Relating Categorical Semantics for Intuitionistic Linear Logic*

Maria Emilia Maietti* Paola Maneggia† Valeria de Paiva‡ Eike Ritter†

*Dipartimento di Matematica Pura ed Applicata, Università di Padova, Italy.
†School of Computer Science, University of Birmingham, UK.
‡Palo Alto Research Center, USA.
maietti@math.unipd.it, {P.Maneggia, E.Ritter}@cs.bham.ac.uk, paiva@parc.com

Abstract

There are several kinds of *linear* typed calculus in the literature, some with their associated notion of *categorical* model. Our aim in this paper is to systematise the relationship between three of these linear typed calculi and their models. We point out that mere soundness and completeness of a linear typed calculus with respect to a class of categorical models are not sufficient to identify the most appropriate class uniquely. We recommend instead to use the notion of *internal language* when relating a typed calculus to a class of models. After clarifying the internal languages of the categories of models in the literature we relate these models via reflections and coreflections.

Key words: intuitionistic linear logic, typed lambda calculus, symmetric monoidal closed categories, symmetric monoidal adjunctions.
AMS classifications: 03G30 03B15 18C50 03B20

1 Introduction

Over the last decade, the proof theory of intuitionistic linear logic and its categorical models have been studied by many researchers. However, the precise relationship between the different notions of model (and their status as models for different calculi) have not been fully worked out, and those results which have been proved are scattered throughout the literature. This paper attempts to rectify this situation by a careful re-examination and comparison of three of these different logics/calculi and their corresponding notions of categorical model. We consider the following typed calculi: Intuitionistic Linear Logic (called ILL [8]), which is the first linear typed calculus for Girard and Lafont's sequent style formulation of Intuitionistic Linear Logic [14, 20], Linear-non-Linear Logic (called LNL [5]) whose models are a considerable simplification of the models of ILL, and Dual Intuitionistic and Linear Logic (called DILL [2]) which fits better the implementational issues that we addressed elsewhere [28]. Of the sound and complete models given in the literature for these typed calculi we consider the following: *linear* categories for ILL ([7, 9]) and *symmetric monoidal* adjunctions, of slightly different kinds, for DILL and LNL (see [2], [5]).

Our leading idea is to relate typed calculi and their categorical models via the notion of internal language. Soundness and completeness theorems are not generally sufficient to identify the most appropriate class of categorical models for a typed calculus, unless the typed calculus

*Research partially supported by EPSRC, grant GR/L28296, "The eXplicit Substitutions Linear Abstract Machine". We would like to thank our two anonymous referees for their help in clarifying our exposition and Mathias Kegelmann for his careful reading and comments on a preliminary version of this paper.

provides an internal language of the models considered. Hence, we propose that a class of sound and complete models of a typed calculus should be called "the categorical semantics" of such a calculus if these models have theories of the given calculus as their internal language.

Our main example of why this more restrictive criterion needs to be used is provided by the typed calculus ILL. It is essentially equivalent to DILL and is sound and complete for both linear categories ([7, 9]) and for a certain class of symmetric monoidal adjunctions [2]. In the literature [5, 26] it is said that these two notions of categorical model for ILL are equivalent. But, after organising each class of models into categories, we show that the two categories of complete models (with morphisms preserving their structure up to isomorphisms) are linked by a reflection and not by a categorical equivalence. Moreover, we prove that only linear categories have ILL-theories as their internal language, while a fragment of LNL provides an internal language for the symmetric monoidal adjunctions which are sound and complete for ILL [2].

This means that to establish a precise notion of equivalence between models, it is essential to consider also model morphisms. As notion of model morphism we take that of a functor preserving the model structure up to isomorphisms, or strictly. Strict preservation corresponds to translation between the internal languages of the models considered. The reason we also consider categories of models with functors preserving structure up to isomorphisms is to be able to relate these categories via reflections or coreflections.

The structure of the paper is as follows. In the next section we describe the chosen linear typed calculi in appropriate detail. In Section 3 we organise all the models of ILL, DILL and LNL into categories and we check whether they satisfy the internal language criterion. Then, we relate the categories of models considered via reflections or coreflections. In particular, after relating linear categories with symmetric monoidal adjunction models via a reflection, we prove that the category of models of DILL with additive conjunction is coreflective in the category of models of LNL with additive conjunction in the linear part.

The systematization given here seems necessary if one intends to extend the work on these calculi to higher order ones, see for example [25]. This systematization makes clear which kind of model should be generalised and why.

2 Linear Typed Calculi

Intuitionistic Linear Logic, introduced by Girard and Lafont [14, 20], has been investigated for its potential applications to functional programming. Several linear typed calculi, which could have been called "the correct" typed linear λ-calculus, have been proposed for these applications.

All these typed calculi have the (non-controversial) !-free fragment of intuitionistic linear logic, called rudimentary linear logic or RLL, in common. To add the connective ! we recall three alternative versions of linear λ-calculus. The linear λ-calculus of Bierman et al. [8] ILL, the linear-non-linear λ-calculus of Benton LNL [5] and finally Barber and Plotkin's Dual Intuitionistic and Linear λ-calculus DILL [2, 1]. There are other typed calculi for this fragment, but we concentrate on the ones with a given categorical model as the aim of this paper is to relate such calculi via their categorical semantics by using the criterion of connecting calculi and models via internal language theorems.

To this end we define the notion of theory and translations for each linear typed calculus and we organise them into a category, the category of theories of the considered calculus.

We start to define all these notions in general for a typed system and then we specialise them for each of the typed calculi.

2.1 General presentation of typed calculi

In this section we describe various linear typed calculi that provide proof-term annotation systems for fragments of propositional intuitionistic linear logic. Hence they deal with simple types only.

A typed system with simple types is described by means of four kinds of judgements

$$\vdash A : \mathsf{type} \qquad \vdash A = B : \mathsf{type} \qquad \Sigma \vdash M : A \qquad \Sigma \vdash M = N : A$$

where the first is a judgement saying that something is of sort type in the empty context, (i.e. we consider simple types only), the second that two simple types are equal, the third that a term M in context Σ is of a type A, the forth that two terms are equal. In this paper we consider calculi where Σ is either a simple context or a double context, i.e. it is formed by two kinds of contexts $\Gamma \mid \Delta$, where Γ is the intuitionistic context and Δ the linear one. For simplicity, in this general presentation we assume that Σ is a simple context and that contexts are lists, considered up to permutation and without repetitions, formed according to the following rules

- $\emptyset$ is a context;
- if Σ is a context and $\vdash A : \mathsf{type}$ is derivable then $\Sigma, x : A$ is a context.

To state that the type equality is an equivalence relation the typed system has the rules

$$\frac{\vdash A : \mathsf{type}}{\vdash A = A : \mathsf{type}} \qquad \frac{\vdash A = B : \mathsf{type}}{\vdash B = A : \mathsf{type}} \qquad \frac{\vdash A = B : \mathsf{type} \quad \vdash B = C : \mathsf{type}}{\vdash A = C : \mathsf{type}}$$

and similarly for term equality:

$$\frac{\Sigma \vdash M : A}{\Sigma \vdash M = M : A} \qquad \frac{\Sigma \vdash M = N : A}{\Sigma \vdash N = M : A} \qquad \frac{\Sigma \vdash M = N : A \quad \Sigma \vdash N = P : A}{\Sigma \vdash M = P : A}$$

The typed system has some type constructors: if F is an n-ary type constructor, then the system has the rule

$$\frac{\vdash A_1 : \mathsf{type} \quad \cdots \vdash A_n : \mathsf{type}}{\vdash F(A_1, \ldots, A_n) : \mathsf{type}}$$

To say that any such type constructor F preserves type equality, the system has the rule

$$\frac{\vdash A_1 = A_1' : \mathsf{type} \quad \cdots \vdash A_n = A_n' : \mathsf{type}}{\vdash F(A_1, \ldots, A_n) = F(A_1', \ldots, A_n') : \mathsf{type}}$$

Analogously, the typed system has some term constructors f for building new terms from other terms according to

$$\frac{\Sigma_1 \vdash M_1 : A_1 \quad \ldots \Sigma_n \vdash M_n : A_n}{\Sigma \vdash f(M_1, \ldots, M_n) : B}$$

which also preserves term equality:

$$\frac{\Sigma_1 \vdash M_1 = M_1' : A_1 \quad \ldots \Sigma_n \vdash M_n = M_n' : A_n}{\Sigma \vdash f(M_1, \ldots, M_n) = f(M_1', \ldots, M_n') : B}$$ [1]

[1] As usual, we identify terms which are equal up to the renaming of bound variables. For a precise treatment of this so-called α-conversion see for example [3].

The system has also identity axioms; in the linear case with simple context there is just one:

$$x : A \vdash x : A$$

Finally, the system has *specific* term equality axioms including β-equalities, η-equalities and also commuting conversions.

Usual presentations of typed systems with simple (non-dependent) types usually ignore equality of types as judgements since without extra equations the equality between types is simply syntactic identity. But we do consider equality of types because, when we come to consider theories, we allow additional equations between types[2]. Ordinary presentations of type systems usually present only the specific equalities of the term constructors, namely β-equalities, η-equalities and commuting conversions, by considering the other equations as implicit (see for example [27]).

Now, we give a definition of a theory based on a typed calculus.

Definition 1. Given a typed calculus $\mathcal{C}$, a typed system $\mathcal{T}$ is a *$\mathcal{C}$-theory* if it is an extension of $\mathcal{C}$ with *proper $\mathcal{T}$-axioms*, namely with new ground type symbols A and corresponding new judgements $\vdash A$ type, new type equality judgements $\vdash A = B$, provided that A and B are types, new term symbols M together with new term judgements $\Sigma \vdash M\colon A$ provided that Σ is a context and A a type symbol, and new equality judgements $\Sigma \vdash M = N\colon A$ provided that $\Sigma \vdash M\colon A$ and $\Sigma \vdash N\colon A$ are derivable judgements.

In other words, a $\mathcal{C}$-theory is an extension of $\mathcal{C}$ with new ground types, new term constants and new equalities between types and between terms. Hence, new inference rules of term constructors are not admitted.

Now, we define the notion of $\mathcal{C}$-translation.

Definition 2. Given a typed calculus $\mathcal{C}$, a *$\mathcal{C}$-translation* L between type theories $\mathcal{T}$ and $\mathcal{T}'$ is a function from types and terms derivable in $\mathcal{T}$ — i.e. from type symbols A such that $\vdash A : \mathsf{type}$ is derivable in $\mathcal{T}$ and from terms M such that there exists a judgement $\Sigma \vdash M : A$ derivable in $\mathcal{T}$ — to type and term of $\mathcal{T}'$ *by preserving type and term judgements*, that is a type A for which $\vdash A : \mathsf{type}$ is derivable in $\mathcal{T}$ is mapped to a type $L(A)$ such that

$$\vdash L(A) : \mathsf{type}$$

is derivable in $\mathcal{T}'$, and a typed term M such that $\Sigma \vdash M : A$ is derivable in $\mathcal{T}$ is sent to a typed term $L(M)$ such that

$$L(\Sigma) \vdash L(M) : L(A)$$

is derivable in $\mathcal{T}'$, where $L(\Sigma) \equiv x_1 : L(A_1), \ldots, x_n : L(A_n)$ if $\Sigma \equiv x_1 : A_1, \ldots, x_n : A_n$, and such that variables are sent to variables, i.e.

$$L(x) = x.$$

Moreover, we also require that L *preserves $\mathcal{C}$-type constructors*, i.e. for any type constructor F of $\mathcal{C}$ and types $A_1, \ldots, A_n$

$$L(F(A_1, \ldots, A_n)) = F(L(A_1), \ldots, L(A_n))$$

[2]We follow here Martin-Löf's presentation of a typed system and also Quine's slogan "*No entity without identity.*"

and that L *preserves $\mathcal{C}$-term constructors*, i.e. for any term constructor f of $\mathcal{C}$ and terms $M_1,\ldots,M_n$

$$L(f(M_1,\ldots,M_n)) \;=\; f(L(M_1),\ldots,L(M_n))$$

Finally, we require that L *preserves the type equality judgements of* $\mathcal{T}$, i.e. for any judgement $\vdash A = B : \mathsf{type}$ derivable in $\mathcal{T}$

$$\vdash L(A) = L(B) : \mathsf{type}$$

is derivable in in $\mathcal{T}'$, and that L *preserves the term equality judgements of* $\mathcal{T}$, i.e. for any judgement $\Sigma \vdash M = N : A$ derivable in $\mathcal{T}$

$$L(\Sigma) \vdash L(M) = L(N) : L(A)$$

is derivable in $\mathcal{T}'$.

Remark 3. Note that we can extend the definition of $\mathcal{C}$-translation L to all judgements of $\mathcal{T}$ by putting

$$\begin{array}{rcl} L(\vdash A : \mathsf{type}) & \equiv & \vdash L(A) : \mathsf{type} \\ L(\Sigma \vdash M : A) & \equiv & L(\Sigma) \vdash L(M) : L(A) \\ L(\vdash A = B : \mathsf{type}) & \equiv & \vdash L(A) = L(B) : \mathsf{type} \\ L(\Sigma \vdash M = N : A) & \equiv & L(\Sigma) \vdash L(M) = L(N) : L(A) \end{array}$$

Moreover, note that a $\mathcal{C}$-translation is determined by its definition on the new types and terms added to the base calculus $\mathcal{C}$.

For any typed calculus $\mathcal{C}$, we introduce the category of $\mathcal{C}$-theories and $\mathcal{C}$-translations. Composition of two $\mathcal{C}$-translations is simply composition of functions which is again a $\mathcal{C}$-translation:

Definition 4. For any typed calculus $\mathcal{C}$, we call $\mathsf{Th}(\mathcal{C})$ the category having $\mathcal{C}$-theories as objects and $\mathcal{C}$-translations as morphisms.

In the following, when describing a linear typed system, we just mention the specific type constructors, typed term constructors and specific term equalities.

2.2 Rudimentary Linear Logic RLL

We describe the modality-free (or exponential-free in Girard's terminology) fragment of intuitionistic linear logic, i.e. we consider only tensor products, their unit and linear function spaces. A context Δ used in a judgement consists of a list of variable declarations of the form $a\colon A$ up to permutation and without repetitions according to the rules in subsection 2.1.

The RLL type constructor rules are the following:

$$\vdash\ I : \mathsf{type} \qquad \frac{\vdash A_1 : \mathsf{type} \quad \vdash A_2 : \mathsf{type}}{\vdash A_1 \otimes A_2 : \mathsf{type}} \qquad \frac{\vdash A_1 : \mathsf{type} \quad \vdash A_2 : \mathsf{type}}{\vdash A_1 \multimap A_2 : \mathsf{type}}$$

where $A \multimap A$ denotes the linear function space, $A \otimes A$ tensor products and I the unit for the tensor.

