# Internal Models of Linear Type Theories

**Sam L. Speight**
School of Computer Science, University of Birmingham, UK
**Niels van der Weide**
iCIS, Radboud University Nijmegen, The Netherlands

## 1 Introduction

The advancement of linear logic has witnessed its core ideas being woven into various forms of type theory, such as System F [2] and dependent type theory [13, 21]. To provide general semantics for these systems, categorical structures have been proposed: linear hyperdoctrines in the case of System F [2] and linear comprehension categories in the case of linear dependent type theory [15, 20]. These notions bear a fundamental similarity in that they are variations on linear-non-linear (LNL) adjunctions [3].

In this work, we make these similarities precise. Specifically, we define a **general notion of linear-non-linear adjunction** that unifies categorical semantics of different linear theories. The general notion is a reformulation of the familiar notion internal to 2-categories with finite products. We recover the familiar notion when the ambient 2-category is that of categories, functors and natural transformations. We also show that pseudofunctors between 2-categories preserve linear non-linear adjunctions whenever they preserve finite products. As an application, we study externalization of internal categories, demonstrating how various concrete models arise this way, including realizability models [2, 20] and the discrete families model [21].

**Overview** We start in Section 2 by defining our general notion of LNL adjunction in a 2-category with finite products. We observe how categorical semantics for different linear theories can be obtained by suitable choice of the ambient bicategory. Section 3 shows that LNL adjunctions are preserved by pseudofunctors between 2-categories that preserve finite products. Before concluding, in Section 4, we present a case study. Specifically, we study LNL adjunctions of internal categories, which we call 'internal LNL models'. We discuss how various concrete models in the literature can be recovered as externalizations of internal LNL models.

## 2 Internal linear-non-linear adjunctions

Our first observation is that the notion of LNL adjunction can be formulated internal to 2-categories with (strict) finite products. Recall that an LNL model [3] is given by a symmetric monoidal adjunction as follows.

$$\begin{tikzcd}
\mathcal{L} \arrow[r, bend right, "\mathsf{M}"'] \arrow[r, phantom, "\perp"] & \mathcal{M} \arrow[l, bend right, "\mathsf{L}"']
\end{tikzcd}$$

Here, $\mathcal{L}$ is required to be a symmetric monoidal category and $\mathcal{M}$ is required to be a Cartesian monoidal category. For simplicity, we do not consider closedness of either category.

In order to formulate the notion of LNL-model 2-categorically, we use two established notions from 2-category theory: Cartesian objects [4] and symmetric pseudomonoids [5, Definition 17]. These notions generalise categories with finite products and symmetric monoidal categories, respectively. In the remainder of this section, let $\mathsf{B}$ be a 2-category with strict finite products.

▶ **Definition 2.1.** *A **Cartesian object** [4] is an object $x$ for which the maps $\Delta : x \to x \times x$ and $! : x \to 1$ have right adjoints. We display Cartesian objects as follows.*

$$\begin{tikzcd}
x \times x \arrow[r, shift right=1.5ex, "m"'] \arrow[r, phantom, "\perp"] & x \arrow[l, shift right=1.5ex, "\Delta"'] \arrow[r, shift left=1.5ex, "!"] \arrow[r, phantom, "\perp"] & 1 \arrow[l, shift left=1.5ex, "u"]
\end{tikzcd}$$

▶ **Definition 2.2.** *A **symmetric pseudomonoid** [5, Definition 17] in $\mathsf{B}$ consists of an object $x$ together with 1-cells $u : 1 \to x$ and $m : x \times x \to x$, as well as invertible 2-cells as displayed below.*

$$\begin{tikzcd}
x \arrow[r, "\langle !,\mathsf{id}\rangle"] \arrow[rrd, "\mathsf{id}"'] & 1 \times x \arrow[r, "u \times \mathsf{id}"] & x \times x \arrow[d, "m"] \arrow[ld, Rightarrow, shorten=2ex, "\mathsf{l}"] \\
& & x
\end{tikzcd}
\qquad
\begin{tikzcd}
x \arrow[r, "\langle \mathsf{id},! \rangle"] \arrow[rrd, "m"'] & x \times 1 \arrow[r, "\mathsf{id} \times u"] & x \times x \arrow[d, "m"] \arrow[ld, Rightarrow, shorten=2ex, "\mathsf{r}"] \\
& & x
\end{tikzcd}$$

$$\begin{tikzcd}
(x \times x) \times x \arrow[rr, "\simeq"] \arrow[dd, "m \times \mathsf{id}"'] & & x \times (x \times x) \arrow[d, "\mathsf{id} \times m"] \\
& \arrow[d, Rightarrow, "\mathsf{a}"] & x \times x \arrow[d, "m"] \\
x \times x \arrow[rr, "m"'] & \, & x
\end{tikzcd}
\qquad
\begin{tikzcd}
x \times x \arrow[rr, "\langle \pi_2,\pi_1 \rangle"] \arrow[rd, "m"'] & \arrow[r, phantom, ""] & x \times x \arrow[ld, "m"] \\
& x &
\end{tikzcd}
\quad \overset{\mathsf{s}}{\Rightarrow}$$

