Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science

Electronic Notes in Theoretical Computer Science 323 (2016) 143–161

www.elsevier.com/locate/entcs

Fibrational Modal Type Theory

Valeria de Paiva [1,2]

Natural Language and AI Research Lab
Nuance Communications, USA
Sunnyvale, USA

Eike Ritter[3]

School of Computer Science
University of Birmingham
Birmingham, UK

Abstract

This paper describes a fibrational categorical semantics for the modal necessity-only fragment of constructive modal type theory, both with and without dependent types. Constructive type theory does not usually discuss logical modalities, and modalities tend to be mostly studied within classical logic, not within type theory. But modalities should be very useful in type theory, as they are very useful in modelling theoretical computing systems. Providing constructive versions of modal logics and their associated Curry-Howard modal type theories is also a very productive program, e.g. helpful when dealing with computational effects, staged computation, and functional reactive types, for example. There seems to be renewed interest in the notion of constructive modal type theory (and in notions of linear type theory), in part because of the interest in homotopy type theory. The modal type theory presented here uses dependent types, in the style of Ritter's categorical models of the Calculus of Constructions. To build up to these, we first discuss the kinds of constructive modal type theory in the literature. Then we provide a non-dependent modal type theory, introduced in previous work, that we generalize to dependent types in the following section. Dependent type theories are usually but not always given categorical semantics in terms of *fibrations*. We provide semantics in terms of fibrations for both the non-dependent and the dependent type systems discussed and prove them sound and complete, thereby providing evidence that the type theory is meaningful. These fibrational models should be also applicable to the homotopy type theory setting.

Keywords: modal logic, fibrations, categorical models

1 Introduction

Modal logic is the formal logic system that extends propositional (or predicate) logic to include operators expressing modality, mostly intensional notions of possibility and necessity, but also temporal, deontic, provability and other kinds of modalities.

[1] We would like to thank the editors Mário Benevides and René Thiemann for their patience beyond the call of duty with typesetting issues.

[2] Email: valeria.depaiva@nuance.com

[3] Email: exr@cs.bham.ac.uk

http://dx.doi.org/10.1016/j.entcs.2016.06.010

Modal logic, originally conceived as the logic of necessity and possibility, has evolved into a mathematical discipline of its own that deals with (restricted) description languages for talking about various kinds of relational structures.

Modal logic is the explicit logical system most used in Computer Science, but almost all of its uses are based on *classical* modal logic, that is on modal logic over a classical propositional logic basis, while in this paper we deal with constructive modal logic. Constructive (or intuitionistic, we will use the terms interchangebly) modal logic starts from an intuitionistic propositional basis and add modalities to it. This process, like any of 'constructivizing' a logical system, is usually a one-to-many one, with a single classical concept giving rise to many possible constructive versions, which one needs to compare, contrast and choose from. All kinds of mathematical, philosophical and esthetical criteria can be used for choosing your favorite constructive version of a classical concept.

The basic unary (1-place) modal operators are usually written $\Box$ for *necessarily the case* and $\Diamond$ for it is *possibly the case*. In a classical modal logic, each of these operators can be expressed using the other using negation: $\Diamond P \leftrightarrow \neg\Box\neg P$; $\Box P \leftrightarrow \neg\Diamond\neg P$. In the case of constructive modal logics this interdefinability of the operators is not expected nor desired. The same way as we do not have in intuitionistic logic the interdefinability between universal and existential quantifiers $\forall x.P(x) \leftrightarrow \neg\exists x\neg P(x)$, $\exists x.P(x) \leftrightarrow \neg\forall x\neg P(x)$ we do not expect it as far as the modalities are concerned.

Some times different methods of constructivization in logic led researchers to the same systems. In particular the work of Fitch [18], Fisher-Servi [17], Ewald [15], Plotkin and Stirling and especially Simpson [29] has resulted in the most well-known and successful systems of intuitionistic modal logic, based on the system IK. Simpson's systems use a labelled deduction method, where the semantic intuitions of possible worlds are codified in the syntax, via assertions about accessibility between worlds. Following a very different path Prawitz also provided intuitionistic versions for systems S4 and S5 in his seminal work in Natural Deduction [26]. This work gave rise to a collection of associated work, especially on intuitionistic versions of S4, [8,7,16,2], where the guiding intuitions come from the Curry-Howard correspondence. These systems are here dubbed the systems CS4, for constructive modal logic.

Some of the research on these constructive modal systems based on Prawitz-style Natural Deduction were inspired by Girard's Linear Logic and these benefit from later developments in linear lambda calculi, such as the Dual Intuitionistic and Linear lambda-calculus (DILL) [4] of Barber and Plotkin. Our previous work on constructive modal logic provides a Dual Intuitionistic Modal lambda-calculus (DIML) [19] in the style of DILL. Since the work on DIML was basically motivated by the implementation of functional languages using categorical combinators, the main work on the DIML calculus was to make sure that the *explicit substitutions* necessary for the implementation of the categorical combinators worked. But we also presented the basics of the constructive modal type theory for necessity-only S4, which we reproduce in the next section.

The main goal of this paper is to show a fibrational categorical semantics for this constructive modal type theory and to extend it with dependent boxes. We have known about this fibrational categorical semantics for a while, but it has remained in our "to think about" folder of ideas, as we had no compelling application for it. Now there seems to be renewed interest in the notions of constructive modal type theory and linear type theory, in part caused by the interest in homotopy type theory [30]. Thus it seems a good idea to write up how to integrate S4 box-only modalities with dependent types, making the boxes dependent constructors.

2 Modal Type Theories

The somewhat outdated survey [12] explains that there are several proposals for constructive modal logics and associated modal type theories in the literature. They come from different motivations and intuitions, from logic, mathematics and computer science, drawn from the different backgrounds of their authors.

The pioneering work on constructive modal logics has resulted in the most well-known and successful systems of intuitionistic modal logic, based on the system IK of Simpson[29]. Simpson's systems use a labelled deduction method, where the semantic intuitions of possible worlds are codified in the syntax, via assertions about accessibility between worlds.

These systems are to be contrasted with the systems that originated with Prawitz, who also provided intuitionistic versions for systems S4 and S5 in his seminal work in Natural Deduction [26]. Prawitz work gave rise to a collection of associated work, especially on intuitionistic versions of S4, [8,7,16], where the guiding intuitions come from the Curry-Howard correspondence. These systems are dubbed the systems CS4, for constructive modal logic. Unlike Simpson's systems, these systems do not use information about possible worlds semantics in their syntax, and do not satisfy the distribution of possibility over disjunction, as exemplified by the implications

$$\Diamond(A \vee B) \to \Diamond A \vee \Diamond B \tag{1}$$

$$\Diamond \bot \to \bot \tag{2}$$

The first version of modal type theory we know of is Moggi's seminal work on the Computational Lambda-calculus [24], explained by [7], [21] as a rather degenerate modal type theory. There is the doctoral work of Borghuis [9] which presents a *classical* modal theory on top of Barendregt's cube. Borghuis uses Fitch's variant of Natural Deduction, which is equivalent to Gentzen-Prawitz Natural deduction for first-order classical logic. But the modal type theory he considers is classical, the modalities are interdefinable, which does not make much sense, if the goal is obtaining a Curry-Howard correspondence.

