Revisiting the Logical Framework for Locally Cartesian Closed Categories

Zhixuan Yang

***Abstract*—In categorical logic, the construction of *classifying categories* for various notions of algebraic theories bridges syntactic presentations of algebraic theories and semantic categorical structures. In this paper, we give a detailed proof of the construction of *classifying locally cartesian closed categories* for syntactic theories defined in a logical framework that has been previously used to study meta-theoretic properties of type theories and programming languages.**

***Index Terms*—logical frameworks, categorical logic, type theory, locally cartesian closed categories**

I. Introduction

A. Presentations of Locally Cartesian Closed Categories

A well known result in categorical logic due to Seely and Hofmann is that *extensional Martin-Löf type theory* can be interpreted in any *locally cartesian closed categories* (LCCCs) [58, 30, 14], similar to how finitary algebraic theories can be interpreted in any finitely complete categories, and how simply typed λ-calculus can be interpreted in any cartesian closed categories. Moreover, Clairambault and Dybjer showed that there is a biequivalence

$$\mathrm{LCCC} \underset{\longrightarrow}{\overset{\longleftarrow}{\cong}} \mathrm{CwF}^{\mathrm{I}_{\mathrm{ext}}\Sigma\Pi}_{\mathrm{dem}} \tag{1}$$

between LCCCs and *democratic categories with families with* Σ, Π *and extensional equality types*, which can be seen as an algebraic formulation of type theories extending extensional Martin-Löf type theory with type and term constants [11]. The main practical consequence of this biequivalence is that if we have an LCCC $\mathscr{C}$, then we can manipulate constructions in $\mathscr{C}$ using the convenient type-theoretic language afforded by its corresponding CwF, i.e. the internal language of $\mathscr{C}$.

However, if our task is to *define* an LCCC, the biequivalence between LCCCs and $\mathrm{CwF}^{\mathrm{I}_{\mathrm{ext}}\Sigma\Pi}_{\mathrm{dem}}$'s (1) cannot help us much: to define a CwF, we need to specify its contexts, substitution, types, terms, etc, which are not easier to present than the data of an LCCC. Instead, what we usually want is to present a LCCC or a CwF using some generators and equations, similarly to, for example, how we can present the Abelian group $\langle \mathbb{Z}, +, 0\rangle$ with one generator 1, or how we present the tensor $A \otimes B$ of two Abelian groups by generators $(a, b) \in A \times B$ and equations

$$(a_1, b) + (a_2, b) = (a_1 + a_2, b)$$
$$(a, b_1) + (a, b_2) = (a, b_1 + b_2).$$

Unsurprisingly, there are already general frameworks to present LCCCs. One of them is *LCC-sketches*, a special case of Kinoshita et al.'s general framework of T*-sketches* for finitary (enriched) monads $T : \mathscr{C} \to \mathscr{C}$ on a locally finitely presentable enriched category $\mathscr{C}$ [41]. LCC-sketches are the case where T is the free-LCCC monad over the (groupoid-enriched) category $\mathrm{CAT}_{\cong}$ of small categories, functors, and natural isomorphisms. An exposition of LCC-sketches and their application in the study of type theories are given by Gratzer and Sterling [24].

Another approach to present LCCCs/CwFs is using syntactic *logical frameworks* (LFs) [28, 51, 67, 54, 9], which are type theories designed for presenting logics/theories. In particular, Sterling introduced a logical framework in his thesis [60, Chapter 1]. Sterling did not name this LF, and for convenience we will refer to it by LccLF in this paper. Roughly, LccLF is a dependent type theory with

- a universe $\mathbb{J}$ closed under the type connectives of extensional Martin-Löf type theory (the unit type, Σ, Π, and extensional equality types), and
- type connectives outside $\mathbb{J}$ are restricted so that logics/theories definable in this LF present LCCCs.

To define a theory using LccLF is exactly to define a context (or equivalently a closed type) in LccLF. For example,

- the context $A : \mathbb{J}$ is the theory with one generating object;
- the context $(A : \mathbb{J}, \mu : A \times A \to A)$ is the last theory with additionally a generating binary operation μ;
- the context $(A : \mathbb{J}, \mu : A \times A \to A, \alpha : (a, b, c : A) \to (\mu\ a\ (\mu\ b\ c)) = \mu\ (\mu\ a\ b)\ c)$ is the last theory with additionally the associativity law. The equality sign here denotes the extensional equality type on A.

Syntactic LFs (such as LccLF) have two advantages over categorical frameworks (such as sketches). Firstly, they are especially intuitive for presenting the syntactic category of logics or programming languages following Harper et al.'s *judgements-as-types* principle [28], of which we will see a number of examples later. Secondly, they work seamlessly with internal languages – if $\mathscr{C}$ is an LCCC with a *universe* U [7, 47, 64], to construct a model of the theory S in $\mathscr{C}$, it is sufficient to define each variable of S in the internal language of $\mathscr{C}$ with $\mathbb{J}$ replaced by U. In this way, both defining theories and constructing models are done in a type-theoretic language. For the reasons, LccLF has been used by several authors to study the meta-theoretic properties of a number of type theories and programming languages [63, 61, 26, 49, 62].

B. Contributions

However, the status of LccLF in the literature is still a 'ghost': the syntax of LccLF is given in Sterling's thesis; there is a consensus on how it is *supposed* to work, and people have built useful work on top of it, but the actual mathematics 'implementing' LccLF is still missing in the literature.

The purpose of the present paper is to fill the gap. In particular, for every theory S in LCCLF and LCCC $\mathscr{C}$, we will define a groupoid $S\text{-}\textsc{Mod}(\mathscr{C})$ of models of S in $\mathscr{C}$ and isomorphisms of models. We will then prove the *raison d'être* theorem of LCCLF: there is an LCCC $\textsc{Jdg}\, S$ and an equivalence of groupoids

$$S\text{-}\textsc{Mod}(\mathscr{C}) \;\cong\; \textsc{LCCC}_{\cong}(\textsc{Jdg}\, S, \mathscr{C}) \tag{2}$$

between the groupoid of S-models in $\mathscr{C}$ and the groupoid of LCC-functors from $\textsc{Jdg}\, S$ to $\mathscr{C}$ and natural isomorphisms between those functors. In other words, $\textsc{Jdg}\, S$ is the *classifying* locally cartesian closed category for the theory S.

In the course of this paper, we will also show a number of examples from programming language theory, so this paper may also serve as a tutorial of how to use logical frameworks similar to LCCLF to study programming languages.

C. Related Work

Beside the work on sketches [41, 24, 71], the most related work is other 'semantic logical frameworks' in the terminology of Harper [27] that permit (not necessarily decidable) equations in the user-defined theories and highlight semantic models of the theories. Examples of such semantic LFs are (first-order) algebraic theories (in the sense of universal algebra) [12, 1], second-order algebraic theories [18, 19, 20], generalised algebraic theories [9], second-order generalised algebraic theories (SOGATs) [67, 68, 36].

Among them, Uemura's framework for SOGATs is the closest to LCCLF that we study in this paper. There are two crucial differences between SOGATs and LCCLF. Firstly, in an SOGAT, only a set of distinguished judgements (e.g. x being a term of type A) can be hypothesised by other judgements (i.e. in the context of other judgements), whereas LCCLF allows unrestricted hypothesising. Categorically, SOGATs correspond to *categories with representable maps*, or CwRs, which are finitely-complete categories equipped with a (pullback-stable) family of morphisms along which Π-types exist, whereas in an LCCC, Π-types along every morphism exist. Secondly, Uemura essentially gave two notions of models of an SOGAT: maps between CwRs (called *second-order models* by Kaposi and Xie [36]) and CwF-style models (called *first-order models* by Kaposi and Xie and just *models* by Uemura himself). The models for LCCLF that we consider in this paper correspond to second-order models of SOGATs.

D. Outline

In Section II, we introduce the syntax of LCCLF and explain how a programming language can be defined as a *theory/signature* in LCCLF (which is formally a context in LCCLF) following the motto of *judgements as types* [28].

In Section III, we describe the *category of judgements* $\textsc{Jdg}\, S$ of a signature S. Functors $M : \textsc{Jdg}\, S \to \mathscr{C}$ into an LCCC $\mathscr{C}$ preserving the locally cartesian closed structure then provide *functorial models* of S in $\mathscr{C}$ in the tradition of functorial semantics pioneered by Lawvere [44].

In Section IV, we define a notion of *(diagrammatic) models* of an LCCLF-signature S. Unlike a functorial model $M : \textsc{Jdg}\, S \to \mathscr{C}$, which specifies the interpretation of *all* the judgements generated by the signature S in a coherent way, a diagrammatic model only needs to specify the interpretation of the generating operations in the signature S, so diagrammatic models are the notion that we actually want to work with when concretely defining a model of a signature. Due to type dependency in LCCLF, the definition of diagrammatic models is much more involved than that of, e.g. finitary algebraic theories. The main technical tool that we will need is the concept of *universes* in categories, and the theorem that for every LCCC $\mathscr{C}$, there is a universe in $\textsc{Pr}\,\mathscr{C}$ that classifies the Yoneda embedding of objects and morphisms of $\mathscr{C}$.

In Section V, we the desired equivalence of groupoids $S\text{-}\textsc{Mod}(\mathscr{C}) \cong \textsc{LCCC}_{\cong}(\textsc{Jdg}\, S, \mathscr{C})$, where $S\text{-}\textsc{Mod}(\mathscr{C})$ contains diagrammatic models of a signature S and isomorphisms between them, and $\textsc{LCCC}_{\cong}(\textsc{Jdg}\, S, \mathscr{C})$ is the groupoid of functorial models and natural isomorphisms. To define the groupoid $S\text{-}\textsc{Mod}(\mathscr{C})$, a technique similar to, but at one dimensional higher than, Altenkirch et al.'s [4] syntactic translation for *setoid type theory* is used. The reason that only isomorphisms instead of homomorphisms are considered that LCCLF has dependent function types so type expressions are not necessarily covariant with respect to variables in it.