*Finally, we require these 2-cells to satisfy equations reminiscent of symmetric monoidal categories (see the above reference).*

The 2-cells $\mathsf{l}$ and $\mathsf{r}$ are 'unitality constraints'; likewise, $\mathsf{a}$ is the 'associativity constraint' and $\mathsf{s}$ the 'symmetry constraint'.

Just as every category with finite products induces a symmetric monoidal category, we have the following.

▶ **Proposition 2.3.** *Every Cartesian object induces a symmetric pseudomonoid.*

Finally, we need the notion of symmetric monoidal adjunction for our context. Adjunctions can be formulated in any 2-category—so it is a matter of picking the right one. For this, we use the following fact.

▶ **Proposition 2.4.** *Symmetric pseudomonoids in $\mathsf{B}$ organise into a 2-category $\mathsf{PSm}_{\mathsf{sym}}(\mathsf{B})$.*

Everything is now in place for our main definition.

▶ **Definition 2.5.** *An **internal LNL adjunction** in a 2-category $\mathsf{B}$ with strict finite products consists of a symmetric pseudomonoid $\mathcal{L}$, a Cartesian object $\mathcal{M}$ and functors $\mathsf{L} : \mathcal{M} \to \mathcal{L}$ and $\mathsf{M} : \mathcal{L} \to \mathcal{M}$ together with the data of an adjunction $\mathsf{L} \dashv \mathsf{M}$ in $\mathsf{PSm}_{\mathsf{sym}}(\mathsf{B})$.*

## 2.1 Models of linear theories

We instantiate the notion of LNL adjunction to different 2-categories $\mathsf{B}$ to find that it generalizes several notions of model in the literature.

▶ **Example 2.6.** It is well-known that the 2-category $\mathsf{Cat}$ has finite products. Cartesian objects correspond to categories with finite products and symmetric pseudomonoids correspond to symmetric monoidal categories. Internal LNL adjunctions in $\mathsf{Cat}$ thus correspond to the usual notion of LNL model.

▶ **Example 2.7.** Let $\mathsf{C}$ be a category. Then we have a 2-category $\mathsf{Fib}_{\mathsf{split}}(\mathsf{C})$ of split fibrations $\mathcal{E} \to \mathsf{C}$. This 2-category has finite products, which are given fibrewise. Internal LNL adjunctions in $\mathsf{Fib}_{\mathsf{split}}(\mathsf{C})$ correspond to LNL hyperdoctrines [2].

▶ **Example 2.8.** Given a Cartesian closed category $\mathcal{V}$, we have a 2-category $\mathcal{V}\mathsf{Cat}$ of $\mathcal{V}$-enriched categories, and this 2-category has finite products. Internal LNL adjunctions in $\mathcal{V}\mathsf{Cat}$ generalise the notion of enriched combined linear-non-linear (ECLNL) model [14].

▶ **Example 2.9.** Since the 2-category of (strict) double categories has finite products, we also obtain a double categorical version of LNL adjunctions.

## 3 Preservation of linear-non-linear adjunctions

Our second observation is that LNL adjunctions are preserved by pseudofunctors between 2-categories that preserve finite products. Throughout this section, let $F : \mathsf{B}_1 \to \mathsf{B}_2$ be a pseudofunctor that preserves finite products. We first note that every pseudofunctor preserves adjunctions.

▶ **Proposition 3.1.** *Every pseudofunctor preserves adjunctions.*

To show that $F$ also preserves LNL adjunctions, we first show that $F$ lifts to the 2-category of symmetric pseudomonoids. This guarantees that $F$ preserves adjunctions between symmetric pseudomonoids.

▶ **Proposition 3.2.** *$F$ lifts to a pseudofunctor* $\mathsf{PSm}_{\mathsf{sym}}(F) : \mathsf{PSm}_{\mathsf{sym}}(\mathsf{B}_1) \to \mathsf{PSm}_{\mathsf{sym}}(\mathsf{B}_2)$*.*

Since one of the objects in an LNL adjunction is Cartesian, we also need that $F$ preserves Cartesian objects.

