# Theory and Applications of Linear Dependent Types
## First Year Report, DPhil

Matthijs Vákár

Oxford, UK, October 6, 2014

**Abstract**

This report outlines a proposed 2-year research programme towards the degree of Doctor of Philosophy, building on the exploratory research conducted by the author in the past year. We propose to pursue the following two related goals.

A) Explaining the flow of information in dependent type theory;

B) Generating and studying new models of type dependency motivated by various scientific disciplines, particularly models that are naturally linear in character.

In practice, this will be achieved by investigating the *theory and applications of linear dependent types*, from the angles of (i) syntax, (ii) abstract categorical semantics, and (iii) concrete relevant models. We have a particular interest in models motivated from computer science (e.g. coherence space semantics, game semantics, semantics for quantum computation) and geometry (e.g. (stable) homotopy theory, tangles). We believe we are in an excellent position to meet the challenge of this project, given the extensive expertise present in Oxford in the fields of game semantics, semantics for quantum computation, stable homotopy theory, and type theory, combined with the author's knowledge of categorical logic, quantum theory, and geometry.

# Contents

# 0 Introduction

Recently, there has been a surge of interest in dependent type theory in both the mathematics and computer science communities, as a result of the appeal of the homotopy model of the theory. Although a lot of progress seems to have been made in the mathematics community, for instance formalising a large part of homotopy theory in the formalism of type theory, from the point of view of computer science, several of the elementary analyses one would normally perform for a type theory are still missing for dependent types. Therefore, the time seems ripe for a deeper analysis of this flavour.

In particular, we believe it would be desireable to better understand the flow of information in dependent type theory, for instance by developing a game semantics. Closely related is the objective of giving a linear refinement of dependent type theory, in the same way that linear logic refines intuitionistic logic. Its proof nets, put differently: its model in terms of tangles, should provide a similar perspective on information flow as a game semantics. Moreover, various flavours of semantics that are of interest from the point of view of theoretical computer science are naturally linear in flavour: e.g. coherence space semantics, game semantics, and semantics for quantum computation. Furthermore, several fundamental topics in computer science that go beyond common mathematical practice, such as concurrency and mutable state, are best described in the more general world of linear types. With this in mind, a linear analysis should be one of the basic desiderata, if we are to obtain a thorough understanding of dependent types from a theoretical computer science perspective. Finally, it should be noted that such a linear analysis could well feed back into the work of the mathematical community by providing a better understanding of homotopy type theory, possibly through a model in stable homotopy theory.

Therefore, this report will sketch a 2-year research programme on the topic of linear dependent types, which should lead towards the degree of Doctor of Philosophy. We start with a literature review, covering among other things the syntax, abstract categorical semantics, and some of the relevant models for linear logic and dependent type theory, as well as a brief discussion of the existing work on combining linear and dependent types. Next, we outline our proposed research programme in more detail and describe how this should take the shape of a thesis to be submitted to the University of Oxford to qualify for the degree of Doctor of Philosophy. We end with a list of references.

# 1 Literature Review

## 1.1 Linear Logic

Linear logic was introduced by Girard in [1] as a resource sensitive refinement of intuitionistic logic, which was inspired by the structure present in certain models for system F. From a modern perspective, we can see the essence of linear logic, or rather that of its proof term calculus, the linear $\lambda$-calculus, to already be present in [2]. Put simply[1], the linear $\lambda$-calculus provides an internal language for symmetric monoidal closed categories in the same way that the (simply typed) $\lambda$-calculus does for cartesian closed categories. The system is resource sensitive in the sense that a possibly non-cartesian monoidal structure does not generally admit copying and deleting morphisms. This means that, in the corresponding logic or $\lambda$-calculus, we lose the structural rules of contraction and weakening, respectively. This results in an exposure of the frequency at which assumptions are used in proofs in logic gives us a better grip on complexity in the $\lambda$-calculus.

To be precise, the logic that is arises from this linear $\lambda$-calculus via a Curry-Howard correspondence is referred to as *(multiplicative) intuitionistic linear logic*. This system is strictly more general than the *(multiplicative-additive-exponential) classical linear logic* studied by Girard. This latter system differs from the former in three significant ways.

1. It admits a *classical* duality in the sense that there is a dualising object $\perp$ for the implication $\multimap$. At the same time it still admits a non-trivial term calculus. This is one of the historically surprising aspects of the system, in the light of the Joyal lemma (see e.g. [3]), which states that a cartesian closed category with a dualising object is a preorder.

2. It comes equipped with a comodality !, called the *exponential*, which recovers the structural rules.

3. It comes equipped with a additional notion of conjunction, called the *additive conjunction*, written &, as opposed to the multiplicative conjunction $\otimes$ from multiplicative intuitionistic linear logic. It represents an internal choice, rather than a simultaneous occurence of resources.

It will be the level of greater generality of (multiplicative) intuitionistic linear logic, including the more specific cases of systems à la Girard, that we think of when we refer to *linear logic*.

### 1.1.1 Syntax

A convenient flavour of ILTT to have in mind before we continue to our system of linear dependent types is the so-called *dual intuitionistic linear logic (DILL)* of [4]. The system is a conservative extension of intuitionistic linear logic. Its attractive features for us are its homogeneous treatment of all type formers in a natural deduction style and its separation of contexts into a linear and an intuitionistic region, which puts ! on equal footing with the other connectives. This separation of the context will be important in our extension to linear dependent types as types will only be allowed to depend on intuitionistic assumptions.

Briefly put, DILL is a propositional type theory in which contexts consist of a linear and an intuitionistic region. In the latter, the usual structural rules of contraction and weakening apply, while they do not in the former. It has several optional type formers with natural deduction rules: nullary ones ($I$, $\top$, 0), a unary one (!), and binary ones ($\otimes$, $\multimap$, &, $\oplus$). We will not say any more about the syntax, as [4] features an excellent treatment of both the logic and the corresponding proof term calculus.

### 1.1.2 Categorical Semantics

Intuitionistic linear logic admits a relatively simple, though not historically uncontroversial, sound and complete categorical semantics which we will describe here briefly. Our principal reference will be [4]. Some more background is provided in [5] and [6].

There are several notions of model in use that are equivalent, which only differ in their interpretation of !. For our purposes, the notion of a linear-non-linear model of [7] is the best fit.

[1] Here, we are ignoring some subtleties to do with the presence of conjunctions in the language, or the internal homs arising from a monoidal structure.

All systems interpret the structural core of intuitionistic linear logic by a symmetric multicategory $\mathcal{D}$. If we have $I$- and $\otimes$-types, the multicategory structure can be taken to arise from a symmetric monoidal structure and we can treat $\mathcal{D}$ as a category. $\multimap$-types correspond to closure of $\mathcal{D}$, as a multicategory (or as a symmetric monoidal category). $\top$- and &-types correspond to finite products in $\mathcal{D}$, while 0- and $\oplus$-types correspond to finite distributive coproducts. What remains is the interpretation of !. As a comodality, ! should be interpreted as a certain kind of comonad on $\mathcal{D}$. Now, the different notions of model in use differ precisely in the way they formulate the necessary and sufficient conditions on this comonad to obtain a system that is equivalent to the syntax. We will use the condition that it arises from a *linear-non-linear adjunction*, that is, a lax symmetric monoidal adjunction (i.e. an adjunction in the 2-category of symmetric monoidal categories and lax symmetric monoidal functors)

$$\begin{tikzcd} (\mathcal{C}, 1, \times) \arrow[r, shift left, "L"] \arrow[r, phantom, "\perp"] & (\mathcal{D}, I, \otimes) \arrow[l, shift left, "M"] \end{tikzcd}$$

to a cartesian monoidal category $\mathcal{C}$. An equivalent condition for the adjunction $L \dashv M$ to be lax symmetric monoidal is for the functor $L$ to be strong symmetric monoidal, in which case the symmetric oplax structure on $L$ transfers along the adjunction to a symmetric lax structure on $M$.

### 1.1.3 Models

In this section, we describe some of the references on models of linear logic that are of particular interest to us, as we plan to extend them to model linear dependent types.

**1.1.3.1 Coherence Spaces** Historically it was the model in *coherence spaces* that motivated Girard to formulate linear logic, this model still is very appealing due to its simplicity, while still illustrating many of the important aspects of multiplicative additive exponential classical linear logic. It can be thought to give a linear refinement of the stable domain theory of [8]. References include [9, 5].

We briefly recall some of the definitions in Girard's coherence space model of (classical) linear logic. The model is given by the category Coh of coherence spaces and cliques. Its objects are coherence spaces $(A, \coh_A)$ (or, graphs): a set $A$ of vertices, thought of as information tokens, with a reflexive relation $\coh_A$, called the coherence relation, thought of as a compatibility of two tokens. We shall write $\frown_A$ for the irreflexive part of $\coh_A$, $\smile_A$ for the complement of $\coh_A$, and $\asymp_A$ for the complement of $\frown_A$. Before we define the morphisms of the category, we describe a few operations on objects.

Given a coherence space $A$, we define its *linear negation* $A^\perp$ as the space with the same underlying set $A$ of vertices and coherence relation $\asymp_A$:

$$a \coh_{A^\perp} a' := \neg(a \frown_A a').$$

Given coherence spaces $A$ and $B$, we define their *multiplicative conjunction* $A \otimes B$ as having underlying set the product $A \times B$ of the underlying sets of $A$ and $B$ and coherence relation

$$(a, b) \coh_{A \otimes B} (a', b') := a \coh_A a' \wedge b \coh_B b'.$$

We can then define their *multiplicative disjunction* $A \parr B$, through De Morgan duality,

$$A \parr B := (A^\perp \otimes B^\perp)^\perp,$$

and their *(multiplicative) linear implication* $A \multimap B$,

$$A \multimap B := A^\perp \parr B.$$

(Explicitly, $(a, b) \frown_{A \parr B} (a', b') := a \frown_A a' \vee b \frown_B b'$ and $(a, b) \frown_{A \multimap B} (a', b') := a \coh_A a' \Rightarrow b \frown_B b'$.)

We can define their *additive disjunction* $A \oplus B$ as the disjoint union of coherence spaces, where never $a \coh_{A \oplus B} b$ if $a \in A$ and $b \in B$ and $\coh_{A \oplus B}$ restricts to $\coh_A$ and $\coh_B$, and we can define their *additive conjunction* $A \& B$, through De Morgan duality, as

$$A \& B := (A^\perp \oplus B^\perp)^\perp.$$

(Explicitly, always $a \frown_{A\&B} b$ for $a \in A$, $b \in B$ and $\frown_{A\&B}$ restricts to $\frown_A$ and $\frown_B$.)

The operations $\otimes$, $⅋$, $\oplus$, and $\&$ also have neutral elements which we shall denote by $I$, $\bot$, $0$, and $\top$, respectively. Indeed, $I = \bot = \{*\}$ and $0 = \top = \varnothing$ are easily seen to do the trick. (These identities between the units can be seen as degeneracies of this model of linear type theory, as they do not follow from the syntax.)