There is also the highly influential work on modal type theory of Frank Pfenning and collaborators [25,11], following the Curry-Howard correspondence for constructive S4 and Martin-Löf's methodology of distinguishing judgements from propositions. This work, contrary to [8,7] has not, by and large, being concerned with

categorical semantics. In the next section we describe a system that like [8] has its origins in Prawitz work, and takes as guiding intuition the categorical modelling of the Curry-Howard correspondence in the style of [6]. Thus we think of modal systems as defined on the same level as their intuitionistic counterparts. Our original motivation was the re-use of the machinery dealing with reduction of the simply typed lambda-calculus.

3 Dual Intuitionistic Modal Logic (DIML)

We recall the proof system for constructive necessity called Dual Intuitionistic Modal Logic (DIML) introduced in our previous work [19] which is based upon Barber (and Plotkin's) Dual Intuitionistic Linear Logic (DILL) [4]. The original system DILL manages to keep reduction systems for intuitionistic propositional and intuitionistic linear logic, side-by-side, by making use of a direct "translation" to logic of its proposed categorical semantics. The modal system DIML does the same for a necessity modality in intuitionistic logic. Note that despite the fact that we are calling it a logic, the system is a type theory that is Curry-Howard correspondent to the logic, that is, if we erase the terms, we end up with a Natural Deduction formulation of the logic. We recap the details of the type theory, as generalizing it to dependent types is our main goal.

3.1 The Typing Judgements of DIML

The types of DIML are ground types, function types $A \to B$ and box-types $\Box A$. Variables are tagged x_M or x_I to indicate whether they are *modal* or *intuitionistic* and the raw expressions of DIML are

$$\begin{aligned} t ::= {} & x_M \mid x_I \mid \lambda x\colon A.t \mid tt \\ & \mid \Box t \mid \texttt{let}\ t\ \texttt{be}\ \Box x\ \texttt{in}\ t \end{aligned}$$

where the x's are variables. (The tags on variables are sometimes omitted to increase legibility.)

We identify α-equivalent terms. A context is a sequence of the form $x_1\colon A_1, \ldots, x_n\colon A_n$ where the x's are distinct variables and the A's are types. A DIML context consists of two contexts Γ and Δ containing disjoint sets of variables and is written $\Gamma|\Delta$. We call the variables in Γ *modal* variables and the variables in Δ *intuitionistic* variables.

The typing judgements of DIML are of the form $\Gamma|\Delta \vdash t\colon A$ and are generated by the inference rules in Figure 1. These are the rules for the simply typed lambda-calculus plus an introduction and elimination rule for the $\Box$ modality. Note that the introduction of a necessary $\Box$ type requires an empty intuitionistic context, corresponding to the intuitive idea that we can say a proposition is necessary, if all the assumptions it depends on are already necessary, i.e. the set of non-necessarily-modal assumptions (its intuitionistic context) is empty. The elimination rule for

the $\Box$ modality is usual, following Schroeder-Heister's style of elimination rules.

Weakening for both kinds of variables is admissible because of the typing rule for these variables. Contraction of both kinds of variables follows from the additive way contexts are combined. Exchange is also admissible. Note that we could also have rules for logical conjunctions and disjunctions, but we prefer to concentrate on the essential connectives implication and modality.

$$\Gamma, x\colon A, \Gamma' | \Delta \vdash x_M\colon A \qquad \Gamma | \Delta, x\colon A, \Delta' \vdash x_I\colon A$$

$$\frac{\Gamma | \Delta, x\colon A \vdash t\colon B}{\Gamma | \Delta \vdash \lambda x\colon A.t\colon A \to B}\ (\to I) \qquad \frac{\Gamma | \Delta \vdash t\colon A \to B \ \ \Gamma | \Delta \vdash u\colon A}{\Gamma | \Delta \vdash tu\colon B}\ (\to E)$$

$$\frac{\Gamma |_{\_} \vdash t\colon A}{\Gamma | \Delta \vdash \Box t\colon \Box A}\ (\Box I) \qquad \frac{\Gamma | \Delta \vdash t_i\colon \Box A_i \ \Gamma, x\colon A | \Delta \vdash u\colon B}{\Gamma | \Delta \vdash \mathtt{let}\ t\ \mathtt{be}\ \Box x\ \mathtt{in}\ u\colon B}\ (\Box E)$$

$$\frac{\Gamma | \Delta, x\colon A \vdash t\colon B \ \ \Gamma | \Delta \vdash u\colon A}{\Gamma | \Delta \vdash (\lambda x\colon A.t)u = t[u/x]\colon B} \qquad \frac{\Gamma | \Delta \vdash t\colon A \to B}{\Gamma | \Delta \vdash \lambda x\colon A.tx = t\colon A \to B}\ (x \notin FV(t))$$

$$\frac{\Gamma |_{\_} \vdash t\colon A \ \ \Gamma, x\colon A | \Delta, y\colon B \vdash C}{\Gamma | \Delta \vdash \mathtt{let}\ \Box t\ \mathtt{be}\ \Box x\ \mathtt{in}\ u = u[t/x]\colon C}$$

$$\frac{\Gamma \Delta \vdash t\colon \Box A \ \ \Gamma | \Delta, y\colon \Box A \vdash t'\colon C}{\Gamma | \Delta \vdash \mathtt{let}\ t\ \mathtt{be}\ \Box x\ \mathtt{in}\ t'[\Box(x)/y] = t'[t/y]}$$

Fig. 1. Typing Judgements for DIML (Dual Intuitionistic and Modal Logic)

The more complex nature of contexts in DIML results in a more subtle meta-theory. Just as there are two rules for typing variables, so there are two substitution lemmas depending on the nature of the variable being substituted. As we learned from the linear lambda-calculus we need an appropriate, double substitution lemma, catering for both kinds of variables, see below. Observe that forgetting about the terms in the lemma below, we obtain two forms of the cut rule, which we can show are admissible in DIML.

Lemma 3.1 (Substitution) *If* $\Gamma | \Delta \vdash t\colon A$ *and* $\Gamma | x_I\colon A, \Delta \vdash s\colon B$, *then* $\Gamma | \Delta \vdash s[t/x_I]\colon B$, *where* $[t/x_I]$ *denotes the traditional meta-level substitution. Similarly if* $\Gamma |\_ \vdash t\colon A$ *and* $\Gamma, x_M\colon A | \Delta \vdash s\colon B$ *then* $\Gamma | \Delta \vdash s[t/x_M]\colon B$.

We can regard the introduction and elimination rules as being inverse to each other and hence derive a β and an η-equality judgement for each type constructor. This then leads to a full Curry-Howard correspondence for DIML, which can be

showed equivalent to the type theory in [8] and thence to the axiomatic definition of $\Box$-only S4, sometimes written as CS4.

Theorem 3.2 *There is an axiomatic $\Box$-only intuitionistic S4 modal derivation of $\Gamma \to A$ iff there is a DIML term t and a DIML judgement $\_|\Gamma \vdash t : A$.*

Note the empty modal context in $\_|\Gamma \vdash t : A$, modal variables can always be moved to the intuitionistic part of the context, with the $\Box$ prefixed to them. The proof is an easy adaptation of Barber's proof, so is omitted.

Soundness and completeness theorems for $\Box$-only constructive S4, as proved in [8] are available, both via the translation between DIML and CS4 and directly, as below. We do not expand on these remarks but instead move on directly to reduction in the type theory DIML.

3.2 The reduction calculus for DIML

Reduction in DIML consists only of β-redexes and is the least congruence on raw expressions containing the basic reductions

$$(\lambda x\colon A.t)u \Rightarrow t[u/x]$$

$$\texttt{let}\ \Box u\ \texttt{be}\ \Box y,\ \texttt{in}\ s \Rightarrow s[u/y]$$

Two terms t and u which are equivalent in the equational theory generated by the DIML reduction relation are called DIML equivalent and this is written $t \cong u$. Subject reduction means that typing information is preserved by reduction and now we consider only those reductions whose redexes are well-typed terms — by subject reduction the reducts are also well-typed.

