# A diagram model of linear dependent type theory

**Martin Lundfall**
Stockholm University
martin@dapp.org

## Abstract

We present a type theory dealing with non-linear, "ordinary" dependent types (which we will call *cartesian*), and *linear types*, where both constructs may depend on terms of the former. In the interplay between these, we find the new type formers $\sqcap_{x:A} B$ and $\sqsubset_{x:A} B$, akin to $\Pi$ and $\Sigma$, but where the dependent type $B$, (and therefore the resulting construct) is a linear type. These can be seen as internalizing universal and existential quantification over linear propositions, respectively. We also consider two modalities, $M$ and $L$, transforming linear types into cartesian types and vice versa.

The theory is interpreted in a split comprehension category $\pi : \mathcal{T} \to \mathcal{C}^{\to}$ [4], accompanied by a split monoidal fibration, $q : \mathcal{L} \to \mathcal{C}$. We interpret $\mathcal{C}$ as a category of contexts, which for any $\Gamma \in \mathcal{C}$, determines the fibers $\mathcal{T}_\Gamma$ and $\mathcal{L}_\Gamma$. We interpret $\mathcal{T}_\Gamma$ as category of the cartesian types over $\Gamma$, and $\mathcal{L}_\Gamma$ as the monoidal category of linear types in $\Gamma$. In this setting, the type formers $\sqcap_{x:A}$ and $\sqsubset_{x:A}$ are understood as right and left adjoints of the monoidal reindexing functor $\pi_A^* : \mathcal{L}_\Gamma \to \mathcal{L}_{\Gamma.A}$ corresponding to the weakening projection $\pi_A : \Gamma.A \to \Gamma$ in $\mathcal{C}$. The operators $M$ and $L$ give rise to a fiberwise adjunction $L \dashv M$ between $\mathcal{L}$ and $\mathcal{T}$, where we understand the traditional exponential modality as the comonad $! = LM$.

We provide a model of this theory called the *Diagram model*, which extends the groupoid model of dependent type theory [3] to accommodate linear types. Here, cartesian types over a context $\Gamma$ are interpreted as a family of groupoids indexed over the groupoid $\Gamma$, while linear types are interpreted as diagrams over groupoids, $A : \Gamma \to \mathcal{V}$ in any symmetric monoidal category $\mathcal{V}$. We show that the *diagrams model* can under certain conditions support a linear analogue of the univalence axiom, and provide some discussion on the higher-dimensional nature of linear dependent types.

**2012 ACM Subject Classification** Theory of computation → Linear logic; Theory of computation → Type theory

**Keywords and phrases** Dependent type theory, linear type theory, diagram model, monoidal categories, groupoid model

**Digital Object Identifier** 10.4230/LIPIcs.CVIT.2016.23

**Related Version** `http://kurser.math.su.se/pluginfile.php/16103/mod_folder/content/0/2017/2017_47_report.pdf`

**Acknowledgements** I want to thank Peter Lumsdaine for his constant guidance and inspiration.

## Contents

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 1 Introduction & summary of results

Lately, there has been an increasing interest in combining linear and dependent types [12], [5], [13], [8]. The idea is that such a theory would inherit the higher-order nature of dependent types, while maintaining a careful account of how assumptions are used in a derivation. It is not completely clear, however, what the synthesis looks like, since in dependent type theory, variables may appear in both terms and types, but linear type theory only allows each variable to appear freely exactly once. Here, we take an approach inspired by [5] and [13], in which we distinguish between non-linear, dependent types (which we call *cartesian*), and linear types, and circumvent the issue by only allowing cartesian terms to appear in types (both cartesian and linear).

The theory splits contexts into two parts, divided by a semicolon, where the first part contains cartesian assumptions, for which weakening and contraction is admissible, while the second part contains linear assumptions, for which only exchange is allowed. We introduce two new type formers, $\sqcap_{x:A}B$ and $\sqsubset_{x:A} B$, akin to $\Pi$ and $\Sigma$, but where the dependent type $B$ (and therefore the resulting construct) is a linear. The traditional ! modality is deconstructed as a comonad arising from the adjoint pair $L \dashv M$, where $L$ is a functor (or modality) sending cartesian types into linear, and $M$ sends linear types to cartesian. We have $\Pi_{x:A}B_M \cong (\sqcap_{x:A}B)_M$, for linear $B$, and, assuming a few additional rules, a linear isomorphism $(\Sigma_{x:A}C)_L \cong \sqsubset_{x:A} C_L$ for cartesian $C$.

Compared to ordinary dependent type theory, we get additional elimination and computation rules for both $\Sigma$ and $Id$-types when eliminating into a linear type.

We postulate the existence of two universes, $L$ and $U$, containing codes of linear and cartesian types, respectively and assumed to be closed under all type formers.

We develop categorical semantics for the theory by defining a model as a *comprehension category* [4], $\pi : \mathcal{T} \to \mathcal{C}$ equipped with a *split monoidal fibration* $q : \mathcal{L} \to \mathcal{C}$ over the same base. A split monoidal fibration has just enough structure to make the fibers $\mathcal{L}_\Gamma$ over a context $\Gamma \in \mathcal{C}$ into monoidal categories, and reindexing functors (strict) monoidal functors. The traditional linear type formers $\&, \oplus, 0, \top, \multimap$ correspond to the existence of binary products and coproducts, initial and terminal object and internal homs in each fiber, such that these are preserved under reindexing. The new type formers $\sqcap_{x:A}B$ and $\sqsubset_{x:A} B$ correspond to right and left adjoints to the reindexing functor $\pi_A^* : \mathcal{L}_\Gamma \to \mathcal{L}_{\Gamma.A}$, while the modalities $L$ and $M$ give rise to a fiber adjunction between $\mathcal{L}$ and $\mathcal{T}$. The new rules for $\Sigma$ are automatically

$$\Sigma\text{-E}_1\ \frac{\begin{array}{c}\Gamma, t : \Sigma_{x:A}B \vdash C \ \text{ type} \\ \Gamma, x : A, y : B \vdash c : C[(x,y)/t] \\ \Gamma \vdash s : \Sigma_{x:A}B\end{array}}{\Gamma \vdash \hat{c}[s] : C[s/t]} \qquad \Sigma\text{-E}_2\ \frac{\begin{array}{c}\Gamma, t : \Sigma_{x:A}B \vdash C \ \text{ linear} \\ \Gamma, x : A, y : B; \Xi \vdash c : C[(x,y)/t] \\ \Gamma \vdash s : \Sigma_{x:A}B\end{array}}{\Gamma; \Xi[pr_1(s)/x][pr_2(s)/y] \vdash \hat{c}[s] : C[s/t]}$$

$$=\text{-E}_1\ \frac{\begin{array}{c}\Gamma, x, y : A, p : x =_A y \vdash C \ \text{ type} \\ \Gamma, z : A \vdash c : C[z/x, z/y, \text{refl}(z)/p] \\ \Gamma \vdash M, N : A \\ \Gamma \vdash P : M =_A N\end{array}}{\Gamma \vdash R^{Id} : C[M/x, N/y, P/p]} \qquad =\text{-E}_2\ \frac{\begin{array}{c}\vdash \Gamma, x, y : A, p : x =_A y; \Xi \ \text{ctxt} \\ \Gamma, x, y : A, p : x =_A y \vdash C \ \text{ linear} \\ \Gamma, z : A; \Xi[z/x, z/y, \text{refl}(z)/p] \vdash c : C[z/x, z/y, \text{refl}(z)/p] \\ \Gamma \vdash M, N : A \\ \Gamma \vdash P : M =_A N\end{array}}{\Gamma; \Xi[M/x, N/y, P/p] \vdash R^{Id} : C[M/x, N/y, P/p]}$$

**Figure 1** Elimination rules for $\Sigma$ and $Id$

satisfied by the semantic interpretation of $\Sigma_A$ as a left adjoint to the reindexing functor $\pi_A^* : \mathcal{T}_\Gamma \to \mathcal{T}_{\Gamma.A}$. The new rules for $Id$-types impose an additional condition on the semantic interpretation of $Id$, which are always fulfilled if our identity types are extensional.