The RLL typed term constructor rules are given by:

$$a : A \vdash a : A$$

$$\frac{\Delta, a : A \vdash M : B}{\Delta \vdash \lambda a^A.M : A \multimap B} \qquad \frac{\Delta_1 \vdash M : A \multimap B \quad \Delta_2 \vdash N : A}{\Delta_1, \Delta_2 \vdash M(N) : B}$$

$$\frac{\Delta_1 \vdash M : A \quad \Delta_2 \vdash N : B}{\Delta_1, \Delta_2 \vdash M \otimes N : A \otimes B} \qquad \frac{\Delta_1 \vdash M : A \otimes B \quad \Delta_2, a : A, b : B \vdash N : C}{\Delta_1, \Delta_2 \vdash \mathsf{let}\ M\ \mathsf{be}\ a \otimes b\ \mathsf{in}\ N : C}$$

$$\_ \vdash \bullet : I \qquad \frac{\Delta_1 \vdash M : I \quad \Delta_2 \vdash N : C}{\Delta_1, \Delta_2 \vdash \mathsf{let}\ M\ \mathsf{be}\ \bullet\ \mathsf{in}\ N : C}$$

Each RLL type constructor gives rise to both β (on the left) and η (on the right) equality rules as follows:

$$\frac{\Delta_1, a\colon A \vdash M\colon B \quad \Delta_2 \vdash N\colon A}{\Delta_1, \Delta_2 \vdash (\lambda a^A.M)(N) = M[N/a] : B} \qquad \frac{\Delta \vdash M\colon A \multimap B}{\Delta \vdash \lambda a^A.M(a) = M\colon A \multimap B}$$

$$\frac{\Delta_1, a\colon A, b\colon B \vdash R\colon C \quad \Delta_2 \vdash M\colon A \quad \Delta_3 \vdash N\colon B}{\Delta_1, \Delta_2, \Delta_3 \vdash \mathsf{let}\ M \otimes N\ \mathsf{be}\ a \otimes b\ \mathsf{in}\ R = R[M/a, N/b]\colon C} \qquad \frac{\Delta_1, c\colon A \otimes B \vdash M\colon C \quad \Delta_2 \vdash N\colon A \otimes B}{\Delta_1, \Delta_2 \vdash \mathsf{let}\ N\ \mathsf{be}\ a \otimes b\ \mathsf{in}\ M[a \otimes b/c] = M[N/c]\colon C}$$

$$\frac{\Delta \vdash M\colon A}{\Delta \vdash \mathsf{let}\ \bullet\ \mathsf{be}\ \bullet\ \mathsf{in}\ M = M\colon A} \qquad \frac{\Delta_1, a\colon I \vdash M\colon A \quad \Delta_2 \vdash N\colon I}{\Delta_1, \Delta_2 \vdash \mathsf{let}\ N\ \mathsf{be}\ \bullet\ \mathsf{in}\ M[\bullet/a] = M[N/a]\colon A}$$

Note that our presentation of the equational system, using generalised η rules [12], differs from the standard one (e.g. [9]). The standard system has two η rules, namely $\mathsf{let}\ M\ \mathsf{be}\ x \otimes y\ \mathsf{in}\ x \otimes y = M$ and $\mathsf{let}\ M\ \mathsf{be}\ \bullet\ \mathsf{in}\ \bullet = M$ plus commuting conversions instead of the generalised η rules above[3].

Definition 5. An RLL-theory is a calculus obtained by extending RLL with new ground types and corresponding type judgements, new type equality judgements, new term symbols and corresponding term judgements, and new term equality judgements as in Definition 1.

Definition 6. Given two RLL-theories T_1 and T_2, an RLL-translation L is a translation from types and terms of T_1 to types and terms of T_2 preserving type and term judgements in the sense of Definition 2, preserving RLL type constructors, i.e

$$L(I) = I \qquad L(A \otimes B) = L(A) \otimes L(B) \qquad L(A \multimap B) = L(A) \multimap L(B)$$

preserving RLL term constructors, i.e

$$\begin{aligned} L(\bullet) &= \bullet \\ L(\lambda a^A.M) &= \lambda a^{L(A)}.L(M) \\ L(M \otimes N) &= L(M) \otimes L(N) \\ L(\mathsf{let}\ M\ \mathsf{be}\ a \otimes b\ \mathsf{in}\ N) &= \mathsf{let}\ L(M)\ \mathsf{be}\ a \otimes b\ \mathsf{in}\ L(N) \\ L(\mathsf{let}\ M\ \mathsf{be}\ \bullet\ \mathsf{in}\ N) &= \mathsf{let}\ L(M)\ \mathsf{be}\ \bullet\ \mathsf{in}\ L(N) \end{aligned}$$

and preserving the type and term equality judgements of T_1.

[3] The ensuing typed calculus with corresponding suitable reduction rules is still strongly normalising and confluent [13].

Definition 7. The category $\mathsf{Th}(\mathrm{RLL})$ has RLL-theories as objects and RLL-translations as morphisms.

2.3 Intuitionistic Linear typed calculus ILL

The typed calculus ILL was originally presented by Benton, Bierman, de Paiva and Hyland in [8]. Like for RLL, in ILL a context Δ used in the judgements consists of a list of variable declarations of the form $a\colon A$ up to permutation and without repetitions.

The ILL type constructor rules are those of RLL plus a rule for the modality type constructor $!A$:

$$\frac{\vdash A : \mathsf{type}}{\vdash !A : \mathsf{type}}$$

The ILL typed term constructor rules include those of RLL with the addition of the following rules:

$$\frac{\Delta \vdash M\colon !A}{\Delta \vdash \mathsf{derelict}(M)\colon A} \qquad \frac{\Delta_1 \vdash M\colon !A \quad \Delta_2 \vdash N\colon B}{\Delta_1, \Delta_2 \vdash \mathsf{discard}\ M\ \mathsf{in}\ N\colon B} \qquad \frac{\Delta_1 \vdash M\colon !A \qquad \Delta_2, a\colon !A, b\colon !A \vdash N\colon B}{\Delta_1, \Delta_2 \vdash \mathsf{copy}\ M\ \mathsf{as}\ a, b\ \mathsf{in}\ N\colon B}$$

$$\frac{\Delta_1 \vdash M_1\colon !A_1, \ldots, \Delta_k \vdash M_k\colon !A_k \qquad a_1 : !A_1, \ldots, a_k : !A_k \vdash N : B}{\Delta_1, \Delta_2, \ldots, \Delta_k \vdash \mathsf{promote}\ M_i\ \mathsf{for}\ a_i\ \mathsf{in}\ N\colon !B}$$

This calculus has explicit rules (and terms) for copying and discarding assumptions, following [23]. It also has explicit substitutions in the "promotion" rule, to cope with the lack of substitutivity in previous systems. Since the number of equations is lengthy we refer the reader to Bierman's thesis [9] or [2, 1].

Definition 8. An ILL-theory is a calculus obtained by extending ILL with new ground types and corresponding type judgements, new type equality judgements, new term symbols and corresponding term judgements, and new term equality judgements as in Definition 1.

Definition 9. Given two ILL-theories T_1 and T_2, an ILL-translation L is a translation from types and terms of T_1 to types and terms of T_2 preserving type and term judgements in the sense of Definition 2, preserving ILL type constructors, i.e

$$L(I) = I \qquad L(A \otimes B) = L(A) \otimes L(B) \qquad L(A \multimap B) = L(A) \multimap L(B) \qquad L(!A) = !L(A)$$

and preserving ILL typed term constructors, i.e

$$\begin{array}{rcl}
L(\bullet) & = & \bullet \\
L(\lambda a^A.M) & = & \lambda a^{L(A)}.L(M) \\
L(M \otimes N) & = & L(M) \otimes L(N) \\
L(\mathsf{let}\ M\ \mathsf{be}\ a \otimes b\ \mathsf{in}\ N) & = & \mathsf{let}\ L(M)\ \mathsf{be}\ a \otimes b\ \mathsf{in}\ L(N) \\
L(\mathsf{let}\ M\ \mathsf{be}\ \bullet\ \mathsf{in}\ N) & = & \mathsf{let}\ L(M)\ \mathsf{be}\ \bullet\ \mathsf{in}\ L(N) \\
L(!M) & = & !L(M) \\
L(\mathsf{promote}\ M_i\ \mathsf{for}\ a_i\ \mathsf{in}\ N) & = & \mathsf{promote}\ L(M_i)\ \mathsf{for}\ a_i\ \mathsf{in}\ L(N) \\
L(\mathsf{derelict}(M)) & = & \mathsf{derelict}(L(M)) \\
L(\mathsf{discard}\ M\ \mathsf{in}\ N) & = & \mathsf{discard}\ L(M)\ \mathsf{in}\ L(N) \\
L(\mathsf{copy}\ M\ \mathsf{as}\ a, b\ \mathsf{in}\ N) & = & \mathsf{copy}\ L(M)\ \mathsf{as}\ a, b\ \mathsf{in}\ L(N)
\end{array}$$

and preserving the type and term equality judgements of T_1.

Definition 10. The category $\mathsf{Th}(\mathrm{ILL})$ has ILL-theories as objects and ILL-translations as morphisms.

2.4 Linear-non-Linear typed calculus LNL

Benton's typed calculus LNL [5, 6] was obtained from its corresponding notion of categorical model, a sensible category theoretical generalisation of the notion of a model for ILL. It attaches equal prominence to both linear and intuitionistic logic, instead of giving linear logic the status of a basis. Hence, instead of defining weakening and contraction via special linear terms (like "copy M as a, b in N"), the typed calculus is split into a linear part and an intuitionistic part. The intuitionistic part is the simply typed λ-calculus with products. The type constructor !, which regulates when weakening and contraction are applicable, is replaced by an appropriate relation between the linear and the intuitionistic parts of the typed calculus. This special relation between the intuitionistic and the linear parts of the typed calculus is given by two type constructors F and G which map the intuitionistic typed calculus into the linear one and vice versa, and terms which describe how these type constructors interact.

Thus, LNL has type and term judgements and corresponding type equality and term equality judgements regarding the intuitionistic part of the form

$$\vdash_{\mathcal{I}} \tau : \mathsf{type} \qquad \vdash_{\mathcal{I}} \tau = \sigma : \mathsf{type} \qquad \Gamma \vdash_{\mathcal{I}} t : \tau \qquad \Gamma \vdash_{\mathcal{I}} t = s : \tau$$

and the corresponding type and term judgements with their equalities regarding the linear part of the form

$$\vdash_{\mathcal{L}} A : \mathsf{type} \qquad \vdash_{\mathcal{L}} A = B : \mathsf{type} \qquad \Gamma \mid \Delta \vdash_{\mathcal{L}} M : A \qquad \Gamma \mid \Delta \vdash_{\mathcal{L}} M = N : A$$

where Γ is the intuitionistic (or non-linear) context and Δ the linear one.

To make the distinction between these parts more visible, we write $\sigma, \tau, \ldots$ for intuitionistic types, Γ, Γ' for intuitionistic contexts, $x, y, \ldots$ for intuitionistic variables and $t, s, \ldots$ for intuitionistic terms. We write $A, B, \ldots$ for linear types, Δ, Δ' for linear contexts, $a, b, \ldots$ for linear variables and $M, N, \ldots$ for linear terms.

Hence, a (non-linear) context Γ consists of a list of variable declarations of the form $x\colon \sigma$ up to permutation and without repetitions according to the rules in subsection 2.1, whereas a linear context Δ consists of a list of variable declarations of the form $a\colon A$ up to permutation and without repetitions according to the rules in subsection 2.1.

The LNL type constructor rules are:

$$\vdash_{\mathcal{I}} 1 : \mathsf{type} \qquad \frac{\vdash_{\mathcal{I}} \sigma : \mathsf{type} \quad \vdash_{\mathcal{I}} \tau : \mathsf{type}}{\vdash_{\mathcal{I}} \sigma \times \tau : \mathsf{type}}$$

$$\frac{\vdash_{\mathcal{I}} \sigma : \mathsf{type} \quad \vdash_{\mathcal{I}} \tau : \mathsf{type}}{\vdash_{\mathcal{I}} \sigma \to \tau : \mathsf{type}} \qquad \frac{\vdash_{\mathcal{L}} A : \mathsf{type}}{\vdash_{\mathcal{I}} G(A) : \mathsf{type}}$$

$$\frac{}{\vdash_{\mathcal{L}} I : \mathsf{type}} \qquad \frac{\vdash_{\mathcal{L}} A_1 : \mathsf{type} \quad \vdash_{\mathcal{L}} A_2 : \mathsf{type}}{\vdash_{\mathcal{L}} A_1 \otimes A_2 : \mathsf{type}}$$

$$\frac{\vdash_{\mathcal{L}} A_1 : \mathsf{type} \quad \vdash_{\mathcal{L}} A_2 : \mathsf{type}}{\vdash_{\mathcal{L}} A_1 \multimap A_2 : \mathsf{type}} \qquad \frac{\vdash_{\mathcal{I}} \sigma : \mathsf{type}}{\vdash_{\mathcal{L}} F(\sigma) : \mathsf{type}}$$

The LNL typed term constructor rules are:

$$\Gamma, x:\sigma, \Gamma' \vdash_{\mathcal{I}} x:\sigma \qquad\qquad \Gamma \vdash_{\mathcal{I}} *:1$$

$$\frac{\Gamma \vdash_{\mathcal{I}} s:\sigma \quad \Gamma \vdash_{\mathcal{I}} t:\tau}{\Gamma \vdash_{\mathcal{I}} \langle s,t\rangle : \sigma\times\tau} \qquad \frac{\Gamma \vdash_{\mathcal{I}} p:\sigma\times\tau}{\Gamma \vdash_{\mathcal{I}} \mathsf{Fst}(p):\sigma} \qquad \frac{\Gamma \vdash_{\mathcal{I}} p:\sigma\times\tau}{\Gamma \vdash_{\mathcal{I}} \mathsf{Snd}(p):\tau}$$

$$\frac{\Gamma, x:\sigma \vdash_{\mathcal{I}} t:\tau}{\Gamma \vdash_{\mathcal{I}} \lambda x^{\sigma}.t:\sigma\to\tau} \qquad \frac{\Gamma \vdash_{\mathcal{I}} f:\sigma\to\tau \quad \Gamma \vdash_{\mathcal{I}} s:\sigma}{\Gamma \vdash_{\mathcal{I}} f(s):\tau}$$

$$\Gamma \mid a:A \vdash_{\mathcal{L}} a:A$$

$$\frac{\Gamma \mid \Delta, a:A \vdash M:B}{\Gamma \mid \Delta \vdash_{\mathcal{L}} \lambda a^{A}.M : A\multimap B} \qquad \frac{\Gamma \mid \Delta_1 \vdash_{\mathcal{L}} M: A\multimap B \quad \Gamma \mid \Delta_2 \vdash_{\mathcal{L}} N:A}{\Gamma \mid \Delta_1, \Delta_2 \vdash_{\mathcal{L}} M(N):B}$$

$$\frac{\Gamma \mid \Delta_1 \vdash_{\mathcal{L}} M:A \quad \Gamma \mid \Delta_2 \vdash_{\mathcal{L}} N:B}{\Gamma \mid \Delta_1,\Delta_2 \vdash_{\mathcal{L}} M\otimes N : A\otimes B} \qquad \frac{\Gamma \mid \Delta_1 \vdash_{\mathcal{L}} M:A\otimes B \quad \Gamma \mid \Delta_2, a:A, b:B \vdash_{\mathcal{L}} N:C}{\Gamma \mid \Delta_1,\Delta_2 \vdash_{\mathcal{L}} \mathsf{let}\ M\ \mathsf{be}\ a\otimes b\ \mathsf{in}\ N:C}$$

$$\Gamma \mid \_ \vdash_{\mathcal{L}} \bullet : I \qquad \frac{\Gamma \mid \Delta_1 \vdash_{\mathcal{L}} M:I \quad \Gamma \mid \Delta_2 \vdash_{\mathcal{L}} N:C}{\Gamma \mid \Delta_1,\Delta_2 \vdash_{\mathcal{L}} \mathsf{let}\ M\ \mathsf{be}\ \bullet\ \mathsf{in}\ N:C}$$

$$\frac{\Gamma \vdash_{\mathcal{I}} t: G(A)}{\Gamma \mid \_ \vdash_{\mathcal{L}} \mathsf{derelict}(t):A} \qquad \frac{\Gamma \mid \_ \vdash_{\mathcal{L}} M:A}{\Gamma \vdash_{\mathcal{I}} G(M):G(A)}$$

$$\frac{\Gamma \vdash_{\mathcal{I}} t:\sigma}{\Gamma \mid \_ \vdash_{\mathcal{L}} F(t):F(\sigma)} \qquad \frac{\Gamma \mid \Delta_1 \vdash_{\mathcal{L}} M:F(\sigma) \qquad \Gamma, x:\sigma \mid \Delta_2 \vdash_{\mathcal{L}} N:B}{\Gamma \mid \Delta \vdash_{\mathcal{L}} \mathsf{let}\ M\ \mathsf{be}\ F(x)\ \mathsf{in}\ N:B}$$

The β and η equations for intuitionistic terms, (defined using the judgement $\vdash_{\mathcal{I}}$) are those for the simply typed lambda calculus (see for example [15]) with connectives 1, $\times$ and $\to$.