Lemma 3.3 (DIML Subject Reduction) *If there is a DIML typing judgement $\Gamma|\Delta \vdash t\colon A$ and a rewrite $t \Rightarrow t'$, then there is also a typing judgement $\Gamma|\Delta \vdash t'\colon A$.*

A second key property of DIML reduction is that it is contained within the equality judgements of DIML.

Lemma 3.4 *If $\Gamma|\Delta \vdash t : A$ and $t \Rightarrow u$, then there is an equality judgement $\Gamma|\ \Delta \vdash t = u : A$ in DIML.*

Theorem 3.5 *The DIML reduction relation $\Rightarrow$ is strongly normalising and confluent.*

Proof. Strong normalisation is proved by standard reducibility methods while confluence then follows from local confluence which may easily be checked. $\Box$

3.3 DIML Semantics

The presentation of DIML just given is based on typing and equality judgements so as to make the connection with the semantics as smooth as possible. Our reduction calculus is designed to model the computational process and so is restricted to only

the β-redexes for each type constructor, no η conversions. The upshot is a well-behaved type theory, for which we can provide an operational semantics, if desired.

A traditional 1-categorical semantics for DIML is also easy to state and prove sound and complete, as the type theory was originally (in the linear logic case [4]) derived from it.

Definition 3.6 We define a DIML category $\mathcal{M}$ to be a pair of cartesian closed categories (S, C), with two symmetric monoidal functors $F : C \to S$ and $G : S \to C$ which are adjoint, $G \dashv F$.

The intuition is that S models the modal contexts and modal terms, and C the intuitionistic contexts and intuitionistic terms. The adjunction is used to model the $\Box$-modality.

Define an interpretation $[\![\_]\!] : DIML \to \mathcal{M}$ which takes the types and sequents of DIML (over a basic set of types) to a model $\mathcal{M}$ as follows:

$$\begin{aligned} [\![X]\!] &= I(X) \text{ for } X \text{ a base type} \\ [\![A \to B]\!] &= [\![A]\!] \to [\![B]\!] \\ [\![\Box A]\!] &= FG([\![A]\!]) \end{aligned}$$

We extend this interpretation to contexts $(A_1, ...A_n|\ B_1, ...B_n)$ by adding boxes to the modal assumptions and defining $[\![A_1, \ldots, A_n|\ B_1, \ldots, B_n]\!] = \Box[\![A_1]\!] \times \cdots \times \Box[\![A_n]\!] \times\ [\![B_1]\!] \times \cdots \times [\![B_n]\!]$. The interpretation will take a sequent $\Gamma|\Delta \vdash t : A$ to an arrow $[\![\Gamma|\Delta \vdash t : A]\!] : [\![\Gamma|\Delta]\!] \to [\![A]\!]$ in the DIML category.

The DIML type theory can be soundly interpreted in a DIML category as defined above.

Theorem 3.7 *The type theory DIML has **sound** models provided by the structures $\mathcal{M}$ defined above. In other words, given a DIML category $\mathcal{M}$, using the above interpretation, the following hold:*

- *Assume $\Gamma|\Delta \vdash t : A$ in DIML. Then $[\![\Gamma|\Delta \vdash t\colon A]\!]$ is a morphism in S with domain $[\![\Gamma|\Delta]\!]$ and codomain $[\![A]\!]$;*
- *Assume $\Gamma \vdash t = s\colon A$. Then $[\![\Gamma \vdash t\colon A]\!] = [\![\Gamma \vdash s\colon A]\!]$.*

We also have completeness of DIML categories:

Theorem 3.8 *The DIML models are **complete** in the appropriate sense for the type theory DIML. This is to say, if we have equality of the interpretations $[\![\Gamma \vdash t\colon A]\!] = [\![\Gamma \vdash s\colon A]\!]$ (where $[\![\ ]\!]$ is the interpretation defined above) in the DIML category $\mathcal{M}$ for any derived sequents $\Gamma \vdash t\colon A$ and $\Gamma \vdash s\colon A$ then we can derive the equation in the type theory DIML $\Gamma \vdash t = s\colon A$.*

The proof is *mutatis mutandis* the one in Barber's thesis [4]. We construct the term model and show that the basic rules of the calculus are mapped to appropriate morphisms using the categorical constructs.

Thus we have a relatively easy model of DIML at hand, however in the next section we will explain a fibration model for DIML. Why? Why do we embark on

using fibrations to model this type theory? The reason is that we want to model modal extensions of Martin-Löf's dependent type theory (DTT), which will need the more involved, fibrational categorical semantics. Fortunately one of us wrote a doctoral thesis [28] on modelling dependent type theories and implementing them in efficient ways, so most of the work was in place, needing only to be adapted to the case of necessity-modality constructor.

4 Simple DIML Fibrational Semantics

We can also model the DIML type theory via fibrations, instead of adjunctions. The usual fibration modelling framework was developed for theories with dependent types. One can use this framework for modelling theories without dependent types by imposing a strong restriction, discussed below.

Recall that the basic modelling of dependent types (products and sums) using fibrations is complicated somewhat by the mismatch between syntax and semantics, in that substitution in the syntax is "strict", that is up to syntactic equality, while substitution in the semantics tends to be modelled "up to isomorphism". The discussion in the n-Lab (`http://ncatlab.org/nlab/show/categorical+model+of+dependent+types`) explains the existence of several almost equivalent formulations and some of the pros and cons of each. Jacobs [20] provides a detailed exposition of dependent type theories and their categorical modelling.

We choose in this paper a formulation based on indexed categories $E\colon \mathcal{B}^{op} \to$ **Cat** which models strict substitutions. To model the necessity modality in the dependent setting, we follow the intuitions of our previous work on Linear Logic, more specifically we follow the work on modelling ILT [23], an intuitionistic linear lambda-calculus, where linear and intuitionistic implications co-exist. The main point is that a necessity $\Box$ modality, like the linear logic bang ! requires proof-theoretically a special form of assumptions, these have to be of the form $!A$. Similarly, to introduce a modal type $\Box B$ all its assumptions have to be of the form $\Box A_i$.

We first recap the modelling of dependent types using D-categories, then we restrict ourselves to non-dependent types, but add the necessity modality.

4.1 Dependent Products

To model simply dependent products, a generalisation of function spaces, we use a variant of Ehrhard's D-category [14], which goes back to Lawvere's idea of hyperdoctrines satisfying the comprehension axiom, see [28] for a fuller explanation.

Here we simply recap the dependent type theory syntax that we want to model and the choices we have made. The types are ground types and dependent products $\Pi_{x\colon A}B$. The raw expressions are

$$A ::= G \mid \Pi_{x\colon A}B$$
$$t ::= x \mid \lambda x\colon A.t \mid tt$$

where the x's are variables.