We consider two concrete models, the first being the *families model*, in which cartesian types consist of families of sets, indexed by their context set $\Gamma$, and a linear type in the context $\Gamma$ is a $\Gamma$-indexed family of objects in a given symmetric monoidal category $\mathcal{V}$. Examples of suitable $\mathcal{V}$ supporting all type formers present in our syntax are $\mathbf{AbGrp}$, $\mathbf{GCTop}_*$, $\mathbf{Vect}_F$, i.e. the category of abelian groups, the category of compact generated, pointed topological space and the category of vector spaces over a field $F$, respectively.

Generalizing the families model, we get the *diagrams model*, in which contexts are interpreted as groupoids, and cartesian types over a groupoid $\Gamma$ are diagrams in $\mathbf{Gpd}$ over $\Gamma$, and linear types over $\Gamma$ are diagrams in a given symmetric monoidal category $\mathcal{V}$ over $\Gamma$. Just as the groupoid model [3] can be shown to support a univalent universe, we construct a linear analogue of the univalence axiom and show that it holds in the diagrams model if the adjunction $L \dashv M$ factors through sets.

## 2 Syntax

As cartesian type formers, we use the standard $\Sigma$, $\Pi$, and identity type formers as well as universe types $U$ and $L$ for linear and cartesian types, respectively. The purely linear part of our type theory contains all the type formers of intuitionistic linear logic; the additive connectives $\&, \oplus, \mathbf{0}, \top$ and the multiplicatives $\otimes, \mathbf{1}, \multimap$. In addition to these, we have the new type formers $\sqsubset$, $\sqcap$, which play a role analogous to that of $\Sigma$ and $\Pi$ in the cartesian setting. Finally, we have the two modalities, $M$ and $L$, which turns linear types into cartesian, and vice versa. A detailed presentation of our syntax can be found in [7]. For the familiar, “purely” dependent or linear type formers, our presentation offer no significant surprises, except for a couple of additional rules for $\Sigma$ and the identity type. Therefore, we focus on presenting the syntax for the new type formers $\sqsubset$, $\sqcap$ and the modalities $M$ and $L$.

### 2.1 Auxiliary elimination rules

Besides the traditional rules for $\Pi$, $\Sigma$ and the identity type, we find that since we can now eliminate into linear types, we must introduce an extra elimination and computational rule for each one. These additional rules are presented next along with the traditional elimination rules in Figure 1.

$$\frac{\Gamma \vdash A \text{ type}}{\Gamma \vdash A_L \text{ linear}}\ \text{L-F} \qquad \frac{\Gamma \vdash B \text{ linear}}{\Gamma \vdash B_M \text{ type}}\ \text{M-F}$$

$$\frac{\Gamma \vdash a : A}{\Gamma; \cdot \vdash a_L : A_L}\ \text{L-I} \qquad \frac{\Gamma; \cdot \vdash b : B}{\Gamma \vdash \sigma(b) : B_M}\ \text{M-I}$$

$$\frac{\begin{array}{c}(\Gamma \vdash B \text{ linear})\\ (\vdash \Gamma; \Xi' \text{ ctxt})\\ \Gamma; \Xi \vdash y : A_L \quad \Gamma, x : A; \Xi' \vdash t : B\end{array}}{\Gamma; \Xi, \Xi' \vdash \text{let } x \text{ be } y \text{ in } t : B}\ \text{L-E} \qquad \frac{\Gamma \vdash t : B_M}{\Gamma; \cdot \vdash \sigma^{-1}(t) : B}\ \text{M-E}$$

$$\frac{\Gamma; \Xi \vdash \text{let } x \text{ be } s_L \text{ in } t : B}{\Gamma; \Xi \vdash \text{let } x \text{ be } s_L \text{ in } t \equiv t[s/x] : B}\ \text{L-C} \qquad \frac{\Gamma \vdash \sigma(b) : B_M}{\Gamma; \cdot \vdash \sigma^{-1}(\sigma(b)) \equiv b : B}\ \text{M-C}_1$$

$$\frac{\begin{array}{c}\Gamma; y : A_L, \Xi \vdash t : B\\ \Gamma; \Xi' \vdash a : A_L\end{array}}{\Gamma; \Xi, \Xi' \vdash \text{let } x \text{ be } a \text{ in } t[x_L/y] \equiv t[a/y] : B}\ \text{L-U} \qquad \frac{\Gamma; \cdot \vdash \sigma^{-1}(t) : B_M}{\Gamma \vdash \sigma(\sigma^{-1}(t)) \equiv t : B_M}\ \text{M-C}_2$$

**Figure 2** Typing rules for $M$ and $L$

## 2.2 The modalities $M$ and $L$

We introduce two the modal operators $M$ and $L$, which transfers a linear type/term to its cartesian counterpart and vice versa. Semantically, this will establish a fiberwise monoidal adjunction between the categories of linear and cartesian types:

$$\begin{tikzcd}
\mathcal{L}_\Gamma \arrow[r, bend right, "M"'] \arrow[r, phantom, "\perp"] & \mathcal{T}_\Gamma \arrow[l, bend right, "L"']
\end{tikzcd}$$

where the exponential modality from traditional linear logic is understood as the comonad $! = LM$. The decomposition of the exponential into an adjunction goes back to at least [1], and is given an interesting new light in [6], where it is seen as a particular case of a more general procedure of encoding structure in contexts.

The rules for the operators $M$ and $L$ are presented in Figure 2. The interpretation of $L$ and $M$ as an adjoint pair is already present at the syntactic level. We can show that they form instances of a Haskell-like **Functor** class, by constructing terms: $\texttt{fmapM} : (A \multimap B)_M \to A_M \to B_M$ and $\texttt{fmapL} : L(A \to B) \multimap (LA \multimap LB)$, satisfying the functor laws.

Furthermore, we can construct a "counit" $\epsilon : LM \implies 1$ satisfying the universal property of adjunction (thanks to L-U). The syntactic formulation of the statement becomes:

▶ **Theorem 1** ($L \dashv M$). *There is a term $\Gamma; \beta_1 : B_{LM} \vdash \epsilon_B : B$ with the following property: For any term: $\Gamma; y : A_L \vdash f : B$, there is a unique term $\Gamma, x : A \vdash g : B_M$ such that $\Gamma; y : A_L \vdash \epsilon_B[\mathit{let}\ x\ \mathit{be}\ y\ \mathit{in}\ g_L/\beta_1] \equiv f : B$.*

Based on this knowledge we expect the right adjoint $M$ to preserve limits, and indeed we find an isomorphism: $A_M \times B_M \cong (A\&B)_M$. We can now also reformulate some common results about the exponential modality using $! = LM$, such as $(A\&B)_{LM} \cong A_{LM} \otimes B_{LM}$.