In Section VI, we discuss the related work and conclude.

Assumption. In this paper, we will assume familiarity with (extensional) dependent type theory and its categorical semantics in the form of category with families [10, 16]. An excellent exposition on dependent type theory is Angiuli and Gratzer's book [5]; an elementary account of the categorical semantics is Hofmann [31], which is sufficient for our needs in this paper; a comprehensive textbook account is Jacobs [35].

II. Syntax of the Logical Framework

In brief, the logical framework LCCLF is a dependent type theory with the unit type 1, Σ-types, and a Tarski-style universe type $\mathbb{J}$ such that

1) the universe $\mathbb{J}$ is closed under the unit type and Σ-types;
2) the universe $\mathbb{J}$ is closed under *extensional* equality types;
3) there are Π-types $\Pi\ A\ B$ provided that A is in the universe $\mathbb{J}$. If the codomain B is a type family valued in $\mathbb{J}$, the Π-type $\Pi\ A\ B$ is also in $\mathbb{J}$.

In other words, $\mathbb{J}$ is a universe having all the connectives of extensional Martin-Löf type theory (MLTT), while types outside $\mathbb{J}$ have only the unit type, Σ-types, and restricted Π-types whose domain must be in $\mathbb{J}$.

The precise type formation rules of the logical framework are in Figure 1, and the term formation rules for the universe $\mathbb{J}$ are in Figure 2. All other rules, including the rules for contexts, substitutions, term formations, and judgemental equalities (β and η equalities for all type formers) are the same as the usual extensional MLTT [31, 51, 46, 5] and thus omitted here.

Remark 1. There is a subtle difference between the logical framework defined here and the one in Sterling's thesis [60]: op.

$$\frac{}{\Gamma \vdash 1 \text{ type}} \qquad \frac{\Gamma \vdash A \text{ type} \qquad \Gamma, a : A \vdash B \text{ type}}{\Gamma \vdash \Sigma\ A\ B \text{ type}}$$

$$\frac{}{\Gamma \vdash \mathbb{J} \text{ type}} \qquad \frac{\Gamma \vdash A : \mathbb{J}}{\Gamma \vdash \mathrm{El}\ A \text{ type}}$$

$$\frac{\Gamma \vdash A : \mathbb{J} \qquad \Gamma \vdash a : \mathrm{El}\ A \qquad \Gamma \vdash b : \mathrm{El}\ A}{\Gamma \vdash \mathrm{Eq}(a, b) \text{ type}}$$

$$\frac{\Gamma \vdash A : \mathbb{J} \qquad \Gamma, a : \mathrm{El}\ A \vdash B \text{ type}}{\Gamma \vdash \Pi\ A\ B \text{ type}}$$

Fig. 1. Type formation rules for the logical framework

$$\frac{}{\Gamma \vdash \widehat{1} : \mathbb{J}} \qquad \frac{\Gamma \vdash A : \mathbb{J} \qquad \Gamma, a : \mathrm{El}\ A \vdash B : \mathbb{J}}{\Gamma \vdash \widehat{\Sigma}\ A\ B : \mathbb{J}}$$

$$\frac{\Gamma \vdash A : \mathbb{J} \qquad \Gamma \vdash a : \mathrm{El}\ A \qquad \Gamma \vdash b : \mathrm{El}\ B}{\Gamma \vdash \widehat{\mathrm{Eq}}(a, b) : \mathbb{J}}$$

$$\frac{\Gamma \vdash A : \mathbb{J} \qquad \Gamma, a : \mathrm{El}\ A \vdash B : \mathbb{J}}{\Gamma \vdash \widehat{\Pi}\ A\ B : \mathbb{J}}$$

Fig. 2. Codes of types in the universe $\mathbb{J}$

cit. *all types* have extensional equality types, not just those in $\mathbb{J}$. Those extensional equality types play the role of *sort equations* in Cartmell's generalised algebraic theories [9]. Although they are sometimes handy when specifying type theories in the LF, they necessitate considerations of strict equalities between objects in a category when considering categorical models of theories specified in the LF, making the notion of models not invariant under equivalences of categories. They also complicate the definition of isomorphisms of models. For these reasons, they are left out in the LF here.

Notation 2. To make working with type theories as natural as working with ordinary maths, we impose the following conventions, which resemble the concrete syntax of Agda [52].

- Dependent function types, i.e. Π-types, are written as $(a : A) \to B$, or $A \to B$ when B does not depend on A. We will use *implicit function types* $\{a : A\} \to B$, whose function application and abstraction are elided when they can be inferred or have a unique choice.
- Dependent pair types, i.e. Σ-types, are written as usual as $\Sigma(a : A).\ B$, or $A \times B$ if B does not depend on A. Pairing is (a, b) and projections are $\pi_1\ p$ and $\pi_2\ p$. We also use the record syntax for iterative Σ-types.
- We will use the same notation for type formers and their codes in universes. The decoding operator El of universes will be elided, as if we are working with Russell-style universes.
- The extensional equality type $\mathrm{Eq}(a, b)$ will be written as simply $a = b$, and its only constructor is $\mathit{refl} : a = a$.
- An identifier that contains underscores '_' is used as an operator. For example, if $\_+\_ : A \to A \to A$, we can write $a + b$ for $a, b : A$. However, a single underscore '_' that appears alone just means a 'wildcard' that take the place of something inferable or irrelevant.

A type theory is defined in the logical framework as a context, or equivalently a closed type since an LF-context $(a_1 : A_1, \ldots, a_n : A_n)$ can be packed into a record type with fields $a_i : A_i$. The idea is the *judgements-as-types* principle of the Edinburgh Logical Framework [28]: *judgements* of the object type theory (e.g. something being a type) are declared as types in the universe $\mathbb{J}$ in the logical framework; *inference rules* are declared as functions between judgements; *deductions* are then terms of judgements that make use of the previously declared judgements and inference rules.

To avoid confusion with concepts in object type theories, we will call LF contexts *signatures* or *theories*. In the study of algebra theories, typically a signature means the type of the operations, and a theory means a signature plus equations, but in LccLF equations are declared using equality types, so 'theories' and 'signatures' can be used interchangeably. The variables of an LF context are referred to as *declarations* of the signature. LF types in the universe $\mathbb{J}$ will be called *judgements*.

Example 3. The signature of *barebone type theory*,

$$ty : \mathbb{J} \qquad\qquad tm : ty \to \mathbb{J}$$

has two declarations which are respectively the judgement for *something being a type* and the family of judgements for *something being a term of a type*. This signature alone is not very interesting but it serves as a basic building block for more complex type theories.

In the traditional turnstile-and-gamma presentation, elements $a : ty$ and $t : tm\ a$ of these judgements would be written as

$$\Gamma \vdash a \text{ type} \qquad\qquad \Gamma \vdash t : a$$

Note that the LF presentation of ty and tm does not have explicit contexts Γ because contexts can be handled using function abstraction in the LF – the so-called *higher-order abstract syntax* (HOAS). If $a : ty$, the judgement $x : a \vdash x : a$ in the traditional presentation would be represented as the function $\lambda(x : tm\ a).\ x$ of type $tm\ a \to tm\ a$ in the LF.

Example 4. The signature of *simply typed λ-calculus* (STLC) extends barebone type theory (Example 3) with the following:

$$\begin{gathered}
\iota : ty \qquad\qquad _\Rightarrow_ : ty \to ty \to ty \\
abs : \{a, b : ty\} \to (tm\ a \to tm\ b) \to tm\ (a \Rightarrow b) \\
app : \{a, b : ty\} \to tm\ (a \Rightarrow b) \to (tm\ a \to tm\ b) \\
_ : \{a, b : ty\} \to \{f : tm\ a \to tm\ b\} \to app\ (abs\ f) = f \\
_ : \{a, b : ty\} \to \{g : tm\ (a \Rightarrow b)\} \to abs\ (app\ g) = g
\end{gathered}$$

The declaration $\iota : ty$ corresponds to the inference rule of a base type in STLC, and the declaration $\_\Rightarrow\_$ corresponds to the inference rule of (non-dependent) function types. Terms of

function types are specified using HOAS via an isomorphism with the function space in the logical framework. With these declarations, we can define STLC terms such as

$$abs\ (\lambda f.\ abs\ (\lambda x.\ app\ f\ (app\ f\ x))) : tm\ ((\iota \Rightarrow \iota) \Rightarrow (\iota \Rightarrow \iota)).$$

Let us compare the LF presentation of a type theory and the traditional gamma-and-turnstile presentation.

(1) Σ-types in the LF can be used to pack two judgements together. This is implicit in traditional presentations. For example, an LF-function $(\Sigma(a : ty).\ tm\ a) \to J$ is just

$$\frac{a\ \text{type} \qquad t : a}{J}$$

(2) Equality types of the LF are used to specify the equational theory of the object type theory, and since equality types are respected by all constructions of the LF, there is no need to manually specify any congruence rules.

(3) A more noticeable difference is that dependent function types in the universe $\mathbb{J}$ (which may be called *higher-order judgements* as we call elements of $\mathbb{J}$ judgements) uniformly handles two different things in traditional presentations: *contexts of hypotheses* $\Gamma \vdash J$ and *schematic inference rules* $\frac{J}{K}$. Taking the rule of function abstraction in STLC for example, the traditional presentation is

$$\frac{\Gamma, x : a \vdash t : b}{\Gamma \vdash \lambda x.\ t : a \Rightarrow b}$$

while in the LF presentation (Example 4), this rule is

$$abs : \{\, a, b : ty \,\} \to (tm\ a \to tm\ b) \to tm\ (a \Rightarrow b)$$

The higher-order judgement $tm\ a \to tm\ b$ corresponds to a deduction of $t : b$ with a new hypothesis $x : a$ in the context, and $(tm\ a \to tm\ b) \to tm\ (a \Rightarrow b)$ corresponds to the inference rule. In the traditional presentation, contexts Γ can only contain certain basic judgements, such as $x : a$, but not

$$\Gamma, x : (a\ \text{type}) \vdash \cdots \quad \text{or} \quad \Gamma, x : (\Gamma, x : a \vdash t : b) \vdash \cdots,$$

so there need to be two layers of entailment, $\Gamma \vdash J$ and $\frac{J}{K}$. In contrast, both of them are handled as higher-order judgements in the logical framework.

