable $u : \mathbb{U}$ is added to the right of Γ. This corresponds to what happens in MTT, where $u : \mathbb{U}$ is added to the shape context and Γ is marked fresh.

The β-law states:

$$\begin{aligned} \mathsf{prmod}_{\between\, u} \cdot^{\mathrm{a}} (\mathsf{mod}^{\mathrm{t}}_{\between\, u}\, a) &= a[\mathsf{unmer}{\downarrow}^{\mathrm{at}}_{\bullet}], \\ \mathsf{unmer}_u \cdot^{\mathrm{f}} (\mathsf{mer}^{\mathrm{a}}_u\, a) &= a[\mathsf{const}_u{\uparrow}^{\mathrm{af}}_{\bullet}], \\ \mathsf{unmer}(u.\mathsf{mer}\, u\, a) &= a. \end{aligned}$$

In the naïve system, no substitution of a is necessary. In MTT, we substitute with $\forall\, u \circ \lrcorner\, u \Leftarrow 1 : \mathsf{const}_u \dashv \mathsf{unmer}_u : \forall\, u \circ \between\, u \Rightarrow 1$, which always exists and is an isomorphism if $\mathbb{U}$ is cancellative and affine.

The η-law states:

$$\begin{aligned} t &= \mathsf{mod}^{\mathrm{t}}_{\between\, u}\, (\mathsf{prmod}_{\between\, u} \cdot^{\mathrm{a}} (t[\mathsf{reidx}{\downarrow}^{\bullet}_{\mathrm{ta}}])), \\ t &= \mathsf{mer}^{\mathrm{a}}_u\, (\mathsf{unmer}_u \cdot^{\mathrm{f}} (t[\mathsf{app}_u{\uparrow}^{\bullet}_{\mathrm{fa}}])), \\ t &= \mathsf{mer}\, u\, (\mathsf{unmer}(v.t[v/u, {}^u\delta\, v/\delta])). \end{aligned}$$

In the naïve system, we have to apply a substitution $(v/u, {}^u\delta\, v/\delta) : (\Gamma, \forall\, u.\Delta, v : \mathbb{U}) \to (\Gamma, u : \mathbb{U}, \Delta)$ to $\Gamma, u : \mathbb{U}, \Delta \vdash t : \between\, u.A$. This corresponds to the substitution $\mathsf{app}_u{\uparrow}^{\bullet}_{\mathrm{fa}} : (\lrcorner\, u^{\,\mathrm{f}'}\, (\Gamma), \Delta) \to \lrcorner\, u^{\,\mathrm{f}}\, \Big(\forall\, u^{\,\mathrm{a}}\, \Big(\lrcorner\, u^{\,\mathrm{f}'}\, (\Gamma), \Delta\Big)\Big)$ in MTT.

7.3. **Internal transposition.** In section 2.3, we proved the isomorphism

$$(\forall (u : \mathbb{U}).A) \to B \quad \cong \quad \forall (u : \mathbb{U}).(A \to \between\, u.B). \tag{7.1}$$

The MTT equivalent,

$$(\forall\, u \mathbin{\shortmid}^{\mathrm{a}} A) \to B \quad \cong \quad \big\langle \forall\, u \mid^{\mathrm{a}} A \to \big\langle \between\, u \mid^{\mathrm{t}} B[\mathsf{unmer}_u^{-1}{\downarrow}^{\bullet}_{\mathrm{at}}] \big\rangle \big\rangle$$

is not in general true (or even statable) but for cancellative and affine multipliers it follows from the following instance of proposition 3.4 (which holds in general)

$$\big\langle \between\, u \mid^{\mathrm{t}} (\forall\, u \mathbin{\shortmid}^{\mathrm{a}} A[\mathsf{reidx}_u{\downarrow}^{\bullet}_{\mathrm{ta}}]) \to B \big\rangle \quad \cong \quad \big(A \to \big\langle \between\, u \mid^{\mathrm{t}} B \big\rangle\big) \tag{7.2}$$

by applying $\langle \forall\, u \mid \sqcup \rangle$ to both sides and using the quantification theorem 6.27.

7.4. **Higher-dimensional pattern matching.** Deriving HDPM from internal transposition for general multipliers is a bit more involved than in the naïve system because we have to use eq. (7.2) instead of eq. (7.1). However, we can construct an isomorphism $i : \langle \forall\, u \mid^{\mathrm{a}} A \uplus B \rangle \cong \langle \forall\, u \mid^{\mathrm{a}} A \rangle \uplus \langle \forall\, u \mid^{\mathrm{a}} B \rangle$ directly by translating from section 2.4. Again, one direction is trivial by pattern matching (which is still the original eliminator for modal types!). For the other direction, we again list the MTT versions in lockful and lockless notation together with the naïve version for comparison:

$$i : \langle \forall\, u \mid^{\mathrm{a}} A \uplus B \rangle \to \langle \forall\, u \mid^{\mathrm{a}} A \rangle \uplus \langle \forall\, u \mid^{\mathrm{a}} B \rangle \qquad \boxed{\text{MTT, lockful}}$$

$$i\,\hat{c} = \mathsf{prmod}_{\Diamond\, u} \cdot^{\mathrm{a}} \mathsf{case}\,(\mathsf{prmod}_{\forall\, u} \cdot^{\mathrm{f}} (\hat{c}\,\mathsf{const}_u{\downarrow_{\mathrm{af}}}))\,\mathsf{of}\left\{ \begin{array}{l} \mathsf{inl}\,a \mapsto \mathsf{mod}^{\mathrm{t}}_{\Diamond\, u}\,(\mathsf{inl}\,(\mathsf{mod}^{\mathrm{a}'}_{\forall\, u}\,(a\,\mathsf{reidx}_u{\downarrow_{\mathrm{ta}'}}))) \\ \mathsf{inr}\,b \mapsto \mathsf{mod}^{\mathrm{t}}_{\Diamond\, u}\,(\mathsf{inr}\,(\mathsf{mod}^{\mathrm{a}'}_{\forall\, u}\,(b\,\mathsf{reidx}_u{\downarrow_{\mathrm{ta}'}}))) \end{array} \right\}$$

$$i : \langle \forall\, u \mid^{\mathrm{f}} A \uplus B \rangle \to \langle \forall\, u \mid^{\mathrm{f}} A \rangle \uplus \langle \forall\, u \mid^{\mathrm{f}} B \rangle \qquad \boxed{\text{MTT, lockless}}$$

$$i\,\hat{c} = \mathsf{unmer}_u \cdot^{\mathrm{f}} \mathsf{case}\,(\mathsf{app}_u \cdot^{\mathrm{e}} (\hat{c}\,\mathsf{drop}_u{\uparrow_{\mathrm{ef}}}))\,\mathsf{of}\left\{ \begin{array}{lcl} \mathsf{inl}\,a & \mapsto & \mathsf{mer}^{\mathrm{a}}_u\,(\mathsf{inl}\,(\Lambda^{\mathrm{f}'}_u\,(a\,\mathsf{app}_u{\uparrow_{\mathrm{f}'\mathrm{a}}}))) \\ \mathsf{inl}\,b & \mapsto & \mathsf{mer}^{\mathrm{a}}_u\,(\mathsf{inl}\,(\Lambda^{\mathrm{f}'}_u\,(b\,\mathsf{app}_u{\uparrow_{\mathrm{f}'\mathrm{a}}}))) \end{array} \right\}$$

$$i : (\forall\, u.A \uplus B) \to (\forall\, u.A) \uplus (\forall\, u.B) \qquad \boxed{\text{naïve system}}$$

$$i\,\hat{c} = \mathsf{unmer}\left(u.\mathsf{case}\,\hat{c}\,u\,\mathsf{of}\left\{ \begin{array}{lcl} \mathsf{inl}\,a & \mapsto & \mathsf{mer}\,u\,(\mathsf{inl}\,(\lambda v.{}^u a\,v)) \\ \mathsf{inr}\,b & \mapsto & \mathsf{mer}\,u\,(\mathsf{inr}\,(\lambda v.{}^u b\,v)) \end{array} \right\}\right).$$

8. Additional Typing Rules

In this section, we add a few extensions to MTT in order to reason about boundaries (definition 6.5) in the *type* theory, rather than in the shape theory, and in order to recover all known presheaf operators in section 10.

8.1. **Subobject classifier.** We add a universe of propositions (semantically the subobject classifier) $\mathsf{Prop} : \mathsf{U}_0$, with implicit encoding and decoding operations à la Coquand. This universe is closed under logical operators and weak DRAs [Nuy20a, §6.5]. This is necessary to talk about Ψ and Φ. We identify all proofs of the same proposition.

8.2. **Boundary predicate.** Similar to how we write $(\Xi, u : \mathbb{U})$ for $\Xi \ltimes \mathbf{y}U$, we will write $(\Xi, u : \partial\mathbb{U})$ for $\Xi \ltimes \partial U$ for the direct boundary (definition 6.22). Write $(u \in \partial\mathbb{U})$ for the presheaf morphism that includes $(\Xi, u : \partial\mathbb{U})$ in $(\Xi, u : \mathbb{U})$. We add a predicate of the same name $\Xi, u : \mathbb{U} \mid \cdot \vdash u \in \partial\mathbb{U} : \mathsf{Prop}$ corresponding in the model to this subobject $(\Xi, u : \partial\mathbb{U}) \subseteq (\Xi, u : \mathbb{U})$. Note that, since the direct boundary was *not* defined by pullback, the boundary predicate is not preserved by shape substitution $\sigma : \Xi_1 \to \Xi_2$, i.e. $\langle \Omega(\sigma, u := u) \mid^{\mathrm{o}} (u \in \partial\mathbb{U})_{\Xi_2} \rangle$ is not in general isomorphic to $(u \in \partial\mathbb{U})_{\Xi_1}$.

If we had modal type formers for *left* adjoints, then we could define the boundary predicate as $(\top, \blacksquare_{\Omega(u \in \partial\mathbb{U})})$ or $\Sigma(u \in \partial\mathbb{U})\,(\top)$ in lockless notation. However, MTT does not support such type formers and we do not know how to do this[20] so we simply axiomatize

[20] It is worth noting that $\Sigma^{\sigma|}$ is a parametric right adjoint so the work by Gratzer et al. [GCK+21] may be relevant.

the predicate by decreeing for every type $\Xi, u : \mathbb{U} \mid \Gamma \vdash A\,\mathsf{type}$ an isomorphism

$$(u \in \partial\mathbb{U}) \to A \quad \cong \quad \langle \Pi(u \in \partial\mathbb{U}) \mid^{\mathrm{p}} \langle \Omega(u \in \partial\mathbb{U}) \mid^{\mathrm{o}} A[\mathsf{const}_u{\downarrow}^{\bullet}_{\mathrm{po}}] \rangle \rangle \qquad \text{(lockful)}$$

$$(u \in \partial\mathbb{U}) \to A \quad \cong \quad \langle \Pi(u \in \partial\mathbb{U}) \mid^{\mathrm{o}} \langle \Omega(u \in \partial\mathbb{U}) \mid^{\mathrm{s}} A[\mathsf{drop}_u{\uparrow}^{\bullet}_{\mathrm{so}}] \rangle \rangle \qquad \text{(lockless)}.$$

In practice, for concrete systems, we will want axioms based on our findings in section 6.3, e.g. in a binary cubical system we would decree $(i \in \partial\mathbb{I}) \cong (i \equiv_{\mathbb{I}} 0) \vee (i \equiv_{\mathbb{I}} 1)$ where the latter two predicates could be axiomatized as above.