## 2.3 $\sqcap$ and $\sqsubset$

Since we allow linear types to depend on terms of cartesian types, we can form new versions of the $\Pi$- and $\Sigma$-types, denoted $\sqcap$ and $\sqsubset$, respectively. The typing rules for these are presented in Figure 3. The sense in which $\sqcap$ and $\sqsubset$ are "linear analogues" of $\Pi$ and $\Sigma$ can be formalized in the following way:

$$\dfrac{\Gamma \vdash A \ \text{type} \quad \Gamma, x : A \vdash B \ \text{linear}}{\Gamma \vdash \sqcap_{x:A} B \ \text{linear}} \ \sqcap\text{-F} \qquad \dfrac{\Gamma \vdash A \ \text{type} \quad \Gamma, x : A \vdash B \ \text{linear}}{\Gamma \vdash \sqsubset_{x:A} B \ \text{linear}} \ \sqsubset\text{-F}$$

$$\dfrac{\begin{array}{c} \vdash \Gamma; \Xi \ \text{ctxt} \\ \Gamma, x : A; \Xi \vdash b : B \end{array}}{\Gamma; \Xi \vdash \lambda x.b : \sqcap_{x:A} B} \ \sqcap\text{-I} \qquad \dfrac{\Gamma \vdash s : A \quad \Gamma; \Xi \vdash b : B[s/x]}{\Gamma; \Xi \vdash (s, b) : \sqsubset_{x:A} B} \ \sqsubset\text{-I}$$

$$\dfrac{\Gamma; \Xi \vdash t : \sqcap_{x:A} B \quad \Gamma \vdash a : A}{\Gamma; \Xi \vdash t(a) : B[a/x]} \ \sqcap\text{-E} \qquad \dfrac{\begin{array}{c} \vdash \Gamma; \Xi' \ \text{ctxt} \\ \Gamma \vdash C \ \text{linear} \\ \Gamma; \Xi \vdash t : \sqsubset_{x:A} B \quad \Gamma, x : A; \Xi', y : B \vdash c : C \end{array}}{\Gamma; \Xi, \Xi' \vdash \text{let } x, y \text{ be } t \text{ in c : C}} \ \sqsubset\text{-E}$$

$$\dfrac{\Gamma; \Xi \vdash \lambda x.b(a) : \sqcap_{x:A} B}{\Gamma; \Xi \vdash \lambda x.b(a) \equiv b[a/x] : B[a/x]} \ \sqcap\text{-C} \qquad \dfrac{\Gamma; \Xi \vdash \text{let } x, y \text{ be } (s, t) \text{ in c : C}}{\Gamma; \Xi \vdash \text{let } x, y \text{ be } (s, t) \text{ in c} \equiv c[s/x][t/y] : C} \ \sqsubset\text{-C}$$

**Figure 3** Typing rules for $\sqsubset$ and $\sqcap$

▶ Proposition 2. For all $\Gamma \vdash A$ type and $\Gamma, x : A \vdash B$ linear, there is an isomorphism:

$$\Pi_{x:A} B_M \cong (\sqcap_{x:A} B)_M$$

We would like to show a similar result relating $\Sigma$ and $\sqsubset$, but for this we need a couple of additional rules. First, we assume uniqueness rules for $\Sigma$ and $\sqsubset$, asserting that the elimination rule followed by the introduction rule is the identity. In other words, for any $p : \Sigma_{x:A} B$ and $q : \sqsubset_{x:C} D$, we have $(pr_1(p), pr_2(p)) \equiv p$ and let $x, y$ be $q$ in $(x, y) \equiv q$ [1]: Second, we assume a kind of naturality rule for the $L$ modality:

$$\dfrac{\begin{array}{c} \Gamma; \Xi, y : B \vdash e : C \\ \Gamma, x : A; \Xi' \vdash u : B \\ \Gamma; \Xi'' \vdash t : A_L \end{array}}{\Gamma; \Xi, \Xi', \Xi'' \vdash e[\text{let } x \text{ be } t \text{ in } u/y] \equiv \text{let } x \text{ be } t \text{ in } e[u/y] : C} \ \text{Nat}_L$$

▶ Proposition 3. Assuming the $\text{Nat}_L$ and the uniqueness rules for $\Sigma$ and $\sqsubset$, there is a linear isomorphism[2]

$$(\Sigma_{x:A} B)_L \cong \sqsubset_{x:A} B_L$$

As outlined in section 3.2, the semantic interpretation of the type formers $\Pi$, $\sqcap$ and $\Sigma$, $\sqsubset$ are as right and left adjoints to reindexing functors respectively. Based on this interpretation we can understand these equivalence results through the diagram:

$$\begin{tikzcd}[column sep=huge, row sep=huge]
\mathcal{L}_{\Gamma.A} \arrow[r, bend right, "M_{\Gamma.A}"'] \arrow[r, phantom, "\perp"] \arrow[d] & \mathcal{T}_{\Gamma.A} \arrow[l, bend right, "L_{\Gamma.A}"'] \arrow[d] \\
\mathcal{L}_{\Gamma} \arrow[u, bend left, "\sqsubset_A"] \arrow[u, bend right, "\sqcap_A"'] \arrow[u, phantom, "\dashv \ \dashv"] \arrow[r, bend right, "M_{\Gamma}"'] \arrow[r, phantom, "\perp"] & \mathcal{T}_{\Gamma} \arrow[l, bend right, "L_{\Gamma}"'] \arrow[u, bend left, "\Sigma_A"] \arrow[u, bend right, "\Pi_A"'] \arrow[u, phantom, "\dashv \ \dashv"]
\end{tikzcd}$$

[1] The former is provable as a propositional identity [10, Corollary 2.7.3]. Perhaps it is possible to obtain a similar result for $\sqsubset$, using the "surrogate equality" described in the end of Section 2.2

[2] Here a linear isomorphism, $A \cong B$, means a pair $f : A \multimap B$, $g : B \multimap A$ such that the composite is judgmentally equal to the identity. We discuss the weaker notion of linear equivalence in Section 4.3

# 3 Semantics

## 3.1 Structural semantic core

Our semantic exploration of linear dependent type theory begins with the notion of a model. For the cartesian fragment of our theory, we follow [4] and ask for a *comprehension category*, $\pi : \mathcal{T} \to \mathcal{C}^{\to}$, where $\mathcal{C}$ is a category of context with terminal object, and the fibrations $\mathcal{T}_\Gamma$ contains the cartesian types over $\Gamma$. For the linear fragment of our theory, we would like a fibration $q : \mathcal{L} \to \mathcal{C}$ where each fiber $\mathcal{L}_\Gamma$ is a symmetric monoidal category and the reindexing functors are symmetric monoidal. This is captured in the notion of a *(lax) monoidal fibration*:

▶ **Definition 4.** A **lax monoidal fibration** [14] is a fibration $p : E \to B$ along with

**1.** Two functors $\otimes : E \times_B E \to E$ and $I : B \to E$ fitting into the following diagram:

$$\begin{tikzcd}
E \times_B E \arrow[r, "\otimes"] \arrow[dr] & E \arrow[d, "p"] & B \arrow[l, "I"'] \arrow[dl, "1_B"] \\
& B &
\end{tikzcd}$$

**2.** Three fibred natural isomorphisms $\alpha, \lambda$ and $\rho$ associated with the diagrams:

$$\begin{tikzcd}
E \times_B E \times_B E \arrow[r, "1_E \times_B \otimes"] \arrow[d, "\otimes \times_B 1_E"'] & E \times_B E \arrow[d, "\otimes"] \\
E \times_B E \arrow[r, "\otimes"'] \arrow[ur, phantom, "\overset{\alpha}{\Longrightarrow}"] & E
\end{tikzcd}$$

and

$$\begin{tikzcd}
B \times_B E \arrow[r, "I \times_B 1_E"] \arrow[dr, "\pi_2"'] & E \times_B E \arrow[d, "\otimes"] & E \times_B B \arrow[l, "1_E \times I"'] \arrow[dl, "\pi_1"] \\
& E &
\end{tikzcd}$$

with $\overset{\lambda}{\Longrightarrow}$ and $\overset{\rho}{\Longleftarrow}$ in the two triangles.

**3.** such that $\alpha$, $\lambda$ and $\rho$ satisfies the pentagon and triangle identities in each fiber.

**4.** for every $b \in B$, $\rho_{I_b} = \lambda_{I_b}^{-1} : I_b \otimes I_b \to I_b$

To avoid any coherence problems, we require the both the comprehension category and the monoidal fibration to be *split*.