However, unless we take the LF presentation of a type theory as the definitive formulation of the type theory, the existence of the higher-order judgements in the LF raises the question that whether a type theory defined in the LF is the same as its traditional presentation, since higher-order judgements *a priori* may introduce new terms to base judgements. This question is called the *adequacy* of LF presentations [28]. Gratzer and Sterling [24] showed the adequacy of LCC-sketches with respect to Uemura's logical framework [67, 68], which can be seen as a faithful formulation of the traditional presentations of type theories. Their gluing argument should be able to be adapted to LCCLF but we will not go into this in this paper.

Example 5. It is a common pattern that a type former in the object type theory is specified by internalising an LF judgement via an isomorphism, which is precisely the purpose of introducing logical frameworks. Thus for convenience we define the judgement of isomorphisms given $A, B : \mathbb{J}$ as

$$\begin{aligned}
&\textsc{record}\ A \cong B : \mathbb{J}\ \textsc{where}\\
&\quad fwd : A \to B\\
&\quad bwd : B \to A\\
&\quad _ \quad : (a : A) \to bwd\ (fwd\ a) = a\\
&\quad _ \quad : (b : B) \to fwd\ (bwd\ b) = b
\end{aligned}$$

Using isomorphisms to LF types to specify object type theories does not entail that object type theories are restricted to sublanguages of the logical framework. The following two examples shows how general recursion and impredicative polymorphism can be specified in this way, although LCCLF does not have general recursion or any impredicativity.

Example 6. The signature of PCF [56] extends STLC in Example 4 with a fixed-point combinator at every type

$$Y : \{\, a : ty \,\} \to (tm\ a \to tm\ a) \to tm\ a$$

as well as some new base types and terms

$$\begin{gathered}
0 : tm\ \iota \qquad\qquad succ, pred : tm\ \iota \to tm\ \iota\\
o : ty \qquad tt, ff : tm\ o \qquad iszero : tm\ \iota \to tm\ o\\
\supset : \{\, a : ty \,\} \to tm\ o \to tm\ a \to tm\ a
\end{gathered}$$

and also the following equational declarations (whose names are irrelevant and omitted):

$$\begin{gathered}
\{\, n : tm\ \iota \,\} \to pred\ (succ\ n) = n\\
pred\ 0 = 0 \qquad\qquad iszero\ 0 = tt\\
\{\, n : tm\ \iota \,\} \to iszero\ (succ\ n) = ff\\
\{\, a : ty \,\}\ \{\, x, y : tm\ a \,\} \to (\supset\ tt\ x\ y = x)\ \times\ (\supset\ ff\ x\ y = y)\\
\{\, a : ty \,\} \to \{f : tm\ a \to tm\ a\} \to f\ (Y\ f) = Y\ f
\end{gathered}$$

Within the logical framework, in the context of this signature, we can write programs such as addition of numbers:

$$\begin{aligned}
&_+_ : tm\ (\iota \Rightarrow \iota \Rightarrow \iota)\\
&_+_ = abs\ (\lambda n.\ Y\ (\lambda rec.\ abs\ (\lambda m.\\
&\qquad \supset\ (iszero\ m)\ n\ (succ\ (app\ rec\ (pred\ m))))))
\end{aligned}$$

The equational axioms in the signature implies, for example, that $0 + succ\ 0$ is judgementally equal to $succ\ 0$ in the logical framework. Note that in this way PCF is formulated as an equational theory rather than a reduction system of terms, viz a small-step operational semantics.

Example 7. The signature of *System F* [21, 57] extends the one of STLC in Example 4 with the following declarations:

$$\begin{gathered}
\forall : (ty \to ty) \to ty\\
\forall\text{-}iso : \{\, A : ty \to ty \,\} \to tm\ (\forall\ A) \cong ((\alpha : ty) \to tm\ (A\ \alpha))
\end{gathered}$$

where $\_\cong\_$ is the judgement of isomorphisms (Example 5). Since polymorphic (and ordinary) functions in this signature are specified by function types of the logical framework, they inherit the β and η equalities of LF function types. As an example of terms of System F, letting $Abs = \forall\text{-}iso.bwd$, we can define Church numerals:

$$CNum : ty$$
$$CNum = \forall\ (\lambda a.\ (a \Rightarrow a) \Rightarrow a \Rightarrow a)$$
$$C_2 : tm\ CNum$$
$$C_2 = Abs\ (\lambda a.\ abs\ (\lambda f.\ abs\ (\lambda x.\ app\ f\ (app\ f\ x))))$$

Remark 8. Because of *extensional* equality types in $\mathbb{J}$, LCCLF does not enjoy decidable type checking, this is not a problem for using LCCLF as a paper-and-pencil logical framework. However, if we are after mechanically checking the signatures and terms defined in LCCLF, we can 'approximate' LCCLF using existing proof assistants implementing intensional equality types, such as Agda, with the axioms of uniqueness of identity proofs (UIP) and function extensionality (FUNEXT) as postulates, since extensional Martin-Löf type theory is conservative with respect to intensional Martin-Löf type theory augmented with UIP and FUNEXT [29, 39]. The (unavoidable) cost of this is that some extra transportations along intensional equalities must be inserted manually.

III. Functorial Semantics of Signatures

Type theories, like other flavours of algebraic theories, or languages in general, are invented for *talking about things*, either mathematical objects or intuitive objects in 'the physical world'. Therefore a logical framework ought to provide a notion of *models* of theories defined in it. In this section, we show how this is done for LCCLF by means of *functorial semantics* à la Lawvere theories [44], except that categories with finite products are replaced by locally cartesian closed categories.

A. CwFs with the LF Connectives

The syntax of LCCLF, quotiented by judgemental equalities, forms a *category with families* (CwF) with the extra structures in Figure 1, henceforth called an LF-CwF. It can further be proven to be *initial* among all LF-CwFs, similarly to the initiality results of many other dependent type theories in the literature [15, 65, 38, 55]. In outline, we first define a *partial* interpretation of the raw syntax for every LF-CwF by induction on the raw syntax, and then we show that the partial interpretation is defined on well typed terms by induction on the typing derivation, and finally we show that the interpretation respects judgemental equalities.

Alternatively, from a more abstract point of view, we can view the typing rules of LCCLF as a *generalised algebraic theory* (GAT) [9, 59] or the signature of a *quotient inductive-inductive type* (QIIT) [37, 42, 3]. Then we can directly take the initial model of this GAT or this QIIT as the definition of the syntax of LCCLF. The existence of initial models of GATs is shown by Cartmell [9] in a set-theoretic metatheory, and the existence of QIITs is shown by Kaposi et al. [37] in a type-theoretic metatheory assuming the existence of a specific QIIT of QIIT-signatures; see also Kovács's thesis [42]. In this way, there is no need to prove initiality manually, since we are essentially using existing logical frameworks (GATs or QIITs) to define our logical framework. In fact, it is folklore that every elementary topos with a natural number object has finitary quotient inductive-inductive types [42, §4.6], although I am not aware of a complete proof in the literature.

Either (i) by constructing the abstract syntax of LCCLF from raw syntax and proving the initiality manually or (ii) by taking the initial model as the definition of the abstract syntax of the LF, in what follows we write LFSIG for the category of LF-contexts and substitutions between contexts, i.e.,

$$\textsc{Obj}\,\textsc{LfSig} = \{\vdash \Gamma\ \mathrm{ctx}\}$$
$$\textsc{Hom}_{\textsc{LfSig}}(\Delta, \Gamma) = \{\Delta \vdash \gamma : \Gamma\},$$

and Ty_{LF} for the presheaf of LF-types over contexts,

$$Ty_{LF} : \textsc{LfSig}^{\mathrm{op}} \to \textsc{Set}, \quad Ty_{LF}(\Gamma) = \{\Gamma \vdash A\ \mathrm{type}\},$$

and Tm_{LF} for the presheaf of terms over the category of elements of Ty_{LF}, i.e. for every context Γ and $A \in Tm_{LF}(\Gamma)$,

$$Tm_{LF} : (\textstyle\int Ty_{LF})^{\mathrm{op}} \to \textsc{Set}, \quad Tm_{LF}(\Gamma; A) = \{\Gamma \vdash a : A\}.$$

The context extension of $\Gamma \in \textsc{LfSig}$ with $A \in Ty_{LF}(\Gamma)$ is written as just $\Gamma.A \in \textsc{LfSig}$, together with the projection substitution $p : \Gamma.A \to \Gamma$.

B. Categories of Judgements and Functorial Models

The classifying LCCC of a signature S is actually very concrete to describe – it is the category of judgements of S.

Definition 9. For every signature S, i.e. a context, of LCCLF, its *category of judgements* $\textsc{Jdg}\,S$ is the full subcategory of the slice category $\textsc{LfSig}/S$ spanned by projection maps $p : S.A \to S$ of context extensions for $S \vdash A : \mathbb{J}$.

The objects of $\textsc{Jdg}\,S$ can be identified with judgements $S \vdash A : \mathbb{J}$ in the context of S; the morphisms $t : A \to B$ from $S \vdash A : \mathbb{J}$ to $S \vdash B : \mathbb{J}$ can be identified with functions $S \vdash f : A \to B$. Since the universe $\mathbb{J}$ is closed under precisely the connectives of extensional MLTT (1, Σ, Π, and extensional equality types), the category $\textsc{Jdg}\,S$ is the category of types for extensional MLTT with the additional constants from S.