▶ **Proposition 3.3.** *$F$ maps Cartesian objects to Cartesian objects.*

All in all, we conclude that $F$ preserves LNL adjunctions.

▶ **Proposition 3.4.** *Every pseudofunctor $F$ that preserves finite products also preserves LNL adjunctions.*

## 4 Case study: externalization

As an application of Proposition 3.4, we focus on **externalization of internal categories** [11]. In the remainder, we fix a category $\mathsf{C}$ with finite limits. Recall that an internal graph in $\mathsf{C}$ is given by morphisms $s, t : m \to o$. An internal category [9] is an internal graph together with morphisms $i : o \to m$ and $c : m\,{}_t{\times}_s\, m \to m$ that satisfy the axioms of a category expressed diagramatically (see [11, Definition 7.1.1], for instance). We denote internal categories by their underlying graph. Note that we have a 2-category of internal categories, and that this 2-category has finite products.

▶ **Example 4.1.** Let $\mathsf{C}$ be a category with finite limits. Then we have a 2-category $\mathsf{Cat}(\mathsf{C})$ of categories internal to $\mathsf{C}$, and $\mathsf{Cat}(\mathsf{C})$ has finite products. We call LNL adjunctions in $\mathsf{Cat}(\mathsf{C})$ **internal LNL models**.

Every internal category $s, t : m \to o$ in $\mathsf{C}$ induces a split fibration over $\mathsf{C}$, which we call its **externalization**. Objects over $x$ in the externalization are given by morphisms $x \to o$. Externalization gives rise to a pseudofunctor $\mathsf{Ext} : \mathsf{Cat}(\mathsf{C}) \to \mathsf{Fib}_{\mathsf{split}}(\mathsf{C})$, which preserves finite products. By Proposition 3.4 we can directly conclude the following proposition.

▶ **Proposition 4.2.** *The externalization of an internal LNL model is an LNL hyperdoctrine over the ambient category.*

We can also use externalization to construct **linear comprehension categories** [15, 20]. Briefly, a linear comprehension category is a fibred symmetric monoidal adjunction between a comprehension category and a symmetric monoidal fibration.

▶ **Proposition 4.3.** *Let* $\mathsf{C}$ *be a locally Cartesian closed category. Every LNL model of internal categories where* $\mathcal{M}$ *is a full internal subcategory of* $\mathsf{C}$ *induces a linear comprehension category with category of contexts* $\mathsf{C}$.

Propositions 4.2 and 4.3 allow us to recover various models from the literature. Two of our examples are based on realizability, and use the notion of **linear combinatory algebra** (LCA) [1]. For the exact definition, we refer the reader to the literature. We recall that every LCA $A$ induces a (cartesian) combinatory algebra $A_!$. Then there are categories $\mathsf{Asm}(A)$ and $\mathsf{Asm}(A_!)$ of assemblies, as well as categories $\mathsf{Mod}(A)$ and $\mathsf{Mod}(A_!)$ of modest sets, and categories $\mathsf{PER}(A)$ and $\mathsf{PER}(A_!)$ of partial equivalence relations (symmetric and transitive relations).

▶ **Example 4.4.** Assume that we have a Grothendieck universe $\mathcal{U}$. With this, we have a category $\mathsf{Asm}(A_!)$ of large assemblies, inside which we have an internal category $\mathsf{asm}(A_!)$ of small assemblies [16] (i.e., assemblies whose carrier lies in $\mathcal{U}$). In fact, we have an internal LNL model.

$$\begin{tikzcd}
\mathsf{asm}(A) \arrow[r, bend right, "\mathsf{M}"'] \arrow[r, phantom, "\perp"] & \mathsf{asm}(A_!) \arrow[l, bend right, "\mathsf{L}"']
\end{tikzcd}$$

The externalization of this internal LNL model is the linear comprehension category described by Speight and Van der Weide [20, Example 4.11]. The comprehension category part of this linear comprehension category is that of uniform families of cartesian assemblies; the symmetric monoidal fibration is that of uniform families of linear assemblies.

▶ **Example 4.5.** Let $A$ be an LCA. We have the following internal LNL model in $\mathsf{Asm}(A_!)$ (where the PERs live in the same universe as the assemblies).

$$\begin{tikzcd}
\mathsf{PER}(A) \arrow[r, bend right, "\mathsf{M}"'] \arrow[r, phantom, "\perp"] & \mathsf{PER}(A_!) \arrow[l, bend right, "\mathsf{L}"']
\end{tikzcd}$$

Its externalization is the PER linear hyperdoctrine due to Abramsky and Lenisa [2, Theorem 2.2]. But it is known that $\mathsf{PER}(A_!)$ is a full internal subcategory of $\mathsf{Asm}(A_!)$ (this works for any PCA). So by Proposition 4.3, we furthermore obtain a linear comprehension category. Combining Example 4.4 and Example 4.5, there is even more to say.