▶ **Definition 5.** A **model for linear dependent type theory** consists of a split comprehension category $\pi : \mathcal{T} \to \mathcal{C}^{\to}$ and a split symmetric monoidal fibration $q : \mathcal{L} \to \mathcal{C}$, as illustrated in the following picture:

$$\begin{tikzcd}
\mathcal{L} \arrow[dr, "q"] & \mathcal{T} \arrow[r, "\pi"] \arrow[d, "p"] & \mathcal{C}^{\to} \arrow[dl, "\text{cod}"] \\
& \mathcal{C} &
\end{tikzcd}$$

This provides the necessary machinery to interpret all the structural rules of our theory as well as the rules for $\otimes$ and $I$, by constructing an interpretation function $[\![-]\!]$, which sends:

- Cartesian contexts $\Gamma$ to objects of $\mathcal{C}$, considered up to judgmental equality and renaming of bound variables.

- Linear contexts $\Xi = a_1 : A_1, a_2 : A_2, \dots a_n : A_N$ in $\Gamma$ to objects $[[\Xi]] = \bigotimes_{i=1}^{n}[[A_i]]$ of $\mathcal{L}_{[[\Gamma]]}$.
- Cartesian types $A$ in $\Gamma$ to objects of $\mathcal{T}_{[[\Gamma]]}$.
- Linear types $B$ in $\Gamma$ to objects of $\mathcal{L}_{[[\Gamma]]}$.
- Cartesian terms $M : A$ in $\Gamma$ to sections of the projection morphism $\pi([[A]]) : [[\Gamma, A]] \to [[\Gamma]]$.
- Linear terms $b : B$ in $\Gamma; \Xi$ to morphisms $[[b]] : [[\Xi]] \to [[B]]$.

## 3.2 Semantic type formers

Equipped with the baseline structure of a model in which we can interpret the structural rules of our theory, we formulate the conditions under which such models support various type formers. From now on, we will assume that the comprehension category comprising the core of our syntax is full, i.e. that the functor $\pi : \mathcal{T} \to \mathcal{C}^{\to}$ is full and faithful. This simplifies the semantic interpretation of many type formers.

The interpretation of the purely linear type formers $\otimes, I, \multimap, \&, \oplus, \top$ and 0 in symmetric monoidal categories is well known. See for instance [9]. Notice that $\otimes$ and $I$ types are supported in any model. For a model to support the type formers $\multimap, \&, \oplus, \top$ and 0, correspond to the condition that the fibers of $\mathcal{L}$ have weak versions of internal homs, binary products and coproducts, and terminal and initial object, and that these are stable under reindexing functors.

### 3.2.1 $\Pi$ and $\Sigma$

What it means for a model of linear dependent type theory to support $\Pi$-types is directly inherited from the standard, non-linear case; we require right adjoints to reindexing functors satisfying a Beck-Chevalley condition.

As the rules $\Sigma$ contains one more eliminator than usual ($\Sigma$-E$_2$ from Section 2.1), one might wonder whether this poses additional clauses in the definition of the semantic type former. But as it turns out, the relevant condition will always hold in any model supporting $\Sigma$-types:

▶ **Definition 6.** A model of LDTT **supports $\Sigma$-types** if it satisfies the following:

1. For all $A \in \mathcal{T}_\Gamma$, the induced functor $\pi_A^* : \mathcal{T}_\Gamma \to \mathcal{T}_{\Gamma.A}$ has a left adjoint, $\Sigma_A$,
2. (Beck-Chevalley) such that for all pullbacks

$$\begin{tikzcd}
\Gamma.E \arrow[r, "q_{E,E'}"] \arrow[d, "\pi_E"] & \Delta.E' \arrow[d, "\pi_{E'}"] \\
\Gamma \arrow[r, "f"] & \Delta
\end{tikzcd}$$

the natural transformation: $\Sigma_E q^* \to f^* \Sigma_{E'}$ is a natural isomorphism, and

3. the induced map *pair* $: \Gamma.A.B \to \Gamma.\Sigma_A B$ is an isomorphism

We will denote the inverse of *pair* by $(pr_1, pr_2)$, when it exists. This structure is sufficient to support new elimination rule ($\Sigma$-E$_2$).

▶ **Theorem 7.** *If a model of LDTT supports $\Sigma$-types, then for every object $C \in \mathcal{L}_{\Gamma.\Sigma_A B}$, morphism $c : \Xi \to (pair_{A,B})^* C$ in $\mathcal{L}_{\Gamma.A.B}$ and section $s : \Gamma \to \Gamma.\Sigma_A B$, there exists a morphism $\hat{c}_s : s^*(pr_1, pr_2)^*\Xi \to s^* C$ such that given sections $a : \Gamma \to \Gamma.A$ and $b : \Gamma.A \to \Gamma.A.B$ we have $\hat{c}_{(a,b)} = a^* b^* c : a^* b^* \Xi \to a^* b^* C$.*

**Proof.** Let $\hat{c}_s = c(pr_1, pr_2)^* s^* c$. Given sections $a : \Gamma \to \Gamma.A$ and $b : \Gamma.A \to \Gamma.A.B$ we compose with $pair$ to get the section $(a, b) = pair \circ ba : \Gamma \to \Gamma.\Sigma_A B$. We have:

$$\hat{c}_{(a,b)} = (pr_1, pr_2)^*(a, b)^* c = (pr_1, pr_2)^* pair^* b^* a^* c = b^* a^* c$$

◀

### 3.2.2 Identity types

The situation for Id-types requires a bit more care. If one wants to keep the theory intensional, we need to add condition (2) to make sure that the semantic identity types satisfy the added elimination rule, =-E$_2$.

▶ **Definition 8** (Id-types)**.** A model of LDTT **supports Id-types** if, for all $A \in \mathcal{T}_\Gamma$, there exists an object $Id_A \in \mathcal{T}_{\Gamma.A.\pi_A^* A}$ and a morphism $r_A : \Gamma.A \to \Gamma.A.\pi_A^* A.Id_A$ such that $\pi_{Id_A} \circ r_A = v_A$, and:

1. For any commutative diagram:

$$\begin{tikzcd} \Gamma.A \arrow[r] \arrow[d, "r_A"] & \Delta.C \arrow[d, "\pi_C"] \\ \Gamma.A.\pi_A^* A.Id_A \arrow[r] & \Delta \end{tikzcd}$$

   there exists a lift $J : \Gamma.A.\pi_A^* A.Id_A \to \Delta.C$ making the two triangles commute.
2. For any pair of objects, $C, \Xi \in \mathcal{L}_{\Gamma.A.\pi_A^* A.Id}$, sections $M, N : \Gamma \to \Gamma.A$, $P : \Gamma \to \Gamma.M^*(N^+)^* Id_A$, and morphism $c : r_A^* \Xi \to r_A^* C$, there exists a morphism $\hat{c}_{[M,N,P]} : M^*(N^+)^* P^* \Xi \to M^*(N^+)^* P^* C$ such that $\hat{c}_{[M,M,\mathrm{refl}]} = M^* c$.

Notice that if our type theory has extensional id-types, in the sense that $a =_A b$ implies $a \equiv b$, then the third condition is always met.

### 3.2.3 $\sqcap$- and $\sqsubset$-types

The semantic type formers for the linear dependent $\sqcap$ and $\sqsubset$ is akin to that of $\Pi$ and $\Sigma$. They are given by adjoints to the functors between fibers of $\mathcal{L}$ induced by the projection maps in $\mathcal{C}$.

▶ **Definition 9.** A model of LDTT **supports $\sqcap$-types** if, for all $A \in \mathcal{T}_\Gamma$, the induced monoidal functor $\pi_A^* : \mathcal{L}_\Gamma \to \mathcal{L}_{\Gamma.A}$ has a monoidal right adjoint, $\sqcap_A$ satisfying the following Beck-Chevalley condition:

For all pullback squares in $\mathcal{C}$ of the following form:

$$\begin{tikzcd} \Gamma.E \arrow[r, "q_{E,E'}"] \arrow[d, "\pi_E"] & \Delta.E' \arrow[d, "\pi_{E'}"] \\ \Gamma \arrow[r, "f"] & \Delta \end{tikzcd}$$

the canonical natural transformation $f^* \sqcap_{E'} \to \sqcap_E q_{E,E'}^*$ is a natural isomorphism.