8.3. **Strictness axiom.** The strictness axiom [OP18] allows to extend a partial type T to a total type if T is isomorphic to a total type A, effectively strictifying the isomorphism:

$$\frac{\Xi \mid \Gamma \vdash \varphi : \mathsf{Prop} \qquad \Xi \mid \Gamma \vdash A : \mathsf{U}_\ell \qquad \Xi \mid \Gamma, \_ : \varphi \vdash T : \mathsf{U}_\ell \qquad \Xi \mid \Gamma, \_ : \varphi \vdash i : A \cong T}{\begin{array}{l}\Xi \mid \Gamma \vdash \mathsf{Strict}\{A \cong (\varphi\ ?\ T\ ;\ i)\} : \mathsf{U}_\ell \qquad \Xi \mid \Gamma \vdash \mathsf{strict}\{\varphi\ ?\ i\} : A \cong \mathsf{Strict}\{A \cong (\varphi\ ?\ T\ ;\ i)\} \\ \text{where } \Gamma, \_ : \varphi \vdash \mathsf{Strict}\{A \cong (\varphi\ ?\ T\ ;\ i)\} = T : \mathsf{U}_\ell \qquad \Gamma, \_ : \varphi \vdash \mathsf{strict}\{\varphi\ ?\ i\} = i : A \cong T\end{array}}$$

9. Investigating the Transpension Type

In section 2.2, we have briefly investigated the structure of the transpension type in the naïve type system. In this section, we investigate the structure of the transpension type $\langle \between u \mid^{\mathrm{t}} A \rangle$ that exists in our specific instance of MTT.

9.1. **Poles.** Our first observation is that on the boundary, the transpension type is trivial. Let $\top : \Xi_1 \to \Xi_2$ be the modality, between any two modes, which maps any presheaf to the terminal presheaf. We clearly have $\top \circ \mu = \top$ for any μ, but also $\mu \circ \top \cong \top$ because all internal modalities are right adjoints and therefore preserve the terminal object.

Theorem 9.1 (Pole)**.** *We have* $\Omega(u \in \partial\mathbb{U}) \circ \between u \cong \top$*. We can thus postulate a term* $\Xi, u : \mathbb{U} \mid \Gamma, \_ : u \in \partial\mathbb{U} \vdash \mathsf{pole} : \langle \between u \mid^{\mathrm{t}} T \rangle$ *for any* $\Xi \mid \Gamma, \mathbin{🔒}^{\mathrm{t}}_{\between u} \vdash T\,\mathsf{type}$*, with an* η*-rule* $\Xi, u : \mathbb{U} \mid \Gamma, \_ : u \in \partial\mathbb{U} \vdash t = \mathsf{pole} : \langle \between u \mid^{\mathrm{t}} T \rangle$.

Sketch of proof. The left adjoints $\forall(u : \mathbb{U}) \circ \Sigma(u \in \partial\mathbb{U})$ and $\bot$ of the concerned modalities are isomorphic because $\forall(u : \mathbb{U}).(u \in \partial\mathbb{U})$ is false. We give a full proof in the technical report [Nuy20b]. □

Definition 6.5 of the boundary relied on the notion of dimensional splitness. The following result shows that it was a good one: the transpension is *only* trivial on the boundary:

Theorem 9.2 (Boundary)**.** *In the model, we have* [Nuy20b]

$$\Xi, u : \mathbb{U} \mid \Gamma \vdash (u \in \partial\mathbb{U}) \cong \langle \between (u : \mathbb{U}) \mid^{\mathrm{t}} \mathsf{Empty} \rangle.$$

9.2. **Meridians.** As all our modalities are proper DRAs [BCM+20], the modal introduction rule is invertible in the model. This immediately shows that sections[21] of the transpension type

$$\Xi \mid \Gamma \vdash f : \left\langle \forall(u : \mathbb{U}) \mid^{\mathrm{a}} \left\langle \textcircled{\S}(u : \mathbb{U}) \mid^{\mathrm{t}} T \right\rangle \right\rangle \qquad \text{(lockful)}$$

$$\Xi \mid \Gamma \vdash f : \left\langle \forall(u : \mathbb{U}) \mid^{\mathrm{f}} \left\langle \textcircled{\S}(u : \mathbb{U}) \mid^{\mathrm{a}} T \right\rangle \right\rangle \qquad \text{(lockless)}$$

(which we call meridians) are in 1-1 correspondence with terms

$$\Xi \mid \Gamma, \blacklozenge^{\mathrm{a}}_{\forall\, u}, \blacklozenge^{\mathrm{t}}_{\textcircled{\S}\, u} \vdash t : T \qquad \text{(lockful)}$$

$$\Xi \mid \forall\, u^{\,\mathrm{a}} \left(\exists\, u^{\,\mathrm{f}} \left(\Gamma \right) \right) \vdash t : T \qquad \text{(lockless).}$$

If it were not for the locking of the context, this characterization in terms of poles and meridians would make the transpension type look quite similar to a dependent version of the suspension type in HoTT [Uni13], whence our choice of name. If $\mathbb{U}$ is cancellative and affine, then the applied locks are actually isomorphic to the identity lock (theorem 6.27). In any case, regardless of the properties of $\mathbb{U}$, proposition 3.3 tells us that the let-rule for $\textcircled{\S}(u : \mathbb{U})$ has the same power as

$$\mathsf{prmod}_{\textcircled{\S}(u:\mathbb{U})} : (\forall(u : \mathbb{U}) \shortmid^{\mathrm{a}} \left\langle \textcircled{\S}(u : \mathbb{U}) \mid^{\mathrm{t}} T \right\rangle) \to T[\mathsf{unmer}_u{\downarrow}^{\mathrm{at}}_{\bullet}] \qquad \text{(lockful)}$$

$$\mathsf{unmer}_u : (\forall(u : \mathbb{U}) \shortmid^{\mathrm{f}} \left\langle \textcircled{\S}(u : \mathbb{U}) \mid^{\mathrm{a}} T \right\rangle) \to T[\mathsf{const}_u{\uparrow}^{\mathrm{af}}_{\bullet}] \qquad \text{(lockless)}$$

which extracts meridians. If $\mathbb{U}$ is cancellative and affine, then the 2-cell unmer_u is invertible (theorem 6.27) and we can also straightforwardly *create* meridians from elements of $T[\mathsf{unmer}_u{\downarrow}^{\mathrm{at}}_{\bullet}]$.

9.3. **Pattern matching.** The eliminator $\mathsf{prmod}_{\textcircled{\S}(u:\mathbb{U})}$ is only capable of eliminating *sections* of the transpension type. If the kernel theorem 6.24 applies to $\mathbb{U}$, we can eliminate locally by pattern matching:

Theorem 9.3. *If $\mathbb{U}$ is cancellative, affine and connection-free, then the rule* TRANSP:ELIM *in fig. 8 is sound* [Nuy20b].

The elimination rule is best understood by looking at the lockless notation. We get a context Γ depending on $u : \mathbb{U}$, a type A depending on sections of Γ, a type C depending on u and $r : \langle \textcircled{\S}\, u \mid^{\mathrm{a}} A \rangle$, and an argument t of type $\langle \textcircled{\S}\, u \mid^{\mathrm{a}} A \rangle$. To obtain a value of type C, we need to give an action c_{pole} on the boundary, where t is necessarily pole (pole theorem 9.1), and a compatible action on sections of the transpension type, i.e. meridians, which live over sections of Γ but are themselves essentially elements of A (quantification theorem 6.27), producing sections of C (but the quantifier $\forall\, u$ has been brought to the left as $\blacklozenge_{\forall\, u} = \exists\, u$). Thanks to connection-freedom, we know that everything that is not a section[22], is on the boundary, so this suffices.

The computation rule for meridians fires when all of Γ is fresh for u. In this situation, the judgement for t is $\Xi, u : \mathbb{U} \mid \exists\, u^{\,\mathrm{f}}\, (\Delta) \vdash t : \langle \textcircled{\S}\, u \mid^{\mathrm{a}} A \rangle$ which by transposition boils down to $\Xi \mid \Delta \vdash t' : \left\langle \forall\, a \mid^{\mathrm{f}} \langle \textcircled{\S}\, u \mid^{\mathrm{a}} A \rangle \right\rangle$, i.e. it fires when t can actually be seen as a full section of the transpension type, so that we can apply the action on sections given by c_{mer}. This computation rule is in a non-general context and needs to be forcibly closed under

[21] By a section of a dependent type, we mean a dependent function with the same domain as the type.

[22] or a 'dimensional section' in case of spooky base categories

In lockful MTT notation:

TRANSP:ELIM

$$\frac{\begin{array}{l}\Xi, u : \mathbb{U} \mid \Gamma\ \mathsf{ctx} \\ \Xi \mid \Gamma, \mathbin{🔒}^{\mathrm{t}}_{\between u} \vdash A\ \mathsf{type} \\ \Xi, u : \mathbb{U} \mid \Gamma, r : \langle \between u \mid^{\mathrm{t}} A \rangle \vdash C\ \mathsf{type} \\ \Xi, u : \mathbb{U} \mid \Gamma, \_ : u \in \partial\mathbb{U} \vdash c_{\mathsf{pole}} : C[\mathsf{pole}/r] \\ \Xi, u : \mathbb{U} \mid \Gamma, \mathbin{🔒}^{\mathrm{t}}_{\between u}, x : A, \mathbin{🔒}^{\mathrm{a}}_{\forall u} \vdash c_{\mathsf{mer}} : C\left[\mathrm{id}_\Gamma, \mathsf{reidx}_u{\downarrow}^{\bullet}_{\mathrm{ta}}, \mathsf{mod}^{\mathrm{t}'}_{\between u}\,(x\,\mathsf{unmer}_u^{-1}{\downarrow}_{\mathrm{at}'})/r\right] \\ \Xi, u : \mathbb{U} \mid \Gamma, \mathbin{🔒}^{\mathrm{t}}_{\between u}, x : A, \mathbin{🔒}^{\mathrm{a}}_{\forall u}, \_ : u \in \partial\mathbb{U} \vdash c_{\mathsf{mer}} = c_{\mathsf{pole}}[\mathrm{id}_\Gamma, \mathsf{reidx}_u{\downarrow}^{\bullet}_{\mathrm{ta}}] : C[\mathrm{id}_\Gamma, \mathsf{reidx}_u{\downarrow}^{\bullet}_{\mathrm{ta}}, \mathsf{pole}/r] \\ \Xi, u : \mathbb{U} \mid \Gamma \vdash t : \langle \between_u \mid^{\mathrm{t}} A \rangle \end{array}}{\begin{array}{l}\Xi, u : \mathbb{U} \mid \Gamma \vdash c := \mathsf{case}\, t\, \mathsf{of} \left\{\ \mathsf{pole} \mapsto c_{\mathsf{pole}} \mid \mathsf{mer}(\mathrm{t}, x, \mathrm{a}) \mapsto c_{\mathsf{mer}}\ \right\} : C[t/r] \\ \quad \text{where } c[\mathsf{pole}/t] = c_{\mathsf{pole}} \\ \qquad \Xi, u : \mathbb{U} \mid \Delta, \mathbin{🔒}^{\mathrm{a}'}_{\forall u} \vdash c = c_{\mathsf{mer}}[\mathrm{id}_\Delta, \mathsf{unmer}_u{\downarrow}^{\mathrm{a}'\mathrm{t}}_{\bullet}, \mathsf{prmod}_{\between u}\,\cdot^{\mathrm{a}'}\, t/x, {\downarrow}^{\mathrm{a}}_{\mathrm{a}'}] : C\end{array}}$$