Consequently, $\textsc{Jdg}\,S$ is locally cartesian closed. For every $S \vdash A : \mathbb{J}$, in the slice category $\textsc{Jdg}\,S/A$,

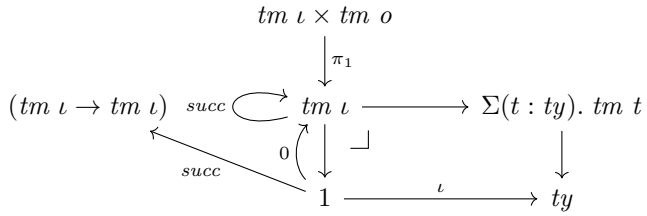
- the terminal object is $(\lambda a.\ a) : A \to A$, where we omit $S \vdash$ for clarity;
- the product of $f : B \to A$ and $g : C \to A$ is $\lambda p.\ f\ (\pi_1\ p) : P \to A$ where

$$P := \Sigma(b : B).\ \Sigma(c : C).\ (f\ b = g\ c);$$

- the exponential of $f : B \to A$ and $g : C \to A$ is $\pi_1 : E \to A$ where

$$E := \Sigma(a : A).\ B_a \to C_a,$$
$$B_a := \Sigma(b : B).\ f\ b = a,$$
$$C_a := \Sigma(c : C).\ g\ c = a.$$

Example 10. Consider the signature of PCF in Example 6. Some examples of objects and morphisms of the category of judgements for PCF are



This category is not the same as the usual category of *contexts* for PCF, since it contains higher-order judgements such as $tm\ \iota \to tm\ \iota$ or $(tm\ \iota \to tm\ \iota) \to tm\ \iota$ that do not correspond to any PCF-contexts. However, the adequacy of the LF encoding of PCF implies that the category of PCF-contexts can be fully faithfully embedded in the category of judgements. Namely, it is the full subcategory spanned by finite products of objects of the form $\mathrm{PCF} \vdash tm\ a : \mathbb{J}$ for some $\mathrm{PCF} \vdash a : ty$.

Definition 11. Let S be a signature in LCCLF and $\mathscr{C}$ a locally cartesian closed category (LCCC). A *(functorial) model of S in $\mathscr{C}$* is a functor $M : \textsc{Jdg}\, S \to \mathscr{C}$ that preserves the locally cartesian closed structure (up to isomorphisms).

Because the objects of the category of judgements $\textsc{Jdg}\, S$ are generated by a quite intricate induction, it is not straightforward to construct functorial models manually. We will solve this by using internal languages later, but for now let us sketch a partial example for some intuition.

The following lemma about presheaf categories is well known (see e.g. [50]) and will be used in the example.

Lemma 12. For every small category $\mathscr{C}$ and presheaf $A : \mathscr{C}^{\mathrm{op}} \to \textsc{Set}$, there is an equivalence $(\textsc{Pr}\,\mathscr{C})/A \cong \textsc{Pr}\,(\textstyle\int A)$ between the slice category $(\textsc{Pr}\,\mathscr{C})/A$ and the presheaf category $\textsc{Pr}\,(\textstyle\int A)$ over the category of elements of A.

Example 13. Consider the signature of STLC in Example 4. Let $\mathscr{C}$ be a small cartesian closed category. The category $\mathscr{C}$ has enough structure for interpreting the category of *contexts* of STLC [43] but not enough for interpreting the category of *judgements* of STLC, since $\mathscr{C}$ may not be locally cartesian closed. However, we can interpret all judgements in the presheaf category $\textsc{Pr}\,\mathscr{C}$, which is always locally cartesian closed.

Firstly, we define $Ty \in \textsc{Pr}\,\mathscr{C}$ to be the constant presheaf

$$Ty(\Gamma) = \textsc{Obj}\,\mathscr{C} \qquad\qquad Ty(\gamma) = id,$$

and $Tm \in \textsc{Pr}\,\mathscr{C}$ to be the presheaf defined by

$$Tm(\Gamma) = \{(A, f) \mid A \in \textsc{Obj}\,\mathscr{C}, f : \Gamma \to A\}$$
$$Tm(\gamma) = (A, f) \mapsto (A, \Delta \xrightarrow{\gamma} \Gamma \xrightarrow{f} A)$$

for all $\Gamma, \Delta \in \mathscr{C}$ and $\gamma : \Delta \to \Gamma$. As the names suggest, the projection map $p : Tm \to Ty$ is going to be the interpretation of the family of judgements $tm : ty \to \mathbb{J}$.

The interpretation of the declaration $\iota : ty$ can be any global element $1 \to Ty$, i.e. any object of $\mathscr{C}$. The interpretation of the declaration $\_\Rightarrow\_ : ty \to ty \to ty$ in $\textsc{Pr}\,\mathscr{C}$ is given by the adjunct of the natural transformation $F : Ty \times Ty \to Ty$:

$$F_\Gamma\,(A, B) = B^A \text{ for all } (A, B) \in (Ty \times Ty)\,(\Gamma).$$

To interpret the isomorphism pair abs and app, following the interpretation of MLTT in LCCCs [31, 58], we need to construct in the slice category $\textsc{Pr}\,\mathscr{C}/(Ty \times Ty)$ an isomorphism between the object $F^*p : F^*\,Tm \to Ty \times Ty$, obtained by pulling back p along F,

$$\begin{array}{ccc} F^*\,Tm & \longrightarrow & Tm \\ {\scriptstyle F^*p}\downarrow & \lrcorner & \downarrow{\scriptstyle p} \\ Ty \times Ty & \xrightarrow{F} & Ty \end{array}$$

and the object $\pi_1^*p \Rightarrow \pi_2^*p$, obtained by taking the exponential of π_1^*p and π_2^*p in $\textsc{Pr}\,\mathscr{C}/(Ty \times Ty)$, where π_i^*p is respectively obtained by the pullback of p along $\pi_i : Ty \times Ty \to Ty$. We can construct this isomorphism pointwise for each object $\Gamma \in \mathscr{C}$. An element of $(Ty \times Ty)(\Gamma)$ is a pair (A, B) of $\mathscr{C}$-objects, so the presheaf $F^*\,Tm$ at Γ is the set

$$\{(A, B, f) \mid A, B \in \mathscr{C},\ f : \Gamma \to B^A\}. \tag{3}$$

The object $\pi_1^*p \Rightarrow \pi_2^*p$ is harder to compute, but by using Lemma 12 and the end formula of exponentials in presheaf categories, we can compute that the fiber of $\pi_1^*p \Rightarrow \pi_2^*p$ over $(A, B) \in (Ty \times Ty)(\Gamma)$ is the set

$$\begin{aligned} & \textstyle\int_{\Delta \in \mathscr{C}} \prod_{\mathscr{C}(\Delta, \Gamma)} \mathscr{C}(\Delta, A) \Rightarrow \mathscr{C}(\Delta, B) \\ \cong\ & \{\text{powering in SET is the same as exponentiating}\} \\ & \textstyle\int_{\Delta \in \mathscr{C}} \mathscr{C}(\Delta, \Gamma) \Rightarrow \mathscr{C}(\Delta, A) \Rightarrow \mathscr{C}(\Delta, B) \\ \cong\ & \{\text{by uncurrying}\} \\ & \textstyle\int_{\Delta \in \mathscr{C}} \mathscr{C}(\Delta, \Gamma) \times \mathscr{C}(\Delta, A) \Rightarrow \mathscr{C}(\Delta, B) \\ \cong\ & \{\text{by the universal property of products}\} \\ & \textstyle\int_{\Delta \in \mathscr{C}} \mathscr{C}(\Delta, \Gamma \times A) \Rightarrow \mathscr{C}(\Delta, B) \\ \cong\ & \{\text{by Yoneda embedding}\} \\ & \mathscr{C}(\Gamma \times A,\ B) \end{aligned}$$

which is indeed isomorphic to the fiber of $F^*\,Tm$ (3) over (A, B) in a canonical way.

We have given the interpretation of the generating judgements of STLC in $\textsc{Pr}\,\mathscr{C}$, but this is still not a complete definition of an LCC-functor $M : \textsc{Jdg}\,\mathrm{STLC} \to \textsc{Pr}\,\mathscr{C}$ because M has to map *all* the judgements to $\textsc{Pr}\,\mathscr{C}$ (preserving the LCC structure), not just the generating ones. Because of type dependency in LCCLF, extending the interpretation from the generating judgements to all the judgements is not trivial, and we will study the general situation in the next section.

IV. Diagrammatic Semantics of Signatures

Since the category of judgement $\textsc{Jdg}\, S$ is comprised of syntactic entities that are *inductively generated*, it is natural to expect that as soon as we give the interpretation of the generating judgements of S in a category $\mathscr{C}$, the interpretation can be extended to all the derivable judgements under S. The

situation should generalise that of Lawvere theories – if L is the Lawvere theory generated by a signature S of some operations, as soon as we provide an object in $\mathscr{C}$ equipped with the operations in S, the interpretation can be extended to a product-preserving functor $L \to \mathscr{C}$.

Taking the example of STLC (Example 4) again, it is natural to expect that LCCC-functors $\textsc{Jdg}\,\mathrm{STLC} \to \mathscr{C}$ correspond to diagrams in $\mathscr{C}$ of the shape

$$\begin{array}{ccccc} \pi_1^* p \Rightarrow \pi_2^* p & \longrightarrow & Tm & & \\ \downarrow & \lrcorner & \downarrow{\scriptstyle p} & & \\ Ty \times Ty & \xrightarrow{F} & Ty & \xleftarrow{\iota} & 1 \end{array}$$

where $\pi_1^* p \Rightarrow \pi_2^* p$ is the exponential of $\pi_1^* p$ and $\pi_2^* p$ in the slice $\mathscr{C}/(Ty \times Ty)$, and the square must be a pullback. The goal of this section is to define a notion of *diagrammatic models* like this for all signatures S in LCCLF.