▶ **Definition 10.** It **supports $\sqsubset$-types** if, for all $A \in \mathcal{T}_\Gamma$, the functor every $\pi_A^*$ has a monoidal left adjoint, satisfying the following:

1. (Beck-Chevalley): For all pullbacks squares as above, the natural transformation $\sqsubset_E q^* \to f^* \sqsubset_{E'}$ is a natural isomorphism.
2. (Frobenius reciprocity): For all objects $\Xi \in \mathcal{L}_\Gamma$ and $B \in \mathcal{L}_{\Gamma.A}$, the canonical morphism $\sqsubset_A (\Xi\{\pi_A\} \otimes B) \to \Xi \otimes \sqsubset_A B$ is an isomorphism.

### 3.2.4 The operators $M$ and $L$

▶ **Definition 11.** A model of LDTT with unit **supports the operators** $M$ **and** $L$ if there exists functors $M : \mathcal{L} \leftrightarrow \mathcal{T} : L$ which are cartesian with respect to the fibrations $p : \mathcal{T} \to \mathcal{C}$ and $q : \mathcal{L} \to \mathcal{C}$, such that

- $L \dashv M$ is a fibred adjunction,
- $L(1) \cong I$
- and there is an isomorphism of hom-sets:

$$\mathcal{L}_{\Gamma.A}(\pi_A^*(\Xi'), \pi_A^*(B)) \cong \mathcal{L}_\Gamma(LA \otimes \Xi', B)$$

.

Recall that a fibred adjunction implies that there are natural isomorphisms making the following diagram commute:

$$\begin{tikzcd}
\mathcal{L}_{\Gamma.A} \arrow[d, "\pi_A^*"] \arrow[r, bend right, "M_{\Gamma.A}"'] & \mathcal{T}_{\Gamma.A} \arrow[l, bend right, "L_{\Gamma.A}"'] \arrow[d, "\pi_A^*"] \\
\mathcal{L}_\Gamma \arrow[r, bend right, "M_\Gamma"'] & \mathcal{T}_\Gamma \arrow[l, bend right, "L_\Gamma"']
\end{tikzcd}$$

which from a syntactic perspective ensures that $M$ and $L$ commute with substitution.

Note that the interpretation of a term $\Gamma\cdot \vdash \sigma(a) : A_M$ arises from the adjunction via $\sigma : \mathcal{L}_\Gamma(I, A) \cong \mathcal{L}_\Gamma(L(1), A) \cong \mathcal{T}_\Gamma(1, M(A))$.

The final condition of the definition is what yields the elimination and computation rules L-U, and while it might appear somewhat unnatural semantically, it does turn out to hold in a broad variety of models, due to the following result:

▶ **Theorem 12.** *In a model of LDTT that supports $\multimap$ type formers, then any fibred adjunction $L \dashv M$ where $L(1) \cong I$ satisfies $\mathcal{L}_{\Gamma.A}(\pi_A^*(\Xi'), \pi^*(B)) \cong \mathcal{L}_\Gamma(LA \otimes \Xi', B)$.*

**Proof.** A model supporting internal homs must have reindexing functions which preserve these. That is, we have an isomorphism $\pi_A^*[\Xi, B] \cong [\pi_A^*\Xi, \pi_A^*B]$. We get a chain of isomorphisms:

$$\mathcal{L}_\Gamma(LA \otimes \Xi, B) \cong \mathcal{L}_\Gamma(LA, [\Xi, B]) \cong \mathcal{T}_\Gamma(A, M_\Gamma[\Xi, B]) \cong$$
$$\mathcal{T}_{\Gamma.A}(1, \pi_A^*(M_\Gamma[\Xi, B])) \cong \mathcal{T}_{\Gamma.A}(1, M_{\Gamma.A}\pi_A^*[\Xi, B])) \cong \mathcal{L}_{\Gamma.A}(L_{\Gamma.A}(1), \pi_A^*[\Xi, B])) \cong$$
$$\mathcal{L}_{\Gamma.A}(I, \pi_A^*[\Xi, B])) \cong \mathcal{L}_{\Gamma.A}(I, [\pi_A^*\Xi, \pi_A^*B])) \cong \mathcal{L}_{\Gamma.A}(\pi_A^*\Xi, \pi_A^*B).$$

◀

## 4 Diagram Model

The main novelty of this paper is the Diagram model of linear dependent type theory. This model extends the groupoid model of dependent type theory [3] to support linear types, while still maintaining a higher dimensional interpretation of the identity type. Most interestingly, perhaps, it provides a model in which we can support univalent universes, both for cartesian and linear types. The diagram model can be seen as a natural generalization of the set indexed families model described by [13]. We briefly recall the set indexed families model below as a useful comparison to the diagrams model.

## 4.1 Set indexed families

▶ **Definition 13** ($Fam(\mathcal{C})$)**.** For an arbitrary category $\mathcal{C}$, let $Fam(\mathcal{C})$ denote the category whose objects consists of pairs $(S, f)$ where $S$ is a set and $f$ is a function $f : S \to Ob(\mathcal{C})$. Morphisms of $Fam(\mathcal{C})$ are pairs $(u, \alpha) : (S, f) \to (S', g)$ where $u : S \to S'$ and $\alpha : S \to \mathrm{Mor}(\mathcal{C})$ such that $\alpha(s) : f(s) \to g(u(s))$ for all $s \in S$.

By projecting a family to its indexing set, we get a fibration $p : Fam(\mathcal{C}) \to \mathbf{Set}$ and a comprehension category by defining $\pi(S, f) = fst : \{(s, t) \mid s \in S, t : \top \to f(s)\} \to S$. [3]

Letting $\mathcal{C} = \mathbf{Set}$ thus gives us a (full, split) comprehension category, forming the cartesian part of our model. For the linear part, we can for any symmetric monoidal category $\mathcal{V}$ form a monoidal fibration by a simple pointwise construction, giving us the following picture:

$$\begin{tikzcd}
Fam(\mathcal{V}) \arrow[dr, "q"] & Fam(\mathbf{Set}) \arrow[d, "p"] \arrow[r, "\pi"] & \mathbf{Set}^{\to} \arrow[dl, "cod"] \\
 & \mathbf{Set} &
\end{tikzcd}$$

In this setting, most type formers will be given by a simple pointwise construction which are preserved under reindexing. It turns out that the families model supports the type formers $\otimes, I, \multimap, \oplus, 0, \&$, and $\top$ if $\mathcal{V}$ is a monoidal category which is closed, has binary coproducts, initial object, binary products and terminal object respectively.

It supports $\sqcap$-types if $\mathcal{V}$ has small products, and $\sqsubset$ if $\mathcal{V}$ has small coproducts that distribute over $\otimes$ (Frobenius reciprocity).

The families model of course also supports $\Pi$ and $\Sigma$-types, and since its identity types are extensional, the extra condition posed on our semantic identity types poses no additional difficulty.

Whenever $\mathcal{V}$ is a concrete category, the adjunction $F \dashv U$ will induce a fiber adjunction between the corresponding fibrations, which forms support for the operators $M$ and $L$ (as long as $F(\mathbf{1}) \cong I$).