▶ Remark 4.6. The externalizations of $\mathsf{PER}(A_!)$ and $\mathsf{PER}(A)$ are full subfibrations of, respectively, uniform families of cartesian assemblies and uniform families of linear assemblies. Both of these subfibrations are complete, as shown by Speight and Van der Weide [20, Example 4.18]. A result of Hyland, Robinson and Rosolini states that an internal category is internally complete if and only if its externalization is complete as a fibration [10, Proposition 4.4]. Hence we may conclude that *both* $\mathsf{PER}(A)$ and $\mathsf{PER}(A_!)$ are complete small categories in $\mathsf{Asm}(A_!)$. This is why there is an impredicative universe (given by the set of PERs on $A$) in the model of Speight and Van der Weide.

▶ **Example 4.7.** For this example, we assume that we have two nested Grothendieck universes $\mathcal{U}_1 \in \mathcal{U}_2$. Let $\mathcal{V}$ be a symmetric monoidal category such that every coproduct of its unit $\mathbb{1}$ exists. We have the following internal LNL model in the category $\mathbf{Set}_{\mathcal{U}_2}$.

$$\begin{tikzcd}
\mathcal{V} \arrow[r, bend right, "\mathsf{M}"'] \arrow[r, phantom, "\perp"] & \mathbf{Set}_{\mathcal{U}_1} \arrow[l, bend right, "\mathsf{L}"']
\end{tikzcd}$$

Here $\mathsf{M}$ maps objects $x$ to the set $\mathbb{1} \to x$ and $\mathsf{L}$ maps sets $X$ to the coproduct $\amalg_{x \in X} \mathbb{1}$. The externalization of this model is the discrete families model by Vakar [21]. Specifically, the so-obtained LNL adjunction is internal to the category of sets in $\mathcal{U}_2$, and $\mathcal{V}$ is required to live in $\mathcal{U}_1$.

If we have enough Grothendieck universes, then we can phrase various standard examples of LNL models as internal LNL models, such as the relational model, Scott model, and the lifting model on directed complete partial orders. Our framework provides a general way of constructing models of complex linear theories from simpler models.

## 5 Conclusion

We defined a general notion of LNL adjunction, which unifies models of various linear type theories. We showed that LNL adjunctions are preserved by pseudofunctors that preserve finite products. This gives a method for constructing models of complex linear type theories from simpler models, reducing the workload in individual cases. To substantiate this claim, we showed how various concrete models of linear type theories can be obtained via externalization.

As future work, we plan to study type formers in the externalization. The notions of linear hyperdoctrine and linear comprehension category model the structural core of the relevant linear type theory. Type formers require additional structure. Specifically, our goal is to identify conditions on internal categories that allow us to conclude that their externalization supports certain standard type formers. The result of Hyland, Robinson and Rosolini discussed in Remark 4.6 is a start: completeness of an internal category yields dependent products and equalizer types in the externalization.

We also plan to study further applications. For instance, both the Yoneda embedding and change of base for enriched categories preserve finite products, and there might also be applications within game theory [17].

Proposition 3.4 also has other instantiations of interest. For instance, we have a pseudofunctor sending a double category to its horizontal 2-category. Since this pseudofunctor preserves finite products, we see that every double-categorical LNL adjunction (Example 2.9) gives rise to a 2-categorical model of linear logic. However, we cannot apply Proposition 3.4 to obtain bicategorical models [6, 8, 18, 12, 7], since there is no bicategory of bicategories. The notion of internal LNL adjunction might still be of interest for this purpose. This is because we get a notion of LNL models in the 2-category of pseudo double categories, and because monoidal pseudo double categories can be used to conveniently construct monoidal bicategories [19]. It remains to verify that LNL adjunctions of pseudo double categories give rise to bicategorical models of linear logic.

## References

**1** Samson Abramsky, Esfandiar Haghverdi, and Philip J. Scott. Geometry of interaction and linear combinatory algebras. *Mathematical Structures in Computer Science*, 12(5):625–665, 2002. `doi:10.1017/S0960129502003730`.

**2** Samson Abramsky and Marina Lenisa. Linear realizability and full completeness for typed lambda-calculi. *Annals of Pure and Applied Logic*, 134(2-3):122–168, 2005. `doi:10.1016/j.apal.2004.08.003`.