We now define the morphisms:

$$\mathrm{Coh}(A,B) := \mathrm{cliques}(A \multimap B),$$

where a clique $\sigma$ in $C$ is a subset $\sigma \subset C$ such that $c, c' \in \sigma \Rightarrow c \frown_C c'$. We think of $(a,b) \in \sigma \in \mathrm{cliques}(A \multimap B)$, as the information token $b$ being produced from the token $a$. We compose cliques as relations, which gives us the identity relations (which are cliques!) as the identities of our category.

We finally note that $(I, \otimes, \multimap)$ make Coh into a symmetric monoidal closed category, that $\top$ and $\&$ are our nullary and binary products, and that $0$ and $\oplus$ are our nullary and binary distributive coproducts. We note that we have obtained a model of intuitionistic linear logic with $I$-, $\otimes$-, $\multimap$-, $\top$-, $\&$-, $0$-, and $\oplus$-types. In fact, as $((-)^\bot)^\bot \cong \mathrm{id}_{\mathrm{Coh}}$, we even have a model of classical linear logic.

Recall that we have a monoidal adjunction between the category Stable of domains (taken here to mean Scott domains with pullbacks[2]) and continuous continuous stable functions and the category Coh of coherence spaces, $L \dashv M$:

$$(\mathrm{Stable}, 1, \times) \underset{M}{\overset{L}{\rightleftarrows}} (\mathrm{Coh}, I, \otimes).$$

$M$ takes the domain of cliques on objects and sends a clique $\sigma$ in $A \multimap B$ to the continuous stable function $d \mapsto \{b \mid \exists_{a \in d}(a,b) \in \sigma\}$. $L$ sends a domain $\Delta$ to the coherence space with set of vertices the compact elements of $\Delta$ and coherence relation $s \frown_{L\Delta} t := \exists_{u \in \Delta} s \le u \wedge t \le u$ and sends a continuous stable function $\Delta' \xrightarrow{f} \Delta$ to the clique $\{(x,y) \mid y \le f(x) \wedge \forall_{x' \le x} y \le f(x') \Rightarrow x = x'\}$.

Recall that we have the following bijection of homsets

$$\begin{array}{ccc}
\sigma & \longmapsto & d \mapsto \{c \mid \exists_{d' \le d}(d',c) \in \sigma\} \\
\mathrm{Coh}(L\Delta, C) & \underset{\text{trace}}{\overset{\text{fun}}{\underset{\cong}{\rightleftarrows}}} & \mathrm{Stable}(\Delta, MC) \\
\{(d,c) \mid c \in f(d) \wedge \forall_{d' \le d} c \in f(d') \Rightarrow d' = d\} & \longleftarrow\!\shortmid & f.
\end{array}$$

This induces a linear exponential comonad $! := LM$ on Coh. Explicitly, $!A$ has set of vertices $\mathrm{fin} - \mathrm{cliques}(A)$ and coherence relation

$$s \frown_{!A} s' := (s \cup s' \in !A).$$

This shows that our model Coh of classical linear logic also supports !-types.

**1.1.3.2 Games** Of particular importance are *game semantics*, or interpretations in categories of 2-player games and strategies, of various type theories, due to their close connection to the syntax, which often takes the form of full completeness, full abstraction, and definability results. There are two main branches of game semantics, the AJM-approach [10, 11] and the HO-approach [12, 13]. We will primarily be interested in the former, as it is technically less involved and has a closer connection with coherence semantics.

In a nutshell, what game semantics adds to the picture of coherence semantics is a time dimension, which takes the form of a partial order relation on information tokens: some tokens are still in the process of being determined and some are a possible extension in time of others. Although this is not the usual treatment of game semantics, [14] gives a detailed exposition of how the usual category of games admits a full and faithful embedding into a category of partially ordered coherence spaces. The intuition is that we get a new (finer) level of atom of description: moves. Even-length plays replace tokens, strategies replace

[2]If we want the category to be cartesian closed, we should take some full subcategory, like Berry's dI-domains.

cliques. Classical duality is interpreted by interchanging the roles of the players (player and opponent), while $\otimes$ is interpreted by playing two games simultaneously, where only opponent is allowed to switch between games. Finally, identity morphisms are interpreted by copycat-strategies and composition is interpreted by interaction of strategies. Good introductions to AJM-style game semantics can be found in [15] and [16] (Chapters 1-3).

**1.1.3.3 Modules, Databases, Probability Theory, and Quantum Mechanics** The author could not find suitable references for the following material. However, he is sure it can be considered common knowledge.

Let $R$ be a commutative rig (a ring without additive inverses). Write AbMon for the category of abelian monoids. Note that we can see $R$ as a AbMon-enriched category. Then, we can study $R$ through its enriched Yoneda embedding into the category $\mathrm{Mod}_R := \mathrm{AbMon}-\mathrm{Cat}(R, \mathrm{AbMon})$ of $R$-modules. Clearly, as a category of AbMon-valued presheaves, $\mathrm{Mod}_R$ has (distributive) biproducts and is symmetric monoidal closed with respect to the (non-cartesian[3]) Day-tensor structure arising from the usual symmetric monoidal structure on AbMon. Hence, we have a model of multiplicative additive intuitionistic linear logic. For the interpretation of exponentials, there are several possibilities. One rather boring option arises from the free-forgetful adjunction to Set. One can conceive of more interesting constructions relating to Fock space in quantum mechanics, but it remains to be seen how generalisable these constructions are. See e.g. [17]. The connection with quantum mechanics, here, arises if we take $R$ to be $\mathbb{C}$, the complex numbers. In that case, $\mathrm{Mod}_{\mathbb{C}}$ is equivalent to $\mathrm{FreeMod}_{\mathbb{C}} = \mathrm{Vect}_{\mathbb{C}}$, as $\mathbb{C}$, being a field, does not have any non-trivial quotients. More generally, the categories $\mathrm{FreeMod}_R$ of free $R$-modules and $\mathrm{ProjMod}_R$ of projective $R$-modules are also easily seen to give submodels of multiplicative additive intuitionistic linear logic. This includes the category $\mathrm{FreeMod}_{\mathbb{B}} = \mathrm{Rel}$ of sets and relations, where $\mathbb{B}$ is the rig of Booleans, which can be seen as a setting for (relational) database theory. Other interesting examples arise when we use the rig of positive reals, the tropical rig, or provenance rigs (also called provenance semirings).

We can also fit various kinds of probability theory (e.g. possibility theory, ordinary probability theory, signed probability theory) into this framework. For this purpose, we first note that we can give the following alternative presentation of the category $\mathrm{f.g.}-\mathrm{FreeMod}_R$ of finitely generated free $R$-modules. Let us write $\mathrm{Rel}_R$ for this presentation, before we note it to be equivalent to $\mathrm{f.g.}-\mathrm{FreeMod}_R$. It has finite sets - let us even work with a skeleton, and take numerals - as objects and a morphism

$$X \xrightarrow{A} Y$$

will be a map

$$X \times Y \xrightarrow{\widetilde{A}} R.$$

We compose morphisms according to

$$X \xrightarrow{A} Y \xrightarrow{A'} Z$$

as

$$X \times Z \xrightarrow{\widetilde{A' \circ A}} R$$

$$\widetilde{A' \circ A}(x,z) = \sum_{y \in Y} A(x,y) \cdot A'(y,z).$$

This category has a simple symmetric monoidal structure $(\otimes, I) = (\times, \{0\})$. On morphisms we define $\widetilde{A \otimes A'} = \widetilde{A} \cdot \widetilde{A'}$. Of course, $R$ lives in this category as the endomorphisms of the unit. We call $\mathrm{Rel}_R(\{*\}, X) = R^X$ the set of states of $X$. We can embed $\mathrm{Rel}_R$ into $\mathrm{Mod}_R$ as follows[4]:

$$\mathrm{Rel}_R \xrightarrow{R^{(-)}} \mathrm{Mod}_R$$

$$X \xrightarrow{A} Y \longmapsto R^X \xrightarrow{R^A} R^Y,$$

[3]In the case of $R = \mathbb{C}$, this will give us the existence of entangled states in quantum mechanics.

[4]We identify an object with its states and a morphism with its action on states. The fact that this defines an embedding justifies the name 'state'.

where $R^A(A')$ acts by post-composition with $A$ when we interpret $R^X$ as $\mathrm{Rel}_R(\{*\}, X)$. We note that this is a full embedding of monoidal categories that identifies $\mathrm{Rel}_R$ with a skeleton of f.g. $-\,\mathrm{FreeMod}_R$: we obtain a pseudo-inverse to the inclusion $\mathrm{Rel}_R \subset$ f.g. $-\,\mathrm{FreeMod}_R$ by choosing (i.e. relying on AC) generators for each module. Note that, for $R = \mathbb{C}$ , $\mathrm{Rel}_{\mathbb{C}}$ should therefore be viewed as the category of finite dimensional Hilbert spaces and linear maps.

Now, we can define $\mathrm{Prob}_R$ to be the non-full subcategory of $\mathrm{Rel}_R$ defined by the restriction on morphisms $X \xrightarrow{A} Y$ that

$$\forall_{x \in X} \quad \sum_{y \in Y} \tilde{A}(x, y) = 1.$$

We can note that $\mathrm{Prob}_R$ inherits the symmetric monoidal structure. We have monoidal closure if $\Sigma_{1 \leq k \leq n} 1$ is (multiplicatively) invertible in $R$ for all $n \in \mathbb{N}$.

Finally, we make a remark about certain extra degeneracies that $\mathrm{Rel}_R$ has. Remembering the elementary fact from commutative algebra that, for a finitely generated free[5] module $M$, $M^{**} \cong M$, where we write $M^*$ for $M \multimap R$. This means not only that $\mathrm{Rel}_R$ is a model of classical linear logic, but it is saying more, in fact: the dualising object is the tensor unit. Actually, an even stronger statement is true: we have a compact closed category, meaning that the natural morphism $M^* \otimes N^* \longrightarrow (M \otimes N)^*$ we obtain is an iso. In fact, $\mathrm{Rel}_R$ also has duals for morphisms. These are easily seen to arise if we start out with a $*$-rig $R$, i.e. a rig $R$ equipped with an involution $R \xrightarrow{*} R$. Examples include the booleans and reals (with the trivial involution) and the complexes with conjugation. Then, for $X \xrightarrow{A} Y$, we can define $Y \xrightarrow{A^\dagger} X$ by $\widetilde{A^\dagger} := Y \times X \xrightarrow{\mathrm{swap}} X \times Y \xrightarrow{\tilde{A}} R \xrightarrow{*} R$. The two notions of duality are easily seen to be compatible so we have, in fact, a dagger compact closed category.

A good account on the relation between dagger compact closed categories and quantum computation can be found in [18, 19]. An exposition on how to deal with mixed states and open quantum systems in the same framework is given in [20].

**1.1.3.4 Pointed Sets and Stable Homotopy Theory** Another important exemplary class of models for linear type theory arises when we consider pointed objects in various categories (which we may want to think of as algebras over the maybe monad $X \mapsto X + 1$). We will discuss the case of the category $\mathrm{Set}_*$ of pointed sets first, but the construction can, in fact, be generalised to the category of pointed objects (i.e. $1/\mathcal{C}$) in an arbitrary symmetric monoidal closed category $\mathcal{C}$ with finite limits and colimits, i.e. a model of multiplicative additive intuitionistic linear logic. [21] Many of the observations here will be of value when thinking of the more involved situation of spectra of stable homotopy theory. In particular, we will see that we do not always end up with a cartesian monoidal structure after applying the construction $1/-$, even if we start out with one. In the light of the previous section, another suggestive way of looking at this example might be to view pointed sets as modules over the field with one element.