In lockless MTT notation:

$$\frac{\begin{array}{l}\Xi, u : \mathbb{U} \mid \Gamma\ \mathsf{ctx} \\ \Xi \mid \forall(u : \mathbb{U})^{\,\mathrm{a}}\,(\Gamma) \vdash A\ \mathsf{type} \\ \Xi, u : \mathbb{U} \mid \Gamma, r : \langle \between u \mid^{\mathrm{a}} A \rangle \vdash C\ \mathsf{type} \\ \Xi, u : \mathbb{U} \mid \Gamma, \_ : u \in \partial\mathbb{U} \vdash c_{\mathsf{pole}} : C[\mathsf{pole}/r] \\ \Xi, u : \mathbb{U} \mid \lrcorner\, u^{\,\mathrm{f}}\,(\forall(u : \mathbb{U})^{\,\mathrm{a}}\,(\Gamma), x : A) \vdash c_{\mathsf{mer}} : C\left[\mathsf{app}_u{\uparrow}^{\bullet}_{\mathrm{fa}}(\mathrm{id}_\Gamma), \mathsf{mer}^{\mathrm{a}'}_u\,(x\,\mathsf{const}_u^{-1}{\uparrow}_{\mathrm{a}'\mathrm{f}})/r\right] \\ \Xi, u : \mathbb{U} \mid \lrcorner\, u^{\,\mathrm{f}}\,(\forall(u : \mathbb{U})^{\,\mathrm{a}}\,(\Gamma), x : A), \_ : u \in \partial\mathbb{U} \vdash c_{\mathsf{mer}} = c_{\mathsf{pole}}[\mathsf{app}_u{\uparrow}^{\bullet}_{\mathrm{fa}}(\mathrm{id}_\Gamma)] : C[\mathsf{app}_u{\uparrow}^{\bullet}_{\mathrm{fa}}(\mathrm{id}_\Gamma), \mathsf{pole}/r] \\ \Xi, u : \mathbb{U} \mid \Gamma \vdash t : \langle \between_u \mid^{\mathrm{a}} A \rangle \end{array}}{\begin{array}{l}\Xi, u : \mathbb{U} \mid \Gamma \vdash c := \mathsf{case}\, t\, \mathsf{of} \left\{\ \mathsf{pole} \mapsto c_{\mathsf{pole}} \mid \mathsf{mer}(\mathrm{a}, x, \mathrm{f}) \mapsto c_{\mathsf{mer}}\ \right\} : C[t/r] \\ \quad \text{where } c[\mathsf{pole}/t] = c_{\mathsf{pole}} \\ \qquad \Xi, u : \mathbb{U} \mid \lrcorner\, u^{\,\mathrm{f}'}\,(\Delta) \vdash c = c_{\mathsf{mer}}[{\uparrow}^{\mathrm{f}}_{\mathrm{f}'}(\mathsf{const}_u{\uparrow}^{\mathrm{af}'}_{\bullet}(\mathrm{id}_\Delta), \mathsf{unmer}_u\,\cdot^{\mathrm{f}'}\, t/x)] : C\end{array}}$$

Figure 8: Transpension elimination by pattern matching (sound if $\mathbb{U}$ is cancellative, affine and connection-free).

substitution (we have not found a better way to phrase this computation rule, but neither do we exclude that there exists one).

10. Recovering Known Operators

In this section, we explain how to recover the amazing right adjoint $\surd$ [LOPS18], BCM's Φ and Ψ combinators [BCM15, Mou16], Glue [CCHM17, NVD17], Weld [NVD17] and mill [ND18b] and locally fresh names [PMD15] from the transpension type, the strictness axiom [OP18] and certain pushouts.

10.1. **The amazing right adjoint $\surd$.** Licata et al. [LOPS18] use presheaves over a cartesian base category of cubes and introduce $\surd$ as the right adjoint to the non-dependent exponential $\mathbb{I} \to \sqcup$. We generalize to *semicartesian* systems and look for a right adjoint to $\mathbb{U} \multimap \sqcup$, which decomposes as substructural quantification after cartesian weakening $\forall(u : \mathbb{U}) \circ \Omega(u : \mathbb{U})$. Then the right adjoint is obviously $\surd_{\mathbb{U}} := \Pi(u : \mathbb{U}) \circ \between(u : \mathbb{U})$. The type constructor has type $\langle \surd_{\mathbb{U}} \mid \sqcup \rangle : (\surd_{\mathbb{U}} \,{}^{\mathrm{r}}\, \mathsf{U}_\ell) \to \mathsf{U}_\ell$ and the transposition rule is as in proposition 3.4. This is an improvement in two ways: First, we have introduction, elimination and computation rules, so that we do not need to postulate functoriality of $\surd_{\mathbb{U}}$ and invertibility of transposition.

Secondly, we have no need for a global sections modality $\flat$. Instead, we use the modality $\surd_{\mathbb{U}}$ to escape Licata et al.'s no-go theorems.

Our overly general mode theory does contain a global sections modality $\flat : () \to ()$ acting in the empty shape context, and we can use this to recover Licata et al.'s axioms for the amazing right adjoint. Let us write $\mathsf{spconst}_u : 1 \Rightarrow \forall\, u \circ \Omega\, u$ and $\mathsf{spunmer}_u : \Pi\, u \circ \rotatebox{180}{$\flat$}\, u \Rightarrow 1$ for the 2-cells built by transposition from either spoil_u or $\mathsf{cospoil}_u$ (theorem 6.27), which are each other's transposite. The following are isomorphisms:

$$\kappa := \mathsf{spconst}_u \star 1_\flat : \flat \cong \forall\, u \circ \Omega\, u \circ \flat,$$
$$\zeta := 1_\flat \circ \mathsf{spunmer}_u : \flat \circ \Pi\, u \circ \rotatebox{180}{$\flat$}\, u \cong \flat.$$

For κ, this is intuitively clear from the fact that we are considering $\mathbb{U}$-cells in a discrete presheaf produced by $\flat$. For ζ, this is similarly clear after taking the left adjoints:

$$\mathsf{spconst}_u \star 1_\int : \int \cong \forall\, u \circ \Omega\, u \circ \int$$

where $\int = \text{🔒}_\flat$ is the connected components functor which also produces discrete presheaves. Write $\varepsilon : \flat \Rightarrow 1$ for the co-unit of the comonad $\flat$. We can define

$$\mathbb{U}\surd\sqcup : (\flat \mid^{\mathrm{b}} \mathsf{U}) \to \mathsf{U} \qquad \mathbb{U} \multimap \sqcup : \mathsf{U} \to \mathsf{U}$$
$$\mathbb{U}\surd^{\mathrm{b}} A = \left\langle \Pi\, u \circ \rotatebox{180}{$\flat$}\, u \mid^{\mathrm{pt}} A[\zeta^{-1}\!\downarrow^{\mathrm{b}}_{\mathrm{bpt}}][\varepsilon\!\downarrow^{\mathrm{b}}_{\bullet}, \downarrow^{\mathrm{pt}}_{\mathrm{pt}}] \right\rangle \qquad \mathbb{U} \multimap A = \langle \forall\, u \circ \Omega\, u \mid^{\mathrm{ao}} A[\mathsf{spconst}_u\!\downarrow^{\bullet}_{\mathrm{ao}}] \rangle.$$

Let two global types $\top \mid \Gamma, \text{🔒}^{\mathrm{b}}_\flat \vdash A, B\ \mathsf{type}$ be given. Applying the non-dependent version of proposition 3.4 to $\forall\, u \circ \Omega\, u \dashv \surd_{\mathbb{U}} = \Pi\, u \circ \rotatebox{180}{$\flat$}\, u$ yields:

$$\begin{aligned}
&\left(A[\varepsilon\!\downarrow^{\mathrm{b}}_{\bullet}] \to \mathbb{U}\surd^{\mathrm{b}} B\right) \\
&\cong \left\langle \surd_{\mathbb{U}} \mid^{\mathrm{pt}} (\forall\, u \circ \Omega\, u \mid^{\mathrm{ao}} A[\varepsilon\!\downarrow^{\mathrm{b}}_{\bullet}][\mathsf{const}_u\!\downarrow^{\bullet}_{\mathrm{po}}][\downarrow^{\mathrm{p}}_{\mathrm{p}}, \mathsf{reidx}_u\!\downarrow^{\bullet}_{\mathrm{ta}}, \downarrow^{\mathrm{o}}_{\mathrm{o}}]) \to B[\zeta^{-1}\!\downarrow^{\mathrm{b}}_{\mathrm{bpt}}][\varepsilon\!\downarrow^{\mathrm{b}}_{\bullet}, \downarrow^{\mathrm{pt}}_{\mathrm{pt}}] \right\rangle \\
&= \left\langle \surd_{\mathbb{U}} \mid^{\mathrm{pt}} (\forall\, u \circ \Omega\, u \mid^{\mathrm{ao}} A[\mathsf{spconst}_u\!\downarrow^{\bullet}_{\mathrm{ao}}])[\zeta^{-1}\!\downarrow^{\mathrm{b}}_{\mathrm{bpt}}][\varepsilon\!\downarrow^{\mathrm{b}}_{\bullet}, \downarrow^{\mathrm{pt}}_{\mathrm{pt}}] \to B[\zeta^{-1}\!\downarrow^{\mathrm{b}}_{\mathrm{bpt}}][\varepsilon\!\downarrow^{\mathrm{b}}_{\bullet}, \downarrow^{\mathrm{pt}}_{\mathrm{pt}}] \right\rangle \\
&= \mathbb{U}\surd^{\mathrm{b}} ((\mathbb{U} \multimap A) \to B).
\end{aligned}$$

Equality of the substitutions applied to A is proven by transposing $\forall\, u \circ \Omega\, u$ to the left as $\Pi\, u \circ \rotatebox{180}{$\flat$}\, u$. Then the unit $\mathsf{const}_u \circ (1_{\Pi\, u} \star \mathsf{reidx}_u \star 1_{\Omega\, u}) : 1 \Rightarrow \Pi\, u \circ \rotatebox{180}{$\flat$}\, u \circ \forall\, u \circ \Omega\, u$ becomes $1_{\Pi\, u \circ \rotatebox{180}{$\flat$}\, u}$, leaving just $\varepsilon : \flat \Rightarrow 1$, whereas $\mathsf{spconst}_u : 1 \Rightarrow \forall\, u \circ \Omega\, u$ becomes $\mathsf{spunmer}_u : \Pi\, u \circ \rotatebox{180}{$\flat$}\, u \Rightarrow 1$ and cancels against ζ^{-1}, again leaving just ε. Applying the $\flat$ modality to both sides of the isomorphism and using ζ yields transposition functions as given by Licata et al. [LOPS18].

We refer back to section 6.7 for the opposite construction: a transpension type for a cartesian multiplier can be constructed from an amazing right adjoint [Yet87].

10.2. **The Φ-combinator.** In fig. 9, we *state* BCM's Φ-rule [BCM15, Mou16], also known as extent [CH20]; both a slight reformulation adapted to the naïve system (section 2) and the rule PHI adapted to the current MTT system in lockful and lockless notation.