To define diagrammatic models for all signatures S, we need to perform an induction on the syntax of LCCLF, since a signature S is nothing other than an LF context. An induction on the LF is the same as constructing a CwF with the type connectives of the LF. However, we cannot directly use the LCCC $\mathscr{C}$ as the (underlying category of) the CwF, since $\mathscr{C}$ does not have the structure for interpreting the universe $\mathbb{J}$. The solution is passing to the presheaf category $\textsc{Pr}\,\mathscr{C}$, in which we can construct a *universe* containing (the Yoneda embedding of) objects of $\mathscr{C}$.

A. Universes in Categories

The concept of universes in *toposes* dates back to Bénabou [7] and Maurer [47], and was developed later by Streicher [64]. A more general account of universes in categories is developed by Voevodsky [69, 70] and Kapulkin and Lumsdaine [40] in the study of homotopy type theory; see also Gratzer's thesis [23, §3.2, 3.3] for an exposition.

In this subsection, we will have a brief digression on universes without going into much technical detail, ending with the theorem that a universe of $\mathscr{C}$-objects can be constructed in the presheaf category $\textsc{Pr}\,\mathscr{C}$ (Theorem 18).

Definition 14. A *universe* in a category $\mathscr{C}$ is simply a morphism $p : \tilde{U} \to U$ equipped with chosen pullbacks $A^* p : \Gamma.A \to \Gamma$ along every morphism $A : \Gamma \to U$. A morphism $\Delta \to \Gamma$ in $\mathscr{C}$ is said to be *classified* by a universe $p : \tilde{U} \to U$ if it is a pullback of p along some (not necessarily unique) morphism $\Gamma \to U$.

A universe may additionally be equipped with logical structures such as Π-types and Σ-types; see [40, §1.4] or [23, §3.2] for details. For example, *binary product* on a universe $p : \tilde{U} \to U$ is a pair of morphisms $prod : U \times U \to U$ and $pair : \tilde{U} \times \tilde{U} \to \tilde{U}$ forming a pullback:

$$\begin{array}{ccc} \tilde{U} \times \tilde{U} & \xrightarrow{pair} & \tilde{U} \\ {\scriptstyle p \times p}\downarrow & \lrcorner & \downarrow{\scriptstyle p} \\ U \times U & \xrightarrow{prod} & U \end{array} \tag{4}$$

Example 15. Universes abound in logic and type theory.

1) In the category of sets, every Grothendieck universe U determines a universe $\pi_1 : \tilde{U} \to U$ where $\tilde{U}$ is the set of pointed U-small sets:
$$\tilde{U} = \{(A, a) \mid A \in U, a \in A\},$$
and $\pi_1(A, a) = A$ is the projection function.
2) For a small category $\mathscr{C}$, every Grothendieck universe U of sets can be lifted to a universe $\pi_1 : \tilde{V} \to V$ in the presheaf category $\textsc{Pr}\,\mathscr{C}$ by the *Hofmann-Streicher lifting* [33]: V maps every $\Gamma \in \mathscr{C}$ to the set of U-valued presheaves over $\mathscr{C}/\Gamma$, and $\tilde{V}$ maps every $\Gamma \in \mathscr{C}$ to the set
$$\{(A, a) \mid A \in V(\Gamma), a : 1 \to A \in \textsc{Pr}\,(\mathscr{C}/\Gamma)\}.$$
The actions of V and $\tilde{V}$ on morphisms $\gamma : \Delta \to \Gamma$ is given by precomposing with $(\gamma \cdot -) : (\mathscr{C}/\Delta)^{\mathrm{op}} \to (\mathscr{C}/\Gamma)^{\mathrm{op}}$. However, the universe V constructed in this way is not what we want for interpreting the universe of judgements $\mathbb{J}$, because V classifies all (U-small) presheaves rather than just (Yoneda-embedding of) $\mathscr{C}$-objects.
3) Liftings of Grothendieck universes of sets to sheaf toposes in general [64, 25], categories of assemblies, and realizability toposes [64] also exist.
4) A syntactic example of universes is the map $\pi_1 : (A : \mathbb{J}, a : A) \to (A : \mathbb{J})$ in the category LFSIG of LF-signatures (III-A). The pullback of π_1 along an arbitrary morphism $B : S \to (A : \mathbb{J})$ can be chosen to be simply
$$\begin{array}{ccc} S.B & \longrightarrow & (A : \mathbb{J}, a : A) \\ \downarrow & \lrcorner & \downarrow \\ S & \xrightarrow[B]{} & (A : \mathbb{J}) \end{array}$$

All the examples above can be equipped with structures of Σ-types, Π-types, and extensional equality types.

Definition 16. For a universe $p : \tilde{U} \to U$ in a category $\mathscr{C}$, its *externalisation* $[p : \tilde{U} \to U]$, or simply $[U]$, is the fibration over $\mathscr{C}$ whose fiber category $[U]_\Gamma$ over every object $\Gamma \in \mathscr{C}$ has as objects $\mathscr{C}$-morphisms $A : \Gamma \to U$. Morphisms $A \to B$ in the fiber $[U]_\Gamma$ are $\mathscr{C}$-morphisms $h : \Gamma.A \to \Gamma.B$ making the following diagram commute:

$$\begin{array}{ccc} \Gamma.A & \xrightarrow{h} & \Gamma.B \\ & {\scriptstyle A^* p}\searrow \quad \swarrow{\scriptstyle B^* p} & \\ & \Gamma & \end{array}$$

where $\Gamma.A$ and $\Gamma.B$ arise from pulling back p along A and B respectively. The reindexing functor $\gamma^* : [U]_\Gamma \to [U]_\Delta$ for a morphism $\gamma : \Delta \to \Gamma$ is precomposition:

$$y^*(\Gamma \xrightarrow{A} U) = (\Delta \xrightarrow{\gamma} \Gamma \xrightarrow{A} U).$$

For an LF-signature S, the category $\textsc{Jdg}\,S$ of judgements of S from Definition 9 is precisely the fiber over S of the externalisation of the universe $\pi_1 : (A : \mathbb{J}, a : A) \to (A : \mathbb{J})$ in the category LFSIG of LF-contexts from Section III-A.

When a universe is additionally equipped with logical structures, these structures can be carried over to the externalisation. For example, if the universe is equipped with binary product $prod : U \times U \to U$ as in (4), the cartesian product of two objects $A, B : \Gamma \to U$ in the fiber $[U]_\Gamma$ is

$$\Gamma \xrightarrow{\langle A,B\rangle} U \times U \xrightarrow{prod} U$$

Moreover, this choice of cartesian products are strictly preserved by the reindexing functor $\gamma^* : [U]_\Gamma \to [U]_\Delta$ since reindexing is defined to be precomposition.

If the universe U is equipped with Σ-types, Π-types, and extensional equality types, each fiber category of $[U]$ has an LCCC structure given in a way similar to Section III-B. The reindexing functor strictly preserves the LCCC structure.

Example 13 contains another example of universes: starting from a cartesian closed category $\mathscr{C}$, we constructed in the presheaf category $\mathrm{PR}\,\mathscr{C}$ a morphism $p : Tm \to Ty$ that as a universe classifies exactly (the Yoneda embedding of) projection morphisms $\Gamma \times A \to \Gamma$ in $\mathscr{C}$. However, p inherently only supports *simple types*: For an arbitrary type family over Γ, i.e. an arbitrary morphism $f : B \to \Gamma$ in $\mathscr{C}$, there may be no $\lceil B \rceil : \mathrm{Y}\Gamma \to Ty$ making $\mathrm{Y}f$ a pullback of p along $\lceil B \rceil$.

The good news is that the dependently typed version of the universe $p : Tm \to Ty$ in Example 13 exists. Perhaps unexpectedly, what we need is exactly Hofmann's [30] technique for turning locally cartesian closed categories into (strict) models of extensional MLTT. Hofmann op. cit. showed how to construct a CwF (with the type connectives of extensional MLTT) over an arbitrary LCCC $\mathscr{C}$, but as pointed out by Fiore [17] and Awodey [6], a CwF structure $\langle Ty, Tm\rangle$ over $\mathscr{C}$ is precisely a universe $p : Tm \to Ty$ in $\mathrm{PR}\,\mathscr{C}$ that is a *representable morphism*, a perspective known as *natural models* of dependent type theories [6, 48].

Definition 17. A morphism $p : Tm \to Ty$ in a presheaf category $\mathrm{PR}\,\mathscr{C}$ is called *representable* if for every $\Gamma \in \mathscr{C}$ and $A : \mathrm{Y}\Gamma \to Ty$, there is an object $\Gamma.A \in \mathscr{C}$ and a morphism $f : \Gamma.A \to \Gamma$ in $\mathscr{C}$ making a pullback square:

$$\begin{array}{ccc} \mathrm{Y}\Gamma.A & \longrightarrow & Tm \\ {\scriptstyle \mathrm{Y}f}\downarrow & \lrcorner & \downarrow{\scriptstyle p} \\ \mathrm{Y}\Gamma & \xrightarrow[A]{} & Ty \end{array}$$

Putting aside the type connectives, Hofmann's construction can be seen as an instance of Bénabou's [8] construction from a fibration to an equivalent split fibration (see also Streicher [66, Theorem 3.1] and Jacobs [35, §5.2]), applied to the codomain fibration $\mathscr{C}^{\to} \to \mathscr{C}$ of an LCCC $\mathscr{C}$. This splitting technique is the right adjoint to the forgetful functor from split fibrations to fibrations. There is also a left adjoint due to Giraud [22, I 2.4.3]; see also Streicher [66, page 13]. Lumsdaine and Warren [45] and Awodey [6] showed that the left-adjoint splitting can also be used to construct (strict) models of type theories from weak models, known as the *local universe* construction.

Using either Hofmann's construction or the local universe construction, we have the following result.