An interesting feature of the pointed set model $\mathrm{Set}_*$ of linear logic is that we have a natural notion of internal hom $\multimap$ (the set of basepoint-respecting functions, pointed by the function that is constantly the basepoint in the codomain), while the appropriate tensor structure is not a-priori clear. As a left adjoint to $\multimap$, this is uniquely determined, however, and it turns out to be the smash product, which we know from algebraic topology.

More generally, let $(\mathcal{C}, I, \otimes, \multimap)$ be a symmetric monoidal closed category with finite limits and colimits. Let $1 \xrightarrow{x} X$ and $1 \xrightarrow{y} Y$ be objects in $1/\mathcal{C}$. Then take the pullback

$$\begin{tikzcd}
X \multimap_* Y \arrow[r] \arrow[d] & 1 \arrow[d, "y"] \\
X \multimap Y \arrow[r, "x \multimap \mathrm{id}_Y"'] & 1 \multimap Y.
\end{tikzcd}$$

Now, $1 \xrightarrow{y \circ !_X} X \multimap Y$ defines a map $1 \xrightarrow{x \multimap_* y} X \multimap_* Y$. It turns out that this is an internal hom in $1/\mathcal{C}$. The corresponding monoidal structure - let's call it, again, a smash product and let's write it $\wedge$ - can be

[5] We could also have said projective modules here.

expressed as the pushout

$$\begin{tikzcd}
(X \otimes 1) + (1 \otimes Y) \arrow[r] \arrow[d, "\mathrm{id}_X \otimes y + x \otimes \mathrm{id}_Y"'] & 1 \arrow[d, "x \wedge y"] \\
X \otimes Y \arrow[r] & X \wedge Y.
\end{tikzcd}$$

A good survey of stable homotopy theory can be found in [22]. As in many examples, we have a candidate for the linear exponential !: the infinite loop space $\Omega^\infty$, followed by the infinite suspension spectrum $\Sigma^\infty$. Curiously, it appears to be rather the inverse composition $\Omega^\infty\Sigma^\infty$, which is closely related to the Goodwillie exponential, that is commonly studied by topologists. Seeing that linear dependent type theory has a canonical candidate for !, we are hoping this will clarify some things.

**1.1.3.5 Tangles** Recently, there has been a lot of interest in intuitive presentations of initial models of various type theories, in terms of simple geometric objects called tangles (or graphical calculi or string diagrams or diagrammatic languages). Roughly, tangles are generalisations of knots which may have open ended wires to arbitrary dimensions. One can think of Feynman diagrams [23], Penrose diagrams [24], string diagrams in category theory [25], Girard's proof nets for classical linear logic [26], the Baez-Dolan cobordism (which was recently resolved: [27]) and tangle hypotheses [28], and graphical calculi used in categorical quantum mechanics [29]. The idea seems to be quite robust and has been adapted to yield initial models for a whole range of different type theories. A good survey of some of the standard graphical calculi for various sorts of monoidal categories can be found in [30], while [31] gives a good survey of the early history of such diagrammatic languages.

In many ways, semantics in terms of 1-dimensional tangles is similar to game semantics. The wires in the tangles effectively are equivalent to copycat strategies and represent the flow of information in the type theory. It is therefore not surprising that both approaches to semantics have had similar success in precisely representing the syntax of type theories: wires/copycats keep track of how information flows, in the same way that variable names do in the syntax. Again, tangle semantics are naturally linear in character, unless we add copying and deleting wires (i.e. wires that split and end respectively).

**1.1.3.6 Linguistics** *Non-commutative* linear logic, or (free) non-symmetric monoidal categories, can be used to provide a framework for natural language syntax, essentially abstracting away from and rephrasing the basic ideas that were already present in the work of Panini in the fifth century BCA. This idea was perhaps first employed in [32]. More modern accounts of this idea can be found in [33, 34]. The essence of the project can be summarised in Lambek's crucial observation that logic is precisely distinguished from grammar by the presence of the three structural rules of exchange, contraction, and weakening.

One of the main advantages, therefore, of this approach to natural language is that it puts natural language syntax and semantics on equal footing: both can be described by sequent calculi, one with and one without structural rules. This idea of giving natural language semantics in terms of formal logic was first worked out in [35], although it must have existed for much longer. The syntax-semantics interface, in this framework, takes the form of a homomorphism between the two sequent calculi (or a monoidal functor between the corresponding categories of contexts): semantics is compositional.

Although this idea is very elegant, there is a large flaw in Montegue's work from a modern perspective: his semantics is about truth of sentences, rather than their meaning, which surely goes beyond mere truth! One can compare this with the proof theoretic point of view that logic surely is about more than truth: it might be better to identify a proposition with its type of proofs or some other denotation which is less simplistic than a truth value. Luckily, the syntax-semantics framework of a monoidal functor Syn $\longrightarrow$ Sem, where Syn and Sem are certain monoidal categories representing syntax and semantics, is expressive enough to accommodate much more than just truth value semantics. A good candidate for Sem comes from distributional semantics: the idea, which perhaps originated in [36], that a large part of the meaning of a word is the nothing more than the company it commonly keeps. One of the first computational ideas of this idea by representing words by vectors can be found in [37]. This was finally translated into the statement that a category of (finite dimensional) vector spaces or Hilbert spaces can be used for Sem (see [38, 39]).

One very interesting, perhaps coincidental, aspect of this idea is that the syntax-semantics relation in natural language is modeled in a way that strongly reminds us of a topological quantum field theory. Indeed, according to the cobordism hypothesis, both are in a sense monoidal functors to Hilbert spaces, from a certain free monoidal category with duals. Does this mean that cobordisms are the (commutative) syntax of a topological quantum field theory?

## 1.2 Dependent Type Theory

Dependent type theory is an extension of simple intuitionistic type theory in which types are allowed to depend on terms. It arose, historically, by extending the propositions-as-types correspondence to predicate logic, after [40] had established the result for propositional logic. This was done independently by [41] and [42]. Particularly well known is the system of [43], which is considerably more expressive than the earlier systems, through several new type formers, and is usually referred to as Martin-Löf type theory. A remarkable feature of this type theory is the combination of its strong-normalisation property with its expressive power, allowing a wide range of recursive functions to be defined. For this reason, it is often referred to as a programming logic.

### 1.2.1 Syntax

The structural core of dependent type theory (DTT) can easily be described as the system where we have

1. contexts and context morphisms;
2. a copy of a propositional intuitionistic type theory in each context and operations substituting context morphisms in terms *and types*, mapping between the copies of propositional type theory in various contexts;
3. an empty context $\cdot$ and an operation that allows us to extend a context $\Gamma$ by a variable declaration $x : A$ of a type $A$ in context $\Gamma$ such that context morphisms into $\Gamma, x : A$ are uniquely obtained as pairs of a context morphism into $\Gamma$ and a term of $A$.

Systems without feature 3. are also of interest, as they are suitable for expressing *external quantification* (and represent an internal language for strict indexed categories). Feature 3. is a crucial aspect of what we will call dependent type theory, however, as it will ensure the system's typical *internal* quantification (if we assume to have $\Sigma$- and/or $\Pi$-types). In presence of feature 3., we can give a practically equivalent formulation of the system without mentioning context morphisms, as these arise of lists of terms.

Various optional type formers are used in dependent type theory. One class is formed by the usual type formers from propositional type theory, which we can demand to exist in every context. New are the so-called $\Sigma$- and $\Pi$-*types*, which will be the type theoretic equivalents of existential and universal quantification, respectively, and the Id-*types*, which represent an internalisation of equality. A somewhat surprising new feature often studied in dependent type theory are so-called $W$-*types*, which can be thought of as a new kind of quantifier. They are types of well-founded trees and can be used to represent types defined through well-founded induction (that is, if we are in the extensional version of Martin-Löf type theory: see below). It is surprising to see that the dual notion, $M$-*types*, types of non-well-founded trees, which can be used to represent a wide range of types defined through coinduction, has hardly been studied. [44, 45] Another defining feature of Martin-Löf's system of dependent type theory are so-called *universes*, which play the role of *types of types*, which are usually demanded to be closed under various operations such as predicative or impredicative quantification. See e.g. [46, 47].

One important aspect of dependent type theory we'd like to touch on is the distinction between *extensional* and *intensional* type theory. We usually demand formation, introduction, elimination, computation, and, *possibly*, uniqueness rules for our various type formers. We (unambiguously) refer to type formers with a uniqueness rule as extensional type formers and sometimes, somewhat ambiguously, refer to the case without such a rule as intensional ones. Similarly, extensional Martin-Löf type theory refers to the system where all type formers are extensional. It turns out that, in the resulting system, type checking is undecidable, due to the uniqueness rule for Id-types. Therefore, there has been increased interest in the system without this rules, which is, again a bit ambiguously, referred to as intensional Martin-Löf type theory. This ambiguity is caused by the fact that, often, intensionality is not seen as the mere negation of extensionality. Rather, it is common to demand certain *intensionality criteria* of the system, replacing extensionality. See [48] for more details.

Recently, there has been a popular offshoot of Martin-Löf type theory, called homotopy type theory. [49] Inspired by the homotopy model of dependent type theory, which we will describe below, Voevodsky added one axioms, called the *univalence axiom*, to Martin-Löf type theory and observed that the resulting system reproducing much of homotopy theory. The univalence axiom is an axiom on a type-universe that forces the type of homotopy equivalences between types to be homotopy equivalent to the Id-type of the universe. It should be noted that $W$-types are not sufficient for defining types by well-founded induction in type theory with intensional Id-types. [50] Another indication that intensional Id-types should change our point of view on induction is the new possibility of *higher inductive types*, which are "freely" generated not only by certain of their terms but possibly also by some of the inhabitants of their iterated Id-types (see [49]).

The variant of the syntax that we are thinking of is described in [51]. For a more comprehensive account of the syntax of dependent type theory, we refer the reader to [46].

### 1.2.2 Categorical Semantics

There are various more or less equivalent flavours of semantics for the structural core of dependent type theory, all of which boil down to certain kinds of indexed or fibred categories: e.g. comprehension categories [52], categories with attributes [51]. It is the exact formulation of the conditions on these categories that differ between the various semantic frameworks. A good comparison in the language of fibred categories of the various semantic frameworks can be found in [52].

We should mention that the semantics in terms of locally cartesian closed categories [53, 54] is very elegant, but unsuitable for our purposes, as it is specific to the situation with extensional Id-types. The closest connection with the syntax in probably found in the categories with families of [46], which are strictly more general. In many ways, however, the yet more general comprehension categories of [52] provide more elegant descriptions of various type formers and are more suitable for a generalisation to linear dependent type theory. Within this setting, it is the so-called full strict comprehension categories that are in precise correspondence with the syntax. Therefore, it is this notion of semantics we shall use or, rather, a slight reformulation of it.