In the binary version of the BCM system, or in the naïve system with an interval shape as in cubical type theory, the Φ-combinator allows us to define functions of naïve type $\forall\, i.(y : B\, i) \to C\, i\, y$ from an action c_ϵ at every endpoint ϵ and a compatible action $c_\forall$ on sections $\forall\, i.B\, i$. When the resulting function $\Phi\, c_0\, c_1\, c_\forall$ is applied to an endpoint $\epsilon : \mathbb{I}$ it just reduces to the corresponding action $\lambda y.c_\epsilon$. When it is applied to an interval variable $i : \mathbb{I}$

Binary and destrictified reformulation of the original Φ-rule [BCM15, Mou16]**:**

$$\frac{\begin{array}{l}\Delta, i : \mathbb{I} \vdash B\,\mathsf{type}\\ \Delta, i : \mathbb{I}, y : B \vdash C\,\mathsf{type}\\ \Delta, y : B[\epsilon/i] \vdash c_\epsilon : C[\epsilon/i] \qquad (\epsilon \in \{0,1\})\\ \Delta, h : \forall(i : \mathbb{I}).B, i : \mathbb{I} \vdash c_\forall : C[h\,i/b]\\ \Delta, h : \forall(i : \mathbb{I}).B \vdash c_\forall[\epsilon/i] = c_\epsilon[h\,\epsilon/b] : C[h\,\epsilon/b] \quad (\epsilon \in \{0,1\})\end{array}}{\begin{array}{l}\Delta \vdash \Phi\,c_0\,c_1\,c_\forall : \forall\, i.(y : B) \to C\\ \text{where } \Phi\,c_0\,c_1\,c_\forall\,\epsilon\,b = c_\epsilon[b/y] \qquad (\epsilon \in \{0,1\})\\ \qquad \Delta, i : \mathbb{I} \vdash \Phi\,c_0\,c_1\,c_\forall\,i\,b = c_\forall[\lambda i.b/h]\end{array}}$$

In lockful MTT notation:

$$\frac{\begin{array}{l}\Xi, u : \mathbb{U} \mid \Gamma \vdash C\,\mathsf{type}\\ \Xi, u : \mathbb{U} \mid \Gamma, \_ : u \in \partial\mathbb{U} \vdash c_\partial : C\\ \Xi, u : \mathbb{U} \mid \Gamma, \text{🔒}^{\mathrm{t}}_{\lozenge\, u}, \text{🔒}^{\mathrm{a}}_{\forall\, u} \vdash c_\forall : C[\mathsf{reidx}_u{\downarrow}^{\bullet}_{\mathrm{ta}}]\\ \Xi, u : \mathbb{U} \mid \Gamma, \text{🔒}^{\mathrm{t}}_{\lozenge\, u}, \text{🔒}^{\mathrm{a}}_{\forall\, u}, \_ : u \in \partial\mathbb{U} \vdash c_\forall = c_\partial[\mathsf{reidx}_u{\downarrow}^{\bullet}_{\mathrm{ta}}] : C[\mathsf{reidx}_u{\downarrow}^{\bullet}_{\mathrm{ta}}]\end{array}}{\begin{array}{l}\Xi, u : \mathbb{U} \mid \Gamma \vdash \Phi_u\,c_\partial\,(\mathrm{t}.\mathrm{a}.c_\forall) : C\\ \text{where } \Xi, u : \mathbb{U} \mid \Gamma, \_ : u \in \partial\mathbb{U} \vdash \Phi_u\,c_\partial\,(\mathrm{t}.\mathrm{a}.c_\forall) = c_\partial : C\\ \qquad \Xi, u : \mathbb{U} \mid \Delta, \text{🔒}^{\mathrm{a}'}_{\forall\, u} \vdash \Phi_u\,c_\partial\,(\mathrm{t}.\mathrm{a}.c_\forall) = c_\forall[(\mathsf{unmer}_u \star 1_{\forall u}){\downarrow}^{\mathrm{a}'\mathrm{ta}}_{\mathrm{a}'}] : C\end{array}}$$

In lockless MTT notation:

$$\frac{\begin{array}{l}\Xi, u : \mathbb{U} \mid \Gamma \vdash C\,\mathsf{type}\\ \Xi, u : \mathbb{U} \mid \Gamma, \_ : u \in \partial\mathbb{U} \vdash c_\partial : C\\ \Xi, u : \mathbb{U} \mid \lrcorner\, u^{\,\mathrm{f}}\,(\forall\, u^{\,\mathrm{a}}\,(\Gamma)) \vdash c_\forall : C[\mathsf{app}_u{\uparrow}^{\bullet}_{\mathrm{fa}}]\\ \Xi, u : \mathbb{U} \mid \lrcorner\, u^{\,\mathrm{f}}\,(\forall\, u^{\,\mathrm{a}}\,(\Gamma)), \_ : u \in \partial\mathbb{U} \vdash c_\forall = c_\partial[\mathsf{app}_u{\uparrow}^{\bullet}_{\mathrm{fa}}] : C[\mathsf{app}_u{\uparrow}^{\bullet}_{\mathrm{fa}}]\end{array}}{\begin{array}{l}\Xi, u : \mathbb{U} \mid \Gamma \vdash \Phi_u\,c_\partial\,(\mathrm{a}.\mathrm{f}.c_\forall) : C\\ \text{where } \Xi, u : \mathbb{U} \mid \Gamma, \_ : u \in \partial\mathbb{U} \vdash \Phi_u\,c_\partial\,(\mathrm{a}.\mathrm{f}.c_\forall) = c_\partial : C\\ \qquad \Xi, u : \mathbb{U} \mid \lrcorner\, u^{\,\mathrm{f}'}\,(\Delta) \vdash \Phi_u\,c_\partial\,(\mathrm{a}.\mathrm{f}.c_\forall) = c_\forall[(1_{\lrcorner\, u} \star \mathsf{const}_u){\uparrow}^{\mathrm{faf}'}_{\mathrm{f}'}] : C\end{array}}$$

Figure 9: The Φ-rule (sound if $\mathbb{U}$ is cancellative, affine and connection-free).

and expression b that depends only on i and variables fresh for i, then the variable i can be captured in b yielding a section $\lambda i.b : \forall\, i.B\,i$ to which we can apply the action on sections $c_\forall$.

A few remarks are necessary in order to move to MTT. First of all, note that the fully applied conclusion of the Φ-rule is $\Delta, i : \mathbb{I}, y : B \vdash \Phi\,c_0\,c_1\,c_\forall\,i\,y : C$. Of course the endpoints together constitute the boundary of the interval, so if we want to generalize away from cubical type theory, we can expect something like $\Delta, u : \mathbb{U}, y : B \vdash \Phi\,c_\partial\,c_\forall\,u\,y : C$. We will assume that the shape $\mathbb{U}$ is cancellative, affine and connection-free. In order to translate the context $(\Delta, u : \mathbb{U}, y : B)$ to MTT, recall that in the naïve system (as well as in the BCM system) the variables to the left of u are fresh for u, whereas those to the right need not be. In MTT we put the shape variable in the shape context, so the rest of the context translates to $(\Delta, \text{🔒}^{\mathrm{a}}_{\forall\, u}, y : B) = (\lrcorner\, u^{\,\mathrm{f}}\,(\Delta), y : B)$. In MTT we will treat this entire thing as a single abstract context Γ, so we do not grant the domain of the Φ-function any special status; in a way all of Γ takes the role of B.

Then, completely analogously to BCM, we need a type C in context Γ, an object $c_\partial : C$ whenever u is on the boundary, and a compatible action $c_\forall$ that acts on sections of Γ and

produces sections of C, but the quantifier $\forall\, u$ has been brought to the left as $🔒_{\forall\, u} = \lrcorner\, u$. Again the resulting term $\Phi_u\, c_\partial\, (\mathrm{a}.\mathrm{f}.c_\forall)$ reduces to c_∂ when on the boundary, and to $c_\forall$ when all variables in use are fresh for u. Again, the computation rule for sections is in a non-general context and needs to be forcibly closed under substitution.

Note that the naïve/BCM substitutions $h\, i/b$ and $h\, \epsilon/b$ (i.e. $h\, i/b$ where i is on the boundary) all turn into usages of app_u whereas the variable capturing substitution $\lambda i.b/h$ has turned into $1_{\lrcorner\, u} \star \mathsf{const}_u$ which for cancellative and affine multipliers is inverse to $\mathsf{app}_u \star 1_{\lrcorner\, u}$ and therefore denotes an inverse of application: variable capture. Furthermore, regarding the context of $c_\forall$, we remark that for cancellative and affine multipliers there is an isomorphism of contexts

$$\begin{aligned}(\Delta, 🔒^{\mathrm{a}'}_{\forall\, u}, y : B, 🔒^{\mathrm{t}}_{\langle\!|\, u}, 🔒^{\mathrm{a}}_{\forall\, u}) &\cong_{3.4} (\Delta, 🔒^{\mathrm{a}'}_{\forall\, u}, 🔒^{\mathrm{t}}_{\langle\!|\, u}, \forall\, u \shortmid y :^{\mathrm{a}} B[1_{\forall\, u} \star \mathsf{reidx}_u{\downarrow}^{\mathrm{a}'}_{\mathrm{a}'\mathrm{ta}}], 🔒^{\mathrm{a}}_{\forall\, u}) \\ &\cong_{6.27} (\Delta, \forall\, u \shortmid y :^{\mathrm{a}} B, 🔒^{\mathrm{a}}_{\forall\, u}) \\ \lrcorner\, u^{\mathrm{f}}\left(\forall\, u^{\mathrm{a}}\left(\lrcorner\, u^{\mathrm{f}'}\,(\Delta), y : B\right)\right) &\cong_{3.4} \lrcorner\, u^{\mathrm{f}}\left(\forall\, u^{\mathrm{a}}\left(\lrcorner\, u^{\mathrm{f}'}\,(\Delta)\right), \forall\, u \shortmid y :^{\mathrm{f}} B[\mathsf{app}_u \star 1_{\lrcorner\, u}{\uparrow}^{\mathrm{f}'}_{\mathrm{faf}'}]\right) \\ &\cong_{6.27} \lrcorner\, u^{\mathrm{f}}\,(\Delta, \forall\, u \shortmid y :^{\mathrm{a}} B)\end{aligned}$$

so there really is a close correspondence to what happens in the BCM system.

We remark that if $C = \langle\!\langle\, u \mid^{\mathrm{t}} D\rangle\!\rangle$ and $\mathbb{U}$ is cancellative and affine but not necessarily connection-free, then the Φ-rule follows from the introduction rule of the transpension type by pole theorem 9.1 and quantification theorem 6.27. Indeed, we can then define:

$$\begin{aligned}\Phi_u\, \mathsf{pole}\, (\mathrm{t}.\mathrm{a}.c_\forall) &:= \mathsf{mod}^{\mathrm{t}}_{\langle\!|\, u}\, (\mathsf{prmod}_{\langle\!|\, u} \cdot^{\mathrm{a}} c_\forall) : \langle\!\langle\, u \mid^{\mathrm{t}} D\rangle\!\rangle \\ \Phi_u\, \mathsf{pole}\, (\mathrm{a}.\mathrm{f}.c_\forall) &:= \mathsf{mer}^{\mathrm{a}}_u\, (\mathsf{unmer}_u \cdot^{\mathrm{f}} c_\forall) : \langle\!\langle\, u \mid^{\mathrm{t}} D\rangle\!\rangle.\end{aligned}$$

Theorem 10.1. *If $\mathbb{U}$ is cancellative, affine and connection-free, then the Φ-rule (fig. 9) is sound and indeed derivable from* TRANSP:ELIM *for all C.*

Proof. Use the case-eliminator for $\langle\!\langle\, u \mid^{\mathrm{t}} \mathsf{Unit}\rangle\!\rangle$ (theorem 9.3):

$$\Phi_u\, c_\partial\, (\mathrm{t}.\mathrm{a}.c_\forall) := \mathsf{case}\, (\mathsf{mod}^{\mathrm{t}}_{\langle\!|\, u}\, \mathsf{unit})\, \mathsf{of} \left\{ \begin{array}{lcl} \mathsf{pole} & \mapsto & c_\partial \\ \mathsf{mer}(\mathrm{t}, \_, \mathrm{a}) & \mapsto & c_\forall \end{array} \right\}. \qquad \square$$

10.3. **The Ψ-type.** BCM's Ψ-combinator (fig. 10, also known as Gel [CH20]) constructs a line $\forall(i : \mathbb{I}).\mathsf{U}$ in the universe with endpoints A_ϵ from a relation $R : A_0 \to A_1 \to \mathsf{U}$. A section of the Ψ-type with endpoints a_ϵ is a proof of $R\, a_0\, a_1$. The constructor $\mathsf{in}\Psi$ creates a section $\forall(i : \mathbb{I}).\Psi_i\, A_0\, A_1\, (x_0.x_1.R)$ from the expected inputs. The disappearance of Θ in the premises of Ψ and $\mathsf{in}\Psi$ is entirely analogous to the naïve application rule in fig. 1. The eliminator $\mathsf{out}\Psi$ extracts from a section of the Ψ-type the proof that its endpoints satisfy the relation R.