## 4.2 Diagrams in monoidal categories

For any category $\mathcal{C}$, there is a fibration $cod : \mathbf{Diag}(\mathcal{C}) \to \mathbf{Cat}$, whereby $\mathbf{Diag}(\mathcal{C})$ refers to the category of diagrams in $\mathcal{C}$, i.e. consisting of objects $J : \mathcal{A} \to \mathcal{C}$, and morphisms between $J : \mathcal{A} \to \mathcal{C}$ and $J' : \mathcal{B} \to \mathcal{C}$ are functors $F : \mathcal{A} \to \mathcal{B}$ equipped with a natural transformation $\eta : J \implies J' \circ F$. In other words, the fibers of $\mathbf{Diag}(\mathcal{C})$ are functor categories, which we write $[\Gamma, \mathcal{C}]$, for any small category $\Gamma$. Any functor $F : \mathcal{A} \to \mathcal{B}$ in the base induces a canonical lift $F^* : [\mathcal{B}, \mathcal{C}] \to [\mathcal{A}, \mathcal{C}]$ simply given by precomposition.

When $\mathcal{C}$ has a terminal object $\top$ such that the collections $\mathcal{C}(\top, A)$ are small for any $A \in \mathcal{C}$, we form a comprehension category:

$$\begin{tikzcd}
\mathbf{Diag}(\mathcal{C}) \arrow[r, "\pi"] \arrow[d, "dom"] & \mathbf{Cat}^{\to} \arrow[dl, "cod"] \\
\mathbf{Cat} &
\end{tikzcd}$$

where the functor $\pi$ sends a diagram $A : \Gamma \to \mathcal{C}$ to the **Grothendieck construction** for $A$, i.e. the category whose objects are pairs $(\gamma, t_\gamma)$ where $\gamma \in \Gamma$, $t_\gamma : \top \to A(\gamma)$. Morphisms

[3] As long as $\mathcal{C}$ has a terminal object and the hom-sets $\mathcal{C}(\top, A)$ are small for all $A \in \mathcal{C}$.

$(\gamma, t_\gamma) \to (\gamma', t'_{\gamma'})$ consists of morphisms $u : \gamma \to \gamma'$ such that $A(u) \circ t_\gamma = t'_{\gamma'}$. [4]

When $\mathcal{C}$ is any symmetric monoidal category $\mathcal{V}$, there is an obvious symmetric monoidal structure on each fiber $[\Gamma, \mathcal{V}]$, given pointwise.

Restricting the base of the fibration to groupoids instead of categories, and setting $\mathcal{C} = \mathbf{Gpd}$ we get a model of linear dependent type theory which expands the groupoid model by Hofmann and Streicher [3]:

$$\begin{tikzcd}
\mathbf{Diag}(\mathcal{V}) \arrow[drr, "dom"] & & \mathbf{Diag}\,(\mathbf{Gpd}) \arrow[r, "\pi"] \arrow[d, "dom"] & \mathbf{Gpd}^{\rightarrow} \arrow[dl, "cod"] \\
& & \mathbf{Gpd} &
\end{tikzcd}$$

Since $\top$ is the groupoid $\mathbf{1}$ consisting of a single object, we will equate the functor $t_\gamma : 1 \to A(\gamma)$ with an object $a_\gamma$ of $A(\gamma)$, and the natural transformation $\alpha : A(u) \circ t_\gamma \implies t'_{\gamma'}$ with a morphism $\alpha_\gamma : A(u)(a_\gamma) \to a'_{\gamma'}$.

As shown in [3], this model supports $\Pi$ and $\Sigma$ type formers, and provides an interesting interpretation of the identity type $Id_A$ as the arrow category of $A$.

This construction satisfies the additional requirement in our definition of semantic identity types:

▶ **Theorem 14.** *Given a $\Gamma$-indexed groupoid $A$, diagrams $C, \Xi \in [\Gamma.A^{\rightarrow}, \mathcal{V}] \cong [\Gamma.A.\pi_A^* A, Id_A, \mathcal{V}]$, sections $M, N : \Gamma \to \Gamma.A$, $P : \Gamma.A \to \Gamma.Id_A$ and a natural transformation $c : \Xi \circ r_A \implies C \circ r_A$, there exists a natural transformation $\hat{c}_{[M,N,P]} : \Xi \circ P^+ \circ N^+ \circ M \implies C \circ P^+ \circ N^+ \circ M$ such that $\hat{c}_{[M,M,refl]} = c \circ M$* [5]

**Proof.** The key point to observe is that there is always an isomorphism $(\gamma, P_\gamma : M_\gamma \to N_\gamma) \cong (\gamma, 1_{M_\gamma} : M_\gamma \to M_\gamma)$ given by the commutative diagram:

$$\begin{tikzcd}
(\gamma, M_\gamma) \arrow[r, "1_M"] \arrow[d, "P"] & (\gamma, M_\gamma) \arrow[d, "1_M"] \\
(\gamma, N_\gamma) \arrow[r, "P^{-1}"] & (\gamma, M_\gamma)
\end{tikzcd}$$

forming a collection of isomorphisms in $\Gamma.A^{\rightarrow}$ giving rise to a natural isomorphism $\phi : r_A \circ M \implies P^+ \circ N^+ \circ M$. We define $\hat{c}[M, N, P]$ as the composite:

$$\Xi \circ P^+ \circ N^+ \circ M \xrightarrow{\Xi_\phi} \Xi \circ r_A \circ M \xrightarrow{cM} C \circ r_A \circ M \xrightarrow{C_{\phi^{-1}}} C \circ P^+ \circ N^+ \circ M$$

To see that the computation rule holds, we only need to notice that when $M \equiv N$ and $P = \text{refl}(M)$, $\phi$ is the identity natural transformation. ◀

As in the families model, limits and colimits are constructed pointwise, and preserved by precomposition, so the model supports &, $\top$, $\oplus$, 0, if $\mathcal{V}$ has binary products, terminal object, binary coproducts and initial object respectively.

When it comes to $\multimap$, we utilize the following result:

[4] If $\mathcal{C}$ is a 2-category, this can be weakened so that morphisms $(\gamma, t_\gamma) \to (\gamma', t'_{\gamma'})$ are pairs $(u, \alpha)$, where $u : \gamma \to \gamma'$ and $\alpha$ is a 2-cell $\alpha : A(u) \circ t_\gamma \implies t'_{\gamma'}$.

[5] Where the sections $N^+ : \Gamma.A \to \Gamma.A.\pi_A^* A$ and $P^+ : \Gamma.A.\pi_A^* \to \Gamma.A.\pi_A^*.Id_A$ are weakenings of $N$ and $P$, i.e. functors ignoring the additional arguments

▶ **Theorem 15.** *If $\mathcal{V}$ has internal homs and is complete, $[\mathcal{C}, \mathcal{V}]$ also has internal homs, defined for $F, G \in [\mathcal{C}, \mathcal{V}]$ by:*

$$[F, G] := \int_{x \in \mathcal{C}} [Fx, Gx])$$

These are preserved under reindexing, implying that the diagrams model supports $\multimap$ if $\mathcal{V}$ is monoidal closed and complete.

▶ **Definition 16.** For any functor $p : \mathcal{A} \to \mathcal{B}$ in the base, a left or right adjoint to the induced functor $p^* : [\mathcal{B}, \mathcal{V}] \to [\mathcal{A}, \mathcal{V}]$ is called a **left or right Kan extension** of $p$.

We recall the following fact about Kan extensions:

▶ **Theorem 17.** *Left (right) Kan extensions along $p : \mathcal{A} \to \mathcal{B}$ between two arbitrary small categories $\mathcal{A}$ and $\mathcal{B}$ exists if and only if $\mathcal{V}$ has all colimits (limits).*

The result above ensures the existence of left and right adjoints to reindexing functors in the diagrams model as long as $\mathcal{V}$ is co-complete or complete, respectively. Since our reindexing functors are given by precomposition they will always satisfy the Beck-Chevalley condition. Again, in order to support $M$ and $L$, we can lift an adjunction between $\mathcal{V}$ and **Gpd** to fiber adjunction between the respective diagram categories. Therefore, for any diagrams model which supports $\multimap$ to support $M$ and $L$, it suffices to display an adjunction

$$\begin{tikzcd}
\mathcal{V} \arrow[r, bend right, "L_0"'] & \mathbf{Gpd} \arrow[l, bend right, "M_0"']
\end{tikzcd}$$

such that $L(1) \cong I$.