The β and η equations for the linear terms (defined using the judgement $\vdash_{\mathcal{L}}$) are the ones for RLL plus the following β (on the left) and η (on the right) equality rules:

$$\frac{\Gamma \vdash_{\mathcal{I}} t:\sigma \qquad \Gamma, x:\sigma \mid \Delta \vdash_{\mathcal{L}} M:A}{\Gamma \mid \Delta \vdash_{\mathcal{L}} \mathsf{let}\ F(t)\ \mathsf{be}\ F(x)\ \mathsf{in}\ M = M[t/x]:A} \qquad \frac{\Gamma \mid \Delta_1 \vdash_{\mathcal{L}} N:F(\sigma) \qquad \Gamma \mid \Delta_2, a:F(\sigma) \vdash_{\mathcal{L}} M:A}{\Gamma \mid \Delta_1,\Delta_2 \vdash_{\mathcal{L}} \mathsf{let}\ N\ \mathsf{be}\ F(x)\ \mathsf{in}\ M[F(x)/a] = M[N/a]:A}$$

$$\frac{\Gamma \mid \_ \vdash_{\mathcal{L}} M:A}{\Gamma \mid \_ \vdash_{\mathcal{L}} \mathsf{derelict}(G(M)) = M:A} \qquad \frac{\Gamma \vdash_{\mathcal{I}} t:G(A)}{\Gamma \vdash_{\mathcal{I}} G(\mathsf{derelict}(t)) = t:G(A)}$$

The generalised η rule for F is equivalent to the standard η rule ($\mathsf{let}\ M\ \mathsf{be}\ F(x)\ \mathsf{in}\ F(x) = M$) plus commuting conversions [5]. The typed calculus LNL presented here differs from Benton's original one only in that we add the η rules necessary to get a completeness theorem with respect to the models considered by Benton [5]. In the following we call LNL the calculus with η rules.

Definition 11. An LNL-theory is a calculus obtained by extending LNL with new ground linear types A with $\vdash_{\mathcal{L}} A$ type, new ground intuitionistic types σ with $\vdash_{\mathcal{I}} \sigma$ type, new linear type equality judgements $\vdash_{\mathcal{L}} A = B$, for linear types A and B, new intuitionistic type equality judgements $\vdash_{\mathcal{I}} \sigma = \tau$, for intuitionistic types σ and τ, new linear term symbols M and new term judgements $\Gamma \mid \Delta \vdash_{\mathcal{L}} M \colon A$, provided that $\Gamma \mid \Delta$ is a context and A a linear type, new intuitionistic term symbols t and new term judgement $\Gamma \vdash_{\mathcal{I}} t \colon \sigma$, provided that Γ is a context and σ an intuitionistic type, new linear equality judgements $\Gamma \mid \Delta \vdash_{\mathcal{L}} M = N \colon A$, provided that $\Gamma \mid \Delta \vdash_{\mathcal{L}} M \colon A$ and $\Gamma \mid \Delta \vdash N \colon A$ are derivable judgements, and new intuitionistic equality judgements $\Gamma \vdash_{\mathcal{I}} t = s \colon \sigma$, for derivable judgements $\Gamma \vdash_{\mathcal{I}} t \colon \sigma$ and $\Gamma \vdash_{\mathcal{I}} s \colon \sigma$.

Definition 12. Given two LNL-theories T_1 and T_2, an LNL-translation L is a translation from types and terms of T_1 to types and terms of T_2 preserving type and term judgements in the sense of Definition 2, i.e. sending an intuitionistic type σ for which $\vdash_{\mathcal{I}} \sigma : \mathsf{type}$ is derivable in T_1 to an intuitionistic type $L(\sigma)$ such that $\vdash_{\mathcal{I}} L(\sigma) : \mathsf{type}$ is derivable in T_2, sending a linear type A for which $\vdash_{\mathcal{L}} A : \mathsf{type}$ is derivable in T_1 to a type $L(A)$ such that $\vdash_{\mathcal{L}} L(A) : \mathsf{type}$ is derivable in T_2, sending an intuitionistic typed term t such that $\Gamma \vdash_{\mathcal{I}} t : \tau$ is derivable in T_1 to an intuitionistic typed term $L(t)$ such that

$$L(\Gamma) \vdash_{\mathcal{I}} L(t) : L(\tau)$$

is derivable in T_2, sending a linear typed term M such that $\Gamma \mid \Delta \vdash_{\mathcal{L}} M : A$ is derivable in T_1 to a typed term $L(M)$ of T_2 such that

$$L(\Gamma) \mid L(\Delta) \vdash_{\mathcal{L}} L(M) : L(A)$$

is derivable in T_2 — where $L(\Gamma) \equiv x_1 : L(\sigma_1), \ldots, x_n : L(\sigma_n)$ if $\Gamma \equiv x_1 : \sigma_1, \ldots, x_n : \sigma_n$ and $L(\Delta) \equiv a_1 : L(A_1), \ldots, a_n : L(A_n)$ if $\Delta \equiv a_1 : A_1, \ldots, a_n : A_n$ – satisfying in particular

$$L(x) = x \qquad L(a) = a$$

and preserving LNL type constructors, i.e

$$\begin{array}{ccc} L(I) = I & L(A \otimes B) = L(A) \otimes L(B) & L(A \multimap B) = L(A) \multimap L(B) \\ L(1) = 1 & L(A \times B) = L(A) \times L(B) & L(A \to B) = L(A) \to L(B) \end{array}$$
$$L(F(\sigma)) = F(L(\sigma))) \qquad L(G(A)) = G(L(A))$$

and preserving LNL typed term constructors, i.e.

$$\begin{array}{rcl} L(\bullet) & = & \bullet \\ L(\lambda a^A.M) & = & \lambda a^{L(A)}.L(M) \\ L(M \otimes N) & = & L(M) \otimes L(N) \\ L(\mathsf{let}\ M\ \mathsf{be}\ a \otimes b\ \mathsf{in}\ N) & = & \mathsf{let}\ L(M)\ \mathsf{be}\ a \otimes b\ \mathsf{in}\ L(N) \\ L(\mathsf{let}\ M\ \mathsf{be}\ \bullet\ \mathsf{in}\ N) & = & \mathsf{let}\ L(M)\ \mathsf{be}\ \bullet\ \mathsf{in}\ L(N) \\ L(\mathsf{let}\ M\ \mathsf{be}\ F(x)\ \mathsf{in}\ N) & = & \mathsf{let}\ L(M)\ \mathsf{be}\ F(x)\ \mathsf{in}\ L(N) \\ L(G(M)) & = & G(L(M)) \end{array} \qquad \begin{array}{rcl} L(*) & = & * \\ L(\langle t, s \rangle) & = & \langle L(t), L(s) \rangle \\ L(\mathsf{Fst}(t)) & = & \mathsf{Fst}(L(t)) \\ L(\mathsf{Snd}(t)) & = & \mathsf{Snd}(L(t)) \\ L(\lambda x \colon \sigma.t) & = & \lambda x \colon L(\sigma).L(t) \\ L(\mathsf{derelict}(t)) & = & \mathsf{derelict}(L(t)) \end{array}$$

and preserving the type and term equality judgements of T_1.

Definition 13. The category $\mathsf{Th}(\text{LNL})$ has LNL-theories as objects and LNL-translations as morphisms.

We call $^{\not\to}$LNL the calculus LNL where we omit the intuitionistic implication from the intuitionistic (or non-linear) part of LNL. We define the category $\mathsf{Th}(^{\not\to}\text{LNL})$ of $^{\not\to}$LNL-theories and $^{\not\to}$LNL-translations accordingly.

2.5 Dual Intuitionistic and Linear Logic DILL

The typed calculus DILL was first considered by Plotkin and his student Barber [2, 1] more or less at the same time as LNL was investigated by Benton. DILL makes LNL more of a logician's typed calculus, i.e. it makes clear what the roles of F and G are when it comes to distinguishing linear and non-linear behaviour.
DILL type constructor rules are:

$$\vdash\ I : \mathsf{type} \qquad \frac{\vdash A_1 : \mathsf{type} \quad \vdash_{\mathcal{L}} A_2 : \mathsf{type}}{\vdash A_1 \otimes A_2 : \mathsf{type}}$$

$$\frac{\vdash A_1 : \mathsf{type} \quad \vdash A_2 : \mathsf{type}}{\vdash A_1 \multimap A_2 : \mathsf{type}} \qquad \frac{\vdash A : \mathsf{type}}{\vdash !A : \mathsf{type}}$$

DILL typed term constructor rules are:

$$\Gamma \mid a : A \vdash a : A \qquad \Gamma, x\colon A, \Gamma' \mid \_ \vdash x\colon A$$

$$\frac{\Gamma \mid \Delta, a : A \vdash M : B}{\Gamma \mid \Delta \vdash \lambda a^A.M : A \multimap B} \qquad \frac{\Gamma \mid \Delta_1 \vdash M\colon A \multimap B \quad \Gamma \mid \Delta_2 \vdash N\colon A}{\Gamma \mid \Delta_1, \Delta_2 \vdash M(N)\colon B}$$

$$\frac{\Gamma \mid \Delta_1 \vdash M : A \quad \Gamma \mid \Delta_2 \vdash N : B}{\Gamma \mid \Delta_1, \Delta_2 \vdash M \otimes N : A \otimes} \qquad \frac{\Gamma \mid \Delta_1 \vdash M : A \otimes B \quad \Gamma \mid \Delta_2, a : A, b : B \vdash N : C}{\Gamma \mid \Delta_1, \Delta_2 \vdash \mathsf{let}\ M\ \mathsf{be}\ a \otimes b\ \mathsf{in}\ N : C}$$

$$\Gamma \mid \_ \vdash \bullet : I \qquad \frac{\Gamma \mid \Delta_1 \vdash M : I \quad \Gamma \mid \Delta_2 \vdash N : C}{\Gamma \mid \Delta_1, \Delta_2 \vdash \mathsf{let}\ M\ \mathsf{be}\ \bullet\ \mathsf{in}\ N : C}$$

$$\frac{\Gamma \mid \_ \vdash M : A}{\Gamma \mid \_ \vdash !M : !A} \qquad \frac{\Gamma \mid \Delta_1 \vdash M : !A \quad \Gamma, x\colon A \mid \Delta_2 \vdash N\colon B}{\Gamma \mid \Delta_1, \Delta_2 \vdash \mathsf{let}\ M\ \mathsf{be}\ !x\ \mathsf{in}\ N : B}$$

The β and η equations for DILL are the ones for RLL plus the following β (on the left) and generalised η (on the right) rules:

$$\frac{\Gamma \mid \Delta_1 \vdash !M\colon !A \quad \Gamma, x : A \mid \Delta_2 \vdash N\colon B}{\Gamma \mid \Delta_1, \Delta_2 \vdash \mathsf{let}\ !M\ \mathsf{be}\ !x\ \mathsf{in}\ N = N[M/x]} \qquad \frac{\Gamma \mid \Delta_1 \vdash M\colon !A \quad \Gamma \mid a : !A, \Delta_2 \vdash N\colon B}{\Gamma \mid \Delta_1, \Delta_2 \vdash \mathsf{let}\ M\ \mathsf{be}\ !x\ \mathsf{in}\ N[!x/a] = N[M/a]\colon B}$$

Some of the advantages of DILL are:

1. there is no need for verbose syntax to deal with contraction and weakening as in ILL; these are handled implicitly as in intuitionistic logic;
2. the easy formulation of the rule of promotion (which caused all the trouble before in ILL).

Definition 14. A DILL-theory is a calculus obtained by extending DILL with new ground types A with $\vdash A\ \mathsf{type}$, new type equality judgements $\vdash A = B$, for A and B types, new term symbols M and term judgements $\Gamma \mid \Delta \vdash M\colon A$, provided that $\Gamma \mid \Delta$ is a context and A is a type, and new equality judgements $\Gamma \mid \Delta \vdash M = N\colon A$, for derivable judgements $\Gamma \mid \Delta \vdash M\colon A$ and $\Gamma \mid \Delta \vdash N\colon A$.