Because types may contain terms, we require typing judgements for all syntactic categories, namely contexts, types and terms. They are generated by the inference rules in Figure 2. We have omitted the congruence rules for contexts, types and terms.

$$\frac{}{[] \vdash \texttt{Context}} \qquad \frac{\Gamma \vdash \texttt{Context} \quad \Gamma \vdash A\ \texttt{Type}}{\Gamma, x\colon A \vdash \texttt{Context}}$$

$$\frac{\Gamma, x_M\colon A \vdash B\ \texttt{Type}}{\Gamma \vdash \Pi_{x_M\colon A} B\ \texttt{Type}}$$

$$\frac{}{\Gamma, x\colon A, \Gamma' \vdash x\colon A}$$

$$\frac{\Gamma, x\colon A \vdash t\colon B}{\Gamma \vdash \lambda x\colon A.t\colon \Pi_{x\colon A} B}\ (\Pi I) \qquad \frac{\Gamma \vdash t\colon \Pi_{x\colon A} B \quad \Gamma \vdash u\colon A}{\Gamma \vdash tu\colon B[u/x]}\ (\Pi E)$$

$$\frac{\Gamma, x\colon A \vdash t\colon B \quad \Gamma \vdash u\colon A}{\Gamma \vdash (\lambda x\colon A.t)u = t[u/x]\colon B[u/x]} \qquad \frac{\Gamma \vdash t\colon \Pi_{x\colon A} B}{\Gamma \vdash \lambda x\colon A.tx = t\colon \Pi_{x\colon A} B}\ (x \notin FV(t))$$

Fig. 2. Typing Judgements for dependent type theory (DTT)

For basic dependent type theory there is only one reduction rule, namely β-reduction

$$(\lambda x\colon A.t)u \Rightarrow t[u/x]\ .$$

Because there is a non-trivial equality between types, showing that the dependent type theory satisfies standard meta-theoretic results is much harder. In particular, subject reduction is a non-trivial theorem. As part of the proof one has to show that two dependent product types $\Pi_{x\colon A'} B$ and $\Pi_{x\colon A'} B'$ are equal if and only if A and A' and B and B' are equal. For details see [5].

Lemma 4.1 (DTT Subject Reduction) *If there is a typing judgement $\Gamma \vdash t\colon A$ in dependent type theory and a rewrite $t \Rightarrow t'$, then there is also a typing judgement $\Gamma \vdash t'\colon A$.*

Subject reduction is also the key point in showing that reduction in dependent type theory is contained within the equality judgement.

Lemma 4.2 *If $\Gamma \vdash t : A$ and $t \Rightarrow u$, then there is an equality judgement $\Gamma \vdash t = u : A$ in dependent type theory.*

To model this type theory we have an indexed category $E\colon \mathcal{B}^{op} \to \mathbf{Cat}$ where the base category $\mathcal{B}$ models usual contexts Γ. Fibres over an object Γ, $E(\Gamma)$ model

terms whose variables are contained in the context modelled by Γ. We need a terminal object $\top$ in $\mathcal{B}$. Each fibre is a cartesian closed category. We also require that for each morphism f in the base $\mathcal{B}$, $E(f)$ preserves the terminal object. The key construction of this indexed category is the requirement called the "comprehension property": Substitution of variables is modelled by applying the functor $E(f)$, and context extension is modelled by comprehension which yields a bijection between a morphism $f\colon \Gamma \to \Gamma' \cdot A$ in $\mathcal{B}$ and a pair consisting of a morphism $f\colon \Gamma \to \Gamma'$ in $\mathcal{B}$ and a morphism $g\colon 1 \to f^*A$ in $E(\Gamma)$.

A precise definition is as follows:

Definition 4.3 A *D-category* is an indexed category $E\colon \mathcal{B}^{op} \to \mathbf{Cat}$ such that

(i) $\mathcal{B}$ has a terminal object $\top$;

(ii) The functor $\mathcal{I}\colon \mathcal{B} \to Gr(E)$ defined by $\mathcal{I}(\Gamma) = (\Gamma, 1)$ and $\mathcal{I}(f) = (f, \mathsf{Id})$ has a right adjoint G. We write $\Gamma \cdot A$ for $G((\Gamma, A))$, $(\mathsf{Fst}, \mathsf{Snd})$ for the co-unit of this adjunction, and $f \cdot h$ for $G((f, h))$.

(iii) For every pair of morphisms $g\colon \Gamma \cdot A \to \Delta \cdot B$ and $f\colon \Gamma \to \Delta$ such that $\mathsf{Fst} \circ g = f \circ \mathsf{Fst}$ there exists a unique morphism $h\colon A \to f^*B$ in $E(\Gamma)$ such that $g = f \cdot h$;

(iv) For every object Γ of $\mathcal{B}$ and A of $E(\Gamma)$, the functor $\mathsf{Fst}^*_A\colon E(\Gamma) \to E(\Gamma \cdot A)$ has a right adjoint $\Pi_A\colon E(\Gamma \cdot A) \to E(\Gamma)$. We will write in the sequel Cur for the natural isomorphism between $\mathrm{Hom}_{E(\Gamma \cdot A)}(\mathsf{Fst}^*_A(B), C)$ and $\mathrm{Hom}_{E(\Gamma)}(B,\ \Pi_A(C))$ and App for its counit.

(v) The Beck-Chevalley-condition for the adjunctions $\mathsf{Fst}^*_A \dashv \Pi_A$ is satisfied in the strict sense, i.e. the equation $f^*(\mathsf{Cur}_A(h)) = \mathsf{Cur}_A((f \cdot \mathsf{Id})^*(h))$ holds for every $f\colon \Delta \to \Gamma$, $A \in E(\Gamma)$, $B \in E(\Gamma \cdot A)$ and morphism $h\colon 1 \cdot B$ in $E(\Gamma \cdot A)$.

Any dependent type theory can be soundly interpreted in a D-category. The interpretation is defined by induction over the derivation. Variables are interpreted using the comprehension and dependent products using the right adjunction to weakening.

Using the meta-theoretic results from dependent type theory one shows that the interpretation of a judgement is independent of its derivation.

Theorem 4.4 *Given a D-category $E\colon \mathcal{B}^{op} \to \mathbf{Cat}$ under the above interpretation $[\![\,]\!]$ the following facts hold.*

(i) *Assume $\Gamma \vdash \mathtt{Context}$. Then $[\![\Gamma]\!]$ is an object in $\mathcal{B}$;*

(ii) *Assume $\Gamma \vdash A\ \mathtt{Type}$. Then $[\![\Gamma \vdash A\ \mathtt{Type}]\!]$ is an object in $E([\![\Gamma]\!])$;*

(iii) *Assume $\Gamma \vdash M\colon A$. Then $[\![\Gamma \vdash M\colon A]\!]$ is a morphism from 1 to $[\![A]\!]$ in $E([\![\Gamma]\!])$;*

(iv) *Assume $\vdash \Gamma = \Delta$. Then $[\![\Gamma]\!] = [\![\Delta]\!]$;*

(v) *Assume $\Gamma \vdash A = B$. Then $[\![\Gamma \vdash A\ \mathtt{Type}]\!] = [\![\Gamma \vdash B\ \mathtt{Type}]\!]$;*

(vi) *Assume $\Gamma \vdash M = N\colon A$. Then $[\![\Gamma \vdash M\colon A]\!] = [\![\Gamma \vdash N\colon A]\!]$.*

D-categories are also complete, which can be shown by the usual term model construction.

Theorem 4.5 *If* $[\![\Gamma \vdash M\colon A]\!] = [\![\Gamma \vdash N\colon A]\!]$ *where* $[\![\ ]\!]$ *is the above defined interpretation, for every D-category* $E\colon \mathcal{B}^{op} \to \mathbf{Cat}$ *and for every derived sequents* $\Gamma \vdash M\colon A$ *and* $\Gamma \vdash N\colon A$ *then we can derive in the type theory* $\Gamma \vdash M = N\colon A$.