The idea will be that features 1. and 2. of the syntax tell us that our model should be a (preferably strict[6]) indexed cartesian multicategory $\mathcal{C}^{op} \xrightarrow{\mathcal{I}} \mathrm{CMultCat}$ or, if we assume to have 1- and $\times$-types, a strict indexed cartesian monoidal category $\mathcal{C}^{op} \xrightarrow{\mathcal{I}} \mathrm{CMCat}$. It is clear that the syntax consisting of 1. and 2. would be an internal language for such categories, where $\mathcal{C}$ acts as a category of context, $\mathcal{I}(\Delta)$ for $\Delta \in \mathrm{ob}(\mathcal{C})$ represents the propositional type theory in context $\Delta$, and $\mathcal{I}(f)$ for $f \in \mathrm{mor}(\mathcal{C})$ represents a substitution operation between the various type theories.

Feature 3. is easily seen to be equivalent to the so-called comprehension axiom on $\mathcal{I}$ together with the requirement that the comprehension functor is fully faithful (this last part reflects that we are reformulating *full* split comprehension categories). The comprehension axiom says that $\mathcal{C}$ should have a terminal object $\cdot$ and that for each $\Delta \in \mathrm{ob}(\mathcal{C})$ and $A \in \mathrm{ob}(\mathcal{I}(\Delta))$ a representation for the functor

$$x \mapsto \mathcal{I}(\mathrm{dom}(x))(1, A\{x\}) : (\mathcal{C}/\Delta)^{op} \longrightarrow \mathrm{Set}.$$

We will write its representing object $\Delta.A \xrightarrow{\mathbf{p}_{\Delta,A}} \Delta \in \mathrm{ob}(\mathcal{C}/\Delta)$, write $a \mapsto \langle f, a\rangle$ for the isomorphism $\mathcal{I}(\Delta')(1, A\{f\}) \cong \mathcal{C}/\Delta(f, \mathbf{p}_{\Delta,A})$, and write $\mathbf{v}_{\Delta,A} \in \mathcal{I}(\Delta.A)(1, A\{\mathbf{p}_{\Delta,A}\})$ for the universal element. It is a condition that says that we can build objects in our category of contexts $\mathcal{C}$ as lists of objects from the fibre categories and that morphisms into such objects arise as lists of morphisms in the fibres. The comprehension axiom defines an indexed functor $\mathcal{I} \longrightarrow \mathcal{C}/-$, defined in context $\Delta$ as

$$A \xrightarrow{f} B \qquad \longmapsto \qquad \mathbf{p}_{\Delta,A} \xrightarrow{\langle \mathbf{p}_{\Delta,A}, \mathcal{I}(\mathbf{p}_{\Delta,A})(f) \circ \mathbf{v}_{\Delta,A}\rangle} \mathbf{p}_{\Delta,B}.$$

[6] The strictness, here, reflects the fact that substitution in the syntax in a strict operation. We choose indexed categories over their fibrational equivalents as we feel they are closer to the syntax and are very workable in the strict case. Seeing that every indexed category is equivalent to a strict one, however, we can also use this more general notion of model. [55]

, which is demanded to be full and faithful. This last requirement represents the idea that context morphisms (into contexts that are formed by lists of types) are really nothing more than lists of terms.

In this framework, we can simply define the various type formers of propositional type theory in each context as expected: we demand them to be supported in each fibre category $\mathcal{I}(\Delta)$, with the caveat that the syntax dictates they also have to be preserved under substitution operations $\mathcal{I}(f)$.

What is particularly appealing about this framework is that we also have a simple characterisation of $\Sigma$-, $\Pi$-, and extensional Id-types. This goes back to ideas due to [56]. $\Sigma$-types are interpreted by left-adjoints to the substition operations[7] $\mathcal{I}(\mathbf{p}_{\Delta,A})$ together with an axiom called Frobenius reciprocity, which demands proper interaction with $\times$-types, and the usual condition of proper interaction with substitution, which takes the form of a Beck-Chevalley condition. Similarly, $\Pi$-types are interpreted by right adjoints to $\mathcal{I}(\mathbf{p}_{\Delta,A})$ satisfying a dual Beck-Chevalley condition. Extensional Id-types are interpreted as left-adjoints to $\mathcal{I}(\mathrm{diag}_{\Delta,A})$, where $\mathrm{diag}_{\Delta,\mathbf{p}_{\Delta,A}} := \langle \mathrm{id}_{\Delta.A}, \mathbf{v}_{\Delta,A}\rangle$, again satisfying a Beck-Chevalley condition. Stronger versions of $\Sigma$- and Id-types, with dependent or strong elimination rules, are often used. These have a slightly more intricate categorical semantics. A good survey of the semantics of type formers can be found in [57].

As usual, it is only the extensional versions of the various type formers that have a natural formulation in the categorical semantics. In particular, it is difficult to give a simple criterion that precisely captures intensional Id-types. The best we seem to be able to do is a literal translation of the syntactic rules, with which we shall you not bore the reader. Much more fascinating is the connection that has has recently come to light with homotopy theory, starting with [58]. It seems that intensional Id-types can precisely be characterised as certain weak factorisation systems [59, 60, 61, 62]. In particular, disregarding completeness properties for the moment, as they do not seem to have been sufficiently established[8], we have a sound interpretation[9] of intensional Id-types in any simplicial model category in which the cofibrations are precisely the monomorphisms [59], thus generalising Voevodsky's observation [64], which was made around the same time as Awodey and Warren's work (which followed a suggestion by Moerdijk), that simplicial sets with the standard Quillen model structure model intensional identity types. Similarly, there are fascinating connections between certain $(\infty,1)$-categories (presented by these model categories) and intensional Martin-Löf type theory. See e.g. [65, 66]. One concrete result is that intensional Martin-Löf type theories without a universe has a sound interpretation in any locally presentable locally cartesian closed $(\infty,1)$-category. This is shown in [67]: this type theory can be interpreted in what Shulman calls a type-theoretic model category. Moreover these model categories suffice to present all locally presentable locally cartesian closed $(\infty,1)$-categories. One of the pleasant features of this homotopical approach to the semantics of intensional type theory is that many of the type formers can be characterised either in the model categories or $(\infty,1)$-categories as categorical constructions up to homotopy.

It is well-known that various inductively defined types can be represented as initial algebras for endofunctors. In models dependent type theory with $\Sigma$- and $\Pi$-types, there is a natural class of endofunctors that is particularly well-behaved: so-called polynomial endofunctors. These are defined as the functors on the fibre categories $\mathcal{I}(\Delta)$ according to the following construction. Let $A \in \mathrm{ob}(\mathcal{I}(\Delta))$ and $B \in \mathrm{ob}(\mathcal{I}(\Delta.A))$. Then, we can define a functor

$$\mathcal{I}(\Delta) \xrightarrow{\mathcal{I}(\mathbf{p}_{\Delta,A} \circ \mathbf{p}_{\Delta.A,B})} \mathcal{I}(\Delta.A.B) \xrightarrow{\Pi_B} \mathcal{I}(\Delta.A) \xrightarrow{\Sigma_A} \mathcal{I}(\Delta).$$

It is sketched in [68] how their initial algebras precisely represent Martin-Löf's $W$-types, in the extensional type theory. A similar analysis of $M$-types as terminal coalgebras of polynomial endofunctors can be found in [45]. Higher inductive types still seem poorly understood. However, [69] suggests that they arise precisely as homotopy-initial algebras of presentable monads, in the same way that, according to [50], ordinary inductive types arise as homotopy-initial algebras of polynomial endofunctors or, equivalently, homotopy-initial (monad-)algebras of a corresponding algebraically free monad.

Finally, the matter of the semantics of universes needs to be discussed. Here, there is a big distinction between predicative and impredicative universes. For our purposes, it seems the predicative case is of

[7]In the case without feature 3., we can define $\Sigma_f \dashv \mathcal{I}(f)$ and similarly for $\Pi$-types.

[8]In fact, it appears that for a converse, we need to demand at least the prescence of certain higher inductive types in our type theory. See [63].

[9]Well, almost. A caveat is that the Beck-Chevalley condition for Id-types may not hold in the strict sense. See [59].

most value. In many ways, the categorical formulation of these is just a direct translation of the syntactic rules. It is worth noting however, that this lets us arrive at a concept that strongly resembles the concept of a universal fibration which naturally appears in homotopy theory and category theory and which is closely related to the concept of an object classifier in higher category theory (in which case a version of the univalence axiom is also expected to hold). A categorical analysis of various predicative universe constructions can be found in [70].

### 1.2.3 Models

**1.2.3.1 Families** One particularly simple well-known (see e.g. [57]) class of models of dependent type theory can be obtained from models of propositional type theory by cofreely adding type dependency. (See [71] for the statement about cofreeness.)

Suppose $\mathcal{V}$ is a cartesian monoidal category. We can then consider a strict Set-indexed category, defined through the following enriched Yoneda embedding $\mathrm{Fam}(\mathcal{V}) := \mathcal{V}^{-} := \mathrm{SMCat}(-, \mathcal{V})$:

$$\mathrm{Set}^{op} \xrightarrow{\mathrm{Fam}(\mathcal{V})} \mathrm{SMCat} \qquad\qquad S \xrightarrow{f} S' \longmapsto \mathcal{V}^{S} \xleftarrow{-\circ f} \mathcal{V}^{S'}.$$

Note that this definition naturally extends to a functor Fam. Now, it is easily verified that Fam satisfies the comprehension axiom, so it gives us a model of dependent type theory. By studying type formers in Fam, we can see that (when they exist) $\Sigma$-types effectively represent infinitary disjunctions while $\Pi$-types represent infinitary conjunctions.

**1.2.3.2 Domains** In the search for models for his partial type theory (essentially dependent type theory with fixed-point operators in each context), Martin-Löf constructed the following model of type dependency in a category of domains. The best reference on this model probably is [72].

Let Dom be the category of Scott domains with continuous functions[10]. The category of domains has finite products, given by $1 = \{*\}$ and $\times$, the Cartesian product in Set with lexicographic order. It is Cartesian closed, with the exponential $\Rightarrow$ given by the set of continuous functions, ordered pointwise.