**3** Nick Benton. A mixed linear and non-linear logic: Proofs, terms and models. In Leszek Pacholski and Jerzy Tiuryn, editors, *Proceedings of the 8th International Workshop on Computer Science Logic (CSL'94)*, volume 933 of *Lecture Notes in Computer Science*, pages 121–135. Springer, 1995. `doi:10.1007/3-540-58277-0_6`.

**4** A. Carboni, G. M. Kelly, and R. J. Wood. A 2-categorical approach to change of base and geometric morphisms. I. In *Cahiers Topologie Géom. Différentielle Catég.*, volume 32, pages 47–95. 1991.

**5** Brian Day and Ross Street. Monoidal bicategories and Hopf algebroids. *Advances in Mathematics*, 129(1):99–157, 1997. `doi:10.1006/aima.1997.1649`.

**6** M. Fiore, N. Gambino, M. Hyland, and G. Winskel. The cartesian closed bicategory of generalised species of structures. *Journal of the London Mathematical Society*, 77(1):203–220, 2008. URL: `https://londmathsoc.onlinelibrary.wiley.com/doi/abs/10.1112/jlms/jdm096`, `arXiv:https://londmathsoc.onlinelibrary.wiley.com/doi/pdf/10.1112/jlms/jdm096`, `doi:10.1112/jlms/jdm096`.

**7** Marcelo Fiore, Nicola Gambino, and Martin Hyland. Monoidal bicategories, differential linear logic, and analytic functors. *arXiv preprint arXiv:2405.05774*, 2024. `arXiv:2405.05774`.

**8** Zeinab Galal. A Profunctorial Scott Semantics. In Zena M. Ariola, editor, *5th International Conference on Formal Structures for Computation and Deduction, FSCD 2020, June 29-July 6, 2020, Paris, France (Virtual Conference)*, volume 167 of *LIPIcs*, pages 16:1–16:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPICS.FSCD.2020.16`.

**9** Alexander Grothendieck. Techniques de construction et théorèmes d'existence en géométrie algébrique iii : préschémas quotients. In *Séminaire Bourbaki : années 1960/61, exposés 205–222*, number 6 in Séminaire Bourbaki, 1961. Talk no. 212, 20 pages. URL: `https://www.numdam.org/item/SB_1960-1961__6__99_0/`.

**10** J. M. E. Hyland, E. P. Robinson, and G. Rosolini. The discrete objects in the effective topos. *Proceedings of the London Mathematical Society*, s3-60(1):1–36, 01 1990. `arXiv:https://academic.oup.com/plms/article-pdf/s3-60/1/1/4281575/s3-60-1-1.pdf`, `doi:10.1112/plms/s3-60.1.1`.

**11** Bart Jacobs. *Categorical Logic and Type Theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 2001.

**12** Axel Kerinec, Giulio Manzonetto, and Federico Olimpieri. Why are proofs relevant in proof-relevant models? *Proceedings of the ACM on Programming Languages*, 7(POPL), 2023. `doi:10.1145/3571201`.

**13** Neelakantan R. Krishnaswami, Cécilia Pradic, and Nick Benton. Integrating linear and dependent types. In Sriram K. Rajamani and David Walker, editors, *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2015, Mumbai, India, January 15-17, 2015*, pages 17–30. ACM, 2015. `doi:10.1145/2676726.2676969`.

**14** Bert Lindenhovius, Michael W. Mislove, and Vladimir Zamdzhiev. Enriching a linear/non-linear lambda calculus: A programming language for string diagrams. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 659–668. ACM, 2018. `doi:10.1145/3209108.3209196`.

**15** Martin Lundfall. Models of linear dependent type theory. Master's thesis, Stockholm University, 2017.

**16** Zhaohui Luo. *An Extended Calculus of Constructions*. PhD thesis, University of Edinburgh, 1990.

**17** 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.

**18** Federico Olimpieri. Intersection type distributors. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2021)*, pages 1–15, 2021. `doi:10.1109/LICS52264.2021.9470617`.

**19** Michael A. Shulman. Constructing symmetric monoidal bicategories, 2010. URL: `https://arxiv.org/abs/1004.0993`, `arXiv:1004.0993`.

**20** Sam Speight and Niels van der Weide. Impredicativity in linear dependent type theory. *arXiv preprint arXiv:2602.08846*, 2026.

**21** Matthijs Vákár. A Categorical Semantics for Linear Logical Frameworks. In Andrew M. Pitts, editor, *Foundations of Software Science and Computation Structures - 18th International Conference, FoSSaCS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*, volume 9034 of *Lecture Notes in Computer Science*, pages 102–116. Springer, 2015. `doi:10.1007/978-3-662-46678-0_7`.