▶ Remark. When the functor $\mathcal{V}(I, -) : \mathcal{V} \to \mathbf{Set}$ has a left adjoint $F$, we get an adjunction between $\mathbf{Diag}(\mathcal{V})$ and $\mathbf{Diag}(\mathbf{Gpd})$, induced by:

$$\begin{tikzcd}
\mathcal{V} \arrow[r, bend right, "\mathcal{V}(I,-)"'] \arrow[r, phantom, "\perp"] & \mathbf{Set} \arrow[l, bend right, "F"'] \arrow[r, bend right, "\delta"'] \arrow[r, phantom, "\perp"] & \mathbf{Gpd} \arrow[l, bend right, "\pi_0"']
\end{tikzcd}$$

where $\pi_0$ is the functor sending a groupoid to its set of connected components.

▶ **Theorem 18.** *There are models in which $M$ is not faithful.*

**Proof.** Let $\mathcal{V}$ to be **Gpd** so that $L = \delta\pi_0$ and $M = \delta\mathbf{Gpd}(1, -)$. This induces a fiber adjunction $L \dashv M$ where $L(1) = 1$, but $M$ is not faithful. ◀

### 4.2.1 Universes in the diagrams model

To support universes, assuming one inaccessible cardinal allows us to shift our perspective to from the category of small groupoids, **Gpd**, to the category **GPD** of all groupoids. Among the objects of **GPD** we find the core (i.e. maximal sub-groupoid) of **Gpd** and $\mathcal{V}^{core}$. This allows us to define our cartesian and linear universes in any context $\Gamma$ as the functors:

$\mathbb{U} : \Gamma \to \mathbf{GPD}$
$\mathbb{L} : \Gamma \to \mathbf{GPD}$

which are constant at $\mathbf{Gpd}^{core}$ and $\mathcal{V}^{core}$, respectively. Any section $s : \Gamma \to \Gamma.\mathbb{U}$ will determine a functor $\hat{s} : \Gamma \to \mathbf{Gpd}$, which we embed $\mathbf{Gpd} \to \mathbf{GPD}$ to get an interpretation of $El(s)$. Similarly, we get from each section $s : \Gamma \to \Gamma.\mathbb{L}$, a functor $El(s) : \Gamma \to \mathcal{V}$. It is easily seen that defining the linear universe via the core of $\mathcal{V}$ gives rise to the following interesting property, hinting at the possibility of a linear univalence axiom:

▶ Proposition 19. For a linear universe defined as above via $\mathcal{V}^{core}$, and two sections $s, t : \Gamma \to \Gamma.\mathbb{L}$, an isomorphism $\alpha : El(t) \cong El(s)$ gives rise to a section $p : \Gamma \to \Gamma.Id_{\mathbb{L}}\{s\}\{t\}$.

## 4.3 Univalence in linear dependent types

A key feature of the groupoid model is that it provides a model of dependent type theory where there might be nontrivial terms of the identity type. A natural question to ask is whether this higher dimensional feature of type theory can be extended to the linear dependent setting. In particular, we might wish for a linear analogue to the univalence axiom to hold:

$$\frac{\begin{array}{c}\Gamma \vdash A : \mathbb{L}\\ \Gamma \vdash B : \mathbb{L}\\ \Gamma; \cdot \vdash f : El(A) \multimap El(B)\\ \Gamma; \cdot \vdash g : El(B) \multimap El(A)\\ \Gamma; \cdot \vdash h : El(B) \multimap El(A)\\ \Gamma \vdash p : \sigma(g \circ f) =_{(El(A) \multimap El(A))_M} (id_A)_M\\ \Gamma \vdash q : \sigma(f \circ h) =_{(El(B) \multimap El(B))_M} (id_B)_M\end{array}}{\Gamma \vdash ua(f) : A =_{\mathbb{L}} B}\ \text{L-ua-I}$$

To define the corresponding computation rule, we will the make use of a linear version of `transport`, which is easily definable through identity elimination. Given $\Gamma, x : C \vdash D$ linear, and an identity $p : a =_C b$, we get a function:

$$p^* : D[a/x] \multimap D[b/x]$$

which we call the linear **transport along p**. This function has an inverse and thus yields for any $q : A =_L B$ a linear equivalence $El(A) \cong El(B)$.

The computation rules for the univalence axiom asserts that the process creating equivalences from identities forms an inverse to univalence:

$$\frac{\Gamma \vdash ua(f)^* : El(A) \multimap El(B)}{\Gamma \vdash ua(f)^* \equiv f : El(A) \multimap El(B)}\ \text{L-ua-C}_1$$

$$\frac{\Gamma \vdash ua(p^*) : A =_{\mathbb{L}} B}{\Gamma \vdash ua(p^*) \equiv p : A =_{\mathbb{L}} B}\ \text{L-ua-C}_2$$

The semantic interpretation of the procedure of turning an identity to an equivalence becomes the following:

▶ **Lemma 20.** *In the diagram model, given two sections $A, B : \Gamma \to \Gamma.C$, there is an isomorphism between the sections to the projection morphism $\Gamma.B^{+^*}A^*Id_C \to \Gamma$, and natural isomorphisms $A \cong El(B)$ in $[\Gamma, \mathcal{V}]$.*

**Proof.** Since $\Gamma.(B^+)^*A^*Id_{\mathbb{L}} \cong \Gamma.\mathcal{V}^{core\to}$, a section $p : \Gamma \to \Gamma.\mathcal{V}^{core\to}$ defines for every morphism $\alpha : \gamma \to \gamma' \in \Gamma$ a naturality square:

$$\begin{tikzcd}
A(\gamma) \arrow[r, "p(\gamma)"] \arrow[d, "A(\alpha)"] & B(\gamma) \arrow[d, "B(\alpha)"] \\
A(\gamma') \arrow[r, "p(\gamma')"] & B(\gamma')
\end{tikzcd}$$

where $A(\gamma)$ and $B(\gamma)$ are isomorphisms. Conversely, every natural isomorphism $El(A) \cong El(B)$ clearly defines such a section. ◀

▶ **Theorem 21.** *When $M$ factors through* **Sets** *as in Remark 4.2, the linear univalence axiom holds in the diagram model. That is, given the following data:*

- *sections: $A, B : \Gamma \to \Gamma.\mathbb{L}$*
- *morphisms: $f : I \to [El(A), El(B)]$ and $g, h : I \to [El(B), El(A)]$ in $[\Gamma, \mathcal{V}]$,*
- *sections: $p : \Gamma \to \Gamma.(M(gf))^*(M(id_A))^* Id_{(M[El(A),El(A)])}$*
- *and $q : \Gamma \to \Gamma.(M(fg))^*(M(id_B))^* Id_{(M[El(B),El(B)])}$*

*There is a natural isomorphism $El(A) \cong El(B)$.*

**Proof.** The section $p$ selects for every $\gamma \in \Gamma$ an isomorphism between $\sigma(gf)$ and $\sigma(id_A)$ of $M[El(A), El(A)]$. But since $M$ factors through sets, $M[El(A), El(A)]$ is a discrete groupoid, so $\sigma(gf)$ and $\sigma(id_A)$ must be identical, and the same is true for $\sigma(fg)$ and $\sigma(id_B)$. Transporting back through the isomorphisms $[\Gamma, \mathbf{GPD}](1, M[El(A), El(A)]) \cong [\Gamma, \mathcal{V}](I, [El(A), El(A)]) \cong [\Gamma, \mathcal{V}](El(A), El(A))$ we find $gf = 1_{El(A)}$, and similarly $fg = 1_{El(B)}$. ◀

By the lemma above, equivalences $El(A) \cong El(B)$ are in one-to-one correspondence with identities $p : A =_L B$, demonstrating that the linear univalence axiom holds in the diagrams model as long as $M$ factors through sets.