If we let $\mathcal{U}$ be the (large) poset of small Scott domains where $\Delta \trianglelefteq \Delta'$ iff $D \subset D'$ and $\leq_{\Delta'}\big|_{\Delta} = \leq_D$. Then $\mathcal{U}$ is a Scott domain. We define $\mathrm{ob}\,\mathcal{I}(\Delta) := \mathrm{Dom}(\Delta, \mathcal{U})$ (sometimes called parametrisations). Note that $\mathrm{ob}\,\mathcal{I}(\Delta)$ inherits the pointwise operations 1, $\times$, and $\Rightarrow$, from Dom. For $F \in \mathrm{ob}\,\mathcal{I}(\Delta)$ we define the following two posets, which turn out to be Scott domains.

$$\Sigma_{x:\Delta} F(x) := \{(x, y) \mid x \in \Delta, y \in F(x)\}$$

$$(x, y) \leq_{\Sigma_{x:\Delta} F(x)} (x', y') := x \leq_{\Delta} x' \wedge y \leq_{F(x')} y';$$

$$\Pi_{x:\Delta} F(x) := \{\Delta \xrightarrow{f} \Sigma_{x:\Delta} F(x) \mid \mathrm{fst} \circ f = \mathrm{id}_{\Delta} \wedge \ f \text{ is continuous}\}$$

$$f \leq_{\Pi_{x:\Delta} F(x)} f' := \forall_{x \in \Delta} f(x) \leq_{F(x)} g(x).$$

We define $\mathcal{I}(\Delta)(F, G) := \Pi_{x:\Delta}(F \Rightarrow G)(x)$, making $\mathcal{I}(\Delta)$ into a Cartesian closed category. There are natural pullback operations, relating the fibres of $\mathcal{I}$: for $\Delta' \xrightarrow{f} \Delta$, we define $\mathcal{I}(\Delta) \xrightarrow{-\{f\}} \mathcal{I}(\Delta')$ by $(F \xrightarrow{\sigma} G)\{f\} := F \circ f \xrightarrow{\sigma \circ f} G \circ f$. This makes $\mathcal{I}$ into a strict indexed Cartesian closed category

$$\mathrm{Dom}^{op} \xrightarrow{\mathcal{I}} \mathrm{CMCCat}.$$

This turns out to be equipped with a comprehension (we write $\mathbf{p}_{\Delta,F}$ for $\Sigma_{x:\Delta} F(x) \xrightarrow{\mathrm{fst}} \Delta$)

$$\mathcal{I}(\Delta')(1, F\{f\}) := \Pi_{x':\Delta'} F(f(x')) \xrightarrow{\cong} \mathrm{Dom}/\Delta'(\mathrm{id}_{\Delta'}, \mathbf{p}_{\Delta', F \circ f}) \xrightarrow{\cong} \mathrm{Dom}/\Delta(f, \mathbf{p}_{\Delta,F}).$$

[10] Recall that a Scott domain is an algebraic directed complete bounded complete partial order

Now, $\Sigma$ and $\Pi$ actually turn out to represent $\Sigma$- and $\Pi$-types. Indeed, we can use them to construct left and right adjoints, $\Sigma_F$ and $\Pi_F$, respectively, to $-\{\mathbf{p}_{\Delta,F}\}$:

$$
\begin{tikzcd}
\mathcal{I}(\Sigma_{x:\Delta}F(x)) \arrow[rr, "\Sigma_F"] & & \mathcal{I}(\Delta) \\
G \arrow[d, "\sigma"'] \arrow[rr, mapsto] & & x \mapsto \Sigma_{y:F(x)}G(x,y) \arrow[d, "x \mapsto ((y,z)\mapsto (y,\sigma(x)(z)))"] \\
H & & x \mapsto \Sigma_{y:F(x)}H(x,y),
\end{tikzcd}
$$

which satisfies Frobenius reciprocity and the Beck-Chevalley condition,

$$
\begin{tikzcd}
\mathcal{I}(\Sigma_{x:\Delta}F(x)) \arrow[rr, "\Pi_F"] & & \mathcal{I}(\Delta) \\
G \arrow[d, "\sigma"'] \arrow[rr, mapsto] & & x \mapsto \Pi_{y:F(x)}G(x,y) \arrow[d, "x \mapsto (\sigma(x)\circ -)"] \\
H & & x \mapsto \Pi_{y:F(x)}H(x,y),
\end{tikzcd}
$$

which satisfies the dual Beck-Chevalley condition.

We also have (intensional) Id-types. For $F \in \operatorname{ob}\mathcal{I}(\Delta)$ and $f, f' \in \mathcal{I}(\Delta)(1,F)$ , we define $\mathrm{Id}_F(f,f') \in \operatorname{ob}\mathcal{I}(\Delta)$ as $x \mapsto \{y \in F(x) \mid y \leq f(x), f'(x)\}$ with the inherited order from $F(x)$ (F-rule). We define $\mathrm{refl}_f \in \mathcal{I}(\Delta)(1, \mathrm{Id}_F(f,f)$ as $x \mapsto f(x) \in \mathrm{Id}_F(f,f)(x) \subset F(x)$, which takes care of the I-rule. Finally, for the E-rule, we use the obvious extension by restriction.

We conclude that we have a model of dependent type theory with $\top$-, $\wedge$-, $\Rightarrow$-, $\Sigma$-, $\Pi$-, and (intensional) Id-types. We even have a type universe $\mathcal{U}$.

We note that the intensionality of the Id-types, here, comes from the fact that we really have a partial type theory. In case we restrict to the total elements, we will see that our Id-types become extensional. We will return to this issue later.

**1.2.3.3 Homotopy Theory** Starting with [58], which presents a model of intensional Martin-Löf type theory in the category of groupoids, a range of models have been studied in settings for homotopy theory. As mentioned earlier, it was observed independently by Voevodsky ([64]) and Awodey-Warren ([61]) that the model can be extended to simplicial sets, whose standard homotopy category is a common definition of the category of weak $\infty$-groupoids. Awodey-Warren's construction, in fact, is valid in any simplicial model category where the cofibrations are precisely the monomorphisms, thus generalises to a much larger range of homotopy theories.

An explicit description of the simplicial sets model of type theory can be found in [73]. We only briefly discuss the ideas behind the construction. The idea will be to let the usual Quillen model structure on simplicial sets dictate the definitions. $\mathcal{C}$ will be the usual category of simplicial sets. $\mathcal{I}(\Delta)$ will be the full subcategory of $\mathcal{C}/\Delta$ on the Kan fibrations. Now, we have $\Sigma$-types as fibrations are closed under composition. $\Pi$-types are interpreted by spaces of sections. Id-types are interpreted by path-spaces. We can construct a hierarchy of universes through universal Kan fibrations. Finally, [74] shows how $W$-types and $M$-types can be constructed. In fact, [69] shows that it models all higher inductive types.

In the end of the day, the hope is that a system of Martin-Löf type theory with a univalent universe and higher inductive types can replace traditional homotopy theory (or rather the generalised homotopy theory of suitable model categories). The hope is that it can serve as a cleaner constructive foundation of mathematics, replacing ZF(C). This perspective is pursued in [49].

## 1.3 Linear Dependent Type Theory

Although Girard's early work in linear logic (e.g. [1]) already talks about quantifiers, the analysis appears to have stayed rather superficial. In particular, an account of internal quantification, or a linear variant of Martin-Löf's type theory, was missing, let alone a Curry-Howard correspondence. Later, linear types and dependent types were first combined in [75], where a syntax was presented that extends LF (Logical Framework) with linear types (that depend on terms of intuitionistic types). This has given rise to a line of work in the computer science community. See e.g. [76, 77, 78]. All the work seems to be syntactic in nature, however.

On the other hand, very similar ideas, this time at the level of categorical semantics and specific models (coming from homotopy theory, algebra, and mathematical physics), have emerged in the mathematical community lately, perhaps independently so. Relevant literature is e.g. [79, 80, 81, 82].

The syntactic tradition seems to have stayed restricted to the situation of functional type theory, in which one has $\Pi$- and $\multimap$-types, while we are really interested in the general case of algebraic type theory, which is of fundamental importance from the point of view of mathematics where structures seldom admit internal homs. Simultaneously, in the syntactic traditions there seems to be a lack of other type formers (like $\Sigma$-, !-, and Id-types). The semantic tradition, however, does not seem to have given a sufficient account of the notion of comprehension and internal quantification.

It was the aim of [71] to close this gap between both lines of work. This technical report illustrates how linear and dependent types can be combined straightforwardly and in great generality, from a unified point of view combining syntax, semantic, and a class of models.

Recently, a different but roughly equivalent syntactic system was presented in [83]. Where [71] presented a denotational semantics for linear dependent types, [83] presents an operational semantics. Moreover, it extends the calculus with several features (computationally irrelevant quantification, proof irrelevance, and a monad of computations) in order to give a proof-theoretic account of imperative programming.

Where the syntax of [71] adds type dependency to Barber's DILL, [83] does the same for Benton's LNL-calculus. Where the former describes ! as a comodality directly, the latter decomposes it to an adjunction $L \dashv M$. An advantage of the syntax of [83] is that it provides a nice separation between the linear and the intuitionistic world where the type dependency lives entirely on the intuitionistic side. The syntax of [71], however, is more economical, puts ! on equal footing with the other connectives and hence is more homogeneous, and its in more smoothly in the landscape of commonly studied type theories.

### 1.3.1 Syntax

Our canonical reference will be [71]. If the reader is looking for metatheoretic results of the calculus, however, we recommend [75], which presents all the standard results for the $\top\& \multimap \Pi$-fragment of our calculus.

The syntax of intuitionistic linear dependent type theory (ILDTT) is a straightforward blend between dual intuitionistic linear logic of [4] and dependent type theory of, for instance, [46]. It is a system in which linear types are allowed to depend on intuitionistic assumptions, but not linear assumptions. To be precise, its structural core consists of the following elements:

1. intuitionistic contexts and context morphisms;
2. a copy of a propositional intuitionistic linear type theory in each context and operations substituting context morphisms in terms *and types*, mapping between the copies of propositional type theory in various contexts;
3. an empty intuitionistic context $\cdot$ and an operation that allows us to extend an intuitionistic context $\Delta$ by a variable declaration $x : A$ of a linear type $A$ in intuitionistic context $\Delta$ such that context morphisms into $\Delta, x : A$ are obtained from pairs of a context morphism into $\Delta$ and a term of $A$.

Again, one could consider studying systems without feature 3.. Again, in presence of feature 3., we can give a practically equivalent formulation of the system that does not mention context morphisms.

In each context, we can study the usual type formers from linear logic: $I, \otimes, \multimap, \top, \&, 0, \oplus$, and !. Moreover, we have equivalents of the connectives from dependent type theory. If $\Delta, x : A$ is an intuitionistic context and $B$ is a linear type in that context, we can form a linear types $\Sigma_{!x:!A}B$ and $\Pi_{!x:!A}B$ in context $\Delta$. These $\Sigma$- and $\Pi$-types should be thought of as dependent generalisations of $!(-) \otimes (-)$ and $!(-) \multimap (-)$, as that is what they reduce to in case $x$ is not free in $B$. They can be thought of as type theoretic equivalents of multiplicative quantifiers. In the same way, we can form a linear type $\mathrm{Id}_{!A}$ in context $\Delta, x : A, y : A$. This should be thought of as roughly the intuitionistic identity type. It can be seen to be a multiplicative connective, in a sense. An analysis of linear equivalent of $W$- and $M$-types is currently still missing. Generally, it seems that the (co)inductive types in linear type theory are a poorly understood topic.

### 1.3.2 Categorical Semantics

In [71], a categorical semantics is defined that has strong soundness and completeness properties with respect to the syntax (a stronger co-completeness result, in fact, than the comprehension categories semantics for dependent type theory). It can easily be adapted to yield an internal-language-type correspondence.

For the structural core, the idea is to take the semantics for dependent type theory, described in section 1.2.2 and to follow the usual adagio of linearisation: passing from cartesian monoidal structures to more general linear ones. One new feature, here, will be that the comprehension functor does not need to be full or faithful anymore as it represents (one half of) a !-modality. Indeed, our notion of a model of ILDTT will be the following reformulation of what one may describe as a split symmetric monoidal comprehension category (which may not be full!).