Actually from the naïve transpension type in section 2 and a strictness axiom as in section 8.3 we can already implement a stronger Ψ-type, also given in fig. 10, where Θ does

Binary and destrictified reformulation of the original Ψ-type [BCM15, Mou16]**:**

$$\frac{\Delta \vdash A_\epsilon\,\mathsf{type}\quad(\epsilon\in\{0,1\})\qquad \Delta, x_0 : A_0, x_1 : A_1 \vdash R\,\mathsf{type}}{\Delta, i:\mathbb{I}, \theta:\Theta \vdash \Psi_i\, A_0\, A_1\, (x_0.x_1.R)\,\mathsf{type}}$$
$$\text{where } \Psi_\epsilon\, A_0\, A_1\, R = A_\epsilon \quad (\epsilon\in\{0,1\})$$

$$\frac{\Delta \vdash a_\epsilon : A_\epsilon\quad(\epsilon\in\{0,1\})\qquad \Delta \vdash r : R[a_0/x_0, a_1/x_1]}{\Delta, i:\mathbb{I}, \theta:\Theta \vdash \mathsf{in}\Psi_i\, a_0\, a_1\, r : \Psi_i\, A_0\, A_1\, (x_0.x_1.R)}$$
$$\text{where } \mathsf{in}\Psi_\epsilon\, a_0\, a_1\, r = a_\epsilon \quad (\epsilon\in\{0,1\})$$
$$\Delta, i:\mathbb{I} \vdash q = \mathsf{in}\Psi_i\, q[0/i]\, q[1/i]\, (\mathsf{out}\Psi(j.q[j/i]))$$

$$\frac{\Delta, i:\mathbb{I} \vdash q : \Psi_i\, A_0\, A_1\, (x_0.x_1.R)}{\Delta \vdash \mathsf{out}\Psi(i.q) : R[q[0/i]/x_0, q[1/i]/x_1]}$$
$$\text{where } \mathsf{out}\Psi(i.\mathsf{in}\Psi_i\, a_0\, a_1\, r) = r$$

Ψ-type based on the naïve transpension type from section 2:

$$\frac{\Delta, \theta:\Theta[\epsilon/i] \vdash A_\epsilon\,\mathsf{type}\quad(\epsilon\in\{0,1\})\qquad \Delta, \forall\, i.(\theta:\Theta), x_0 : A_0[{}^i\theta\, 0/\theta], x_1 : A_1[{}^i\theta\, 1/\theta] \vdash R\,\mathsf{type}}{\Delta, i:\mathbb{I}, \theta:\Theta \vdash \Psi_i\, A_0\, A_1\, (x_0.x_1.R)\,\mathsf{type}}$$
$$\text{where } \Psi_\epsilon\, A_0\, A_1\, R = A_\epsilon \quad (\epsilon\in\{0,1\})$$

$$\frac{\Delta, \theta:\Theta[\epsilon/i] \vdash a_\epsilon : A_\epsilon\quad(\epsilon\in\{0,1\})\qquad \Delta, \forall\, i.(\theta:\Theta) \vdash r : R[a_0[{}^i\theta\, 0/\theta]/x_0, a_1[{}^i\theta\, 1/\theta]/x_1]}{\Delta, i:\mathbb{I}, \theta:\Theta \vdash \mathsf{in}\Psi_i\, a_0\, a_1\, r : \Psi_i\, A_0\, A_1\, (x_0.x_1.R)}$$
$$\text{where } \mathsf{in}\Psi_\epsilon\, a_0\, a_1\, r = a_\epsilon \quad (\epsilon\in\{0,1\})$$
$$q = \mathsf{in}\Psi_i\, q[0/i]\, q[1/i]\, (\mathsf{out}\Psi(j.q[j/i, {}^i\theta\, j/\theta]))$$

$$\frac{\Delta, i:\mathbb{I} \vdash q : \Psi_i\, A_0\, A_1\, (x_0.x_1.R)}{\Delta \vdash \mathsf{out}\Psi(i.q) : R[q[0/i]/x_0, q[1/i]/x_1]}$$
$$\text{where } \mathsf{out}\Psi(i.\mathsf{in}\Psi_i\, a_0\, a_1\, r) = r$$

In lockful MTT notation:

PSI
$$\frac{\Xi, u:\mathbb{U} \mid \Gamma, \_ : u\in\partial\mathbb{U} \vdash A\,\mathsf{type}\qquad \Xi \mid \Gamma, \hat{x} : (\_ : u\in\partial\mathbb{U}) \to A, \blacksquare^{\mathrm{t}}_{\hat{Q}(u:\mathbb{U})} \vdash R\,\mathsf{type}}{\Xi, u:\mathbb{U} \mid \Gamma \vdash \Psi_u\, A\, (\hat{x}.\mathrm{t}.R)\,\mathsf{type}}$$
$$\text{where } \_ : u\in\partial\mathbb{U} \vdash \Psi_u\, A\, (\alpha.\mathrm{t}.R) = A$$

PSI:INTRO
$$\frac{\Xi, u:\mathbb{U} \mid \Gamma, \_ : u\in\partial\mathbb{U} \vdash a : A\qquad \Xi \mid \Gamma, \blacksquare^{\mathrm{t}}_{\hat{Q}(u:\mathbb{U})} \vdash r : R[\lambda\_.a/\hat{x}, \downarrow^{\mathrm{t}}_{\mathrm{t}}]}{\Xi, u:\mathbb{U} \mid \Gamma \vdash \mathsf{in}\Psi_u\, (\_.a)\, (\mathrm{t}.r) : \Psi_u\, A\, (\hat{x}.\mathrm{t}.R)}$$
$$\text{where } \_ : u\in\partial\mathbb{U} \vdash \mathsf{in}\Psi_u\, (\_.a)\, (\mathrm{t}.r) = a$$
$$q = \mathsf{in}\Psi_u\, (\_.q)\, (\mathrm{t}.\mathsf{out}\Psi \cdot^{\mathrm{a}} q[\mathsf{reidx}_u\downarrow^{\bullet}_{\mathrm{ta}}])$$

PSI:ELIM
$$\frac{\Xi, u:\mathbb{U} \mid \Delta, \blacksquare^{\mathrm{a}}_{\forall(u:\mathbb{U})} \vdash q : \Psi_u\, A\, (\hat{x}.\mathrm{t}.R)}{\Xi \mid \Delta \vdash \mathsf{out}\Psi \cdot^{\mathrm{a}} q : R[\lambda\_.q/\hat{x}, \downarrow^{\mathrm{t}}_{\mathrm{t}}][\mathsf{unmer}_u\downarrow^{\mathrm{at}}_{\bullet}]}$$
$$\text{where } \mathsf{out}\Psi \cdot^{\mathrm{a}} \mathsf{in}\Psi_u\, (\_.a)\, (\mathrm{t}.r) = r[\mathsf{unmer}_u\downarrow^{\mathrm{at}}_{\bullet}]$$

In lockless MTT notation:

$$\frac{\Xi, u:\mathbb{U} \mid \Gamma, \_ : u\in\partial\mathbb{U} \vdash A\,\mathsf{type}\qquad \Xi \mid \forall\, u^{\,\mathrm{a}}\, (\Gamma, \hat{x} : (\_ : u\in\partial\mathbb{U}) \to A) \vdash R\,\mathsf{type}}{\Xi, u:\mathbb{U} \mid \Gamma \vdash \Psi_u\, A\, (\hat{x}.\mathrm{a}.R)\,\mathsf{type}}$$
$$\text{where } \_ : u\in\partial\mathbb{U} \vdash \Psi_u\, A\, (\alpha.\mathrm{a}.R) = A$$

$$\frac{\Xi, u:\mathbb{U} \mid \Gamma, \_ : u\in\partial\mathbb{U} \vdash a : A\qquad \Xi \mid \forall\, u^{\,\mathrm{a}}\, (\Gamma) \vdash r : R[\uparrow^{\mathrm{a}}_{\mathrm{a}}(\lambda\_.a/\hat{x})]}{\Xi, u:\mathbb{U} \mid \Gamma \vdash \mathsf{in}\Psi_u\, (\_.a)\, (\mathrm{a}.r) : \Psi_u\, A\, (\hat{x}.\mathrm{a}.R)}$$
$$\text{where } \_ : u\in\partial\mathbb{U} \vdash \mathsf{in}\Psi_u\, (\_.a)\, (\mathrm{a}.r) = a$$
$$q = \mathsf{in}\Psi_u\, (\_.q)\, \left(\mathrm{a}.\mathsf{out}\Psi \cdot^{\mathrm{f}} q[\mathsf{app}_u\uparrow^{\bullet}_{\mathrm{fa}}]\right)$$

$$\frac{\Xi, u:\mathbb{U} \mid \exists\, u^{\,\mathrm{f}}\, (\Delta) \vdash q : \Psi_u\, A\, (\hat{x}.\mathrm{a}.R)}{\Xi \mid \Delta \vdash \mathsf{out}\Psi \cdot^{\mathrm{f}} q : R[\uparrow^{\mathrm{a}}_{\mathrm{a}}(\lambda\_.q/\hat{x})][\mathsf{const}_u\uparrow^{\mathrm{af}}_{\bullet}]}$$
$$\text{where } \mathsf{out}\Psi \cdot^{\mathrm{f}} \mathsf{in}\Psi_u\, (\_.a)\, (\mathrm{a}.r) = r[\mathsf{const}_u\uparrow^{\mathrm{af}}_{\bullet}]$$

Figure 10: Typing rules for the Ψ-type.

not disappear but gets universally quantified. This is done by strictifying the following:[23]

$$\Psi_i\, A_0\, A_1\, (x_0.x_1.R) :\cong (\hat{x}_0 : (\mathsf{refl} : i \equiv_{\mathbb{I}} 0) \to A_0) \times (\hat{x}_1 : (\mathsf{refl} : i \equiv_{\mathbb{I}} 1) \to A_1) \times \\ \lozenge\, i.(R[{}^i\hat{x}_0\, 0\, \mathsf{refl}/x_0, {}^i\hat{x}_1\, 1\, \mathsf{refl}/x_1])$$

$$\mathsf{in}\Psi_i\, a_0\, a_1\, r :\overset{\cong}{\longleftarrow} (\lambda\mathsf{refl}.a_0, \lambda\mathsf{refl}.a_1, \mathsf{mer}\, i\, r)$$

$$\mathsf{out}\Psi(i.q) :\overset{\cong}{\longleftarrow} \mathsf{unmer}(i.\pi_3(q))$$

The fact that this construction is isomorphic to A_ϵ at endpoint ϵ follows from our findings about poles in section 2.2.