Both theorems were stated and proved for the Calculus of Constructions, which is a special case of dependent type theory, in [27]. This proof is modular and specialises to soundness and completeness of the general dependent type theory.

Moreover, the proof also specialises for the case of non-dependent type theory, the only modification is to insist that the functor $E(f)$ is the identity on objects of $\mathcal{B}$.

4.2 Adding a Necessity Modality

When we want to add box modalities to the dependent types we can use the basic idea of D-categories, but now contexts have two parts (a modal context and an intuitionistic one) that behave very differently. The basic intuition is to follow the modelling in [23], where the base category $\mathcal{B}$ models terms with **only** modal variables as morphisms, and considers modal contexts as objects. Fibres over an object $\Gamma|\Delta$ model terms with both intuitionistic and modal variables where the modal variables are contained in the modal context modelled by Γ. Substitution of modal variables is modelled by applying the functor $E(f)$, and context extension of modal contexts by comprehension which yields now a bijection between a morphism $f\colon \Gamma \cdot A$ in $\mathcal{B}$ and a pair consisting of a morphism $f\colon \Gamma \to \Gamma'$ in $\mathcal{B}$ and a morphism $g\colon \Delta \to A$ in $E(\Gamma)$. This leads us to the following full definition:

Definition 4.6 A *fibred DIML-category* is an indexed category $E\colon \mathcal{B}^{op} \to \mathbf{Cat}$ such that

(i) $\mathcal{B}$ has a terminal object $\top$;

(ii) all fibres are cartesian closed categories and $E(f)$ preserves the cartesian closed structure on the nose for every morphism f in $\mathcal{B}$;

(iii) Each functor $E(f)$ is the identity on objects;

(iv) The functor $\mathcal{I}\colon \mathcal{B} \to Gr(E)$ defined by $\mathcal{I}(\Gamma) = (\Gamma, 1)$ and $\mathcal{I}(f) = (f, \mathsf{Id})$ has a right adjoint G. We write $\Gamma \cdot A$ for $G((\Gamma, A))$ and $(\mathsf{Fst}, \mathsf{Snd})$ for the co-unit of this adjunction;

(v) For all objects Γ of $\mathcal{B}$ the functor $\mathsf{Fst}_A^*\colon E(\Gamma) \to E(\Gamma \cdot A)$ has a left adjoint $\Box_A$. If $f\colon B \to \mathsf{Fst}_A^* C$ is a morphism in $E(\Gamma \cdot A)$, we write $\nu_A(f)$ for the morphism in $E(\Gamma)$ obtained by transposing f across the adjunction.

(vi) Moreover, the Beck-Chevalley condition for the adjunction $\Box_A \dashv \mathsf{Fst}_A^*$ is satisfied in the strict sense, i.e. the equations

$$f^* \circ \Box_A = \Box_A \circ (f \cdot \mathsf{Id})^* \qquad f^*(\nu_A(g)) = \nu_A((f \cdot \mathsf{Id})^*(g))$$

hold for every $f\colon \Delta \to \Gamma$, $B \in ObE(\Gamma)$, $C \in ObE(\Gamma)$ and $g \in \mathrm{Hom}_{E(\Gamma.A)}(B, C)$.

Note that since this framework was developed for theories with dependent types,

to use it for modelling theories without dependent types we impose the restriction that re-indexing is the identity on objects, condition [(*iii*)] in the definition above.

5 Dependent Modal Type Theory

Having defined a fibrational semantics for constructive modal type theory where all the type constructors are simple types, we now want to have modalities that are themselves dependent types over a system of dependent products. This is the main innovation of this paper. We are not aware of any other work which described dependent modalities as these S4 necessity modalities we describe here.

Dependent type theory has a mixed record. In one hand is considered by some "the modern way of doing first-order logic", on the other hand is considered by others an unintelligible way of extending functional programming. Within functional programming itself, dependent types have been used in two very different ways as well explained in [10]: first as case study in theorem proving in constructive logic, as in Coq, LEGO, Agda, Lean, etc, but also as an extension to already existing and ordinary functional language such as Haskell and ML. In the first case non-termination is forbidden, as it would make the logic inconsistent, while in the second case one seeks a language for heterogeneous verification, as expressive as possible, allowing programmers to devote their 'verification budget' to critical sections. Such a language must support general recursion as natively as any functional programming language and hence non-termination is not an issue. In this paper we are definitely in the first camp and want to see what are the mathematical foundations of type systems that allow for intensional notions of modality (specifically the necessity or $\Box$-like S4 modality) within a dependent type universe. In such universe it makes more sense for the modality itself to be a dependent type constructor.

The formal development of the type theory is parallel to the non-dependent case. The raw expressions of the dependent modal type theory are given by

$$t ::= x_M \mid x_I \mid \lambda x\colon A.t \mid tt \mid \Box(t,t) \mid \texttt{let}\ t\ \texttt{be}\ \Box(x_M, x_I)\ \texttt{in}\ t$$

where the x's are variables. (The tags on variables are sometimes omitted to increase legibility.)

The typing judgements for the dependent modal type theory are in Figure 3. We have omitted the congruence rules for contexts, types and terms. The important thing to note is that, given our formulation of box modalities in terms of double contexts, the generalization of a box to a dependent constructor becomes an easy step, somewhat reminiscent, in the semantics, of how to deal with strong sums in traditional Martin-Löf dependent type theory.

Note that non-dependent products, i.e. usual function spaces or logical implica-

$$\frac{}{[] \vdash \texttt{Context}} \qquad \frac{\Gamma \vdash \texttt{Context} \quad \Gamma \vdash A\ \texttt{Type}}{\Gamma, x\colon A \vdash \texttt{Context}}$$

$$\frac{\Gamma \vdash A\ \texttt{Type} \quad \Gamma \vdash B\ \texttt{Type}}{\Gamma \vdash A \to B\ \texttt{Type}} \qquad \frac{\Gamma, x_M\colon A \vdash B\ \texttt{Type}\ \Gamma, x_M\colon A \vdash B\ \texttt{Type}}{\Gamma \vdash \Pi_{x_M\colon A} B\ \texttt{Type} \quad \Gamma \vdash \Box_{x_M\colon A} B\ \texttt{Type}}$$

$$\Gamma, x\colon A, \Gamma' | \Delta \vdash x_M\colon A \qquad \Gamma | \Delta, x_I\colon A, \Delta' \vdash x_I\colon A$$

$$\frac{\Gamma | \Delta, x_I\colon A \vdash t\colon B}{\Gamma | \Delta \vdash \lambda x_I\colon A.t\colon A \to B}\ (\to I) \qquad \frac{\Gamma | \Delta \vdash t\colon A \to B \quad \Gamma | \Delta \vdash u\colon A}{\Gamma | \Delta \vdash tu\colon B}\ (\to E)$$

$$\frac{\Gamma, x_M\colon A | \Delta \vdash t\colon B}{\Gamma | \Delta \vdash \lambda x_M\colon A.t\colon \Pi_{x\colon A} B}\ (\Pi I) \qquad \frac{\Gamma | \Delta \vdash t\colon \Pi_{x\colon A} B \quad \Gamma | \_ \vdash u\colon A}{\Gamma | \Delta \vdash tu\colon B[u/x_M]}\ (\Pi E)$$

$$\frac{\Gamma | \_ \vdash t\colon A \quad \Gamma | \Delta \vdash s\colon B[t/x_M]}{\Gamma | \Delta \vdash \Box(t,s)\colon \Box_{x_M\colon A} B}\ (\Box I)$$