For modelling features 1. and 2., we have a strict indexed symmetric multicategory $\mathcal{C}^{op} \xrightarrow{\mathcal{L}} \mathrm{SMultCat}$. In case we have $I$- and $\otimes$-types, this comes from a strict indexed symmetric monoidal category $\mathcal{C}^{op} \xrightarrow{\mathcal{L}} \mathrm{SMCat}$. Feature 3. again boils down to a comprehension axiom: $\mathcal{C}$ has a terminal object $\cdot$ and for each $\Delta \in \mathrm{ob}(\mathcal{C})$ and $A \in \mathrm{ob}(\mathcal{L}(\Delta))$ a representation for the functor

$$x \mapsto \mathcal{L}(\mathrm{dom}(x))(I, A\{x\}) : (\mathcal{C}/\Delta)^{op} \longrightarrow \mathrm{Set}.$$

We will write its representing object[11] $\Delta.A \xrightarrow{\mathbf{p}_{\Delta,A}} \Delta \in \mathrm{ob}(\mathcal{C}/\Delta)$ and universal element $\mathbf{v}_{\Delta,A} \in \mathcal{L}(\Delta.A)(I, A\{\mathbf{p}_{\Delta,A}\})$. We will write $a \mapsto \langle f, a \rangle$ for the isomorphism $\mathcal{L}(\Delta')(I, A\{f\}) \cong \mathcal{C}/\Delta(f, \mathbf{p}_{\Delta,A})$.

Again, the type formers $I, \otimes, \multimap, \top, \&, 0,$ and $\oplus$ from linear logic get their usual semantics in the fibre categories $\mathcal{L}(\Delta)$ with the caveat that they also have to be preserved under the change of base functors $\mathcal{L}(f)$. $\Sigma$-, $\Pi$-, and (extensional) Id-types are precisely what would be expected based on the semantics of their intuitionistic equivalents: $\Sigma$-types are left adjoint to $\mathcal{L}(\mathbf{p}_{\Delta,A})$, satisfying a form of Frobenius reciprocity and Beck-Chevalley, $\Pi$-types are right adjoint to $\mathcal{L}(\mathbf{p}_{\Delta,A})$, satisfying a form of dual Beck-Chevalley, and (extensional) Id-types are left adjoint to $\mathcal{L}(\mathrm{diag}_{\Delta,A})$, where $\mathrm{diag}_{\Delta,\mathbf{p}_{\Delta,A}} := \langle \mathrm{id}_{\Delta.A}, \mathbf{v}_{\Delta,A} \rangle$, satisfying a Beck-Chevalley condition.

The semantics of !-types requires some thought. What is perhaps unexpected about linear dependent type theory, is that the context indexing determines a unique !-modality, while the choice of a !-modality was an extra degree of freedom in propositional linear type theory. This canonical modality is induced as follows. One way to view $!A$ as $\Sigma_{!A} I$, which we know to be uniquely determined by the categorical structure. In absence of $\Sigma$-types, there are other characterisations. In fact, $\mathcal{L}$ can be seen to induce a (non-strict) model $\mathcal{I} \subset \mathcal{C}/-$ of (intuitionistic) dependent type theory to which the comprehension scheme provides an indexed functor: $\mathcal{L} \xrightarrow{M} \mathcal{I}$. !-types are seen to be supported iff this has a strong monoidal indexed left adjoint $L$. Moreover, relationships can be established between the $\Sigma$-, $\Pi$-, and Id-types in $\mathcal{L}$ and $\mathcal{I}$. Among other things, the categorical semantics suggests generalisations of the Seely isomorphisms to $\Sigma$- and Id-types, introducing a notion of "additive" $\Sigma$- and Id-types that are related to their usual multiplicative counterparts through these isomorphisms. A further investigation of the situation from the point of view of syntax and concrete models is warranted, here.

The status of intensional Id-types is currently poorly understood in linear dependent type theory. This is primarily due to the fact that there does not appear to be a straightforward linear equivalent of the dependent elimination rules for $\Sigma$ and Id-types. This is due to our restriction that types are

[11] Really, $\Delta.MA \xrightarrow{\mathbf{p}_{\Delta,MA}} \Delta$ would be a better notation, where we think of $L \dashv M$ as an adjunction inducing !, but it would be very verbose.

only allowed to depend on intuitionistic assumptions. The lack of a dependent elimination rule for Id-types means that we cannot prove a propositional equivalent of the various uniqueness principles for type formers, which appears to be a crucial feature of homotopy type theory, adding some much-needed extensionality in the absence of a judgemental uniqueness principle. It is a topic of current investigation to overcome this restriction.

### 1.3.3 Models

**1.3.3.1 Families** The families models of dependent type theory transfer without problems to the linear setting. They do raise some interesting new phenomena. [71]

Suppose $\mathcal{V}$ is a symmetric monoidal category. We can then consider a strict Set-indexed category, defined through the following enriched Yoneda embedding $\mathrm{Fam}(\mathcal{V}) := \mathcal{V}^{-} := \mathrm{SMCat}(-, \mathcal{V})$:

$$\mathrm{Set}^{op} \xrightarrow{\mathrm{Fam}(\mathcal{V})} \mathrm{SMCat} \qquad\qquad S \xrightarrow{f} S' \longmapsto \mathcal{V}^{S} \xleftarrow{-\circ f} \mathcal{V}^{S'}.$$

Note that this definition naturally extends to a functor Fam. Now, it is easily verified that Fam satisfies the comprehension axiom, so it gives us a model of ILDTT. By studying type formers in Fam, we can see that (when they exist) $\Sigma$-types effectively represent infinitary equivalents of $\oplus$ while $\Pi$-types represent infinitary equivalents of &. This at first perhaps surprising interaction between additive and multiplicative connectives, in fact, also has an equivalent that can be proved in the syntax. For charactisations of other type formers, we refer the reader to [71].

**1.3.3.2 Coherence Spaces and Games** In work in progress with Samson Abramsky and Radha Jagadeesan, models of ILDTT have been constructed both in categories of coherence spaces and games. We are hoping this will lead to a satisfactory game semantics of dependent type theory.

# 2 Thesis Proposal

## 2.1 Topic and Fundamental Challenges

The suggested topic for this DPhil project and thesis is *the theory and applications of linear dependent types*. There are at least three *angles* from which one can approach linear dependent types: the study of...

SYN) an appropriate syntax;

SEM) abstract categorical semantics;

MOD) concrete models, in terms of e.g. coherence spaces, games, stable homotopy theory, and vector spaces, that can be related to various scientific disciplines like computer science, pure mathematics, and physics.

The weight of the thesis will be on the latter two, as those have, so far, remained practically unexplored in literature. That said, it will also present some new results on the first, that are motivated by the search for the connection with the latter two angles.

The research conducted by the author in the past year has shown that the question of how to combine type dependency and linearity is not a trivial one. An apparent (partial) incompatibility between these two concepts has been observed on all three levels: the *information-as-a-resource* interpretation of linear logic seems to be difficult to combine with type dependency, which seems to behave more according to the idea of *information-as-a-restriction*. This is a phenomenon that one, for instance, does not encounter when combining polymorphism and linearity. [84] It is precisely this limited compatibility, which we will illustrate more in section 2.3.2, that needs to be clarified further through a thorough investigation. This fundamental challenge is closely related to the question of how we should understand the flow of information in dependent type theory.

## 2.2 High-Level Description of Research Goals

This thesis has two closely related high-level research goals, one more theoretical and one more related to applications from various branches of science:

A) Explaining the *flow of information in dependent type theory*;

B) Generating *new models of type dependency* and understanding their properties, specifically models that are naturally linear in character and that are motivated from various scientific disciplines.

In practice, the paths to these two goals are expected to be deeply intertwined. Indeed, goal B), apart from being a goal in itself, will also serve as a means towards achieving goal A).

On a slightly lower level, these goals decompose into the following subgoals:

1. Exploring the extent to which linearity and type dependency can be combined;
2. Developing an appropriate syntax for linear dependent types;
3. Explaining the relationship between linear dependent type theory and (previously studied) linear predicate logic, where the hope is a sound and complete Curry-Howard-type result or its negation;
4. Developing a sound and complete abstract categorical semantics;
5. Analysing families models of linear type dependency;
6. Developing a string-diagram calculus;
7. Developing a coherence semantics for dependent type theory;
8. Developing a game semantics for dependent type theory, hopefully with definability and full abstraction results with respect to the syntax;

9. Studying some topics in theoretical computer science from the point of view of these models;
10. Investigating possible linear refinements to homotopy type theory;
11. Investigating models of linear dependent types in stable homotopy theory, which, hopefully relate to the previous;
12. Investigating models of type dependency related to quantum physics;
13. Investigating models of (non-commutative) linear type dependency for natural language syntax.

Again, many of these subgoals are closely related and they can be grouped and organised in several different ways. Based on these subgoals, the following structure is suggested for the thesis. The reader should keep in mind, though, that new avenues for research will no doubt arise in the next two years, while others might not turn out to be as fruitful as is currently expected.

1. Preliminaries - Linear Dependent Types?
   (a) What do we mean by linear dependent types?
   (b) Why should we care?
   (c) Challenges
   (d) Literature Review
2. Syntax of ILDTT
   (a) Presentation
   (b) Metatheory
   (c) Relationship to Linear Predicate Logic
3. Categorical Semantics of ILDTT
   (a) Structural Core
   (b) Semantic Type Formers
4. Models for Theoretical Computer Science
   (a) Type Dependency in Coherence Spaces
   (b) Game Semantics for Type Dependency
   (c) Linear Dependent Types and Topics in Computer Science
5. Models for Homotopy Theory
   (a) ILDTT and Stable Homotopy Theory
   (b) Linear Homotopy Type Theory?
6. Other Models
   (a) Families Models of ILDTT
   (b) Tangles: a String-Diagram Calculus for ILDTT
   (c) Quantum Models of Type Dependency
   (d) Linear Type Dependency in Linguistics
7. Discussion - Information Flow in Dependent Type Theory

Figure 1: Sketch of Table of Contents for thesis, as currently planned.

## 2.3 State of the Art

### 2.3.1 Limits of the Current Practice

Although some work has been done in the direction of combining linearity and type dependency, the author feels a satisfactory analysis has not been given.

Although Girard's early work in linear logic already talks about quantifiers, the analysis appears to have stayed rather superficial. In particular, an account of internal quantification, or a linear variant of Martin-Löf's type theory, was missing, let alone a Curry-Howard correspondence. Later, linear types and dependent types were first combined in [75], where a syntax was presented that extends LF (Logical Framework) with linear types (that depend on terms of intuitionistic types). This has given rise to a line of work in the computer science community. See e.g. [76, 77, 78]. All the work seems to be syntactic in nature, however.

On the other hand, very similar ideas, this time at the level of categorical semantics and specific models (coming from homotopy theory, algebra, and mathematical physics), have emerged in the mathematical community lately, perhaps independently so. Relevant literature is e.g. [79, 80, 81, 82].