When we move to MTT, once again we translate the context $(\Delta, u : \mathbb{U}, \Theta)$ to $(\Delta, \mathbin{🔒}^{\mathrm{a}}_{\forall\, u}, \Theta) = (\exists\, u^{\,\mathrm{f}}\, (\Delta), \Theta)$, which we treat as a single abstract context Γ. By a reasoning identical to that in section 10.2, applying $\mathbin{🔒}_{\lozenge\, u} = \forall\, u$ only affects the non-fresh part Θ if the shape $\mathbb{U}$ is cancellative and affine. This leads to the MTT rules listed in fig. 10. Note again how substitutions ${}^i\theta\, j/\theta$ in the naïve system give rise to usages of app_u in MTT. The usages of const_u are entirely absent in the naïve system, but for cancellative and affine multipliers, this is an isomorphism anyway.

The eliminator $\mathsf{out}\Psi$ only eliminates sections. For affine, cancellative and connection-free multipliers, the Φ-rule provides a pattern-matching eliminator which lets us treat the boundary and section cases separately.

Theorem 10.2. *For* any *multiplier, the Ψ-type in fig. 10 is implementable from the transpension type and the strictness axiom.*

Proof. We strictify

$$\Psi_u\, A\, (\hat{x}.\mathrm{t}.R) :\cong (\hat{x} : (\_ : u \in \partial U) \to A) \times \left\langle \lozenge\, u \mid^{\mathrm{t}} R \right\rangle,$$

$$\mathsf{in}\Psi_u\, (\_.a)\, (\mathrm{t}.r) :\overset{\cong}{\longleftarrow} (\lambda\_.a, \mathsf{mod}^{\mathrm{t}}_{\lozenge\, u}\, r),$$

$$\mathsf{out}\Psi \cdot^{\mathrm{a}} q :\overset{\cong}{\longleftarrow} \mathsf{prmod}_{\lozenge\, u} \cdot^{\mathrm{a}} \pi_2(q).$$

The fact that this is isomorphic to A on the boundary follows from the pole theorem 9.1 □

Obviously then the transpension type $\left\langle \lozenge\, u \mid^{\mathrm{t}} T \right\rangle$ is also implementable from the Ψ-type as $\Psi_u\, \mathsf{Unit}\, (\_.\mathrm{t}.T)$.

10.4. **Transpensivity.** The Φ-rule is extremely powerful but not available in all systems. However, when the codomain C is a Ψ-type, then the $\mathsf{in}\Psi$-rule is actually quite similar to the Φ-rule if we note that sections of the Ψ-type are essentially elements of R. As such, we take an interest in types that are very Ψ-like. We have a monad (idempotent if $\mathbb{U}$ is cancellative and affine)

$$\bar{\Psi}_u A := \Psi_u\, A\, (\hat{x}.\mathrm{t}.\langle \forall\, u \mid^{\mathrm{a}} A\, \mathsf{ext}\{u \in \partial\mathbb{U}\ ?\ \hat{x}\ \_\}[\mathsf{reidx}_u{\downarrow}^{\bullet}_{\mathrm{ta}}]\rangle) \qquad \text{(lockful)},$$

$$\bar{\Psi}_u A := \Psi_u\, A\, \left(\hat{x}.\mathrm{a}.\left\langle \forall\, u \mid^{\mathrm{f}} A\, \mathsf{ext}\{u \in \partial\mathbb{U}\ ?\ \hat{x}\ \_\}[\mathsf{app}_u{\uparrow}^{\bullet}_{\mathrm{fa}}]\right\rangle\right) \qquad \text{(lockless)},$$

where $A\, \mathsf{ext}\{\varphi\ ?\ a\}$ is the type of elements of A that are equal to a when φ holds:

$$A\, \mathsf{ext}\{\varphi\ ?\ a\} := (x : A) \times ((\_ : \varphi) \to (x \equiv_A a)).$$

[23] For the identity type, we use pattern-matching abstractions to abbreviate the usage of the J-rule. We are in an extensional type system anyway.

Definition 10.3. A type is **transpensive over** u if it is a monad-algebra for $\bar{\Psi}_u$.

For cancellative, affine and connection-free multipliers, Φ entails that all types are transpensive. For other systems, the universe of u-transpensive types will still be closed at least semantically under many interesting type formers, allowing to eliminate *to* these types in a Φ-like way.

10.5. **Glue, Weld, mill.** $\mathsf{Glue}\{A \leftarrow (\varphi\ ?\ T\ ;\ f)\}$ and $\mathsf{Weld}\{A \to (\varphi\ ?\ T\ ;\ g)\}$ are similar to Strict but extend unidirectional functions. Orton and Pitts [OP18] already show that Glue [CCHM17, NVD17] can be implemented by strictifying a pullback along $A \to (\varphi \to A)$ [ND18b] which is definable internally using a Σ-type. Dually, Weld [NVD17] can be implemented if there is a type former for pushouts along $\varphi \times A \to A$ where $\varphi : \mathsf{Prop}$ [Nuy20a, §6.3.3], which is sound in all presheaf categories.

Finally, mill [ND18b] states that $\forall(u : \mathbb{U})$ preserves Weld and is provable by higher-dimensional pattern matching (where $\circledast$ is the applicative operation of the modal type):

$$\mathsf{mill} : \langle \forall\, u \mid^{\mathrm{a}} \mathsf{Weld}\{A \to (\varphi\ ?\ T\ ;\ g)\}\rangle \to \mathsf{Weld}\{\langle \forall\, u \mid^{\mathrm{a}} A\rangle \to (\langle \forall\, u \mid^{\mathrm{a}} \varphi\rangle\ ?\ \langle \forall\, u \mid^{\mathrm{a}} T\rangle\ ;\ (\mathsf{mod}^{\mathrm{a}}_{\forall\, u}\, g) \circledast \sqcup)\}$$

$$\mathsf{mill}\ \hat{w} = \mathsf{prmod}_{\lozenge\, u} \cdot^{\mathrm{a}} \mathsf{case}\,(\mathsf{prmod}_{\forall\, u} \cdot^{\mathrm{f}} (\hat{w}\ \mathsf{const}_u{\downarrow_{\mathrm{af}}}))\ \mathsf{of} \left\{ \begin{array}{l} \mathsf{weld}\, a \mapsto \mathsf{mod}^{\mathrm{t}}_{\lozenge\, u}\,(\mathsf{weld}\,(\mathsf{mod}^{\mathrm{a}'}_{\forall\, u}\,(a\ \mathsf{reidx}_u{\downarrow_{\mathrm{ta}'}}))) \\ \varphi\ ?\ t \mapsto \mathsf{mod}^{\mathrm{t}}_{\lozenge\, u}\,(\mathsf{mod}^{\mathrm{a}'}_{\forall\, u}\,(t\ \mathsf{reidx}_u{\downarrow_{\mathrm{ta}'}})) \end{array} \right\}$$

$$\mathsf{mill} : \left\langle \forall\, u \mid^{\mathrm{f}} \mathsf{Weld}\{A \to (\varphi\ ?\ T\ ;\ g)\} \right\rangle \to \mathsf{Weld}\left\{ \left\langle \forall\, u \mid^{\mathrm{f}} A \right\rangle \to \left(\left\langle \forall\, u \mid^{\mathrm{f}} \varphi \right\rangle\ ?\ \left\langle \forall\, u \mid^{\mathrm{f}} T \right\rangle\ ;\ (\Lambda^{\mathrm{f}}_u\, g) \circledast \sqcup \right) \right\}$$

$$\mathsf{mill}\ \hat{w} = \mathsf{unmer}_u \cdot^{\mathrm{f}} \mathsf{case}\,(\mathsf{app}_u \cdot^{\mathrm{e}} (\hat{w}\ \mathsf{drop}_u{\uparrow_{\mathrm{ef}}}))\ \mathsf{of} \left\{ \begin{array}{l} \mathsf{weld}\, a \mapsto \mathsf{mer}^{\mathrm{a}}_u\,(\mathsf{weld}\,(\Lambda^{\mathrm{f}'}_u\,(a\ \mathsf{app}_u{\uparrow_{\mathrm{f}'\mathrm{a}}}))) \\ \varphi\ ?\ t \mapsto \mathsf{mer}^{\mathrm{a}}_u\,(\Lambda^{\mathrm{f}'}_u\,(t\ \mathsf{app}_u{\uparrow_{\mathrm{f}'\mathrm{a}}})) \end{array} \right\}.$$

In the first clause, we get an element $a : A$ and can proceed as in section 7.4. In the second clause, we are asserted that φ holds (call the witness p) so that the left hand Weld-type equals T, and we are given $t : T$. Then inside the meridian constructor we know that $\langle \forall\, u \mid^{\mathrm{a}} \varphi\rangle$ holds as this is proven by $\mathsf{mod}^{\mathrm{a}'}_{\forall\, u}\,(p\ \mathsf{reidx}_u{\downarrow_{\mathrm{ta}'}})$ (in lockful notation) or $\Lambda^{\mathrm{f}'}_u\,(p\ \mathsf{app}_u{\uparrow_{\mathrm{f}'\mathrm{a}}})$ (in lockless notation); hence the Weld-type in the codomain simplifies to $\langle \forall\, u \mid^{\mathrm{a}} T\rangle$. When φ holds and $t = g\, a$, then the weld-constructor of the right hand Weld-type reduces to $(\mathsf{mod}^{\mathrm{a}}_{\forall\, u}\, g) \circledast \sqcup$ which effectively applies g under the $\mathsf{mod}_{\forall\, u}$-constructor, so that both clauses match as required.

10.6. **Locally fresh names.** Nominal type theory is modelled in the Schanuel topos [Pit14] which is a subcategory of nullary affine cubical sets $\mathrm{Psh}({}^0\mathsf{Cube}_{\square})$ (example 6.12). As fibrancy is not considered in this paper, we will work directly in $\mathrm{Psh}({}^0\mathsf{Cube}_{\square})$. Names can be modelled using the multiplier $\sqcup * (i : \mathbb{I})$. Interestingly, the fresh weakening functor $\exists_{(i:\mathbb{I})}$ is then *inverse* to its left adjoint $\exists_{(i:\mathbb{I})}$. By consequence, we get $\exists\, i \cong \forall\, i$ (the fresh name quantifier) with inverse $\exists\, i \cong \lozenge\, i$. For consistency, we will only use $\forall\, i$ and $\exists\, i$. Note that in lockless notation we then have $\exists\, i \dashv \forall\, i$, $\forall\, i \dashv \exists\, i$, $\mathsf{app}_i^{-1} \dashv \mathsf{app}_i$ and $\mathsf{const}_i^{-1} \dashv \mathsf{const}_i$.