$$\frac{\Gamma | \Delta \vdash t\colon \Box_{x_M\colon A} B \quad \Gamma, x\colon A | \Delta, y\colon B \vdash u\colon C}{\Gamma | \Delta \vdash \texttt{let}\ t\ \texttt{be}\ \Box(x,y)\ \texttt{in}\ u\colon C}$$

$$\frac{\Gamma, x\colon A | \Delta \vdash t\colon B \quad \Gamma | \_ \vdash u\colon A}{\Gamma | \Delta \vdash (\lambda x\colon A.t)u = t[u/x]\colon B[u/x]} \qquad \frac{\Gamma | \Delta \vdash t\colon \Pi_{x\colon A} B}{\Gamma | \Delta \vdash \lambda x\colon A.tx = t\colon \Pi_{x\colon A} B}\ (x \notin FV(t))$$

$$\frac{\Gamma | \Delta, x\colon A \vdash t\colon B \quad \Gamma | \Delta \vdash u\colon A}{\Gamma | \Delta \vdash (\lambda x\colon A.t)u = t[u/x]\colon B} \qquad \frac{\Gamma | \Delta \vdash t\colon A \to B}{\Gamma | \Delta \vdash \lambda x\colon A.tx = t\colon A \to B}\ (x \notin FV(t))$$

$$\frac{\Gamma | \_ \vdash t\colon A \quad \Gamma | \Delta \vdash s\colon B[t/x_M] \quad \Gamma, x\colon A | \Delta, y\colon B \vdash C}{\Gamma | \Delta \vdash \texttt{let}\ \Box(t,s)\ \texttt{be}\ \Box(x,y)\ \texttt{in}\ u = u[t/x, s/y]\colon C}$$

$$\frac{\Gamma | \Delta \vdash t\colon \Box_{x\colon A} B \quad \Gamma | \Delta, y\colon \Box_{x\colon A} B \vdash t'\colon C}{\Gamma | \Delta \vdash \texttt{let}\ t\ \texttt{be}\ \Box(x,a)\ \texttt{in}\ t'[\Box(x,a)/y] = t'[t/y]}$$

Fig. 3. Typing Judgements for dependent DIML (depDIML)

tions are derivable. The reduction rules of depDIML are as expected

$$(\lambda x_M\colon A.t)u \Rightarrow t[u/x_M]$$

$$(\lambda x_I\colon A.t)u \Rightarrow t[u/x_I]$$

$$\texttt{let}\ \Box(t,u)\ \texttt{be}\ \Box(x,y)\ \texttt{in}\ s \Rightarrow s[t/x, u/y]$$

However the meta-theory of dependent type theories is usually very subtle. In particular subject reduction becomes an important theorem. Luckily we can use the same methodology that is used to show these theorems for standard dependent type theory. As before we have two kinds of substitution lemma:

Lemma 5.1 (depDIML Substitution) *If* $\Gamma|\Delta \vdash t\colon A$ *and* $\Gamma|x_I\colon A, \Delta \vdash s\colon B$, *then* $\Gamma|\Delta \vdash s[t/x_I]\colon B$, *where* $[t/x_I]$ *denotes the traditional meta-level substitution. Similarly if* $\Gamma|- \vdash t\colon A$ *and* $\Gamma, x_M\colon A|\Delta \vdash s\colon B$ *then* $\Gamma|\Delta \vdash s[t/x_M]\colon B$.

Lemma 5.2 (depDIML Subject Reduction) *If there is a dependent DIML typing judgement* $\Gamma|\Delta \vdash t\colon A$ *and a rewrite* $t \Rightarrow t'$, *then there is also a typing judgement* $\Gamma|\Delta \vdash t'\colon A$.

Theorem 5.3 *The dependent DIML reduction relation* $\Rightarrow$ *is strongly normalising and confluent.*

6 Dependent Fibrational Semantics

The hard work done at the beginning now pays off. The framework presented in Section 4 can be used to model the dependent DIML type theory simply by removing the restriction that re-indexing is the identity on objects.

Definition 6.1 A *fibred dependent DIML-category* is an indexed category $E\colon \mathcal{B}^{op} \to \mathbf{Cat}$ such that

(i) $\mathcal{B}$ has a terminal object $\top$;

(ii) all fibres are cartesian closed categories and $E(f)$ preserves the cartesian closed structure on the nose for every morphism f in $\mathcal{B}$;

(iii) The functor $\mathcal{I}\colon \mathcal{B} \to Gr(E)$ defined by $\mathcal{I}(\Gamma) = (\Gamma, 1)$ and $\mathcal{I}(f) = (f, \mathsf{Id})$ has a right adjoint G. We write $\Gamma \cdot A$ for $G((\Gamma, A))$, $(\mathsf{Fst}, \mathsf{Snd})$ for the co-unit of this adjunction and $\mathtt{pair}(f, g)$ for morphism obtained by transposing (f, g) across the adjunction.

(iv) For every object Γ of $\mathcal{B}$ and A of $E(\Gamma)$, the functor $\mathsf{Fst}^*_A\colon E(\Gamma) \to E(\Gamma \cdot A)$ has a right adjoint $\Pi_A\colon E(\Gamma \cdot A) \to E(\Gamma)$. We will write in the sequel Cur for the natural isomorphism between $\mathrm{Hom}_{E(\Gamma \cdot A)}(\mathsf{Fst}^*_A(B), C)$ and $\mathrm{Hom}_{E(\Gamma)}(B, \Pi_A(C))$ and App for its counit.

(v) The Beck-Chevalley-condition for the adjunctions $\mathsf{Fst}^*_A \dashv \Pi_A$ is satisfied in the strict sense, i.e. the equation $f^*(\mathsf{Cur}_A(h)) = \mathsf{Cur}_A((f \cdot \mathsf{Id})^*(h))$ holds for every $f\colon \Delta \to \Gamma$, $A \in E(\Gamma)$, $B \in E(\Gamma \cdot A)$ and morphism $h\colon 1 \cdot B$ in $E(\Gamma \cdot A)$.

(vi) For all objects Γ of $\mathcal{B}$ the functor $\mathsf{Fst}^*_A\colon E(\Gamma) \to E(\Gamma \cdot A)$ has a left adjoint $\Box_A$. If $f\colon B \to \mathsf{Fst}^*_A C$ is a morphism in $E(\Gamma.A)$, we write $\nu_A(f)$ for the morphism in $E(\Gamma)$ obtained by transposing f across the adjunction.

(vii) Moreover, the Beck-Chevalley condition for the adjunction $\Box_A \dashv \mathsf{Fst}^*_A$ is satisfied in the strict sense, i.e. the equations

$$f^* \circ \Box_A = \Box_A \circ (f \cdot \mathsf{Id})^* \qquad f^*(\nu_A(g)) = \nu_A((f \cdot \mathsf{Id})^*(g))$$

hold for every $f\colon \Delta \to \Gamma$, $B \in ObE(\Gamma)$, $C \in ObE(\Gamma)$ and $g \in \mathrm{Hom}_{E(\Gamma.A)}(B, C)$

Modal variables will be interpreted via the comprehension and intuitionistic variables by suitable projections in the fibre. The type $A \to B$ is interpreted by the function space in the corresponding fibre, and the dependent product by the right adjoint to weakening. The $\Box$-modality is interpreted by the left adjoint to weakening. More precisely, the clauses for the $\Box$-modality are as follows:

$$\frac{[\![\Gamma|\_ \vdash t\colon A]\!] = f \quad [\![\Gamma|\Delta \vdash s\colon B]\!] = g}{[\![\Gamma|\Delta \vdash \Box(t,s)]\!] = \mathtt{pair}(\mathsf{Id}, f)^*(\nu_A^{-1}(\mathsf{Id}) \circ g)}$$

$$\frac{[\![\Gamma|\Delta \vdash t\colon \Box_{x\colon A} B]\!] = f \quad [\![\Gamma, x\colon A|\Delta, y\colon B \vdash t\colon C]\!] = g}{[\![\Gamma|\Delta \vdash \mathtt{let}\ t\ \mathtt{be}\ \Box(x,y)\ \mathtt{in}\ s]\!] = \nu^{-1}(g) \circ \langle f, \mathsf{Id}\rangle}$$

Using this definition and the same method of proof as before we can now state soundness and completeness of our categorical models. The soundness proof uses a substitution lemma.