Although, in the past months, some suggestions have been made on the nLab and nForum, of possible connections between both lines of work, no account of the correspondence was ever published, as far as the author is aware. Moreover, the syntactic tradition seems to have stayed restricted to the situation of functional type theory, in which one has $\Pi$- and $\multimap$-types, while we are really interested in the general case of algebraic type theory, which is of fundamental importance from the point of view of mathematics where structures seldom admit internal homs. Simultaneously, in the syntactic traditions there seems to be a lack of other type formers (like $\Sigma$-, $!$-, and Id-types). The semantic tradition, however, does not seem to have given a sufficient account of the notion of comprehension.

Finally, interesting models motivated from computer science are currently missing. The same is true for models that exhibit some of the flow of information in dependent type theory, e.g. models in terms of tangles or games.

### 2.3.2 Why is the problem hard?

Although it is a priori not entirely clear what linear dependent type theory should be, one can easily come up with some guiding criteria. My gut reaction was the following.

What we want to arrive at is a version of dependent type theory without weakening and contraction rules. Moreover, we would want an exponential co-modality on the type theory that gives us back the missing structural rules.

When one first tries to write down such a type theory, however, one will run into the following discrepancy.

- The lack of weakening and contraction rules in *linear type theory* forces us to refer to each declared variable precisely once: for a sequent $x : A \vdash t : B$, we know that $x$ has a unique occurence in $t$.
- In *dependent type theory*, types can have free (term) variables: $x : A \vdash B$ type, where $x$ is a free variable in $B$. Crucially, we can then talk about terms of $B$: $x : A \vdash b : B$, where generally $x$ may also be free in $b$. For almost all interesting applications we will need multiple occurences of $x$ to construct $b : B$, at least one for $B$ and one for $b$.

The question now is what it means to refer to a declared variable only once.

Do we not count occurences in types? This point of view seems incompatible with universes, however, which play an important role in dependent type theory. If we do, however, the language seems to lose much of its expressive power. In particular, it prevents us from talking about constant types, it seems.

One way to circumvent the issue is *by restricting to type dependency on terms of intuitionistic types*. In this case, there is no conflict, as those terms can be copied and deleted freely. This approach originated in [75]. One may wonder, though, if this is the whole story.

Even if it is, it raises some new issues. Indeed, it seems to exclude the possibility of strong or dependent elimination rules for various type formers, which are of fundamental importance in the absence of $\eta$-rules or uniqueness principles, as in homotopy type theory. Furthermore, despite the existing work, the flow of information in dependent type theory is still poorly understood, which is closely related to the absence of a game semantics.

One might try to make sense of type dependency on linear types by taking the category of contexts of linear dependent type theory to arise as a co-Kleisli category for a linear exponential comonad !, which is entirely natural from the point of view of the literature on linear logic. It has been demonstrated in [71], however, that this forces our model to support "additive" Σ-types (and Id-types), in the sense of e.g. linear types $\Sigma^{\&}_{A}B$ such that a dependent version of the Seely isomorphisms holds: $!\Sigma^{\&}_{A}B = \Sigma^{\otimes}_{!A}!B$, where $\Sigma^{\otimes}$ denotes the multiplicative Σ-type. While multiplicative Σ-types are entirely unproblematic, the additive ones are not so straightforward. For instance, it is not obvious what a syntactic counterpart should be and they are not supported in the natural models of linear type theory in coherence spaces and games, although the models can be (rather artificially) modified to support them. That is, while additive Σ-types arise from the angle SEM), they are difficult to understand from the angles SYN) and MOD). Technicalities aside, it is also challenging to come up with an intuitive interpretation, e.g. an extension of the resource interpretation of linear logic, for additive Σ-types. Needless to say, more investigation would be very desirable, here.

## 2.4 Novelty

### 2.4.1 New Method

The new method employed by this endeavour into the topic is the combination of the three angles SYN), SEM), and MOD). We believe the analysis of linear dependent types has so far remained rather superficial because of the large focus on syntax. Even where categorical semantics has been considered, it has never properly been related to the syntax. We believe that an integrated approach to the topic from all three angles is more likely to result in a deep understanding of the matter, as the angles have proven to be complementary in important ways, in the past. We are thinking, for instance, of the history of linear logic, where the syntax was initially motivated from the study of concrete models of system F [1], of commutative conversions in syntactical calculi that have a motivation in the category theoretic dictum that rules should be natural transformations [6], and of homotopy type theory, where the syntax of dependent type theory suggests new methods of reasoning in homotopy theory [49].

### 2.4.2 Examples

Already, we have managed to make considerable progress in achieving subgoals 1., 2., 4., 5., 7., and 8., through our methodology. This has resulted in the technical report [71], an abridged version of which has been submitted to FoSSaCS 2015, and in two papers in preparation on coherence space semantics and game semantics for dependent types, in collaboration with Samson Abramsky and Radha Jagadeesan, at least one of which we are planning to submit to LiCS 2015. In the research conducted so far, our methodology has been rather fruitful.

Importantly, we had intuitions on what linear dependent type theory should be from the point of view of all three angles. From the point of view of SYN), there was an appealing possibility of a straightforward blend of Barber's dual intuitionistic linear logic (see [4]) and Martin-Löf's dependent type theory (see e.g. [46]). From the point of SEM), the usual adagio of passing from cartesian to more general symmetric monoidal structures seemed reasonable, leading to a simple combination of Benton's linear-non-linear adjunctions (see e.g. [4]) and (indexed categories equivalent to) Jacobs' comprehension categories (see [52]). Finally, from the point of view of MOD), there was a natural notion of type dependency in Girard's coherence space model of linear logic (as described in e.g. [5]): as observed by Samson Abramsky, the construction of Martin-Löf's domain model of dependent types (see e.g. [72]) could be mirrored.

The crucial sanity check was going to be that those three intuitions would lead us down compatible or even equivalent paths, which has, indeed, turned out to be the case: the suggested syntax and abstract categorical semantics turn out to be equivalent and the coherence space construction turns out to be a specific model. This leaves us feeling rather confident that we are pursuing an appropriate notion of linear dependent type theory.

Another example of our methodology working out is the following. One criterion for linear dependent type theory could be that it is able to express infinitary equivalents of (some of) the usual binary connectives of linear logic, in the same way that intuitionistic dependent type theory can be used to express infinitary disjunctions and conjunctions as special cases of Σ- and Π-types, respectively. Again, this issue was approached from two angles: SYN) and MOD). In the syntax, we were able to prove that, in presence of a discrete type $2^{12}$, multiplicative $\Sigma$- and $\Pi$-types can be used to express additive disjunction and conjunction respectively. We observed a similar phenomenon when studying concrete models of families with values in a symmetric monoidal category $\mathcal{V}$: here multiplicative $\Sigma$- and $\Pi$-types and nothing but possibly infinitary equivalents of $\oplus$ and $\&$. Indeed, in these models, every intuitionistic context was discrete (being a set).

## 2.5 Evaluation Criteria

The results obtained should be evaluated, both in terms of correctness and in terms of relevance. To achieve such evaluation, we suggest the following methods.

To guarantee correctness of the results, we intend to approach various topics from multiple angles and make sure that the same results arise in SYN), SEM), and MOD). There is the possibility that machine verified proofs, in e.g. Coq, will also be applied.

To guarantee relevance, we are pursuing our described research goals, that have been formulated to ensure that the work has both theoretical and more practical relevance. These goals tie into the active areas of research of type theory, homotopy theory, computational semantics, and categorical quantum mechanics. The amount of progress towards these goals can also serve as an evaluation criterion.

Finally, we intend to submit our results to various peer-reviewed conferences and journals. In addition to helping the dissemination of the results, this also serves to evaluate their quality, both in terms of correctness and relevance.

## 2.6 Relevance and Risks

The most obvious relevance of an investigation into linear dependent type theory is that it should lead - one would hope - to a more detailed understanding of intuitionistic dependent type theory, which has recently experienced a huge surge in interest due to the discovery of a model in terms of homotopy theory. A linear analysis of dependent type theory should help narrow down how information flows in time in dependent type theory.

Moreover, many sorts of semantics for type theories that are of interest from the point of view of computer science, e.g. coherence space semantics and games semantics, are naturally linear in character. The open problems of providing such computational semantics for dependent type theory, thereby for instance allowing the extension of model checking techniques to dependently typed languages, are therefore closely related to the study of linear dependent type theory. Simultaneously, one might be interested in quantum computation, which can famously be described by certain models of simple linear type theory. A reasonable question is if these models can be extended in an interesting way to admit type dependency, possibly yielding more expressive type systems for quantum programming. Finally, given the immense interest of homotopy type theory, a linear analysis of this system is warranted, which might tie in with the study of a model of linear dependent type theory in terms of stable homotopy theory.

Although we feel the expertise of Oxford in areas surrounding the topic together with the author's knowledge of categorical logic, quantum theory, and geometry put us in an excellent position to meet the challenge of the planned DPhil project, the endeavour is not without risks. In particular, the question of information flow in dependent type theory is a rather vague one and will not necessarily have a straightforward satisfactory answer. We hope it can be a useful guiding source of curiosity, however, given that it is closely aligned with various very concrete questions like the possibility of linear dependent type theory and its models in terms of games and tangles.

Significant results have already been obtained to indicate that the endeavour of providing a full theory of linear dependent types together with coherence space and game semantics should be successful. Steps in the direction of a tangle model have been taken in [81], which gives us hope. Similarly, investigations into models related to quantum mechanics can be found in [82]. Further considering the expertise the Oxford Quantum Group has in semantics of quantum computation, we feel it should be possible to find interesting models of type dependency in quantum computation. Finally, [79, 80] strongly suggest that stable homotopy theory should model linear dependent type theory, which, given the presence of local

[12] Here, we call a type discrete if it is a copower of $I$, in a strong sense.

expertise in Oxford (of e.g. Kobi Kremnitzer and Chris Douglas), seems like a reasonably safe topic to pursue in a collaboration with the Department of Mathematics.

One final risk in this research project is due to its theoretical nature. Although, it is very plausible, at this stage, that we can obtain a full theory of linear dependent types, including various models, it is still too soon to predict if those models will be rather artificial in nature or if they will indeed, as hoped, turn out to be of practical relevance.