Definition 15. Given two DILL-theories T_1 and T_2, a DILL-translation L is a translation from types and terms of T_1 to types and terms of T_2 preserving type and term judgements in the sense of Definition 2, i.e. sending a type A for which $\vdash A : \mathsf{type}$ is derivable in T_1 to a type $L(A)$ such that $\vdash L(A) : \mathsf{type}$ is derivable in T_2 and sending a typed term M such that $\Gamma \mid \Delta \vdash M : A$ is derivable in T_1 to a typed term $L(M)$ of T_2 such that

$$L(\Gamma) \mid L(\Delta) \vdash L(M) : L(A)$$

is derivable in T_2 — where $L(\Gamma) \equiv x_1 : L(\sigma_1), \ldots, x_n : L(\sigma_n)$ if $\Gamma \equiv x_1 : \sigma_1, \ldots, x_n : \sigma_n$ and $L(\Delta) \equiv a_1 : L(A_1), \ldots, a_n : L(A_n)$ if $\Delta \equiv a_1 : A_1, \ldots, a_n : A_n$ — which satisfies

$$L(x) = x \qquad L(a) = a$$

and preserves DILL type constructors, i.e.

$$L(I) = I \qquad L(A \otimes B) = L(A) \otimes L(B) \qquad L(A \multimap B) = L(A) \multimap L(B) \qquad L(!A) = !L(A)$$

and DILL typed term constructors, i.e.

$$\begin{array}{rcl} L(\bullet) & = & \bullet \\ L(\lambda a^A.M) & = & \lambda a^{L(A)}.L(M) \\ L(M \otimes N) & = & L(M) \otimes L(N) \\ L(\mathsf{let}\ M\ \mathsf{be}\ a \otimes b\ \mathsf{in}\ N) & = & \mathsf{let}\ L(M)\ \mathsf{be}\ a \otimes b\ \mathsf{in}\ L(N) \\ L(\mathsf{let}\ M\ \mathsf{be}\ \bullet\ \mathsf{in}\ N \bullet) & = & \mathsf{let}\ L(M)\ \mathsf{be}\ \bullet\ \mathsf{in}\ L(N) \\ L(!M) & = & !L(M) \\ L(\mathsf{let}\ M\ \mathsf{be}\ !x\ \mathsf{in}\ N) & = & \mathsf{let}\ L(M)\ \mathsf{be}\ !x\ \mathsf{in}\ L(N) \end{array}$$

and that preserves the type and term equality judgements of T_1.

Definition 16. The category $\mathsf{Th}(\mathrm{DILL})$ has DILL-theories as objects and DILL-translations as morphisms.

2.6 Adding additive conjunction

It is well known that we can add additive conjunction & (read as "with") and its unit 1 to DILL so that we obtain ${}_{\&}$DILL.

The type constructor rules of ${}_{\&}$DILL for these types are:

$$\vdash\ 1 : \mathsf{type} \qquad \frac{\vdash A : \mathsf{type} \quad \vdash B : \mathsf{type}}{\vdash A \& B : \mathsf{type}}$$

and the corresponding term constructor rules are:

$$\Gamma \mid \Delta \vdash \circ : 1 \quad \frac{\Gamma \mid \Delta \vdash M : A \quad \Gamma \mid \Delta \vdash N : B}{\Gamma \mid \Delta \vdash (M, N) : A \& B} \quad \frac{\Gamma \mid \Delta \vdash M : A \& B}{\Gamma \mid \Delta \vdash \mathsf{Fst}(M) : A} \quad \frac{\Gamma \mid \Delta \vdash M : A \& B}{\Gamma \mid \Delta \vdash \mathsf{Snd}(M) : B}$$

We need to add the corresponding β and η rules typical of product types (see for example [15]).

Similarly, the rules above, suitably adapted, can be added to ILL, LNL and ${}^{\rightharpoonup}$LNL to get ${}_{\&}$ILL,${}_{\&}$LNL and ${}^{\rightharpoonup}_{\&}$LNL and analogously for these new systems we define categories of theories and translations.

2.7 Relating ILL and DILL

We start by recalling that all the three typed calculi prove exactly the same purely linear theorems, i.e. they cannot be distinguished in terms of provability of purely linear theorems only. But our interest here is to compare them from the point of view of proofs. In this subsection we state precisely the relation between ILL and DILL.

We say that two different type calculi are *equivalent* if there are translations sending well-formed types to well-formed types and well-typed terms to well-typed terms such that to preserve type and term equality judgements and such that when composed give you the identity, in both directions.

Theorem 17 (Barber). *ILL is equivalent to the fragment of DILL with only judgements of the form* $\_ \mid \Delta \vdash M\colon A$*. In the same way* $_{\&}$*ILL is equivalent to the fragment of* $_{\&}$*DILL with only judgements of the form* $\_ \mid \Delta \vdash M\colon A$*.*

The proof of this can be found in Barber's thesis [1], where he provides translations from DILL to ILL and conversely satisfying the above property. Note that this theorem says that DILL and ILL are essentially equivalent, since in DILL every judgement of the form $\Gamma \mid \Delta \vdash M\colon A$ corresponds bijectively to a judgement $\_ \mid !\Gamma, \Delta \vdash M''\colon A$, where $!\Gamma \equiv !A_1, \ldots !A_n$ if $\Gamma \equiv A_1, \ldots A_n$.

The concept of the category of theories helps us to improve the understanding of the relationship between ILL and DILL. Indeed, thanks to Theorem 17 we can prove that the category of DILL-theories is equivalent to that of ILL-theories.

Theorem 18. *The category* Th*(DILL) of DILL-theories is equivalent to the category* Th*(ILL) of ILL-theories. Analogously, the category* Th*(*$_{\&}$*DILL) of* $_{\&}$*DILL-theories is equivalent to the category* Th*(*$_{\&}$*ILL) of* $_{\&}$*ILL-theories.*

Proof. The proof is based on Theorem 17 whose corresponding translations are ϕ_{DILL} : DILL $\to$ ILL and ψ_{ILL} : ILL $\to$ DILL as defined in [1]. Note that such translations are identities on the types.

To define a functor from Th(DILL) to Th(ILL), we associate with any DILL-theory $\mathcal{T}$ the ILL-theory $\mathcal{L}(\mathcal{T})$ obtained by adding to ILL:

1. a new type symbol A and an axiom $\vdash A : \mathsf{type}$ for any proper $\mathcal{T}$-axiom $\vdash A : \mathsf{type}$;
2. a new axiom $\vdash A = B : \mathsf{type}$ for any proper $\mathcal{T}$-axiom $\vdash A = B : \mathsf{type}$;
3. a new term symbol $\overline{M}$ with a new term axiom $\Delta \vdash \overline{M}\colon A$ for any proper $\mathcal{T}$-axiom $\_ \mid \Delta \vdash M\colon A$, and a new term symbol $\overline{\mathsf{let}\ b\ \mathsf{be}\ !z\ \mathsf{in}\ M}$ with a new term axiom

   $$b : !B, \Delta \vdash \overline{\mathsf{let}\ b\ \mathsf{be}\ !z\ \mathsf{in}\ M}\colon A$$

   for any proper $\mathcal{T}$-axiom $z : B \mid \Delta \vdash M\colon A$. In the same way, for any proper $\mathcal{T}$-axiom $\Gamma \mid \Delta \vdash M\colon A$, where Γ has more than one assumption, we add a new term symbol $\overline{\mathsf{let}\ b_1\ \mathsf{be}\ !z_1\ \mathsf{in}\ \ldots \mathsf{let}\ b_n\ \mathsf{be}\ !z_n\ \mathsf{in}\ M}$ and a new term axiom

   $$!\Gamma, \Delta \vdash \overline{\mathsf{let}\ b_1\ \mathsf{be}\ !z_1\ \mathsf{in}\ \ldots \mathsf{let}\ b_n\ \mathsf{be}\ !z_n\ \mathsf{in}\ M} : A$$

   where we overline the term with many nested let corresponding to

   $$\_ \mid !\Gamma, \Delta \vdash \mathsf{let}\ b_1\ \mathsf{be}\ !z_1\ \mathsf{in}\ \ldots \mathsf{let}\ b_n\ \mathsf{be}\ !z_n\ \mathsf{in}\ M\colon A$$

for $\Gamma \equiv z_1 : B_1, \ldots, z_n : B_n$, which is also derivable in $\mathcal{T}$.

Then, we extend ϕ_{DILL} to a translation $\phi_{\mathcal{T}}$ on the types and terms of $\mathcal{T}$ into the types and terms of $\mathcal{L}(\mathcal{T})$ so far described by sending any new type and any new term of $\mathcal{T}$ added to DILL respectively to the corresponding new type and term of $\mathcal{L}(\mathcal{T})$ added to ILL, i.e. for example we define $\phi_{\mathcal{T}}(M) \equiv \overline{M}$.

4. Using the translation $\phi_{\mathcal{T}}$ we add a new equality axiom

$$\Delta \vdash \phi_{\mathcal{T}}(M) = \phi_{\mathcal{T}}(N) \colon A$$

for any proper $\mathcal{T}$-axiom $\_ \mid \Delta \vdash M = N \colon A$ and also a new equality axiom

$$!\Gamma, \Delta \vdash \phi_{\mathcal{T}}(M) = \phi_{\mathcal{T}}(N) \colon A$$

for any proper $\mathcal{T}$-axiom $\Gamma \mid \Delta \vdash M = N \colon A$.

By this last addition, which completes the description of $\mathcal{L}(\mathcal{T})$, the translation $\phi_{\mathcal{T}} : \mathcal{T} \to \mathcal{L}(\mathcal{T})$ turns out to preserve also equality judgements.

Then, the functor $\mathcal{L}$ can be extended on the translations $Tr : \mathcal{T} \to \mathcal{T}'$ as follows. First, we recall that an ILL-translation is determined by its definition on the new types and terms added to ILL to get $\mathcal{L}(\mathcal{T})$. Hence, we define $\mathcal{L}(Tr)$ as $\mathcal{L}(Tr)(A) \equiv Tr(A)$ on the new types A of $\mathcal{L}(\mathcal{T})$ — which have the same names as those of $\mathcal{T}$ — and as $\mathcal{L}(Tr)(\overline{\mathsf{let}\ b\ \mathsf{be}\ !z\ \mathsf{in}\ M}) \equiv \phi_{\mathcal{T}}(Tr(M))$ on the new terms $\overline{M}$ such that $b :!B, \Delta \vdash \overline{\mathsf{let}\ b\ \mathsf{be}\ !z\ \mathsf{in}\ M} \colon A$ is a proper $\mathcal{L}(\mathcal{T})$-axiom and we complete the definition analogously on the other new terms.

Conversely, to any ILL-theory $\mathcal{S}$ we associate the DILL-theory $\mathcal{L}^{-1}(\mathcal{S})$ defined by adding to DILL:

1. a new type symbol A and a new axiom $\vdash A : \mathsf{type}$ for any proper $\mathcal{S}$-axiom $\vdash A : \mathsf{type}$;

2. a new axiom $\vdash A = B : \mathsf{type}$ for any proper $\mathcal{S}$-axiom $\vdash A = B : \mathsf{type}$;

3. a new term symbol $\overline{M}$ with a new term axiom

$$\_ \mid \Delta \vdash \overline{M} \colon A$$

for any proper $\mathcal{S}$-axiom $\Delta \vdash M \colon A$;

then, we extend ψ_{ILL} to a translation $\psi_{\mathcal{S}}$ on the types and terms of $\mathcal{S}$ into the types and terms of $\mathcal{L}^{-1}(\mathcal{S})$ so far described by sending any new type and new term of $\mathcal{S}$ added to ILL respectively to the corresponding new type and term of $\mathcal{L}^{-1}(\mathcal{S})$ added to DILL;

4. a new equality axiom

$$\_ \mid \Delta \vdash \psi_{\mathcal{S}}(M) = \psi_{\mathcal{S}}(N) \colon A$$

for any proper $\mathcal{S}$-axiom $\Delta \vdash M = N \colon A$.

By this last addition, which completes the description of $\mathcal{L}^{-1}(\mathcal{S})$, the translation $\psi_{\text{DILL}} : \mathcal{S} \to \mathcal{L}^{-1}(\mathcal{S})$ turns out to preserve also equality judgements.

The functor $\mathcal{L}^{-1}$ can be extended on the translations $Tr : \mathcal{S} \to \mathcal{S}'$ as follows. We define $\mathcal{L}^{-1}(Tr)$ as $\mathcal{L}^{-1}(Tr)(A\ ) \equiv Tr(A)$ on the new types of $\mathcal{L}^{-1}(\mathcal{S})$ and as $\mathcal{L}(Tr)(\overline{M}) \equiv \psi_{\mathcal{S}'}(Tr(M))$ on the new terms $\overline{M}$ such that $\Delta \vdash \overline{M} \colon A$ is a proper $\mathcal{L}^{-1}(\mathcal{S})$-axiom.

We can then prove that such functors give an equivalence of categories by employing the bijection between judgements of the form $\Gamma \mid \Delta \vdash M \colon A$ and $\_ \mid !\Gamma, \Delta \vdash M' \colon A$ to define the unit and counit isomorphisms.

The proof that $\mathsf{Th}({}_{\&}\text{DILL})$ is equivalent to $\mathsf{Th}({}_{\&}\text{ILL})$ is analogous. □

We will provide a comparison between models of LNL and DILL (and hence ILL) when the systems have the additive conjunction & and corresponding unit 1 in the next section.

Here, we just observe that a faithful translation of ILL into LNL can be provided in a way that preserves proofs (see [5]). In [5] a translation from LNL into ILL is also given but it does not preserve all proof equalities, as the translation of F does not turn out to be necessarily modelled by a a strong monoidal functor.

3 Categorical Models

The typed calculi described in the previous section have associated categorical models which we recapitulate and clarify in this section. The clarification is necessary since morphisms of models were not taken in consideration before in the literature. Thus we discuss models (and morphisms of models thereof) for RLL, ILL, LNL and DILL, as well as for these systems with additive conjunction and we compare them. Before we start with RLL, we digress a little to introduce our main tool to connect typed calculi and their semantics, namely the notion of *internal language*.

3.1 Notion of internal language

To relate linear typed calculi and their categorical models we use the notion of internal language. Introductions to categorical logic such as Lambek and Scott [21] and Pitts [27], give examples of internal languages for specific categories modelling certain typed calculi.

The general idea (emphasising the role of theory-morphisms) is the following. Given a typed calculus $\mathcal{C}$, we consider its category of theories as defined in the previous section. We also assume that we have sound and complete models for such a calculus with a notion of model morphisms forming a category $\mathcal{M}(\mathcal{C})$. Then we say that *a typed calculus $\mathcal{C}$ provides an* internal *language for the models in $\mathcal{M}(\mathcal{C})$ if we can establish an equivalence between the category of $\mathcal{C}$-theories and the category of $\mathcal{C}$-models.*
The functors establishing the above equivalence, let us say $L : \mathcal{M}(\mathcal{C}) \to \mathsf{Th}(\mathcal{C})$ and $C : \mathsf{Th}(\mathcal{C}) \to \mathcal{M}(\mathcal{C})$ assign to a model M in $\mathcal{M}(\mathcal{C})$ its *specific internal language* $L(M)$ as a $\mathcal{C}$-theory and to each $\mathcal{C}$-theory $\mathcal{V}$ its syntactic category $C(\mathcal{V})$ such that $M \simeq_{iso} C(L(M))$ and $\mathcal{V} \simeq_{iso} L(C(\mathcal{V}))$, that is M is isomorphic to $C(L(M))$ and $\mathcal{V}$ is isomorphic to $L(C(\mathcal{V}))$. [4] Many authors, for example Barr and Wells [4], consider only an equivalence $M \simeq C(L(M))$ as the characterizing property of the internal language. Lambek and Scott [21], Pitts [27] and Crole [11] also give the definition of internal language of some class of categories and they also consider the notion of theory-morphism in terms of translation. However, contrary to Lambek and Scott and us, Pitts [27] and Crole [11] define translations between theories preserving type constructors up to isomorphisms.