Lemma 6.2(i) *Assume* $[\![\Gamma|\_ \vdash s\colon A]\!] = f$. *Then we have*

(i) *If* $\Gamma, x\colon A, \Gamma'|\Delta \vdash \mathtt{Context}$, *then also* $[\![\Gamma, \Gamma'[s/x]|\Delta[s/x] \vdash \mathtt{Context}]\!] = (\langle \mathsf{Id}, f\rangle \cdot \mathsf{Id})^*[\![\Gamma, x\colon A, \Gamma'|\Delta \vdash \mathtt{Context}]\!]$.

(ii) *If* $\Gamma, x\colon A, \Gamma'|\Delta \vdash B\ \mathtt{Type}$, *then also* $[\![\Gamma, \Gamma'[s/x]|\Delta[s/x] \vdash B[s/x]\ \mathtt{Type}]\!] = (\langle \mathsf{Id}, f\rangle \cdot \mathsf{Id})^*[\![\Gamma, x\colon A, \Gamma'|\Delta \vdash B\ \mathtt{Type}]\!]$.

(iii) *If* $\Gamma, x\colon A, \Gamma'|\Delta \vdash t\colon B$, *then* $[\![\Gamma, \Gamma'[s/x]|\Delta[s/x] \vdash s[tx]\colon B[s/x]]\!] = (\langle \mathsf{Id}, f\rangle \cdot \mathsf{Id})^*[\![\Gamma, x\colon A, \Gamma'|\Delta \vdash t\colon B]\!]$.

(ii) *Assume* $[\![\Gamma|\Delta \vdash s\colon A]\!] = f$. *Then we have* $[\![\Gamma|\Delta \vdash t[s/y]\colon B]\!] = [\![\Gamma|\Delta, y\colon A \vdash t\colon B]\!] \circ \langle \mathsf{Id}, f\rangle$.

Theorem 6.3 *Given a fibred dependent DIML-category* $E\colon \mathcal{B}^{op} \to \mathbf{Cat}$ *under the above interpretation* $[\![\,]\!]$ *the following facts hold.*

(i) *Assume* $\Gamma|\Delta \vdash \mathtt{Context}$. *Then* $[\![\Gamma|\Delta \vdash \mathtt{Context}]\!]$ *is an object in* $E([\![\Gamma]\!])$;

(ii) *Assume* $\Gamma \vdash A\ \mathtt{Type}$. *Then* $[\![\Gamma \vdash A\ \mathtt{Type}]\!]$ *is an object in* $E([\![\Gamma]\!])$;

(iii) *Assume* $\Gamma|\Delta \vdash t\colon A$. *Then* $[\![\Gamma \vdash t\colon A]\!]$ *is a morphism from* $[\![\Delta]\!]$ *to* $[\![A]\!]$ *in* $E([\![\Gamma]\!])$;

(iv) *Assume* $\vdash \Gamma|\Delta = \Gamma'|\Delta'$. *Then* $[\![\Gamma|\Delta]\!] = [\![\Gamma'|\Delta']\!]$;

(v) *Assume* $\Gamma \vdash A = B$. *Then* $[\![\Gamma \vdash A\ \mathtt{Type}]\!] = [\![\Gamma \vdash B\ \mathtt{Type}]\!]$;

(vi) *Assume* $\Gamma|\Delta \vdash t = s\colon A$. *Then* $[\![\Gamma|\Delta \vdash t\colon A]\!] = [\![\Gamma|\Delta \vdash s\colon A]\!]$.

Proof. The proof proceeds by simultaneous induction over the derivation of all judgements, using the substitution lemma 6.2. □

Theorem 6.4 *If* $[\![\Gamma|\Delta \vdash t\colon A]\!] = [\![\Gamma \vdash s\colon A]\!]$ *where* $[\![\ ]\!]$ *is the above defined interpretation, for every fibred dependent DIML-category* $E\colon \mathcal{B}^{op} \to \mathbf{Cat}$ *and for every derived sequents* $\Gamma|\Delta \vdash t\colon A$ *and* $\Gamma|\Delta \vdash s\colon A$ *then we can derive in the type theory* $\Gamma|\Delta \vdash t = s\colon A$.

Proof. The proof proceeds by constructing a term model from a dependent DIML-theory. Since the interpretation of this dependent DIML-theory in the syntactic model is the identity, completeness immediately follows. □

This framework also supports the addition of extensional identity types as left adjoint to the diagonal. However, the precise formulation of the corresponding type theory is known to be rather subtle and will be left for further work.

There has been renewed interest in DTT from a more foundational perspective recently due to Univalent Foundations Program of Homotopy Type Theory [30]. We do not have much to say about the program, but it is easy to agree with Voevodsky's diagnosis (in http://bit.ly/1sLF0i3) that:

> There is a substantial difficulty in adding the rules for universes to the rules for the dependent products, dependent sums and identity types. These three groups of rules are independent from each other and can be studied separately. The rules for a universe connect the rules from these three groups to each other making it necessary to coordinate their interpretations.

Given the work just presented, we would like to add that there seem to be also *modal rules*, a fifth class of rules, that appear to be independent from the other sets of rules as well. For a fragment of these (the box-only S4 modal rules), together with dependent products we can provide well-behaved type theories as well as well-behaved categorical semantics in terms of (split indexed) fibrations. We do not know whether it helps or hinders the Homotopy Type Theory program.

7 Related and Future Work

Awodey and Bauer consider a system of dependent types with a modality, the so-called "bracket types" in [3]. This unary type constructor $[A]$ was previously considered, as a way of erasing computational content, and formalizing a notion of proof irrelevance in dependent type theory. Considered as a modality, the bracketting of types is a very special example, idempotent and with extremely degenerate type theoretic properties of a diamond modality, definitely not a paradigmatic one. Awodey and Bauer also mention, as future work, extending the work on Moggi's modal type theory to dependent types, which is what we have done here. Xi and Pfenning started a collection of work on a dependent version of ML (DML [32]), with the goal of "practical dependent types". This line of work, where the types are dependent on a collection of indices, seems very interesting, however our goals are somewhat orthogonal to theirs.

Recent work of Krishnaswami, Pradic and Benton [22] combines dependent and linear types into a single system. Like we did here they use the dual context calculus of Benton to set up this integration. But since they are more interested in the application of the calculus to a "proof-theoretic account of imperative programming" they have to deal with a series of extensions that do not concern us here. We expect that some of these extensions can be modelled in our framework as well, but have not attempted to do so, yet.