# 3 References


[1] Girard, J.Y.: Linear logic. Theoretical computer science **50**(1) (1987) 1–101

[2] Eilenberg, S., Kelly, G.M.: Closed categories. In: Proceedings of the Conference on Categorical Algebra, Springer (1966) 421–562

[3] Abramsky, S.: No-cloning in categorical quantum mechanics. Semantical Techniques in Quantum Computation (2009) 1–28

[4] Barber, A., Plotkin, G.: Dual intuitionistic linear logic. University of Edinburgh, Department of Computer Science, Laboratory for Foundations of Computer Science (1996)

[5] Melliès, P.A.: Categorical semantics of linear logic. Panoramas et Syntheses **27** (2009) 15–215

[6] Bierman, G.M.: On intuitionistic linear logic. PhD thesis, Citeseer (1994)

[7] Benton, P.N.: A mixed linear and non-linear logic: Proofs, terms and models. In: Computer Science Logic, Springer (1995) 121–135

[8] Berry, G.: Stable models of typed $\lambda$-calculi. In: Automata, languages and programming. Springer (1978) 72–89

[9] Girard, J.Y., Taylor, P., Lafont, Y.: Proofs and types. Volume 7. Cambridge University Press Cambridge (1989)

[10] Abramsky, S., Jagadeesan, R.: Games and full completeness for multiplicative linear logic. The Journal of Symbolic Logic **59**(02) (1994) 543–574

[11] Abramsky, S., Jagadeesan, R., Malacaria, P.: Full abstraction for pcf. Information and Computation **163**(2) (2000) 409–470

[12] Hyland, J.M.E., Ong, C.H.L.: Fair games and full completeness for multiplicative linear logic without the mix-rule. preprint **190** (1993)

[13] Hyland, J.M.E., Ong, C.H.: On full abstraction for pcf: I, ii, and iii. Information and computation **163**(2) (2000) 285–408

[14] Calderon, A.C., McCusker, G.: Understanding game semantics through coherence spaces. Electronic Notes in Theoretical Computer Science **265** (2010) 231–244

[15] Abramsky, S., et al.: Semantics of interaction: an introduction to game semantics. Semantics and Logics of Computation **14** (1997) 1

[16] Hyland, J.: Game semantics. Semantics and logics of computation **14** (1997) 131

[17] Blute, R., Panangaden, P., Seely, R.A.: Fock space: a model of linear exponential types. Manuscript, revised version of the MFPS paper above (1994)

[18] Abramsky, S., Coecke, B.: Categorical quantum mechanics. Handbook of quantum logic and quantum structures: quantum logic (2008) 261–324

[19] Abramsky, S., Duncan, R.: A categorical quantum logic. Mathematical Structures in Computer Science **16**(3) (2006) 469–489 Preprint available at http://arxiv.org/abs/quant-ph/0512114.

[20] Selinger, P.: Dagger compact closed categories and completely positive maps. Electronic Notes in Theoretical Computer Science **170** (2007) 139–163

[21] Elmendorf, A., Mandell, M.: Permutative categories, multicategories and algebraic k–theory. Algebraic & Geometric Topology **9** (2009) 2391–2441

[22] Elmendorf, A., Kriz, I., Mandell, M., May, J.: Modern foundations for stable homotopy theory. Handbook of Algebraic Topology, edited by IM James, Elsevier Science, Amsterdam (1995) 213–253

[23] Feynman, R.P.: Space-time approach to quantum electrodynamics. Physical Review **76**(6) (1949) 769

[24] Penrose, R.: Applications of negative dimensional tensors. Combinatorial mathematics and its applications **221244** (1971)

[25] Kelly, G.M., Laplaza, M.L.: Coherence for compact closed categories. Journal of Pure and Applied Algebra **19** (1980) 193–213

[26] Girard, J.Y.: Proof-nets: the parallel syntax for proof-theory. Lecture Notes in Pure and Applied Mathematics (1996) 97–124

[27] Lurie, J., et al.: On the classification of topological field theories. Current developments in mathematics **2008** (2009) 129–280

[28] Baez, J.C., Dolan, J.: Higher-dimensional algebra and topological quantum field theory. Journal of Mathematical Physics **36**(11) (1995) 6073–6105

[29] Coecke, B., Kissinger, A.: Quantum Computer Science - Lecture Notes. University of Oxford (2014) Available at https://www.cs.ox.ac.uk/teaching/materials13-14/quantum/.

[30] Selinger, P.: A survey of graphical languages for monoidal categories. In: New structures for physics. Springer (2011) 289–355

[31] Baez, J.C., Lauda, A.: A prehistory of n-categorical physics. Deep Beauty: Understanding the Quantum World rough Mathematical Innovation (2009) 13–128

[32] Lambek, J.: The mathematics of sentence structure. American mathematical monthly (1958) 154–170

[33] Lambek, J.: Logic and grammar. Studia Logica **100**(4) (2012) 667–681

[34] Moortgat, M.: 2 categorial type logics. Handbook of logic and language (2010) 95

[35] Montague, R.: Universal grammar. Theoria **36**(3) (1970) 373–398

[36] De Saussure, F.: Cours de linguistique générale. (1916)

[37] Schütze, H.: Automatic word sense discrimination. Computational linguistics **24**(1) (1998) 97–123

[38] Clark, S., Pulman, S.: Combining symbolic and distributional models of meaning. In: AAAI Spring Symposium: Quantum Interaction. (2007) 52–55

[39] Clark, S., Coecke, B., Sadrzadeh, M.: A compositional distributional model of meaning. In: Proceedings of the Second Quantum Interaction Symposium (QI-2008). (2008) 133–140

[40] Curry, H.B.: Functionality in combinatory logic. Proceedings of the National Academy of Sciences of the United States of America **20**(11) (1934) 584

[41] de Bruijn, N.: AUTOMATH, a language for mathematics. T.H., Department of Mathematics, Eindhoven University of Technology (November 1968)

[42] Howard, W.A.: The formulae-as-types notion of construction. (1995 [Original Manuscript from 1969])

[43] Martin-Lof, P., Sambin, G.: Intuitionistic type theory. Volume 17. Bibliopolis Naples (1984)

[44] De Marchi, F.: On the semantics of coinductive types in martin-löf type theory. In: Algebra and Coalgebra in Computer Science. Springer (2005) 114–126

[45] van den Berg, B., De Marchi, F.: Non-well-founded trees in categories. Annals of Pure and Applied Logic **146**(1) (2007) 40–59

[46] Hofmann, M.: Syntax and semantics of dependent types. In: Extensional Constructs in Intensional Type Theory. Springer (1997) 13–54

[47] Barendregt, H., et al.: Introduction to generalized type systems. J. Funct. Program. **1**(2) (1991) 125–154

[48] Streicher, T.: How intensional is homotopy type theory? (2013)

[49] Program, T.U.F.: Homotopy Type Theory: Univalent Foundations of Mathematics. http://homotopytypetheory.org/book, Institute for Advanced Study (2013)

[50] Awodey, S., Gambino, N., Sojakova, K.: Inductive types in homotopy type theory. In: Proceedings of the 2012 27th Annual IEEE/ACM Symposium on Logic in Computer Science, IEEE Computer Society (2012) 95–104

[51] Pitts, A.M.: Categorical logic. (2001)

[52] Jacobs, B.: Comprehension categories and the semantics of type dependency. Theoretical Computer Science **107**(2) (1993) 169–207

[53] Seely, R.A.: Locally cartesian closed categories and type theory. In: Mathematical proceedings of the Cambridge philosophical society. Volume 95., Cambridge Univ Press (1984) 33–48

[54] Clairambault, P., Dybjer, P.: The biequivalence of locally cartesian closed categories and martin-löf type theories. In: Typed Lambda Calculi and Applications. Springer (2011) 91–106

[55] Power, A.J.: A general coherence result. Journal of Pure and Applied Algebra **57**(2) (1989) 165–173

[56] Lawvere, F.W.: Equality in hyperdoctrines and comprehension schema as an adjoint functor. Applications of Categorical Algebra **17** (1970) 1–14

[57] Jacobs, B.: Categorical logic and type theory. Volume 141. Elsevier (1999)

[58] Hofmann, M., Streicher, T.: The groupoid interpretation of type theory. Twenty-five years of constructive type theory (Venice, 1995) **36** (1998) 83–111

[59] Warren, M.A.: Homotopy theoretic aspects of constructive type theory. PhD thesis, PhD thesis, Carnegie Mellon University (2008)

[60] Gambino, N., Garner, R.: The identity type weak factorisation system. Theoretical Computer Science **409**(1) (2008) 94–109

[61] Awodey, S., Warren, M.A.: Homotopy theoretic models of identity types. In: Mathematical Proceedings of the Cambridge Philosophical Society. Volume 146., Cambridge Univ Press (2009) 45–55

[62] Awodey, S.: Type theory and homotopy. Springer (2012)

[63] Lumsdaine, P.L.: Model structures from higher inductive types. u rl: http://www. mathstat. dal. ca/˜ pl lumsdaine/research/Lumsdaine-Model-strux-from-HITs. pdf (2011)

[64] Voevodsky, V.: A very short note on the homotopy $\lambda$-calculus. Unpublished note (2006)

[65] Lumsdaine, P.L.: Weak $\omega$-categories from intensional type theory. In: Typed lambda calculi and applications. Springer (2009) 172–187

[66] Garner, R., van den Berg, B.: Types are weak omega-groupoids. Proceedings of the London Mathematical Society **102**(2) (2011) 370–394

[67] Shulman, M.: Univalence for inverse diagrams and homotopy canonicity. arXiv preprint arXiv:1203.3253 (2012)

[68] Moerdijk, I., Palmgren, E.: Wellfounded trees in categories. Annals of Pure and Applied Logic **104**(1) (2000) 189–218

[69] Lumsdaine, P., Shulman, M.: Higher inductive types. preparation (2013) 24

[70] Mendler, N.P.: Predictive type universes and primitive recursion. In: Logic in Computer Science, 1991. LICS'91., Proceedings of Sixth Annual IEEE Symposium on, IEEE (1991) 173–184

[71] Vákár, M.: Syntax and semantics of linear dependent types. arXiv preprint arXiv:1405.0033 (2014)

[72] Palmgren, E., Stoltenberg-Hansen, V.: Domain interpretations of martin-löfs partial type theory. Annals of Pure and Applied Logic **48**(2) (1990) 135–196

[73] Streicher, T.: A model of type theory in simplicial sets. url: http://www. mathematik. tu-darmstadt. de/˜ streicher (2011)

[74] Berg, B.v.d., Moerdijk, I.: W-types in homotopy type theory. arXiv preprint arXiv:1307.2765 (2013)

[75] Cervesato, I., Pfenning, F.: A linear logical framework. In: Logic in Computer Science, 1996. LICS'96. Proceedings., Eleventh Annual IEEE Symposium on, IEEE (1996) 264–275

[76] Dal Lago, U., Gaboardi, M.: Linear dependent types and relative completeness. In: Logic in Computer Science (LICS), 2011 26th Annual IEEE Symposium on, IEEE (2011) 133–142

[77] Petit, B., et al.: Linear dependent types in a call-by-value scenario. In: Proceedings of the 14th symposium on Principles and practice of declarative programming, ACM (2012) 115–126

[78] Gaboardi, M., Haeberlen, A., Hsu, J., Narayan, A., Pierce, B.C.: Linear dependent types for differential privacy. In: ACM SIGPLAN Notices. Volume 48., ACM (2013) 357–370

[79] May, J.P., Sigurdsson, J.: Parametrized homotopy theory. Number 132. American Mathematical Soc. (2006)

[80] Shulman, M.: Enriched indexed categories. Theory and Applications of Categories **28**(21) (2013) 616–695

[81] Ponto, K., Shulman, M.: Duality and traces for indexed monoidal categories. Theory and Applications of Categories **26**(23) (2012) 582–659

[82] Schreiber, U.: Quantization via linear homotopy types. arXiv preprint arXiv:1402.7041 (2014)

[83] Krishnaswami, N.R., Pradic, P., Benton, N.: Integrating dependent and linear types. Retrievable from: https://www.mpi-sws.org/ neelk/dlnl-paper.pdf.

[84] Abramsky, S.: Computational interpretations of linear logic. Theoretical Computer Science **111** (1993) 3–57