3.2 Models of RLL

It is well-known that to model the modality-free fragment of intuitionistic multiplicative linear logic, or rudimentary linear logic, RLL we need a symmetric monoidal closed category [22, 18, 5].

Definition 19. *A model of !-free intuitionistic multiplicative linear logic or rudimentary linear logic RLL consists of a symmetric monoidal closed (SMC) category $\mathbb{A}$, also called an autonomous category.*

[4] Here we could require just an equivalence in both cases, but then we would only get a sort of biequivalence (and not an equivalence) between the category of models and that of theories.

Clearly the collection of SMC categories with the notion of morphism preserving the SMC structure, i.e. that of strong symmetric monoidal closed functors [17], forms a category.

We recall that a (symmetric) monoidal functor $(F, \mathsf{m}^{F}, \mathsf{m}^{F}_{I})$ between the (symmetric) monoidal categories $(\mathbb{S}, \otimes, I)$ and $(\mathbb{S}', \otimes', I')$ is *strong* if $\mathsf{m}^{F}_{A,B}$ and m^{F}_{I} are isomorphisms (where $\mathsf{m}^{F}_{A,B} : FA \otimes' FB \to F(A \otimes B)$ and $\mathsf{m}^{F}_{I} : I' \to FI$ are the natural transformation and the morphism satisfying the monoidal equations [17, 5]).

Moreover, we say that a strong monoidal functor is *strict* if $\mathsf{m}^{F}_{A,B}$ and m^{F}_{I} are identities.

Definition 20. *We call* **SMC** *the category whose objects are symmetric monoidal closed (SMC) categories and whose morphisms are* strong *symmetric monoidal closed. Moreover, we call* $\mathbf{SMC}_{st}$ *the category with the same objects, but whose morphisms are* strict *symmetric monoidal closed functors.*

Remark 21. Throughout the paper whenever we give a definition of a category of models we also give, in parallel, a strict version of it. The strict version matches the notion of translation between theories, while the relaxed one has more of a categorical flavour.

We could state and prove the soundness and completeness of the models, but this result is included in the stronger fact that the typed calculus RLL provides *an internal language for autonomous categories.*

Theorem 22. *The typed calculus RLL provides an internal language for SMC categories, i.e.* $\mathsf{Th}(RLL)$ *is equivalent to* $\mathbf{SMC}_{st}$.

Proof. We define a functor C from $\mathsf{Th}(\mathrm{RLL})$ to $\mathbf{SMC}_{st}$ by associating to any RLL-theory the well-known construction of the syntactic category having types as objects and terms as morphisms (see [23]) both modulo the corresponding equalities. Then, to any translation the functor C associates the functor induced on the syntactic categories by associating to a type and to a term their respective translations.

Conversely, the functor L from $\mathbf{SMC}_{st}$ to $\mathsf{Th}(\mathrm{RLL})$ is defined by taking an SMC category $\mathbb{S}$ to its internal language $L(\mathbb{S})$ defined by extending RLL with:

1. new ground types A and corresponding axioms $\vdash A : \mathsf{type}$ for each object A of $\mathbb{S}$, i.e. we name the objects of $\mathbb{S}$ in the calculus and we extend the interpretation of RLL-types in $\mathbb{S}$ by interpreting the new names in the corresponding objects;

2. new terms M and $x : A_1, \dots, x_n : A_n \vdash M : B$ naming each morphism of $\mathbb{S}$ having the interpretation of the context $x : A_1, \dots, x_n : A_n$ as domain and the interpretation of B as codomain;

3. new equality types axioms $\vdash A = B : \mathsf{type}$, provided that the interpretation of A is equal to that of B in $\mathbb{S}$;

4. new equality terms axioms $x : A_1, \dots, x_n : A_n \vdash M = N : B$, provided that the interpretation of $x : A_1, \dots, x_n : A_n \vdash M : B$ in $\mathbb{S}$ is equal to that of $x : A_1, \dots, x_n : A_n \vdash N : B$ by interpreting the new term symbols in the morphisms they name.

Clearly, any strict monoidal closed functor gives rise to a translation.
It is easy to check that the functors so defined establish an equivalence of categories. □

Part of this result, i.e. for every autonomous category $\mathbb{S}$ we have $\mathbb{S} \simeq C(L(\mathbb{S}))$, was first stated in [23], but there the description of the internal language of a symmetric monoidal closed category is not complete since it does not include all the equations semantically sound in the category itself[5].

An internal language for autonomous categories was also presented in [19] but in the style of Gentzen's sequent calculus.

Remark 23. The reason we choose the notion of strong (strict) symmetric monoidal closed functor between SMC categories rather than that of symmetric monoidal closed functor, is to match the notion of translation when proving the internal language theorem 22.

3.3 Models of ILL

Now, we give the definition of a linear category intended to be the categorical semantics for ILL. More details on the definitions and theorems proved in this subsection can be found in [25].

Definition 24. *A* **linear category** *is a symmetric monoidal closed category* $(\mathbb{S}, \otimes, I, \multimap, a, \lambda, \rho, s)$, *together with a symmetric monoidal comonad* $(!, \mathsf{m}^{!}, \mathsf{m}^{!}_{I}, \varepsilon, \delta)$ *such that the monoidal structure induced on the associated Eilenberg-Moore category* $\mathbb{S}^{!}$ *is a finite product structure.*

This definition is new and equivalent to the original definition of linear category given by Bierman et al. in [10, 7, 9]. This is a consequence of the following:

Proposition 25. *A symmetric monoidal closed category* $\mathbb{S}$ *with a symmetric monoidal comonad* $(!, \mathsf{m}^{!}, \mathsf{m}^{!}_{I}, \varepsilon, \delta)$ *is a linear category if and only if*

1. *There are monoidal natural transformations* $e\colon ! \to I$ *and* $d\colon ! \to (!-\otimes !-)$ *such that for every free coalgebra* $(!A, \delta_A)$ *the components* $e_A\colon !A \to I$ *and* $d_A\colon !A \to !A \otimes !A$ *form a commutative comonoid and are coalgebra maps;*

2. *every component of* δ *is a comonoid morphism.*

Proof. By definition, $\mathbb{S}$ is a linear category if the symmetric monoidal structure induced on the category of coalgebras is given by finite products. Hence, each !-coalgebra (A, a) has an obvious comonoid structure given by the map into the terminal object $(I, \mathsf{m}^{!}_{I})$ and the diagonal morphism into the product $(A, a) \otimes (A, a)$. Consider this commutative comonoid at every free coalgebra $(!A, \delta_A)$. This provides the component at A of the natural transformations e and d.

For the other direction, the proof (appearing in [9]) that conditions (1) and (2) imply that the symmetric monoidal structure induced on the Eilenberg-Moore category is given by finite products goes along this outline:

- $\mathbb{S}^{!}$ has a symmetric monoidal structure induced by that of $\mathbb{S}$;

- from e and d with the rest of the given assumptions, one obtains a commutative comonoid on each !-coalgebra (A, a) in $\mathbb{S}^{!}$;

[5] To get the equivalence $\mathbb{S} \simeq C(L(\mathbb{S}))$ a key difficulty is to relate the free SMC structure generated through the rules of the calculus from the axioms naming objects and morphisms of the category with the original SMC structure presented in the category. For that we could use some encoding terms like in [11, 30] or we could add appropriate equalities of types and terms as we do in theorem 22.

- in $\mathbb{S}^!$ the comonoid on the unit coalgebra is the unit comonoid $(I, id_I, \lambda_I^{-1})$ and the comonoid on the tensor $(A, a) \otimes (B, b)$ of two coalgebras is the tensor of the comonoids on (A, a) and on (B, b);

- every coalgebra map is a comonoid map or, equivalently, the families of counit and comultiplications of the comonoids are natural.

These properties force $\mathbb{S}^!$ with the given symmetric monoidal structure to be in fact a category with finite products. Indeed, the comonoid structure on any coalgebra consists of a morphism to the terminal object (for the counit) and in the diagonal morphism of binary products (for the comultiplication). $\square$

Remark 26. From the above proof we can extract the information that the components of the natural transformations e and d are uniquely determined by the finite product structure of the Eilenberg-Moore category $\mathbb{S}^!$: they are respectively the morphism to the terminal object and the diagonal at each free coalgebra $(!A, \delta_A)$. Their existence depends on the choice of the symmetric monoidal comonad ! and need not be regarded as additional structure for which there could be more than one choice.

Properties (1) and (2) of Proposition 25 remain useful tools to check if a candidate symmetric monoidal comonad provides a linear category in concrete examples. They are a simplification of the original definition of linear category given by Bierman et al. [10, 7, 9] as the following lemma shows[6]:

Lemma 27. *Let $(\mathbb{S}, \otimes, I)$ be a symmetric monoidal category with a symmetric monoidal comonad $(!, \mathsf{m}^!, \mathsf{m}^!_I, \epsilon, \delta)$. Assume that there are monoidal natural transformations $e\colon ! \to I$ and $d\colon ! \to (!-\otimes!-)$ whose components are coalgebra morphisms and such that $(!A, e_A, d_A)$ are commutative comonoids. Then, every coalgebra morphism between free coalgebras $f : (!A, \delta_A) \to (!B, \delta_B)$ is a comonoid morphism if and only if every component of δ is a comonoid morphism.*

Proof. The *only if* part is obvious. For the *if* part consider that the naturality of e and d amounts precisely to the requirement that each morphism of the form $!f$ is a comonoid morphism. Each component of δ is a comonoid morphism, that is $e_{!A} \circ \delta_A = e_A$ and $d_{!A} \circ \delta_A = (\delta_A \otimes \delta_A) \circ d_A$ for any $A \in \mathbb{S}$. With these equations in mind, it is easy to check that a coalgebra morphism $f : (!A, \delta_A) \to (!B, \delta_B)$ is a morphism of commutative comonoids. $\square$

We point out that another concise definition of linear category similar to ours and equivalent to the original one given by Bierman appears in [16]. In [16] the property chosen to define a linear category is the fact that the Eilenberg-Moore category for the symmetric monoidal comonad (with the symmetric monoidal structure induced by the comonad) is a category of commutative comonoids in such a way that the comonoid on the unit coalgebra is the identity comonoid and the comonoid on the tensor of two coalgebras is the tensor of the comonoids on each of them.

As explained in [7, 9, 1] linear categories provide sound and complete models for ILL. In the sequel we show that the relationship between ILL and linear categories is even stronger: ILL provides an internal language for linear categories. To be able to state this result we need to give a suitable notion of morphism of linear category and define a category **Lin** of linear categories. The category **Lin** will be a subcategory of the category of comonads and *strict* comonad morphisms:

[6] We thank A. Schalk for pointing this out to us.

Definition 28 (Comonads). *We denote by **Cmad** the category whose objects are comonads* (T, ε, δ) *on an arbitrary category* $\mathbb{B}$*, and whose morphisms from* (T, ε, δ) *on* $\mathbb{B}$ *to* $(T', \varepsilon', \delta')$ *on* $\mathbb{B}'$ *are functors* $K : \mathbb{B} \to \mathbb{B}'$ *satisfying* $KT = T'K$*,* $K\varepsilon = \varepsilon'_K$ *and* $K\delta = \delta'_K$*.*

This notion of comonad morphism is a particular case of the one in [29] given by a natural transformation $\alpha : KT \to T'K$ (here replaced by identity) satisfying suitable conditions on ϵ and δ to preserve the comonad structure.

The reason of our strict notion of morphism between comonads is to correspond in the special case of linear categories to the notion of ILL-translations.

Definition 29. ***Lin*** *(**Lin**$_{st}$) is a subcategory of **Cmad** having as objects linear categories and morphisms from* $(\mathbb{S}, !, \varepsilon, \delta, \mathsf{m}^!, \mathsf{m}^!_I)$ *to* $(\mathbb{S}', !', \varepsilon', \delta', \mathsf{m}^{!'}, \mathsf{m}^{!'}_I)$ *those comonad morphism* K *that are strong (strict) symmetric monoidal closed functors* $(K, \mathsf{m}^K, \mathsf{m}^K_I) \colon \mathbb{S} \to \mathbb{S}'$ *such that* $K! = !'K$ *as* monoidal functors*, that is* $K\mathsf{m}^!_{A,B} \circ \mathsf{m}^K_{!A,!B} = !'\mathsf{m}^K_{A,B} \circ \mathsf{m}^{!'}_{KA,KB}$ *and* $K\mathsf{m}^!_I \circ \mathsf{m}^K_I = !'\mathsf{m}^K_I \circ \mathsf{m}^{!'}_I$*.*

Note that, as we may expect, the natural transformations e and d of Proposition 25 are preserved by a **Lin**-morphism, in the sense of the following lemma.

Lemma 30. *Let* $(K, \mathsf{m}^!, \mathsf{m}^!_I) : \mathbb{S} \to \mathbb{S}'$ *be a **Lin**-morphism and* (e, d)*,* (e', d') *be the natural transformations for* $\mathbb{S}$ *and* $\mathbb{S}'$ *as given in Proposition 25. Then* $Ke = \mathsf{m}^K_I \circ e'_K$ *and* $Kd = \mathsf{m}^K \circ d'_K$*.*

Proof. This is a consequence of the fact that K is strong monoidal, that $K! = !'K$ as monoidal functors and that e and d are uniquely determined by the finite product structure (see Remark 26). □

This proposition also shows that our definition of morphism provides a *linearly distributive functor* in the sense of [16].

Using the notion of linear morphism in definition 29 we can prove:

Theorem 31. *The typed calculus ILL provides an internal language for linear categories, i.e.* Th*(ILL) is equivalent to **Lin**$_{st}$.*

We could provide a direct proof of Theorem 31 based on the soundness theorem proved in Bierman's thesis [9] and on the completeness theorem for DILL in [1]. However, given the number of rules in ILL, we prefer to prove Theorem 31 by using Theorem 18 saying that Th(ILL) is equivalent to Th(DILL) together with the proof that DILL provides an internal language of a category of models equivalent to **Lin**$_{st}$, as we will see in the next subsection.

3.4 Symmetric monoidal adjunctions as models of LNL and DILL

The models that are presented in the literature for LNL and DILL are based on certain symmetric monoidal adjunctions.

When working with symmetric monoidal adjunctions we often employ the useful characterization due to Kelly [17, 5] which says that an adjunction between (symmetric) monoidal categories $F \dashv G$ is (symmetric) monoidal if and only if the left adjoint F is strong (symmetric) monoidal.