### 4.3.1 Examples

To summarize, these are the conditions imposed on $\mathcal{V}$ in order for the diagram model to support all of the type formers of our theory:

- A bicomplete symmetric monoidal closed category $\mathcal{V}$
- An adjunction $L \dashv M$ between $\mathcal{V}$ and **Sets**, such that $L\{*\}$ is isomorphic to the unit of the monoidal structure of $\mathcal{V}$.

Some concrete choices for $\mathcal{V}$ that fulfill these conditions are:

- The category **AbGroups** of abelian groups with the monoidal structure given by the tensor product of abelian groups. Here $L \dashv M$ arises from the free functor on abelian groups.
- More generally, for any commutative ring $R$, the category $R$-**Mod** of modules over $R$ with the free functor/forgetful functor adjunction
- The category $\mathbf{CGTop}_*$, of pointed compactly generated topological spaces, with the smash product as monoidal structure. The functor $M$ is here the forgetful functor which both forgets the base point and the topology, which has a left adjoint given by the discrete topology, and then taking the coproduct with the point to create a pointed space. The unit of $\mathbf{CGTop}_*$ is the two point discrete set $S^0$, which is precisely the image of the point in the adjunction above.

## 4.4 Discussion

Although the diagram model supports the univalence axiom, we are forced to truncate any higher dimensional structure by factoring $M$ through **Set**, just as we can support it in the groupoid model for a universe only containing discrete groupoids. From the perspective of homotopy type theory, we may think of the set-indexed families model as a 0-dimensional model of linear dependent type theory and the diagram model as a 1-dimensional one. We conclude the paper by sketching what a 2-dimensional model might look:

As outlined in [11], there is a symmetric monoidal structure on **SMCat**, the category of small symmetric monoidal categories, symmetric monoidal functors and monoidal natural transformations. [6]

▶ **Definition 22.** Let the **2-dimensional model of LDTT** be given by the diagrams model where $\mathcal{V}$ is the 2-category of small symmetric monoidal categories, symmetric monoidal functors and monoidal natural transformations:

$$\begin{tikzcd}
\mathbf{Diag}(\mathbf{SMCat}) \arrow[dr, "cod"] & \mathbf{Diag}(\mathbf{Gpd}) \arrow[r, "\pi"] \arrow[d, "cod"] & \mathbf{Gpd}^{\to} \arrow[dl, "dom"] \\
& \mathbf{Gpd} &
\end{tikzcd}$$

For two symmetric monoidal categories $\mathcal{A}$ and $\mathcal{B}$, the (monoidal) functor category $[\mathcal{A}, \mathcal{B}]$ between them carries a natural monoidal structure [11], and gives **SMCat** a monoidal closed structure. Since **SMCat** is complete, with limits inherited from **Cat** equipped with a pointwise monoidal structure, we have support for $\sqcap$ and &, and theorem 15 gives us that this model supports $\multimap$ type formers. [7] There is a natural candidate for the adjunction $L \dashv M$, based on the composite:

$$\begin{tikzcd}
\mathbf{SMCat} \arrow[r, bend right, "U"'] \arrow[r, phantom, "\perp"] & \mathbf{Cat} \arrow[l, bend right, "F"'] \arrow[r, bend right, "\text{core}"'] \arrow[r, phantom, "\perp"] & \mathbf{Gpd} \arrow[l, bend right, "U"']
\end{tikzcd}$$

through which one should be able to construct a univalent universe containing nontrivial 1-dimensional linear types.

Eventually, one would like to go all the way up and construct a $\infty$-dimensional formulation of linear dependent type theory. It has been speculated that models of a higher dimensional linear dependent type theory can be expressed through stable homotopy type theory [12], although it is unclear what the syntax for such a theory looks like.

## References

1 P. N. Benton. A mixed linear and non-linear logic: proofs, terms and models (extended abstract). In *Computer science logic (Kazimierz, 1994)*, volume 933 of *Lecture Notes in Comput. Sci.*, pages 121–135. Springer, Berlin, 1995. URL: https://doi.org/10.1007/BFb0022251.

2 A. D. Elmendorf and M. A. Mandell. Permutative categories, multicategories and algebraic $K$-theory. *Algebr. Geom. Topol.*, 9(4):2391–2441, 2009. URL: https://doi.org/10.2140/agt.2009.9.2391.

3 Martin Hofmann and Thomas Streicher. The groupoid interpretation of type theory. In *Twenty-five years of constructive type theory (Venice, 1995)*, volume 36 of *Oxford Logic Guides*, pages 83–111. Oxford Univ. Press, New York, 1998.

4 Bart Jacobs. Comprehension categories and the semantics of type dependency. *Theoret. Comput. Sci.*, 107(2):169–207, 1993. URL: https://doi.org/10.1016/0304-3975(93)90169-T.

[6] Technically, the structure on **SMCat** is not quite symmetric monoidal, as the associators, unitors and symmetry functors are only invertible up to higher homotopy. However, if one applies these homotopies whenever necessary, one does get a model of linear dependent type theory.

[7] Note, however, that we do not have all coproducts in **SMCat**. Therefore, we cannot support $\oplus$ or $\sqsubset$. An alternative to be explored is the category **Mult**, of multicategories, which is a symmetric monoidal closed, complete and co-complete [2]

5 Neelakantan R Krishnaswami, Pierre Pradic, and Nick Benton. Integrating linear and dependent types. In *ACM SIGPLAN Notices*, volume 50, pages 17–30. ACM, January 2015. URL: `http://www.cs.bham.ac.uk/~krishnan/dlnl-paper.pdf`.

6 Daniel R. Licata, Michael Shulman, and Mitchell Riley. A Fibrational Framework for Substructural and Modal Logics. In Dale Miller, editor, *2nd International Conference on Formal Structures for Computation and Deduction (FSCD 2017)*, volume 84 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 25:1–25:22, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: `http://drops.dagstuhl.de/opus/volltexte/2017/7740`, `doi:10.4230/LIPIcs.FSCD.2017.25`.

7 Martin Lundfall. Models of linear dependent type theory. *Stockholm University*, 2017. URL: `http://kurser.math.su.se/pluginfile.php/16103/mod_folder/content/0/2017/2017_47_report.pdf`.

8 Conor McBride. *I got plenty o' nuttin'*, pages 207–233. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer, 3 2016. `doi:10.1007/978-3-319-30936-1_12`.

9 Paul-André Melliès. Categorical semantics of linear logic. In *Interactive models of computation and program behavior*, volume 27 of *Panor. Synthèses*, pages 1–196. Soc. Math. France, Paris, 2009.

10 The Univalent Foundations Program. *Homotopy type theory—univalent foundations of mathematics*. The Univalent Foundations Program, Princeton, NJ; Institute for Advanced Study (IAS), Princeton, NJ, 2013.

11 Vincent Schmitt. Tensor product for symmetric monoidal categories. preprint, 2007. `arXiv:0711.0324`.

12 Urs Schreiber. Quantization via linear homotopy types. preprint, 2014. `arXiv:1402.7041`.

13 Matthijs Vákár. A categorical semantics for linear logical frameworks. In *Foundations of software science and computation structures*, volume 9034 of *Lecture Notes in Comput. Sci.*, pages 102–116. Springer, Heidelberg, 2015. URL: `https://doi.org/10.1007/978-3-662-46678-0_7`.

14 Marek Zawadowski. Lax monoidal fibrations. In *Models, logics, and higher-dimensional categories*, volume 53 of *CRM Proc. Lecture Notes*, pages 341–426. Amer. Math. Soc., Providence, RI, 2011.