The nominal dependent type system FreshMLTT [PMD15] used in Pitts's examples of interest [Pit14] is substantially different from ours:

- It features a name swapping operation that is semantically *not* merely a substitution.
- Freshness for a name i is not a modality or a type, but a judgement that can be derived for an expression t if and only if t is invariant under swapping i with a newly introduced name j. As a consequence, freshness propagates through type and term constructors.
- Many equalities are strict where we can only guarantee an isomorphism.

name	FreshMLTT	MTT, lockful	MTT, lockless
name quantification	$\mathsf{M}[i:N].T$	$\langle \forall(i:\mathbb{I}) \mid^{\mathsf{a}} [\![T]\!] \rangle$	$\langle \forall(i:\mathbb{I}) \mid^{\mathsf{f}} [\![T]\!] \rangle$
name abstraction	$\alpha[i:N].t$	$\mathsf{mod}^{\mathsf{a}}_{\forall(i:\mathbb{I})}\,[\![t]\!]$	$\Lambda^{\mathsf{f}}_i\,[\![t]\!]$
name application	$t@n$	$\mathsf{prmod}_{\forall\, i}\cdot^{\mathsf{f}}\,[\![t]\!]$	$\mathsf{app}_i\cdot^{\mathsf{a}}\,[\![t]\!]$
non-binding quant.	$\langle\langle i:N\rangle\rangle.T$	$\langle \lrcorner\, i \mid^{\mathsf{f}} \langle \forall\, i \mid^{\mathsf{a}} [\![T]\!][\mathsf{app}_i^{-1}\downarrow^{\bullet}_{\mathsf{fa}}] \rangle \rangle$	$\langle \lrcorner\, i \mid^{\mathsf{a}} \langle \forall\, i \mid^{\mathsf{f}} [\![T]\!][\mathsf{app}_i\uparrow^{\bullet}_{\mathsf{fa}}] \rangle \rangle$
non-binding abs.	$\langle i:N\rangle.t$	$\mathsf{app}_i^{-1}\,[\![t]\!] := \mathsf{mod}^{\mathsf{f}}_{\lrcorner\, i}\,(\mathsf{mod}^{\mathsf{a}}_{\forall\, i}\,([\![t]\!][\mathsf{app}_i^{-1}\downarrow^{\bullet}_{\mathsf{fa}}]))$	$\mathsf{app}_i^{-1}\,[\![t]\!] := \mathsf{mod}^{\mathsf{a}}_{\lrcorner\, i}\,(\Lambda^{\mathsf{f}}_i\,([\![t]\!][\mathsf{app}_i\uparrow^{\bullet}_{\mathsf{fa}}]))$
locally fresh name	$\nu[i:N].t$	$\mathsf{prmod}_{\lrcorner\, i}\cdot^{\mathsf{a}}\,[\![t]\!]$	$\mathsf{const}_i^{-1}\cdot^{\mathsf{f}}\,[\![t]\!]$

Figure 11: A heuristic for translating FreshMLTT [PMD15] to the current system.

For these reasons, we do not try to formally state that we can support locally fresh names in the sense of FreshMLTT. Nevertheless, in fig. 11 we give at least a heuristic $[\![\llcorner\!\lrcorner]\!]$ for translating programs in a subsystem of FreshMLTT to programs in the current system. This subsystem does not feature name swapping, but it does feature the *non-binding abstractions* defined in terms of it, as well as locally fresh names.

Ordinary name quantification is simply translated to the modality $\exists\, i \cong \forall\, i$, and as usual application corresponds to the modal projection function. The non-binding abstraction in FreshMLTT abstracts over a name that is already in scope, *without* shadowing, i.e. it is a variable capturing operation. This is translated essentially to the 2-cell $\mathsf{app}_i^{-1} : 1 \Rightarrow \lrcorner\, i \circ \forall\, i$, which also played a variable capturing role in section 10.2. Finally, a locally fresh name abstraction $\nu[i:N]t$ brings a name i into scope in its body t, but requires that t be fresh for i; in our system we would say that t is a subterm of modality $\forall\, i \circ \lrcorner\, i$. The type of $\nu[i:N].t$ is the same as the type of t, which we can justify with the isomorphism $\mathsf{const}_i^{-1} : \forall\, i \circ \lrcorner\, i \cong 1$. This isomorphism is essentially the content of the modal projection function of $\lrcorner\, i$ (since $\forall\, i \cong \exists\, i \dashv \lrcorner\, i$), which we use to translate locally fresh name abstractions.

Note that for general multipliers, $\lrcorner\, i$ does not have an internal left adjoint and hence not a modal projection function either. For the nullary affine interval, however, $\lrcorner\, i \cong \raisebox{0.2ex}{$\between$}\, i$, so the projection function is essentially unmer_i!

Example 10.4. Consider Pitts's implementation of higher dimensional pattern matching [Pit14]:

$$i : (\mathsf{M}[i:N].A \uplus B) \to (\mathsf{M}[i:N].A) \uplus (\mathsf{M}[i:N].B)$$

$$i\,\hat{c} = \nu[i:N].\mathsf{case}\,\hat{c}@i\,\mathsf{of}\left\{ \begin{array}{lcl} \mathsf{inl}\,a & \mapsto & \mathsf{inl}\,(\langle i:N\rangle.a) \\ \mathsf{inr}\,b & \mapsto & \mathsf{inr}\,(\langle i:N\rangle.b) \end{array} \right\}$$

A brainless translation using fig. 11 yields a type mismatch, because the non-binding abstractions will put the freshness constructor inside the coproduct constructors, whereas the translation of the locally fresh name abstraction requires a fresh element of the coproduct. This is related to our earlier remark that in FreshMLTT, freshness silently propagates through type and term constructors, so here we have to manually intervene. This is also necessary to insert invertible 2-cell substitutions in some places, and to check whether we need a modal argument (i.e. the modality is handled at the judgemental level) or we need to explicitly use

the modal constructor as is always done in fig. 11. Doing so, we find

$$i : \langle \forall\, i \mid^{\mathsf{a}} A \uplus B \rangle \to \langle \forall\, i \mid^{\mathsf{a}} A \rangle \uplus \langle \forall\, i \mid^{\mathsf{a}} B \rangle \qquad \text{(lockful)}$$

$$i\,\hat{c} = \mathsf{prmod}_{\lrcorner\, i} \cdot^{\mathsf{a}} \mathsf{case}\,(\mathsf{prmod}_{\forall\, i} \cdot^{\mathsf{f}} (\hat{c}\,\mathsf{const}_i{\downarrow_{\mathrm{af}}}))\ \mathsf{of} \left\{ \begin{array}{l} \mathsf{inl}\, a \mapsto \mathsf{mod}^{\mathsf{f}}_{\lrcorner\, i}\,(\mathsf{inl}\,(\mathsf{mod}^{\mathsf{a}'}_{\forall\, i}\,(a\,\mathsf{app}_i^{-1}{\downarrow_{\mathrm{fa}'}}))) \\ \mathsf{inr}\, b \mapsto \mathsf{mod}^{\mathsf{f}}_{\lrcorner\, i}\,(\mathsf{inr}\,(\mathsf{mod}^{\mathsf{a}}_{\forall\, i}\,(b\,\mathsf{app}_i^{-1}{\downarrow_{\mathrm{fa}'}}))) \end{array} \right\}$$

$$i : \left\langle \forall\, i \mid^{\mathsf{f}} A \uplus B \right\rangle \to \left\langle \forall\, i \mid^{\mathsf{f}} A \right\rangle \uplus \left\langle \forall\, i \mid^{\mathsf{f}} B \right\rangle \qquad \text{(lockless)}$$

$$i\,\hat{c} = \mathsf{const}_i^{-1} \cdot^{\mathsf{f}} \mathsf{case}\,(\mathsf{app}_i \cdot^{\mathsf{a}} (\hat{c}\,\mathsf{const}_i^{-1}{\uparrow_{\mathrm{af}}}))\ \mathsf{of} \left\{ \begin{array}{lcl} \mathsf{inl}\, a & \mapsto & \mathsf{mod}^{\mathsf{a}}_{\lrcorner\, i}\,(\mathsf{inl}\,(\Lambda^{\mathsf{f}'}_u\,(a\,\mathsf{app}_i{\uparrow_{\mathrm{f}'\mathrm{a}}}))) \\ \mathsf{inl}\, b & \mapsto & \mathsf{mod}^{\mathsf{a}}_{\lrcorner\, i}\,(\mathsf{inl}\,(\Lambda^{\mathsf{f}'}_u\,(b\,\mathsf{app}_i{\uparrow_{\mathrm{f}'\mathrm{a}}}))) \end{array} \right\},$$

which is *exactly* what we found in section 7.4 adapted to our convention that we only want to mention $\lrcorner\, i$ and $\forall\, i$.

Example 10.5. Consider Pitts et al.'s *implementation* [PMD15, ex. 2.2] of what is essentially BCM's Φ-rule [Mou16, BCM15] since the boundary is empty:

$$g : ((\mathsf{И}[i : N].A) \to \mathsf{И}[i : N].B) \to \mathsf{И}[i : N].(A \to B)$$
$$g\, f = \alpha[i : N].(\lambda x.(f\,(\langle i : N \rangle.x))@i).$$

We can translate this to the current system using fig. 11:

$$g : (\langle \forall\, i \mid^{\mathsf{a}} A \rangle \to \langle \forall\, i \mid^{\mathsf{a}} B \rangle) \to \langle \forall\, i \mid^{\mathsf{a}} A \to B \rangle \qquad \text{(lockful)}$$

$$g\, f = \mathsf{mod}^{\mathsf{a}}_{\forall\, i}\left(\lambda x.\mathsf{prmod}_{\forall\, i} \cdot^{\mathsf{f}} (f[\mathsf{const}_i{\downarrow^{\bullet}_{\mathrm{af}}}]\,(\mathsf{mod}^{\mathsf{a}'}_{\forall\, i}\,(x\,\mathsf{app}_i^{-1}{\downarrow_{\mathrm{fa}'}}]))))\right)$$

$$g : \left(\left\langle \forall\, i \mid^{\mathsf{f}} A \right\rangle \to \left\langle \forall\, i \mid^{\mathsf{f}} B \right\rangle\right) \to \left\langle \forall\, i \mid^{\mathsf{f}} A \to B \right\rangle \qquad \text{(lockless)}$$

$$g\, f = \Lambda^{\mathsf{f}}_i\left(\lambda x.\mathsf{app}_i \cdot^{\mathsf{a}} (f[\mathsf{const}_i^{-1}{\uparrow^{\bullet}_{\mathrm{af}}}]\,(\Lambda^{\mathsf{f}'}_i\,(x\,\mathsf{app}_i{\uparrow_{\mathrm{f}'\mathrm{a}}})))\right).$$

The effect of conflating $🔒_{\lrcorner\, i} = \exists\, i$ with $🔒_{\emptyset\, i} = \forall\, i$ is that affine function application no longer renders the non-fresh part of the context inaccessible using $\exists\, i$ but instead universally quantifies it using $\forall\, i$, so that we can capture variables as in FreshMLTT. Remarkably, we do *not* need the Φ-rule (fig. 9) for this nor pattern-matching for the transpension type (fig. 8), although these rules hold as the affine cubical interval is cancellative, affine and connection-free (example 6.12).

11. Conclusion

To summarize, the transpension type can be defined in a broad class of presheaf models and generalizes previous internalization operators. For now, we only present an extensional type system without an algorithmic typing judgement. The major hurdles towards producing an intensional version with decidable type-checking, are the following:

- We need to decide equality of 2-cells. Solutions may exist in the literature on higher-dimensional rewriting.
- The substitution modality should ideally reduce like ordinary substitution. Remark 5.5 explores what is needed for this to work.
- We need a syntax-directed way to close the section computation rules of Φ (fig. 9) and transpension elimination (section 9) under substitution.
- We need to decide whether the boundary predicate, or any similar predicate about shape variables such as $i \equiv_{\mathbb{I}} 0$ in cubical type theory, is true. This problem has been dealt with in special cases, e.g. in implementations of cubical type theory [VMA19].

Applications include all applications (discussed in section 1) of the presheaf internalization operators recovered from the transpension type in section 10. Moreover, our modal approach to shape variables via multipliers allows the inclusion of Pinyo and Kraus's twisted prism functor [PK20] as a semantics of an interval variable, which we believe is an important advancement towards higher-dimensional directed type theory.

References

[AGJ14] Robert Atkey, Neil Ghani, and Patricia Johann. A relationally parametric model of dependent type theory. In *Principles of Programming Languages*, 2014. doi:10.1145/2535838.2535852.

[AHH18] Carlo Angiuli, Kuen-Bang Hou (Favonia), and Robert Harper. Cartesian Cubical Computational Type Theory: Constructive Reasoning with Paths and Equalities. In Dan Ghica and Achim Jung, editors, *Computer Science Logic (CSL 2018)*, volume 119 of *LIPIcs*, pages 6:1–6:17, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2018/9673, doi:10.4230/LIPIcs.CSL.2018.6.