Theorem 18 (Hofmann [30], Awodey [6], Lumsdaine and Warren [45])**.** *For every small locally cartesian closed category $\mathscr{C}$, there is a universe $p : \tilde{U}_\mathscr{C} \to U_\mathscr{C}$ in $\mathrm{PR}\,\mathscr{C}$ such that (1) p is representable; (2) every morphism $f : A \to B$ in $\mathscr{C}$ is classified by p; (3) p supports 1, Σ, Π, and extensional equality types.*

In the situation of Theorem 18, by the representability of p, the fiber $[U_\mathscr{C}]_1$ of the externalisation over the terminal object 1 contains only representable objects. And since p classifies (the Yoneda embedding of) all morphisms of $\mathscr{C}$, it in particularly classifies all morphisms $\mathrm{Y}A \to \mathrm{Y}1_\mathscr{C} = 1$. Therefore the fiber category $[U_\mathscr{C}]_1$ is equivalent to the LCCC $\mathscr{C}$.

B. Diagrammatic Models

By Theorem 18, $\mathrm{PR}\,\mathscr{C}$ has the structure for interpreting the universe of judgements $\mathbb{J}$ and the type connectives on $\mathbb{J}$. The presheaf category $\mathrm{PR}\,\mathscr{C}$ can interpret the unit type and Σ-types of the LF as usual [31]. Thus we can turn $\mathrm{PR}\,\mathscr{C}$ into an LF-CwF. Since the abstract syntax of the LF is initial among all LF-CwFs (Section III-A), there is a unique LF-CwF homomorphism, which consists of (1) a functor interpreting LF-signatures (i.e. LF-contexts) as $\mathscr{C}$-presheaves

$$[\![-]\!] : \textsc{LFSig} \longrightarrow \mathrm{PR}\,\mathscr{C} \tag{5}$$

and (2) mappings from LF-types/terms to $\mathrm{PR}\,\mathscr{C}$-types/terms that strictly preserve all operations, also denoted by $[\![-]\!]$.

Definition 19. A *diagrammatic model* of an LF-signature S in an LCCC $\mathscr{C}$ is a global element $m : 1 \to [\![S]\!]$ of the interpretation of S in $\mathrm{PR}\,\mathscr{C}$.

Diagrammatic models are more ergonomic to work with than functorial models because $\mathrm{PR}\,\mathscr{C}$ as a presheaf topos has a very rich structure that we can manipulate using a type theoretic language. In this way, a diagrammatic model of S in $\mathscr{C}$ is a closed element of the record type $[\![S]\!]$ in the internal language of $\mathrm{PR}\,\mathscr{C}$, containing all fields of S and with $\mathbb{J}$ replaced by its interpretation $U_\mathscr{C}$.

Example 20. The signature of barebone type theory from Example 3 is interpreted as the presheaf $[\![\mathrm{BTT}]\!] \in \mathrm{PR}\,\mathscr{C}$ denoted by the record type

$$\begin{array}{l} \textsc{record}\ [\![\mathrm{BTT}]\!]\ \textsc{where} \\ \quad ty\ : U_\mathscr{C} \\ \quad tm : ty \to U_\mathscr{C} \end{array}$$

A closed element of this record consists of (1) a morphism $A : 1 \to U_\mathscr{C}$, which gives rise to an object $\tilde{A}$ in $\mathscr{C}$ by the representability of $\tilde{U}_\mathscr{C} \to U_\mathscr{C}$, and (2) a morphism $B : \mathrm{Y}\tilde{A} \to U_\mathscr{C}$ which gives rise to a morphism $\tilde{B} \to \tilde{A}$ in $\mathscr{C}$:

$$\begin{array}{ccc} \mathrm{Y}\tilde{B} & & \\ \downarrow & \searrow & \\ \mathrm{Y}\tilde{A} & \longrightarrow & \tilde{U}_\mathscr{C} \\ \downarrow & \overset{B}{\searrow} & \downarrow \\ 1 \cong \mathrm{Y}1_\mathscr{C} & \xrightarrow[A]{} & U_\mathscr{C} \end{array}$$

Thus a diagrammatic model of BTT in a locally cartesian closed category $\mathscr{C}$ gives rise to a morphism $\tilde{B} \to \tilde{A}$ in $\mathscr{C}$.

In Definition 9, the category $\textsc{Jdg}\, S$ of judgements for a signature S is defined to be the full subcategory of the slice $\textsc{LFSig}/S$ spanned by projections $S.A \to S$ for judgements $S \vdash A : \mathbb{J}$. Every such judgement is sent by the interpretation functor $[\![-]\!] : \textsc{LFSig} \to \textsc{Pr}\,\mathscr{C}$ to a morphism $[\![A]\!] : [\![S]\!] \to U_{\mathscr{C}}$, which is exactly an object in the fiber of the externalisation $[U_{\mathscr{C}}]$ over $[\![S]\!]$. Since $[\![-]\!]$ is a homomorphism of LF-CwFs, it (strictly) preserves context extensions, so the context projection $S.A \to S$ is sent to the morphism $[\![S]\!].[\![A]\!] \to [\![S]\!]$ in $\textsc{Pr}\,\mathscr{C}$:

$$\begin{array}{ccc} S.A & \longrightarrow & (B : \mathbb{J}, b : B) \\ \downarrow & \lrcorner & \downarrow \\ S & \xrightarrow[A]{} & (B : \mathbb{J}) \end{array} \quad \overset{[\![-]\!]}{\longmapsto} \quad \begin{array}{ccc} [\![S]\!].[\![A]\!] & \longrightarrow & \tilde{U}_{\mathscr{C}} \\ \downarrow & \lrcorner & \downarrow \\ [\![S]\!] & \xrightarrow[[\![A]\!]]{} & U_{\mathscr{C}} \end{array}$$

Therefore $[\![-]\!]$ sends the morphisms of $\textsc{Jdg}\, S$ to morphisms of $[U_{\mathscr{C}}]_{[\![S]\!]}$ as well. In conclusion, for every LF-signature S and LCCC $\mathscr{C}$, we have a functor

$$\mathrm{G}_{\mathscr{C}} : \textsc{Jdg}\, S \longrightarrow [U_{\mathscr{C}}]_{[\![S]\!]},$$

which preserves LCCC structures since the type connectives giving the LCCC structures of $\textsc{Jdg}\, S$ Section III-B are preserved by interpretation.

Definition 21. Every diagrammatic model $m : 1 \to [\![S]\!]$ in an LCCC $\mathscr{C}$ determines a functorial model $F(m) := m^* \circ \mathrm{G}_{\mathscr{C}} : \textsc{Jdg}\, S \to \mathscr{C}$ by composing $\mathrm{G}_{\mathscr{C}}$ with the reindexing functor $m^* : [U_{\mathscr{C}}]_{[\![S]\!]} \to [U_{\mathscr{C}}]_1 \cong \mathscr{C}$:

$$\begin{array}{ccc} \textsc{Jdg}\, S & \xrightarrow{\mathrm{G}_{\mathscr{C}}} & [U_{\mathscr{C}}]_{[\![S]\!]} \\ & \searrow^{F(m)} & \downarrow m^* \\ & & \mathscr{C} \cong [U_{\mathscr{C}}]_1 \end{array}$$

Using $F(m)$ will be the main way of constructing functorial models. Of course, if the LCCC $\mathscr{C}$ that we start with already has a universe U that can model $\mathbb{J}$, for example, when $\mathscr{C}$ is a presheaf topos, then we may also directly construct diagrammatic models classified by U in the language of $\mathscr{C}$ itself, skipping the step of embedding $\mathscr{C}$ into $\textsc{Pr}\,\mathscr{C}$.

C. From Functorial Models to Diagrammatic Models

Conversely, a functorial model $M : \textsc{Jdg}\, S \to \mathscr{C}$ induces a diagrammatic model $D(M)$ as well. This may seem trivial because functorial models assign semantics to all judgements while diagrammatic models only assign semantics to generating judgements. However, this task is complicated by the fact that M as an LCC-functor only needs to preserve the LCC-structure up to isomorphism, so M will not be exactly $F(D(M))$ in general, and we have to simultaneously prove $F(D(M)) \cong M$ for the induction to go through.

Lemma 22. Let S be a signature and $\mathscr{C}$ an LCCC. Every functorial model $M : \textsc{Jdg}\, S \to \mathscr{C}$ determines a diagrammatic model $D(M) : 1 \to [\![S]\!]$ and a natural isomorphism $\phi_M : F(D(M)) \cong M : \textsc{Jdg}\, S \to \mathscr{C}$.

Before proving 22, we first observe that every LF-signature S is isomorphic (in the category $\textsc{LFSig}$ of LF-signatures) to a *standard signature*, which is inductively defined to be either

1) the empty signature,
2) $(T, a : A)$ for some standard T and $T \vdash A : \mathbb{J}$, or
3) $(T, B : A \to \mathbb{J})$ for some standard T and $T \vdash A : \mathbb{J}$.

Moreover, every judgement $S \vdash A : \mathbb{J}$ under a standard signature S is equal (in the equational theory of the LF) to a *standard judgement*, which is inductively defined to be either

1) $S \vdash B\ a : \mathbb{J}$ for some declaration $B : A \to \mathbb{J}$ in S and $S \vdash a : A$, or
2) the type formers 1, Σ, Π, $a = b$ of $\mathbb{J}$ applied to standard judgements.

These claims can be shown by induction on the syntax of the LF. Note that they are not the same as *normalisation* of the LF – we do not claim terms have any standard or normal form (which is in indeed not true because extensional equality types do not enjoy normalisation [29, §3.2.2]).

When proving statements $P(S)$ about LF-signatures S that are invariant under isomorphisms or defining constructions $C(S)$ for LF-signatures that can be transported along isomorphisms, we conveniently only need to prove or construct for standard signatures and consider only standard judgements.