In [2] it is proved that the following structures are sound and complete models for DILL:

Definition 32 (Barber & Plotkin). *A model of DILL is a symmetric monoidal adjunction* $F \dashv G$*, with* $F \colon \mathbb{C} \to \mathbb{S}$ *and* $G \colon \mathbb{S} \to \mathbb{C}$*, where* $\mathbb{S}$ *is a symmetric monoidal closed category and* $\mathbb{C}$ *is a cartesian category.*

In [5] it is proved that the following structures are sound models for LNL:

Definition 33 (Benton). *A model of LNL is a symmetric monoidal adjunction $F \dashv G$, with $F\colon \mathbb{C} \to \mathbb{S}$ and $G\colon \mathbb{S} \to \mathbb{C}$, where $\mathbb{S}$ is a symmetric monoidal closed category and $\mathbb{C}$ is a cartesian closed category.*

For these structures we now need to define suitable morphisms. The corresponding categories of symmetric monoidal adjunctions are subcategories of the category of adjunctions and transformations of adjunctions as defined in [22]:

Definition 34 (Category of Adjunctions). *By **Adj** we denote the category whose objects are adjunctions $F \dashv G : \mathbb{A} \to \mathbb{B}$ between any two categories and whose morphisms from $F \dashv G : \mathbb{A} \to \mathbb{B}$ to $F' \dashv G' : \mathbb{A}' \to \mathbb{B}'$ are* transformations of adjunctions*, that is, pairs of functors (K, H) with $K : \mathbb{B} \to \mathbb{B}'$, $H : \mathbb{A} \to \mathbb{A}'$ such that $KF = F'H$, $HG = G'K$ and such that the natural isomorphisms defining the adjunctions are preserved. That is both squares below commute*

$$\begin{tikzcd}
\mathbb{A} \arrow[r, "H"] \arrow[d, shift right, "F"'] & \mathbb{A}' \arrow[d, shift right, "F'"'] \\
\mathbb{B} \arrow[r, "K"] \arrow[u, shift right, "G"'] & \mathbb{B}' \arrow[u, shift right, "G'"']
\end{tikzcd}$$

and any of the two following equivalent conditions holds: $H\eta = \eta'_H$ or $K\varepsilon = \varepsilon'_K$ where η, η' are the units and ε, ε' the counits of the given adjunctions.

Before giving the definitions of the corresponding categories of symmetric monoidal adjunctions, we recall that a *cartesian closed* functor is a functor preserving the (chosen) finite products and closure up to isomorphisms and a *strict* cartesian closed functor is a functor preserving such a structure on the nose.

Here is the definition of the category of symmetric monoidal adjunctions sound and complete for DILL.

Definition 35. *The category **Sma** (respectively **Sma**$_{st}$) is the subcategory of **Adj** whose objects are symmetric monoidal adjunctions between a cartesian category and a symmetric monoidal closed category and whose morphisms between two such symmetric monoidal adjunctions $F \dashv G$ and $F' \dashv G'$ are transformations of adjoints (K, H) where K is a strong (respectively strict) symmetric monoidal closed functor and H is a (respectively strict) cartesian functor and (K, H) is* monoidal*, that is $K\overset{F}{\mathsf{m}}_{A,B} \circ \overset{K}{\mathsf{m}}_{FA,FB} = F'\overset{H}{\mathsf{m}}_{A,B} \circ \overset{F'}{\mathsf{m}}_{HA,HB}$ and $K\overset{F}{\mathsf{m}}_I \circ \overset{K}{\mathsf{m}}_I = F'\overset{H}{\mathsf{m}}_I \circ \overset{F'}{\mathsf{m}}_I$ and similar conditions for $G'K = HG$.*

Now we give the definition of the category of symmetric monoidal adjunctions sound for LNL.

Definition 36. *The category $^{\to}$**Sma** ($^{\to}$**Sma**$_{st}$) is the subcategory of **Sma** (**Sma**$_{st}$) whose objects are symmetric monoidal adjunctions between a cartesian* closed *category and a symmetric monoidal closed category and whose morphisms between two such symmetric monoidal adjunctions $F \dashv G$ and $F' \dashv G'$ are **Sma** (**Sma**$_{st}$) morphisms (K, H) where K is a strong (strict) symmetric monoidal closed functor and H is a (strict) cartesian closed functor.*

Then, we can prove.

Proposition 37. *The symmetric monoidal adjunctions in* $^{\rightarrow}\boldsymbol{Sma}$ *are sound and* complete *with respect to the typed calculus LNL.*

Proof. In [5] it is proved soundness of all the rules in LNL except for the η rules whose validity follows easily, too. To get completeness we build a symmetric monoidal adjunction between the symmetric monoidal closed category $\mathbb{S}$, whose objects are the linear types and whose morphisms are $\_ \mid a : A \vdash_{\mathcal{L}} M : B$, both modulo the corresponding equalities, and the cartesian closed category $\mathbb{C}$, whose objects are the non-linear types and whose morphisms are $x : \sigma \vdash_{\mathcal{C}} t : \tau$, both modulo the corresponding equalities. The left adjoint on the objects is defined by F and the right adjoint by G. $\square$

In fact we obtain the stronger result below:

Theorem 38. *The typed calculus LNL provides an internal language for the symmetric monoidal adjunctions in* $^{\rightarrow}\boldsymbol{Sma}_{st}$*, i.e.* $\mathsf{Th}(LNL)$ *is equivalent to* $^{\rightarrow}\boldsymbol{Sma}_{st}$*.*

Proof. We define a functor C from Th(LNL) to $^{\rightarrow}\mathbf{Sma}_{st}$ by mapping any LNL-theory to the construction of the syntactic category used in Prop. 37. On morphisms, C takes any translation to the functor induced on the syntactic categories by mapping types and terms to their respective translations.

Conversely, the functor L from $^{\rightarrow}\mathbf{Sma}_{st}$ to Th(LNL) is defined by mapping any symmetric monoidal adjunction $F \dashv G : \mathbb{C} \to \mathbb{S}$ in $^{\rightarrow}\mathbf{Sma}_{st}$ to its internal language $L(F \dashv G)$ defined by extending LNL with:

1. new linear ground types A with the corresponding axiom $\vdash_{\mathcal{L}} A : \mathsf{type}$ for each object of $\mathbb{S}$, i.e. we name the objects of $\mathbb{S}$ in the calculus and we extend the interpretation of linear types of LNL in $\mathbb{S}$ by interpreting the new names in the corresponding objects;

2. new intuitionistic ground types σ and $\vdash_{\mathcal{I}} \sigma : \mathsf{type}$ for each object of $\mathbb{C}$ by naming the objects of $\mathbb{C}$ in the same way as we did for $\mathbb{S}$;

3. new linear terms M and $\Gamma \mid \Delta \vdash_{\mathcal{L}} M : B$ for each morphism of $\mathbb{S}$ having the interpretation of the context $\Gamma \mid \Delta$ as domain and the interpretation of B as codomain;

4. new intuitionistic terms t and $\Gamma \vdash_{\mathcal{I}} t : \sigma$ naming each morphism of $\mathbb{C}$ having the interpretation of the context Γ as domain and the interpretation of σ as codomain;

5. new equality axioms between linear types $\vdash_{\mathcal{L}} A = B : \mathsf{type}$ if the interpretation of A is equal to that of B in $\mathbb{S}$;

6. new equality axioms between intuitionistic types $\vdash_{\mathcal{I}} \sigma = \tau : \mathsf{type}$ if the interpretation of σ is equal to that of τ in $\mathbb{C}$;

7. new equality axiom between linear terms $\Gamma \mid \Delta \vdash_{\mathcal{L}} M = N : B$ if the interpretation of $\Gamma \mid \Delta \vdash_{\mathcal{L}} M : B$ in $\mathbb{S}$ is equal to that of $\Gamma \mid \Delta \vdash_{\mathcal{L}} N : B$ by interpreting the new term symbols in the morphisms they name;

8. new equality axioms between intuitionistic terms $\Gamma \vdash_{\mathcal{I}} t = s : \sigma$ if the interpretation of $\Gamma \vdash_{\mathcal{I}} t : \sigma$ in $\mathbb{C}$ is equal to that of $\Gamma \vdash_{\mathcal{I}} s : \sigma$ by interpreting the new term symbols in the morphisms they name.

Clearly, any morphism in $^{\rightarrow}\mathbf{Sma}_{st}$ gives rise to a translation.
It is easy to check that the functors so defined establish an equivalence of categories. $\square$

Since symmetric monoidal adjunctions in $\mathbf{Sma}_{st}$ are sound and complete for DILL and given that the category of ILL-theories is equivalent to that of DILL-theories, one might expect that their classes of models, $\mathbf{Lin}$ ($\mathbf{Lin}_{st}$) and $\mathbf{Sma}$ ($\mathbf{Sma}_{st}$), are equivalent, too. But this is not true since the first is isomorphic to a proper subcategory of the latter.

However, in [5, 26] it is said that the two axiomatisations of categorical model for ILL, respectively in terms of linear category in def. 24 and in terms of symmetric monoidal adjunction in def. 32, or even in def. 33, are equivalent. The reason is that given any symmetric monoidal adjunction between a symmetric monoidal closed category and a cartesian *closed* category, one obtains a linear category on the SMC category. (Actually, Barber sharpened Benton's result by proving that one does not need to start from a symmetric monoidal adjunction between a SMC category and a cartesian closed category; a symmetric monoidal adjunction between a SMC category and a cartesian category will do.) Conversely, a linear category gives rise by definition to a symmetric monoidal adjunction of def. 32 by considering the Eilenberg-Moore adjunction associated to its comonad, or to a symmetric monoidal adjunction of def. 33 by considering an elaborated construction starting from the Kleisli adjunction associated to its comonad (see def. 43).

Hence, one is led to think that the models in terms of linear categories are equivalent to those in terms of symmetric monoidal adjunctions. Instead, when we consider not only linear categories and symmetric monoidal adjunctions in def. 32 but the whole corresponding categories $\mathbf{Lin}$ and $\mathbf{Sma}$, we can see that $\mathbf{Lin}$ is equivalent to a reflective subcategory of $\mathbf{Sma}$. A similar relation, but via a coreflection, can be proved between linear categories and symmetric monoidal adjunctions of def. 33 in the case they are based on a symmetric monoidal closed category with finite products to model ILL extended with additive conjunction, as we will see in section 3.5.

To show these results, we recall the following facts. If one considers the class of Eilenberg-Moore adjunctions one obtains a full subcategory $\mathbf{Adj}_{em}$ of $\mathbf{Adj}$. Similarly, one obtains a full subcategory $\mathbf{Adj}_k$ of $\mathbf{Adj}$ by considering the Kleisli adjunctions. Then, thanks to the strict notion of comonad morphisms in Definition 28, a comonad functor K can be lifted *both* to a functor between the Eilenberg-Moore categories *and* to a functor between the Kleisli categories. (With Street's general notion of comonad morphism in [29] one can lift a comonad functor K to a functor between the corresponding Eilenberg-Moore categories only.) We then obtain the following results:

Theorem 39 (Cmad-Isomorphisms). *We have the following isomorphisms of categories:*

(i) $\boldsymbol{Cmad}$ *is isomorphic to the subcategory* $\boldsymbol{Adj}_{em}$ *of* $\boldsymbol{Adj}$.

(ii) $\boldsymbol{Cmad}$ *is also isomorphic to the subcategory* $\boldsymbol{Adj}_k$ *of* $\boldsymbol{Adj}$.

Proof. We define the functor $\mathcal{I}_{em} : \mathbf{Cmad} \to \mathbf{Adj}_{em}$ by taking a comonad (T, ε, δ) on $\mathbb{B}$ to its Eilenberg-Moore construction $F^T \dashv G^T : \mathbb{B}^T \to \mathbb{B}$ and a $\mathbf{Cmad}$ functor $K : \mathbb{B} \to \mathbb{B}'$ to the morphism (K, K^*) (where K^* is defined by $K^*(A, a) = (KA, Ka)$ and $K^* f = Kf$). Similarly we define the functor $\mathcal{I}_k : \mathbf{Cmad} \to \mathbf{Adj}_k$ by mapping a comonad (T, ε, δ) on $\mathbb{B}$ to its Kleisli construction $F_T \dashv G_T : \mathbb{B}_T \to \mathbb{B}$ and a $\mathbf{Cmad}$ functor $K : \mathbb{B} \to \mathbb{B}'$ to the morphism (K, K_*) (where K_* is defined by $K^*(TA, \delta_A) = (KTA, K\delta_A)$ and $K_* f = Kf$).

Now, observe that if S is a comonad on $\mathbb{B}$ and T is a comonad on $\mathbb{B}'$, any $\mathbf{Adj}$ morphism (K, H) from $F^S \dashv G^S$ to $F^T \dashv G^T$ has the form (K, K^*) (since the left adjoint of the Eilenberg-Moore construction for a comonad is the forgetful functor, the commutativity required in the definition of $\mathbf{Adj}$ morphism and the preservation of the bijections defining the adjunctions forces H to coincide with K^*). Similarly, any **Adj** morphism (K, H) from $F_S \dashv G_S$ to $F_T \dashv G_T$ has the form (K, K_*).

Therefore, $\mathcal{I}_{em}$ and $\mathcal{I}_k$ are the inverse functors of the restrictions to $\mathbf{Adj}_{em}$ and $\mathbf{Adj}_k$, respectively, of the functor $U : \mathbf{Adj} \to \mathbf{Cmad}$ that assigns to an adjunction the comonad it defines and to an adjunction morphism its first component. □

The following result relates the category $\mathbf{Adj}_{em}$ and $\mathbf{Adj}_k$ to **Adj**. Before stating it, we recall that a reflection (coreflection) is an adjunction $R \dashv I$ ($I \dashv R$) where I is the embedding functor of a subcategory into the corresponding category.