[BBC+19] Lars Birkedal, Aleš Bizjak, Ranald Clouston, Hans Bugge Grathwohl, Bas Spitters, and Andrea Vezzosi. Guarded cubical type theory. *Journal of Automated Reasoning*, 63(2):211–253, 8 2019. doi:10.1007/s10817-018-9471-7.

[BCH14] Marc Bezem, Thierry Coquand, and Simon Huber. A Model of Type Theory in Cubical Sets. In *19th International Conference on Types for Proofs and Programs (TYPES 2013)*, volume 26, pages 107–128, Dagstuhl, Germany, 2014. URL: http://drops.dagstuhl.de/opus/volltexte/2014/4628, doi:10.4230/LIPIcs.TYPES.2013.107.

[BCM15] Jean-Philippe Bernardy, Thierry Coquand, and Guilhem Moulin. A presheaf model of parametric type theory. *Electron. Notes in Theor. Comput. Sci.*, 319:67 – 82, 2015. doi:http://dx.doi.org/10.1016/j.entcs.2015.12.006.

[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. doi:10.1017/S0960129519000197.

[BGC+16] Aleš Bizjak, Hans Bugge Grathwohl, Ranald Clouston, Rasmus Ejlers Møgelberg, and Lars Birkedal. Guarded dependent type theory with coinductive types. In *FOSSACS '16*, 2016. doi:10.1007/978-3-662-49630-5_2.

[BGM17] Patrick Bahr, Hans Bugge Grathwohl, and Rasmus Ejlers Møgelberg. The clocks are ticking: No more delays! In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12, 2017. doi:10.1109/LICS.2017.8005097.

[BM20] Ales Bizjak and Rasmus Ejlers Møgelberg. Denotational semantics for guarded dependent type theory. *Math. Struct. Comput. Sci.*, 30(4):342–378, 2020. doi:10.1017/S0960129520000080.

[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. doi:10.2168/LMCS-8(4:1)2012.

[BT17] Simon Boulier and Nicolas Tabareau. Model structure on the universe in a two level type theory. Working paper or preprint, 2017. URL: https://hal.archives-ouvertes.fr/hal-01579822.

[BV17] Jean-Philippe Bernardy and Andrea Vezzosi. Parametric application. Private communication, 2017.

[Car86] John Cartmell. Generalised algebraic theories and contextual categories. *Ann. Pure Appl. Logic*, 32:209–243, 1986. doi:10.1016/0168-0072(86)90053-9.

[CCHM17] Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. Cubical type theory: A constructive interpretation of the univalence axiom. *FLAP*, 4(10):3127–3170, 2017. URL: http://www.cse.chalmers.se/~simonhu/papers/cubicaltt.pdf.

[CH20] Evan Cavallo and Robert Harper. Internal parametricity for cubical type theory. In *28th EACSL Annual Conference on Computer Science Logic, CSL 2020, January 13-16, 2020, Barcelona, Spain*, pages 13:1–13:17, 2020. doi:10.4230/LIPIcs.CSL.2020.13.

[Che12] James Cheney. A dependent nominal type theory. *Log. Methods Comput. Sci.*, 8(1), 2012. doi:10.2168/LMCS-8(1:8)2012.

[CMS20] Evan Cavallo, Anders Mörtberg, and Andrew W Swan. Unifying Cubical Models of Univalent Type Theory. In Maribel Fernández and Anca Muscholl, editors, *Computer Science Logic (CSL 2020)*, volume 152 of *LIPIcs*, pages 14:1–14:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: https://drops.dagstuhl.de/opus/volltexte/2020/11657, doi:10.4230/LIPIcs.CSL.2020.14.

[GCK+21] Daniel Gratzer, Evan Cavallo, GA Kavvos, Adrien Guatto, and Lars Birkedal. Modalities and parametric adjoints, 2021.

[Gir64] Jean Giraud. Méthode de la descente. *Bull. Soc. Math. Fr., Suppl., Mém.*, 2:115, 1964. doi:10.24033/msmf.2.

[GKNB20] Daniel Gratzer, Alex Kavvos, Andreas Nuyts, and Lars Birkedal. Type theory à la mode. Pre-print, 2020. URL: https://anuyts.github.io/files/mtt-techreport.pdf.

[GKNB21] Daniel Gratzer, G. A. Kavvos, Andreas Nuyts, and Lars Birkedal. Multimodal Dependent Type Theory. *Logical Methods in Computer Science*, Volume 17, Issue 3, July 2021. URL: https://lmcs.episciences.org/7713, doi:10.46298/lmcs-17(3:11)2021.

[Hof97] Martin Hofmann. *Syntax and Semantics of Dependent Types*, chapter 4, pages 79–130. Cambridge University Press, 1997.

[HS97] Martin Hofmann and Thomas Streicher. Lifting grothendieck universes. Unpublished note, 1997. URL: https://www2.mathematik.tu-darmstadt.de/~streicher/NOTES/lift.pdf.

[Hub16] Simon Huber. *Cubical Interpretations of Type Theory*. PhD thesis, University of Gothenburg, Sweden, 2016. URL: http://www.cse.chalmers.se/~simonhu/misc/thesis.pdf.

[KLV12] Chris Kapulkin, Peter LeFanu Lumsdaine, and Vladimir Voevodsky. The simplicial model of univalent foundations. 2012. Preprint, http://arxiv.org/abs/1211.2851.

[LH11] Daniel R. Licata and Robert Harper. 2-dimensional directed type theory. *Electr. Notes Theor. Comput. Sci.*, 276:263–289, 2011. doi:10.1016/j.entcs.2011.09.026.

[LOPS18] Daniel R. Licata, Ian Orton, Andrew M. Pitts, and Bas Spitters. Internal universes in models of homotopy type theory. In *3rd International Conference on Formal Structures for Computation and Deduction, FSCD 2018, July 9-12, 2018, Oxford, UK*, pages 22:1–22:17, 2018. doi:10.4230/LIPIcs.FSCD.2018.22.

[Mou16] Guilhem Moulin. *Internalizing Parametricity*. PhD thesis, Chalmers University of Technology, Sweden, 2016. URL: publications.lib.chalmers.se/records/fulltext/235758/235758.pdf.

[ND18a] 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 *Logic in Computer Science (LICS) 2018, Oxford, UK, July 09-12, 2018*, pages 779–788, 2018. doi:10.1145/3209108.3209119.

[ND18b] Andreas Nuyts and Dominique Devriese. Internalizing Presheaf Semantics: Charting the Design Space. In *Workshop on Homotopy Type Theory / Univalent Foundations*, 2018. URL: https://hott-uf.github.io/2018/abstracts/HoTTUF18_paper_1.pdf.

[ND19] Andreas Nuyts and Dominique Devriese. Dependable atomicity in type theory. In *TYPES*, 2019.

[Nor19] Paige Randall North. Towards a directed homotopy type theory. *Proceedings of the Thirty-Fifth Conference on the Mathematical Foundations of Programming Semantics, MFPS 2019, London, UK, June 4-7, 2019*, pages 223–239, 2019. doi:10.1016/j.entcs.2019.09.012.

[Nuy18a] Andreas Nuyts. Presheaf models of relational modalities in dependent type theory. *CoRR*, abs/1805.08684, 2018. arXiv:1805.08684.

[Nuy18b] Andreas Nuyts. Robust notions of contextual fibrancy. In *Workshop on Homotopy Type Theory / Univalent Foundations*, 2018. URL: https://hott-uf.github.io/2018/abstracts/HoTTUF18_paper_2.pdf.

[Nuy20a] Andreas Nuyts. *Contributions to Multimode and Presheaf Type Theory*. PhD thesis, KU Leuven, Belgium, 8 2020. URL: https://anuyts.github.io/files/phd.pdf.

[Nuy20b] Andreas Nuyts. The transpension type: Technical report. *CoRR*, abs/2008.08530, 2020. version 2. URL: https://arxiv.org/abs/2008.08530, arXiv:2008.08530.

[NVD17] Andreas Nuyts, Andrea Vezzosi, and Dominique Devriese. Parametric quantifiers for dependent type theory. *PACMPL*, 1(ICFP):32:1–32:29, 2017. URL: http://doi.acm.org/10.1145/3110276, doi:10.1145/3110276.

[OP18] Ian Orton and Andrew M. Pitts. Axioms for modelling cubical type theory in a topos. *Logical Methods in Computer Science*, 14(4), 2018. doi:10.23638/LMCS-14(4:23)2018.

[Ort18] Ian Orton. *Cubical Models of Homotopy Type Theory - An Internal Approach*. PhD thesis, University of Cambridge, 2018.

[Pit13a] A. M. Pitts. *Nominal Sets: Names and Symmetry in Computer Science*, volume 57 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2013.

[Pit13b] Andrew M Pitts. *Nominal sets: Names and symmetry in computer science*, volume 57. Cambridge University Press, 2013.

[Pit14] Andrew Pitts. Nominal sets and dependent type theory. In *TYPES*, 2014. URL: `https://www.irif.fr/~letouzey/types2014/slides-inv3.pdf`.

[PK20] Gun Pinyo and Nicolai Kraus. From Cubes to Twisted Cubes via Graph Morphisms in Type Theory. In Marc Bezem and Assia Mahboubi, editors, *25th International Conference on Types for Proofs and Programs (TYPES 2019)*, volume 175 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:18, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. URL: `https://drops.dagstuhl.de/opus/volltexte/2020/13069`, `doi:10.4230/LIPIcs.TYPES.2019.5`.

[PMD15] Andrew M. Pitts, Justus Matthiesen, and Jasper Derikx. A dependent type theory with abstractable names. *Electronic Notes in Theoretical Computer Science*, 312:19 – 50, 2015. Ninth Workshop on Logical and Semantic Frameworks, with Applications (LSFA 2014). URL: `http://www.sciencedirect.com/science/article/pii/S1571066115000079`, `doi:https://doi.org/10.1016/j.entcs.2015.04.003`.

[Rey83] John C. Reynolds. Types, abstraction and parametric polymorphism. In *IFIP Congress*, pages 513–523, 1983.

[RS17] E. Riehl and M. Shulman. A type theory for synthetic ∞-categories. *ArXiv e-prints*, May 2017. `arXiv:1705.07442`.

[Sta19] The Stacks Project Authors. Stacks project. `http://stacks.math.columbia.edu`, 2019. Tags 00VC and 00XF.

[Uni13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. `http://homotopytypetheory.org/book`, IAS, 2013.

[VMA19] Andrea Vezzosi, Anders Mörtberg, and Andreas Abel. Cubical Agda: a dependently typed programming language with univalence and higher inductive types. *PACMPL*, 3(ICFP):87:1–87:29, 2019. `doi:10.1145/3341691`.

[WL20] Matthew Z. Weaver and Daniel R. Licata. A constructive model of directed univalence in bicubical sets. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 915–928. ACM, 2020. `doi:10.1145/3373718.3394794`.

[Yet87] David Yetter. On right adjoints to exponential functors. *Journal of Pure and Applied Algebra*, 45(3):287–304, 1987. URL: `https://www.sciencedirect.com/science/article/pii/0022404987900776`, `doi:https://doi.org/10.1016/0022-4049(87)90077-6`.

This work is licensed under the Creative Commons Attribution License. To view a copy of this license, visit `https://creativecommons.org/licenses/by/4.0/` or send a letter to Creative Commons, 171 Second St, Suite 300, San Francisco, CA 94105, USA, or Eisenacher Strasse 2, 10777 Berlin, Germany