Proof of 22. The required constructions $D(M)$ and ϕ_M in the statement of Lemma 22 can be transported along isomorphisms $S \cong S'$of LF-signatures, so we can assume S is standard. Then we construct $D(M)$, the components of ϕ_M, and show the naturality of ϕ_M by a simultaneous induction on the structure of standard signatures S, standard judgements A in S, and terms $S \vdash a : A$.

Part 1. We first construct D for every signature S.

Case 1.1. If S is the empty signature, $[\![S]\!]$ is the terminal presheaf, so there is a unique choice for $D(M) : 1 \to [\![S]\!]$.

Case 1.2. If $S = (T, a : A)$ for some $T \vdash A : \mathbb{J}$, we have an inclusion functor $i : \textsc{Jdg}\, T \to \textsc{Jdg}\, S$ sending every judgement $(T \vdash B : \mathbb{J}) \in \textsc{Jdg}\, T$ to its weakening $(S \vdash B : \mathbb{J}) \in \textsc{Jdg}\, S$. By composing this functor with $M : \textsc{Jdg}\, S \to \mathscr{C}$ we have a functorial model $M \circ i : \textsc{Jdg}\, T \to \mathscr{C}$ of T, which further gives rise to a diagrammatic model $D(M \circ i)$ of T by induction. Our goal is to construct a morphism $D(M) : 1 \to [\![S]\!]$ making the left triangle below commute:

$$\begin{array}{ccccc} & & [\![S]\!] = [\![T]\!].[\![A]\!] & \longrightarrow & \tilde{U}_{\mathscr{C}} \\ & \nearrow & \downarrow p_S \lrcorner & & \downarrow p \\ 1 & \xrightarrow[D(M\circ i)]{} & [\![T]\!] & \xrightarrow[[\![A]\!]]{} & U_{\mathscr{C}} \end{array} \tag{6}$$

In the category $\textsc{Jdg}\, S$, we have a morphism

$$(T, a : A, 1 \vdash a : A) : (S \vdash 1 : \mathbb{J}) \to (S \vdash A : \mathbb{J}) \tag{7}$$

which is mapped by $M : \textsc{Jdg}\, S \to \mathscr{C}$ to a morphism

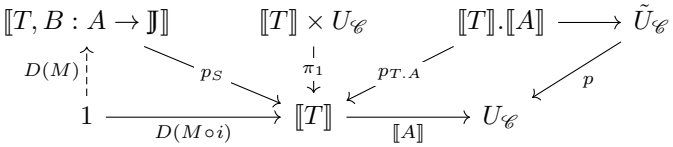
$$M(a) : 1_{\mathscr{C}} \cong M(S \vdash 1 : \mathbb{J}) \to M(S \vdash A : \mathbb{J})$$

in $\mathscr{C} \cong [U_{\mathscr{C}}]_1$. By the inductive hypothesis, we have a natural isomorphism $\phi_{M\circ i} : F(D(M \circ i)) \cong M \circ i$. By Definition 21 of F, $F(D(M \circ i))$ applied to $T \vdash A : \mathbb{J}$ is the pullback of $p : \tilde{U}_{\mathscr{C}} \to U_{\mathscr{C}}$ along $[\![A]\!] \cdot D(M \circ i)$, and $M \circ i$ applied to $T \vdash A : \mathbb{J}$ is simply $M(S \vdash A : \mathbb{J})$. Hence we have pullbacks:

$$\begin{array}{ccccc} M(S \vdash A : \mathbb{J}) & \xrightarrow{g} & [\![S]\!] = [\![T]\!].[\![A]\!] & \longrightarrow & \tilde{U}_{\mathscr{C}} \\ {\scriptstyle M(a)}\uparrow \downarrow & \lrcorner & \downarrow{\scriptstyle p_S} & \lrcorner & \downarrow{\scriptstyle p} \\ M(S \vdash 1 : \mathbb{J}) \cong 1 & \xrightarrow[D(M\circ i)]{} & [\![T]\!] & \xrightarrow[[\![A]\!]]{} & U_{\mathscr{C}} \end{array} \tag{8}$$

We now define the desired diagrammatic model $D(M)$ to be the composite $g \cdot M(a) : 1 \to [\![S]\!]$.

Case 1.3. If $S = (T, B : A \to \mathbb{J})$ for some $T \vdash A : \mathbb{J}$, the projection morphism $p_S : [\![T, B : A \to \mathbb{J}]\!] \to [\![T]\!]$ is the exponential object in the slice category $\mathrm{PR}\,\mathscr{C}/[\![T]\!]$ from $p_{T.A} : [\![T]\!].[\![A]\!] \to [\![T]\!]$ to $\pi_1 : [\![T]\!] \times U_{\mathscr{C}} \to [\![T]\!]$. We would like to define $D(M) : 1 \to [\![T, B : A \to \mathbb{J}]\!]$ making the left triangle below commute,



where $i : \textsc{Jdg}\,T \to \textsc{Jdg}\,S$ is the weakening functor, and $D(M \circ i)$ is the diagrammatic model of the smaller context T obtained from the inductive hypothesis. By the universal property of $[\![T, B : A \to \mathbb{J}]\!]$ as the exponential, we need to construct a morphism $D(M \circ i) \times p_{T.A} \to \pi_1$ in the slice category over $[\![T]\!]$. The product $D(M \circ i) \times p_{T.A}$ is the pullback of $p_{T.A}$ along $D(M \circ i)$, which is isomorphic to the object $M(i(T \vdash A : \mathbb{J}))$ via $\phi_{M\circ i} : F(D(M \circ i)) \cong M \circ i$:

$$\begin{array}{ccccc} M(i(T \vdash A : \mathbb{J})) & \longrightarrow & [\![S]\!] = [\![T]\!].[\![A]\!] & \longrightarrow & \tilde{U}_{\mathscr{C}} \\ \downarrow & \lrcorner & {\scriptstyle p_{T.A}}\downarrow & \lrcorner & \downarrow{\scriptstyle p} \\ 1 & \xrightarrow[D(M\circ i)]{} & [\![T]\!] & \xrightarrow[[\![A]\!]]{} & U_{\mathscr{C}} \end{array}$$

Our goal is to construct a morphism $M(i(T \vdash A : \mathbb{J})) \to U_{\mathscr{C}}$.

Back in the category $\textsc{Jdg}\,S$, we have the judgement $S \vdash \Sigma\, A\, B : \mathbb{J}$, and the projection $\pi_1^{\Sigma AB} : \Sigma\, A\, B \to A$; henceforth we omit the context $S \vdash$ on objects. The projection map is sent by $M : \textsc{Jdg}\,S \to \mathscr{C} \cong [U_{\mathscr{C}}]_1$ to a morphism in $\mathscr{C}$. Since the universe $p : \tilde{U}_{\mathscr{C}} \to U_{\mathscr{C}}$ (weakly) classifies all $\mathscr{C}$-morphisms, the morphism $M\pi_1^{\Sigma AB}$ gives us some $\lceil B \rceil : M(A) \to U_{\mathscr{C}}$ and a pullback square:

$$\begin{array}{ccc} M(\Sigma\, A\, B) & \longrightarrow & \tilde{U}_{\mathscr{C}} \\ {\scriptstyle M\pi_1^{\Sigma AB}}\downarrow & \lrcorner & \downarrow{\scriptstyle p} \\ M(A) & \xrightarrow[\lceil B \rceil]{} & U_{\mathscr{C}} \end{array} \tag{9}$$

The morphism $\lceil B \rceil$ fulfils our goal $M(i(T \vdash A : \mathbb{J})) = M(A) \to U_{\mathscr{C}}$.

Part 2. Now we define the component of $\phi_M : FDM \cong M$ at every (standard) judgement in a (standard) signature S.

Case 2.1. For the judgement $S \vdash B\ a : \mathbb{J}$ where $S = (T, B : A \to \mathbb{J}, R)$ for some T and R, $T \vdash A : \mathbb{J}$, and $S \vdash a : A$, we have a pullback diagram in $\textsc{Jdg}\,S$:

$$\begin{array}{ccc} B\ a & \longrightarrow & \Sigma\, A\, B \\ \downarrow & \lrcorner & \downarrow{\scriptstyle \pi_1^{\Sigma AB}} \\ 1 & \xrightarrow[a]{} & A \end{array}$$

Since M is an LCCC functor, it preserves pullbacks and the terminal object, so we have a pullback square in $\mathscr{C}$:

$$\begin{array}{ccc} M(B\ a) & \longrightarrow & M(\Sigma\, A\, B) \\ \downarrow & \lrcorner & \downarrow{\scriptstyle M\pi_1^{\Sigma AB}} \\ 1 & \xrightarrow[Ma]{} & M(A) \end{array}$$

On the other hand, in Case 1.3 above, we have defined the B-component of the diagrammatic model $D(M)$ to be the code $\lceil B \rceil$ of the morphism $M\pi_1^{\Sigma AB}$ as in the diagram (9). Hence, unfolding the definition of F, $FDM(B\ a)$ will be the pullback of $p : \tilde{U}_{\mathscr{C}} \to U_{\mathscr{C}}$ along the following morphism:

$$1 \xrightarrow{F(D(M))(a)} (FDM)(A) \xrightarrow{(\phi_M)_A} MA \xrightarrow{\lceil B \rceil} U_{\mathscr{C}}$$

Using the naturality of $(\phi_M)_A$, this morphism is the same as $1 \xrightarrow{M(a)} MA \xrightarrow{\lceil B \rceil} U_{\mathscr{C}}$. Therefore both $M(A)$ and $FDM(A)$ are the pullback of p along $\lceil B \rceil \cdot M(a)$, so they are isomorphic in a canonical way.

Case 2.2. For a judgement A that is some type former of $\mathbb{J}$ applied to (smaller) standard judgements, $M(A)$ and $D(F(M))(A)$ are both LCC-functors so they preserve these type formers. By the universal properties of these type formers, $M(A)$ and $D(F(M))(A)$ are isomorphic in a canonical way.