Theorem 40 (Reflection/Coreflection). *Given an adjunction $F \dashv G : \mathbb{A} \to \mathbb{B}$ with unit η and counit ε and the induced comonad $(T = FG, \epsilon, \delta = F\eta G)$:*

*(i) the identity on $\mathbb{B}$ together with the Eilenberg-Moore comparison functor $\mathbb{A} \to \mathbb{B}^T$ is the unit of a reflection $\mathcal{R}_{em} \dashv I$ of **Adj** in $\mathbf{Adj}_{em}$;*

*(ii) the identity on $\mathbb{B}$ together with the Kleisli comparison functor $\mathbb{B}_T \to \mathbb{A}$ is the counit of a coreflection $I \dashv \mathcal{R}_k$ of **Adj** in $\mathbf{Adj}_k$.*

Proof. In the Eilenberg-Moore case one shows that the comparison functor $R : \mathbb{A} \to \mathbb{B}^T$ together with the identity on $\mathbb{B}$ defines a universal arrow from $F \dashv G$ to $\mathcal{I}_{em}(U(F \dashv G))$ in **Adj**. If (K, H) is any **Adj**-map from $F \dashv G$ to $F^S \dashv G^S : \mathbb{C}^S \to \mathbb{C}$ (that is in an object of $\mathbf{Adj}_{em}$), then (K, K^*) is the unique map from $F^T \dashv G^T$ to $F^S \dashv G^S$ such that $(K, K^*) \circ (Id, R) = (K, H)$ (since Rc is defined to be $(Fc, F\eta_c)$, hence $K^*Rc = (KFc, KF\eta_c)$, which can be seen to coincide with Hc by using the assumption $KF = F^S H$ and the fact that (K, H) preserves the natural isomorphisms defining the adjunctions). □

We can combine the last two theorems to either reflect or coreflect **Adj** in **Cmad**. As an application of these theorems we get

Theorem 41. *The category **Lin** ($\mathbf{Lin}_{st}$) is isomorphic to the full subcategory $\mathbf{Sma}^{em}$ ($\mathbf{Sma}^{em}_{st}$) of **Sma** ($\mathbf{Sma}_{st}$) with symmetric monoidal Eilenberg-Moore adjunctions in **Sma**, and the category $\mathbf{Sma}^{em}$ is a full reflective subcategory of **Sma**.*

$$\begin{tikzcd} \mathbf{Sma} \arrow[r, shift right, "R"'] \arrow[r, phantom, "\top"] & \mathbf{Sma^{em}} \arrow[l, shift right, "I"'] \arrow[r, "\simeq"] & \mathbf{Lin} \end{tikzcd}$$

Proof. The proof is based on Theorems 39 and 40 and on the fact that the Eilenberg-Moore adjunction related to the comonad of a linear category is symmetric monoidal and lives in **Sma**, that linear morphisms satisfy lemma 30 and that the Eilenberg-Moore comparison functor preserves products, which follows from the fact that the left adjoint F of a symmetric monoidal adjunction $F \dashv G$ in **Sma** is strong monoidal. □

As a consequence we get that also $\mathbf{Lin}_{st}$ and $\mathbf{Sma}_{st}$ are not equivalent, the first being isomorphic to a proper subcategory of the second. We deduce that *the typed calculus DILL does not provide an internal language for symmetric monoidal adjunctions in $\mathbf{Sma}_{st}$*. Indeed, if DILL provided an internal language for symmetric monoidal adjunctions in $\mathbf{Sma}_{st}$, then we would get that their corresponding categories of models $\mathbf{Sma}_{st}$ and $\mathbf{Lin}_{st}$ are equivalent, too, by Theorem 31, since $\mathsf{Th}(\mathrm{DILL})$ is equivalent to $\mathsf{Th}(\mathrm{ILL})$ by Theorem 18.

The reason why DILL does not provide an internal language for symmetric monoidal adjunctions in $\mathbf{Sma}_{st}$ is that the cartesian category of a symmetric monoidal adjunction in $\mathbf{Sma}_{st}$ may have objects and morphisms which are not in the domain of the functor F and no DILL-theory can provide the syntax corresponding to those objects and morphisms.

However, an internal language for such symmetric monoidal adjunctions is provided by a fragment of LNL without the intuitionistic implication in the intuitionistic part, namely $^{\not\to}$LNL:

Theorem 42. *The typed calculus $^{\not\to}$LNL provides an internal language for symmetric monoidal adjunctions in $\boldsymbol{Sma}_{st}$, i.e.* $\mathsf{Th}(^{\not\to}LNL)$ *is equivalent to $\boldsymbol{Sma}_{st}$.*

Proof. The proof of this is completely similar to that of Theorem 38. □

Of course, as Th(DILL) is equivalent to Th(ILL), one could observe that we already know that the correct models of DILL satisfying the internal language theorem are those in **Lin** as one can deduce thanks to Theorem 31.

However, remember that we did not give a proof of Theorem 31. The reason is that we find easier to show that DILL provides an internal language of suitable symmetric monoidal adjunctions whose category is equivalent to **Lin**. These symmetric monoidal adjunctions are the objects of $^{\to}$**Sma** that best approximate the Kleisli adjunction related to a monoidal comonad on a symmetric monoidal closed category. Indeed, the Kleisli category associated to a monoidal comonad on a symmetric monoidal closed category is not in general closed under products. However, in [5] it is shown that given a linear category $\mathbb{S}$, we can obtain a symmetric monoidal adjunction $F^{!p} \dashv G^{!p}$, with $F^{!p} : \mathbb{S}^p_! \to \mathbb{S}$, living in $^{\to}$**Sma**, since the category $S^p_!$, which is the closure under finite products in the Eilenberg-Moore category $\mathbb{S}^!$ of the Kleisli category, is cartesian *closed*. Here is the precise definition:

Definition 43. *Given a linear category, we define the category $\mathbb{S}^p_!$ (that is $\mathbb{S}^*_!$ in [5]). Its objects are finite lists of free coalgebras related to the comonad $!$ of $\mathbb{S}$ and its morphisms are the coalgebras morphisms between the finite products of the respective lists of free coalgebras in the Eilenberg-Moore category $\mathbb{S}^!$. Then, we define the* finite product symmetric monoidal adjunction *$F^{!p} \dashv G^{!p}$ with $F^{!p} : \mathbb{S}^p_! \to \mathbb{S}$ by essentially restricting the canonical Eilenberg-Moore adjunction: the functor $F^{!p} : \mathbb{S}^p_! \to \mathbb{S}$ sends a list of free coalgebras to the object obtained by applying $F^!$ to the corresponding finite product of the list of free coalgebras and it sends a morphism f in $\mathbb{S}^p_!$ to the morphism $F^!(f)$; the functor $G^{!p}$ is obtained by sending an object A of $\mathbb{S}$ to the list of only the free coalgebra with support A and by sending a morphism f of $\mathbb{S}$ to the coalgebra morphism $G^!(f)$.*

We now consider the following category of finite product symmetric monoidal adjunctions:

Definition 44. *$\boldsymbol{Sma}^{kp}$ ($\boldsymbol{Sma}^{kp}_{st}$) is the full subcategory of $\boldsymbol{Sma}$ ($\boldsymbol{Sma}_{st}$) having as objects the symmetric monoidal adjunctions $F^{!p} \dashv G^{!p}$ with $F^{!p} : \mathbb{S}^p_! \to \mathbb{S}$ and $\mathbb{S}$ a linear category as described in def. 43.*

Next we can show that DILL provides an internal language for finite product symmetric monoidal adjunctions:

Theorem 45. *The typed calculus DILL provides an internal language for symmetric monoidal adjunctions in $\boldsymbol{Sma}^{kp}_{st}$, i.e.* Th*(DILL) is equivalent to $\boldsymbol{Sma}^{kp}_{st}$.*

Proof. We define a functor C from Th(DILL) to $\mathbf{Sma}^{kp}_{st}$. Given any DILL-theory we consider the *syntactic* symmetric monoidal adjunction constructed in Barber's thesis [2, 1]. Note that the syntactic cartesian closed category of this adjunction is isomorphic to $\mathbb{S}^p_!$ with $\mathbb{S}$ being the category of linear types and linear terms with only linear contexts. To prove this, note that morphisms $!A \otimes !B \to C$ in $\mathbb{S}$ are in bijective correspondence with morphisms of $\mathbb{S}^p_!$ from the tensor of the free coalgebras on A and B to the free coalgebra on C. Hence, starting from the syntactic symmetric monoidal closed category, which is a linear category, we can build an object of $\mathbf{Sma}^{kp}_{st}$. Then, on morphisms C takes any translation to the functor induced on the syntactic categories by mapping types and terms to their respective translations.

Conversely, the functor L from $\mathbf{Sma}^{kp}_{st}$ to $\mathsf{Th}(\text{DILL})$ is defined by sending any symmetric monoidal adjunction $F \dashv G$ with $F \colon \mathbb{C} \to \mathbb{S}$ in $\mathbf{Sma}^{kp}_{st}$ to its internal language defined by extending DILL with:

1. new linear ground types A with the corresponding axiom $\vdash A : \mathsf{type}$ for each object of $\mathbb{S}$, i.e. we name the objects of $\mathbb{S}$ in the calculus and we extend the interpretation of linear types of DILL in $\mathbb{S}$ by interpreting the new names in the corresponding objects;

2. new linear terms M and $\Gamma \mid \Delta \vdash M : B$ naming each morphism of $\mathbb{S}$ having the interpretation of the context $\Gamma \mid \Delta$ as domain and the interpretation of B as codomain;

3. new equality axioms between linear types $\vdash A = B : \mathsf{type}$ if the interpretation of A is equal to that of B in $\mathbb{S}$;

4. new equality axioms between linear terms $\Gamma \mid \Delta \vdash M = N : B$ if the interpretation of $\Gamma \mid \Delta \vdash M : B$ in $\mathbb{S}$ is equal to that of $\Gamma \mid \Delta \vdash N : B$, by interpreting the new term symbols in the morphisms they name.

Then, any morphism in ${}^{\to}\mathbf{Sma}_{st}$ gives rise to a translation since a morphism in ${}^{\to}\mathbf{Sma}_{st}$ is determined by its first component.

It is easy to check that the functors so defined establish an equivalence of categories. $\square$

Now, we prove as expected that

Proposition 46. *The category of models of ILL, $\boldsymbol{Lin}_{st}$ is equivalent to the category of models of DILL, $\boldsymbol{Sma}^{kp}_{st}$.*

Proof. The object-part of the equivalence relies on the fact that from any comonad ! on a linear category $\mathbb{S}$ we can build a symmetric monoidal adjunction between $\mathbb{S}$ and $\mathbb{S}^p_!$. Note that a functor from $\mathbb{S}^p_!$ to $\mathbb{T}^p_!$ related to two SMC categories' $\mathbb{S}$ and $\mathbb{T}$ is uniquely determined by a SMC-functor from $\mathbb{S}$ to $\mathbb{T}$. $\square$

Remark 47. Now, thanks to Proposition 46, and Theorem 18 together with Theorem 45 we deduce *a proof of Theorem 31.*

Remark 48. Since $\mathsf{Th}(\text{DILL})$ is equivalent to $\mathsf{Th}(\text{ILL})$, one could think of avoiding DILL. But we chose to prove Theorem 45 and Theorem 46 directly to derive a proof of Theorem 31 since the proofs of soundness and completeness and of the internal language theorem for DILL are neater than those for ILL, as DILL is less verbose than ILL.

Inspired by our generic Theorems 39 and 40 and the fact that the models in $\mathbf{Sma}^{kp}_{st}$ are close to Kleisli adjunctions one could ask whether $\mathbf{Sma}^{kp}_{st}$ coreflects into $\mathbf{Sma}$. But this does not follow without the presence of categorical products in the linear category. The problem is one of coherence (we would need to consider only symmetric monoidal adjunctions $F \dashv G$ where F is strictly monoidal).

3.5 Adding categorical products

In this section we consider the linear type theories ${}_{\&}$ILL, ${}_{\&}$LNL, $\overset{\rightarrow}{\&}$LNL, and ${}_{\&}$DILL, with additive conjunction and its unit, and their respective models, which have categorical products added to the symmetric monoidal closed structure.

Models for ${}_{\&}$ILL are given by the categories in ${}_{\&}\mathbf{Lin}$.

Definition 49. *The category ${}_{\&}\boldsymbol{Lin}$ (${}_{\&}\boldsymbol{Lin}_{st}$) has as objects linear categories with finite products and as morphisms those $\boldsymbol{Lin}$-morphisms ($\boldsymbol{Lin}_{st}$-morphisms) that (strictly) preserve finite products.*

Models of ${}_{\&}$LNL are given by categories in $\overset{\rightarrow}{\&}\mathbf{Sma}$.

Definition 50. *The category $\overset{\rightarrow}{\&}\boldsymbol{Sma}$ ($\overset{\rightarrow}{\&}\boldsymbol{Sma}_{st}$) is defined as the subcategory of ${}^{\rightarrow}\boldsymbol{Sma}$ (${}^{\rightarrow}\boldsymbol{Sma}_{st}$) whose objects are symmetric monoidal adjunctions where the symmetric monoidal closed category has finite products and whose morphisms are ${}^{\rightarrow}\boldsymbol{Sma}$-morphisms (${}^{\rightarrow}\boldsymbol{Sma}_{st}$-morphisms) whose first component (strictly) preserves finite products.*

Models of $\overset{\rightarrow}{\&}$LNL are given by categories in ${}_{\&}\mathbf{Sma}$.

Definition 51. *The category ${}_{\&}\boldsymbol{Sma}$ (${}_{\&}\boldsymbol{Sma}_{st}$) is defined as the subcategory of $\boldsymbol{Sma}$ ($\boldsymbol{Sma}_{st}$) whose objects are symmetric monoidal adjunctions where the symmetric monoidal closed category has finite products and whose morphisms are ${}^{\rightarrow}\boldsymbol{Sma}$-morphisms (${}^{\rightarrow}\boldsymbol{Sma}_{st}$-morphisms) whose first component (strictly) preserves finite products.*

Finally, models for ${}_{\&}$DILL are now simply given by some Kleisli adjunctions and they are much simpler to define than those in Definition 43. The reason is that a linear category with products now yields a Kleisli category associated to its monoidal comonad that has products and actually becomes a cartesian closed category.

Definition 52. *The category ${}_{\&}\boldsymbol{Sma}^{k}$ (${}_{\&}\boldsymbol{Sma}^{k}_{st}$) is the subcategory of $\boldsymbol{Adj}_{k}$ which is also a full subcategory of $\overset{\rightarrow}{\&}\boldsymbol{Sma}$, that is the objects of ${}_{\&}\boldsymbol{Sma}^{k}$ (${}_{\&}\boldsymbol{Sma}^{k}_{st}$) are Kleisli adjunctions related to a monoidal comonad on a symmetric monoidal closed category with finite products – whose associated Kleisli category happens to be cartesian closed – and the morphisms of ${}_{\&}\boldsymbol{Sma}^{k}$ (${}_{\&}\boldsymbol{Sma}^{k}_{st}$) between $F_! \dashv G_!$ with $F_! : \mathbb{S}_! \to \mathbb{S}$ and $F'_! \dashv G'_!$ with $F'_! : \mathbb{S}'_! \to \mathbb{S}'$ are transformations of adjoints (K, H) where $K : \mathbb{S} \to \mathbb{S}'$ is a strong (respectively strict) symmetric monoidal closed functor (strictly) preserving finite products and H is a (respectively strict) cartesian closed functor and (K, H) is* monoidal *as in Definition 35.*

We can then prove that

Proposition 53 (Adding products).

1. *The typed calculus ${}_{\&}$ILL provides an internal language for categories in ${}_{\&}\boldsymbol{Lin}_{st}$, i.e. $\mathsf{Th}({}_{\&}ILL)$ is equivalent to ${}_{\&}\boldsymbol{Lin}_{st}$.*

2. *The typed calculus ${}_{\&}$LNL provides an internal language for symmetric monoidal adjunctions in $\overset{\rightarrow}{\&}\boldsymbol{Sma}_{st}$, i.e. $\mathsf{Th}({}_{\&}LNL)$ is equivalent to $\overset{\rightarrow}{\&}\boldsymbol{Sma}_{st}$.*

3. *The typed calculus $\overset{\rightarrow}{\&}$LNL provides an internal language for symmetric monoidal adjunctions in ${}_{\&}\boldsymbol{Sma}_{st}$, i.e. $\mathsf{Th}(\overset{\rightarrow}{\&}LNL)$ is equivalent to ${}_{\&}\boldsymbol{Sma}_{st}$.*

4. *The typed calculus ${}_{\&}$DILL provides an internal language for symmetric monoidal adjunctions in ${}_{\&}\boldsymbol{Sma}^{k}_{st}$, i.e. $\mathsf{Th}({}_{\&}DILL)$ is equivalent to ${}_{\&}\boldsymbol{Sma}^{k}_{st}$.*

3.5.1 Relating the models with categorical products

Again we can use our generic Theorems 39 and 40 to relate the models of ${}_{\&}$ILL,${}_{\&}$LNL, $\overset{\rightarrow}{\&}$LNL and ${}_{\&}$DILL.