There are other ways of modelling dependent types categorically, e.g., categories with families [13] which are very similar to D-categories. The extension necessary to model the modalities can be done in the same way for categories with families. We should also mention the work of Vákár [31], which starts from similar sources to ours. We actually could have done this kind of modelling in the late nineties, but instead decided to publish the work on ILT [23], because dependent types were not our priority at the time, while re-using the libraries already in place for the simply typed lambda-calculus was, given our goals in the xSLAM project http://www.cs.bham.ac.uk/research/xslam/xslam.html. Now our priorities are related to modal types, more than linear types, but the technology is the same and it was in fact developed for the linear types.

One of our motivations when starting this work was to understand better the limitations of the Curry-Howard correspondence, when it comes to intensional constructors like the modalities. For years we have been involved in the series of workshops called "Intuitionistic Modal Logic and Applications" (IMLA) whose goal is to try to get modal logicians talking to computer scientists, especially functional programmers, so that instead of re-inventing each other's wheels they could benefit from each other's work. There is nowadays more awareness that the problems are common, but still very little substantial incorporation of each other's results. Modal logicians are bound to say that the kind of constructive necessity discussed in this note is too limited to be of any use to them. Similarly functional programmers interested in applied dependent types might say that we have not even scratched the surface of what is required to make these dependent constructive necessity modalities useful to them. This may be so, but we hope that the mathematics developed might be useful for others seeking to understand how fragments of dependent type theories relate to each other. At least it shows that for modalities in the style of S4-necessity, there are no problems either in producing well-behaved type theories or developing appropriate categorical models.

8 Conclusions

We have described modal type theories (a non-dependent) DIML and a (dependent) depDIML for the operator corresponding to constructive necessity in the logic. We showed that these type theories are well-behaved proof-theoretically: their reduction calculus satisfies subject reduction, is confluent and strongly normalising, with modular proofs. We provided fibrational models for both of these modal type theories, which we proved sound and complete, using some established and trusted technology. To sum up, this version of constructive necessity is extremely well-behaved, as witnessed by the fact that both the type-theoretical properties as well as the categorical semantics are modular and parallel to each other. Moreover, explicit substitutions [1] can be added to both dependent and non-dependent versions of the type theory, with appropriate operational semantics, if desired. We left to future work to decide whether this modal type theory is useful for Homotopy Type Theory.

References

[1] Martın Abadi, Luca Cardelli, Pierre-Louis Curien, and Jean-Jacques Lévy. Explicit substitutions. In *Journal of Functional Programming*, volume 6 (2), pages 299–327, 1996.

[2] Natasha Alechina, Michael Mendler, Valeria de Paiva, and Eike Ritter. Categorical and kripke semantics for constructive s4 modal logic. In Laurent Fribourg, editor, *Computer Science Logic*, volume 2142 of *Lecture Notes in Computer Science*, pages 292–307. Springer Berlin Heidelberg, 2001.

[3] Steven Awodey and Andrej Bauer. Propositions as [types]. *Journal of Logic and Computation*, 14(4):447–471, 2004.

[4] Andrew Barber. *Dual intuitionistic linear logic, PhD Thesis*. University of Edinburgh, Department of Computer Science, Laboratory for Foundations of Computer Science, 1996.

[5] Henk Barendregt. Lambda calculi with types. In D. Gabbay S. Abramsky and T. Maibaum, editors, *Handbook of Logic in Computer Science*. Oxford Science Publications, 1992.

[6] P Benton. A mixed linear and non-linear logic: Proofs, terms and models. *Computer Science Logic*, pages 121–135, 1995.

[7] P Nick Benton, Gavin M. Bierman, and Valeria CV De Paiva. Computational types from a logical perspective. *Journal of Functional Programming*, 8(2):177–193, 1998.

[8] Gavin M. Bierman and Valeria CV de Paiva. On an intuitionistic modal logic. *Studia Logica*, 65(3):383–416, 2000.

[9] Valentijn Anton Johan Borghuis. *Coming to terms with modal logic: on the interpretation of modalities in typed λ-calculus.* Eindhoven: Tech. Univ. Eindhoven, 1994.

[10] Chris Casinghino, Vilhelm Sjöberg, and Stephanie Weirich. Combining proofs and programs in a dependently typed language. In *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '14, pages 33–45, New York, NY, USA, 2014. ACM.

[11] Rowan Davies and Frank Pfenning. A modal analysis of staged computation. *Journal of the ACM (JACM)*, 48(3):555–604, 2001.

[12] Valeria de Paiva, Rajeev Goré, and Michael Mendler. Modalities in constructive logics and type theories, 2004.

[13] Peter Dybjer. Internal type theory. In *Types for Proofs and Programs*, number 1158 in LNCS, pages 120–134. Springer Verlag, 2005.

[14] Thomas Ehrhard. A categorical semantics of constructions. In *Proceedings of the Third Annual Symposium on Logic in Computer Science*. Logic in Computer Science, 1988. LICS'88., Proceedings of the Third Annual Symposium on, 1988.

[15] WB Ewald. Intuitionistic tense and modal logic. *The Journal of Symbolic Logic*, 51(01):166–179, 1986.

[16] Matt Fairtlough and Michael Mendler. Propositional lax logic. *Information and Computation*, 137(1):1 – 33, 1997.

[17] G Fisher-Servi. Semantics for a class of intuitionistic modal logic. in dalla chiara, ml (cur.). *Italian Studies in the Philosophy of Science*, pages 59–72, 1981.

[18] Frederick B Fitch. Intuitionistic modal logic with quantifiers. *Portugaliae mathematica*, 7(2):113–118, 1948.

[19] Neil Ghani, Valeria de Paiva, and Eike Ritter. Explicit substitutions for constructive necessity. In KimG. Larsen, Sven Skyum, and Glynn Winskel, editors, *Automata, Languages and Programming*, volume 1443 of *Lecture Notes in Computer Science*, pages 743–754. Springer Berlin Heidelberg, 1998.

[20] Bart Jacobs. *Categorical Logic and Type Theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1999.

[21] Satoshi Kobayashi. Monad as modality. *Theoretical Computer Science*, 175(1):29 – 74, 1997.

[22] Neelakantan R. Krishnaswami, Pierre Pradic, and Nick Benton. Integrating linear and dependent types. In *Proceedings of the 42Nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '15, pages 17–30, New York, NY, USA, 2015. ACM.

[23] Maria Emilia Maietti, Valeria De Paiva, and Eike Ritter. Categorical models for intuitionistic and linear type theory. In *Foundations of Software Science and Computation Structures*, pages 223–237. Springer Berlin Heidelberg, 2000.

[24] Eugenio Moggi. Notions of computation and monads. *Information and computation*, 93(1):55–92, 1991.

[25] Frank Pfenning and Rowan Davies. A judgmental reconstruction of modal logic. *Mathematical structures in computer science*, 11(04):511–540, 2001.

[26] D. Prawitz. Natural deduction: A proof theoretical study. *Almqvist˜ Wiksell, Stockholm*, 1965.

[27] Eike Ritter. *Categorical combinators for the calculus of constructions.* University of Cambridge, Computer Laboratory, 1990.

[28] Eike Ritter. Categorical abstract machines for higher-order typed λ-calculi. *Theoretical Computer Science*, 136(1):125–162, 1994.

[29] Alex K Simpson. The proof theory and semantics of intuitionistic modal logic. *University of Edinburgh. College of Science and Engineering. School of Informatics.*, 1994.

[30] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics.* http://homotopytypetheory.org/book, Institute for Advanced Study, 2013.

[31] M. Vákár. Syntax and semantics of linear dependent types. Manuscript, 2014.

[32] Hongwei Xi and Frank Pfenning. Dependent types in practical programming. In *Proceedings of the 26th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 214–227. ACM, 1999.