We start by noticing that the category of models for ${}_{\&}$ILL is equivalent to that of models for ${}_{\&}$DILL (both with strict and non-strict maps). In the strict case the equivalence can be proved from the fact that the category of ${}_{\&}$DILL-theories and that of ${}_{\&}$ILL-theories are equivalent as proved in Theorem 18 after showing that the chosen models satisfy the internal language theorem.

However, this can also be proved directly:

Proposition 54. *The category ${}_{\&}\boldsymbol{Lin}$ (${}_{\&}\boldsymbol{Lin}_{st}$) of models of ${}_{\&}$ILL is equivalent to the category ${}_{\&}\boldsymbol{Sma}^{k}$ (respectively ${}_{\&}\boldsymbol{Sma}^{k}_{st}$) of models of ${}_{\&}$DILL.*

Proof. This is based on Theorem 39 and the fact that the Kleisli category $\mathbb{S}_{!}$ for the monoidal comonad ! on the SMC category $\mathbb{S}$ with finite products is cartesian closed. Note that any functor from $\mathbb{S}_{!}$ to $\mathbb{T}_{!}$ induced by a SMC cartesian functor from $\mathbb{S}$ to $\mathbb{T}$ in ${}_{\&}\mathbf{Lin}$ (${}_{\&}\mathbf{Lin}_{st}$) is also cartesian closed. ☐

Next, we prove the theorem corresponding to Theorem 41 to relate models of ${}_{\&}$ILL and models of $\overset{\rightarrow}{\&}$LNL:

Theorem 55. *The category ${}_{\&}\boldsymbol{Lin}$ is isomorphic to the full subcategory ${}_{\&}\boldsymbol{Sma}^{em}$ of ${}_{\&}\boldsymbol{Sma}$ with symmetric monoidal Eilenberg-Moore adjunctions in ${}_{\&}\boldsymbol{Sma}$, and the category ${}_{\&}\boldsymbol{Sma}^{em}$ is a reflective subcategory of ${}_{\&}\boldsymbol{Sma}$.*

Proof. The proof is completely similar to Theorem 41. ☐

Then, Theorem 40 specialises both to a coreflection between the (redefined) models of ${}_{\&}$DILL and models of $\overset{\rightarrow}{\&}$LNL and to a coreflection between the models of ${}_{\&}$DILL and models of ${}_{\&}$LNL (all with non-strict morphisms).

Theorem 56. *The category ${}_{\&}\boldsymbol{Sma}^{k}$ is a coreflective subcategory of ${}_{\&}\boldsymbol{Sma}$.*

Proof. The statement is an application of Theorem 40 after substituting ${}_{\&}\mathbf{Sma}$ for $\mathbf{Adj}$ and ${}_{\&}\mathbf{Sma}^{\mathrm{k}}$ for $\mathbf{Adj}_{k}$ and checking that the counit of the coreflection defined as in Theorem 40 is actually in ${}_{\&}\mathbf{Sma}$. Indeed, its component between the cartesian closed categories, which is the Kleisli comparison functor, is cartesian since G preserves finite products as a right-adjoint. For example, given $F \dashv G$ in ${}_{\&}\mathbf{Sma}$ on objects we have

$$K_{!}(A\&B) \equiv G(A\&B) \simeq G(A) \times G(B) \equiv K_{!}(A) \times K_{!}(B)$$

☐

Recalling that ${}_{\&}\mathbf{Sma}^{\mathrm{k}}$ is also a subcategory of $\overset{\rightarrow}{\&}\mathbf{Sma}$, we note that the coreflector of the previous proposition also preserves the intuitionistic function spaces and hence we get that:

Theorem 57. *The category ${}_{\&}\boldsymbol{Sma}^{k}$ is a coreflective subcategory of $\overset{\rightarrow}{\&}\boldsymbol{Sma}$.*

Proof. Following Theorem 40 it remains to check that the comparison functor is a counit morphism in $\overset{\rightarrow}{\&}\mathbf{Sma}$. Actually, based on Theorem 56 we just need to prove that the Kleisli comparison functor $K_! : \mathbb{S}_! \to \mathbb{C}$ preserves closures up to isomorphisms. Because of closure in $\mathbb{C}$

$$\mathsf{Hom}_{\mathbb{C}}(A, K_!(B) \to K_!(C)) \simeq \mathsf{Hom}_{\mathbb{C}}(A \times K_!(B), K_!(C))$$

and by the definition of the comparison functor and by the adjunction $F \dashv G$ we have

$$\mathsf{Hom}_{\mathbb{C}}(A \times K_!(B), K_!(C)) \equiv \mathsf{Hom}_{\mathbb{C}}(A \times G(B), G(C)) \simeq \mathsf{Hom}_{\mathbb{S}}(F(A \times G(B)), C).$$

Now, since F is strong monoidal and $\mathbb{S}$ closed we have

$$\mathsf{Hom}_{\mathbb{S}}(F(A \times G(B)), C) \simeq \mathsf{Hom}_{\mathbb{S}}(F(A) \otimes F(G(B)), C) \simeq \mathsf{Hom}_{\mathbb{S}}(F(A), F(G(B)) \multimap C)$$

and then, again by the adjunction $F \dashv G$ and the definition of the function space in $\mathbb{S}_!$ we get

$$\mathsf{Hom}_{\mathbb{S}}(F(A), F(G(B)) \multimap C) \simeq \mathsf{Hom}_{\mathbb{C}}(A, G(F(G(B)) \multimap C)) \equiv \mathsf{Hom}_{\mathbb{C}}(A, K_!(B \to C)).$$

Hence, we conclude that

$$\mathsf{Hom}_{\mathbb{C}}(A, K_!(B) \multimap K_!(C)) \simeq \mathsf{Hom}_{\mathbb{C}}(A, K_!(B \to C))$$

and since all these isomorphisms are natural in the first component we obtain what wanted, i.e.

$$K_!(B \to C) \simeq K_!(B) \to K_!(C).$$

Finally, we can easily check that the comparison functor preserves the application morphism by chasing diagrams and considering that G preserves products and that $F \dashv G$ is symmetric monoidal.

$\square$

This final theorem shows that models of ${}_{\&}$DILL can only be a subcategory of models of ${}_{\&}$LNL. Note that we cannot establish a similar comparison between models of LNL and models of DILL (ILL) via Eilenberg-Moore adjunctions, since the category of coalgebras generally lacks closure, that is the function space of two coalgebras is not necessarily a coalgebra.

Finally, we summarise the relationships between the models considered in this subsection in the following picture:

$$\begin{tikzcd}
 & & {}_{\&}\mathbf{Sma} \arrow[ddll, shift left, "\dashv"'] \arrow[ddrr, shift left, "\dashv"] & & & \\
 & & {}^{\to}_{\&}\mathbf{Sma} \arrow[u, hook] \arrow[dll, shift left, "\dashv"] & & & \\
{}_{\&}\mathbf{Sma^k} \arrow[uurr, shift left] \arrow[urr, shift left] \arrow[rrrr, "\simeq"'] & & & & {}_{\&}\mathbf{Sma^{em}} \arrow[uull, shift left] \arrow[r, "\simeq"'] & \mathbf{Lin}
\end{tikzcd}$$

4 Conclusion

In this paper we surveyed and clarified the relationship between linear typed calculi and their models. We remarked that soundness and completeness of categorical models are not enough to uniquely identify the most desirable class of models. We have proposed to call *categorical models* for a typed calculus those categorical models for which the typed calculus provides an internal language. We also emphasised the importance of considering model morphisms that correspond to type-theoretic translations. Using this notion of model morphism we organised all the models discussed into categories and related them.

Checking the internal language theorems, we noticed that ILL and LNL do provide internal languages for their models already in the literature. Instead, the DILL models proposed in [2] do not have DILL-theories as their internal language but theories of a fragment of LNL.

Moreover, the study of the precise relationships between these models led us to a new concise definition of a linear category [25]. This facilitated our comparison via a reflection between the two kinds of sound and complete models of ILL given by linear categories and by specific symmetric monoidal adjunctions first shown sound and complete for DILL (and hence for ILL, as ILL is essentially equivalent to DILL) in [2, 1]. Then, we finished by proving that the models of DILL with additive conjunction expressed in terms of Kleisli symmetric monoidal adjunctions are coreflective in those of LNL with additive conjunction.

In conclusion, we have clarified the situation as far as the 'equivalence' between models of intuitionistic linear logic is considered, at least to allow us to generalise to second-order models. This paper has not investigated the relationship between models based on symmetric monoidal closed categories and (higher-dimension) models based on fibrations/indexed categories [24]. Also models for higher-order linear logics, as well as the relationship to models of other variants of linear logic (such as full intuitionistic linear logic, light linear logic, etc) are left to future work.

We have not considered Lafont's style of modelling ILL, advocated in [26], since it is not clear to us what kind of typed calculus could provide an internal language for these kinds of categories.

Finally, from a more categorical (less logical) perspective, it would be interesting to investigate the setting in which our generic Theorems 39 and 40 are valid if, instead of considering adjunctions and comonad morphisms that satisfy equalities, we consider them up to coherent isomorphisms.

References

[1] A. Barber. *Linear Type Theories, Semantics and Action Calculi.* PhD thesis, University of Edinburgh, 1997.

[2] A. Barber and G. Plotkin. Dual intuitionistic linear logic. Technical report, LFCS, University of Edinburgh, 1997.

[3] H. P. Barendregt. Lambda calculi with types. In *Handbook of logic in computer science*, volume 2 of *Oxford Sci. Publ.*, pages 117–309. Oxford Univ. Press, New York, 1992.

[4] M. Barr and C. Wells. *Category Theory for Computing Science.* Series in computer science. Prentice Hall International, New Your, 1990.

[5] N. Benton. A mixed linear and non-linear logic: Proofs, terms and models. In *Proceedings of Computer Science Logic '94, Kazimierz, Poland.* Lecture Notes in Computer Science No. 933, Berlin, Heidelberg, New York, 1995.

[6] N. Benton. A mixed linear and non-linear logic: Proofs, terms and models. Technical Report 352, University of Cambridge-Computer Laboratory., October 1994.

[7] N. Benton, G. Bierman, V. de Paiva, and M. Hyland. Linear λ-calculus and categorical models revisited. In *Computer science logic (San Miniato, 1992).*, volume 702 of *LNCS*, pages 75–90, 1993.

[8] N. Benton, G. Bierman, V. de Paiva, and M. Hyland. A term calculus for intuitionistic linear logic. In *Proc. of Typed Lambda Calculus and Applications.*, volume 664 of *Lecture Notes in Computer Science*, pages 75–90. Springer Verlag, 1993.

[9] G. Bierman. On intuitionistic linear logic. Technical Report 346, Computer Laboratory, University of Cambridge, August 1994. Ph.D. Thesis.

[10] G. Bierman. What is a categorical model of intuitionistic linear logic? In *Proc. of the Second International Conference on Typed Lambda Calculus and Applications.*, volume 902 of *Lecture Notes in Computer Science*. Springer Verlag, 1994.

[11] R. L. Crole. *Categories for types.* Cambridge Mathematical Textbooks. Cambridge University Press, 1993.

[12] N. Ghani. *Adjoint Rewriting.* PhD thesis, University of Edinburgh, 1995. Published as CST-122-95 and ECS-LFCS-95-339.

[13] N. Ghani, V. de Paiva, and E. Ritter. Linear Explicit Substitutions. *Journal of the IGPL*, 2000.

[14] J. Y. Girard and Y. Lafont. Linear logic and lazy computation. In *TAPSOFT '87*, volume 250 of *Lecture Notes in Computer Science*, pages 52–66, 1987.

[15] J. Y. Girard, Y. Lafont, and P. Taylor. *Proofs and types.*, volume 7 of *Cambridge tracts in Theoretical Computer Science.* Cambridge University Press, 1989.

[16] M. Hyland and A. Schalk. Abstract games for linear logic. Extended Abstract. In *CTCS '99*, volume 29 of *Electronic Notes in Theoretical Computer Science*, 1999.

[17] G. M. Kelly. Doctrinal adjunctions. In *Category Seminar '73*, volume 420 of *Lecture Notes in Mathematics*, pages 257–281, 1974.

[18] G. M. Kelly. *Basic concepts of enriched category theory*, volume 64 of *London Mathematical Society Lecture Note Series.* Cambridge University Press, 1982.

[19] T. W. Koh and C.-H. L. Ong. Explicit substitution internal languages for autonomous and *-autonomous categories (preliminary version). In *Proceedings of the 8th Conference on Category Theory and Computer Science*, volume 29, page 30 pages. Electronic Notes in Theoretical Computer Science, 1999.

[20] Y. Lafont. The linear abstract machine. In *International Joint Conference on Theory and Practice of Software Development. (Pisa, 1987)*, volume 59 of *Theoretical Computer Science*, pages 157–180, 1988.

[21] J. Lambek and P. J. Scott. *An introduction to higher order categorical logic.*, volume 7 of *Studies in Advanced Mathematics.* Cambridge University Press, 1986.

[22] S. Mac Lane. *Categories for the working mathematician*, volume 5 of *Graduate text in Mathematics.* Springer, 1971.

[23] I. Mackie, L. Roman, and S. Abramsky. An internal language for autonomous categories. *Applied Categorical Structures*, 1:311–343, 1993.

[24] M. E. Maietti, V. de Paiva, and E. Ritter. Categorical models for intuitionistic and linear type theory. In J. Tiuryn, editor, *Proc. of Foundations of Software Science and Computation Structures*, LNCS, 2000.

[25] P. Maneggia. *Models of Linear Polymorphism.* PhD thesis, The University of Birmingham, UK, 2004.

[26] P. A. Mellies. Categorical models of linear logic revisited. *submitted to Theoretical Computer Science*, 2002.

[27] A. M. Pitts. Categorical logic. In Oxford University Press, editor, *Logical Methods in Computer Science*, volume VI of *Handbook of Logic in Computer Science*, page To appear, 1995.

[28] E. Ritter and V. de Paiva. Short final summary of EPSRC-grant: The eXplicit linear abstract machine. University of Birmingham. http://www.cs.bham.ac.uk/research/xslam/xslam_summary.ps.gz, June 2000.

[29] R. Street. The formal theory of monads. *Journal of Pure and Applied Algebra*, 2:149–168, 1972.

[30] P. Taylor. *Practical Foundations of Mathematics*, volume 99 of *Cambridge studies in advanced mathematics*. Cambridge University Press, 1997.