Part 3. We also need to show that the family of morphisms

$$\phi_M(A) : FDM(A) \cong M(A)$$

is natural in A. Because $\textsc{Jdg}\,S$ has exponentials, which are preserved by M and FDM, it is sufficient to show that for every $S \vdash A : \mathbb{J}$ and $S \vdash a : A$, there is a commutative triangle:

$$\begin{array}{ccc} 1 & \xrightarrow{FDM(a)} & FDM(A) \\ & {\scriptstyle M(a)}\searrow & \downarrow{\scriptstyle (\phi_M)_A} \\ & & M(A) \end{array}$$

If a is a variable, this follows from the definition of $D(T, a : A)$ in Case 1.2 above. If a is one of the other term formers, it follows from the fact that M and FDM as LCC-functors both preserve these term formers. □

V. Equivalence of Two Notions of Models

We have now mappings F (Definition 21) from diagrammatic models to functorial models, and vice versa D (Lemma 22). Moreover, there is an isomorphism $F(D(M)) \cong M$ for every functorial model $M : \textsc{Jdg}\,S \to \mathscr{C}$. Naturally, we would expect $D(F(m)) \cong m$ for every diagrammatic model m as well, and then diagrammatic and functorial models will be in bijection up

to isomorphisms. But we do not have a notion of isomorphisms of diagrammatic models yet, so we will define it in this section, which turns out to be more interesting a task than it sounds.

Example 23. Let us still begin with some small examples for intuition. Consider the signature $(A : \mathbb{J})$ of one judgement and nothing else. Diagrammatic models of it in an LCCC $\mathscr{C}$ are morphisms $m : 1 \to U_{\mathscr{C}}$ in $\mathrm{PR}\,\mathscr{C}$, which give rise to objects A in $\mathscr{C}$ by Theorem 18 in a surjective way (but different m may give rise to the same object in $\mathscr{C}$). An isomorphism between two models m_1 and m_2 in this case ought to be an isomorphism $i : A_1 \to A_2$ in $\mathscr{C}$ between the $\mathscr{C}$-objects A_1 and A_2 induced by m_1 and m_2 respectively.

Now suppose the signature is extended to $(A : \mathbb{J}, f : (A \to A) \to A)$. Then every diagrammatic model gives rise to an object A together with a morphism $f : (A \Rightarrow A) \to A$ in $\mathscr{C}$, where $A \Rightarrow A$ denotes the exponential. Now an isomorphism between two diagrammatic models should be a $\mathscr{C}$-isomorphism $i : A_1 \to A_2$ that commutes with f:

$$\begin{array}{ccc} A_1 \Rightarrow A_1 & \xrightarrow{f_1} & A_1 \\ {\scriptstyle i^{-1}\Rightarrow i}\downarrow & & \downarrow{\scriptstyle i} \\ A_2 \Rightarrow A_2 & \xrightarrow[f_2]{} & A_2 \end{array}$$

The fact that we have dependent functions in LF signatures is why we only consider isomorphisms rather than homomorphisms of diagrammatic models.

Suppose that the signature is further extended with a family of judgements $B : A \to \mathbb{J}$ indexed by A. A diagrammatic model now further induces an $\mathscr{C}$-object B with a morphism $g : B \to A$. An isomorphism of diagrammatic models should now further include a $\mathscr{C}$-isomorphism $j : B_1 \to B_2$ that commutes with i:

$$\begin{array}{ccc} B_1 & \xrightarrow{j} & B_2 \\ {\scriptstyle g_1}\downarrow & & \downarrow{\scriptstyle g_2} \\ A_1 & \xrightarrow[i]{} & A_2 \end{array}$$

Finally, if a signature is extended with a declaration of an equation, the notion of isomorphisms between diagrammatic models should remain unchanged, since in LCCCs there is not any higher-dimensional coherence between equalities.

Diagrammatic models in an LCCC $\mathscr{C}$ are defined by interpreting an LF-signature S as a presheaf $[\![S]\!] \in \mathrm{PR}\,\mathscr{C}$ (Definition 19). In the internal language of the presheaf topos $\mathrm{PR}\,\mathscr{C}$, a presheaf is a 'set'. Now that we are interested in isomorphisms of models, sets are no longer sufficient, and instead, we would like to use groupoids as our interpretation. More precisely, we plan to interpret every LF-signature S as an *internal groupoid* $\langle [\![S]\!], [\![S]\!]^{\cong} \rangle$ in $\mathrm{PR}\,\mathscr{C}$ whose object part is exactly the presheaf $[\![S]\!]$ in the earlier interpretation (Section IV-B). Then, global elements $i : 1 \to [\![S]\!]^{\cong}$ of the morphism part of the groupoid will be defined as isomorphisms between diagrammatic models $s \cdot i$ and $t \cdot i : 1 \to [\![S]\!]$.

One way to carry out the plan above is to follow Hofmann and Streicher's [32] celebrated *groupoid model* of Martin-Löf type theory internally in the language of the presheaf topos $\mathrm{PR}\,\mathscr{C}$, rather than in the ambient set theory.

In outline, in the language of $\mathrm{PR}\,\mathscr{C}$, there is a groupoid $U_{\mathscr{C}}^{\cong}$ whose objects have the type $U_{\mathscr{C}}$ and the morphisms between $A, B : U_{\mathscr{C}}$ in this groupoid are the isomorphisms $A \cong B$. This groupoid provides the interpretation for the universe $\mathbb{J}$. For every $A : U_{\mathscr{C}}$, the type A can be regarded as a *discrete* groupoid, so $U_{\mathscr{C}}^{\cong}$ still models Π, Σ, and *extensional* equalities. Π and Σ types outside $\mathbb{J}$ in the LF are interpreted in the same way as Hofmann and Streicher [32]. The result of this construction would then be an *internal* LF-CwF in $\mathrm{PR}\,\mathscr{C}$, whose externalisation over the terminal object $1 \in \mathrm{PR}\,\mathscr{C}$ will then the (ordinary) LF-CwF that interprets an LF-signature S as a groupoid of diagrammatic models.

While the approach outlined above is feasible, there is a more direct approach that we will follow instead. First, we notice that in virtually all logical frameworks the notions of homomorphisms/isomorphisms of models of a theory S can be expressed as another theory that contains (1) two copies of the declarations of S and (2) new declarations for the homomorphisms/isomorphisms between the basic sorts of the two copies of S, together with (3) equations asserting the homomorphic properties. This is also the case for LCCLF. Take the signature $S = (A : \mathbb{J}, f : (A \to A) \to A)$ for example; in the LF we have the following signature $S^{\cong}$ of *isomorphisms of S-models*:

$$\begin{array}{cc} A_1 : \mathbb{J} & f_1 : (A_1 \to A_1) \to A_1 \\ A_2 : \mathbb{J} & f_2 : (A_2 \to A_2) \to A_2 \\ i : A_1 \cong A_2 & \_ : (i.fwd \cdot f_1) = (\lambda g.\ f_2\ (i.fwd \cdot g \cdot i.bwd)) \end{array}$$

where $A_1 \cong A_2$ is the judgement of isomorphisms defined in Example 5 and $f \cdot g$ means function composition $\lambda x.\ f\ (g\ x)$. In the category LFSIG of LF-signatures, there are two morphisms $s, t : S^{\cong} \to S$ that project out (A_1, f_1) and (A_2, f_2) respectively. Similarly, we have morphisms $inv, id, comp$ in LFSIG for the inverse, identity, composition of S-isomorphisms, assembling to an internal groupoid:

$$\begin{array}{ccccc} & & \overset{inv}{\circlearrowright} & & \\ S^{\cong} \times_S S^{\cong} & \xrightarrow[comp]{} & S^{\cong} & \xleftarrow[id]{} & S \\ & & \downarrow{\scriptstyle \langle s,t \rangle} & & \\ & & S \times S & & \end{array}$$

where $S^{\cong} \times_S S^{\cong}$ is the signature of three copies of S and two isomorphisms in between. The interpretation of this internal groupoid by $[\![-]\!] : \mathrm{LFSIG} \to \mathrm{PR}\,\mathscr{C}$ for every LCCC $\mathscr{C}$ is precisely the internal groupoid that we wanted.

Motivated by the discussion above, we would like to define an internal groupoid $\langle S, S^{\cong} \rangle$ for every LF-signature S, which can be done inductively together with some related conditions on judgements and terms in a signature.

Lemma 24. LCCLF satisfies the following statements:

1) Every LF-signature S determines a judgement of *S-isomorphisms*

$$M_1 : S,\ M_2 : S \vdash S^{\cong} : \mathbb{J}$$

in which we reuse the name S for the iterative Σ-type of the fields of S. Moreover, there are terms inv, id, $comp$ of the following types and they satisfy (judgementally in the LF) the axioms of groupoids:

$$\begin{aligned} M_1, M_2 : S \vdash\ & inv_S\ : S^{\cong}[M_1, M_2] \to S^{\cong}[M_2, M_1] \\ M_1 : S \vdash\ & id_S\ : S^{\cong}[M_1, M_1] \\ M_1, M_2, M_3 : S \vdash\ & comp_S : S^{\cong}[M_1, M_2] \to S^{\cong}[M_2, M_3] \\ & \to S^{\cong}[M_1, M_3] \end{aligned}$$

where square brackets mean substitution.

2) Every judgement $S \vdash A : \mathbb{J}$ over a signature S determines a term coe_A for *coercing* along S-isomorphisms

$$M_1, M_2 : S \vdash coe_A : S^{\cong}[M_1, M_2] \to A[M_1] \to A[M_2]$$

such that coe_A is functorial, i.e., it preserves id_S and $comp_S$ of $S^{\cong}$. Conceptually, this amounts to say that every judgement in S determines a $\mathbb{J}$-valued presheaf over the groupoid $S^{\cong}$ in the LF.

3) Every term $S \vdash a : A$ of a judgement $S \vdash A : \mathbb{J}$ over a signature S determines a term coh_a showing the *coherence* of coercion:

$$\begin{aligned} M_1, M_2 : S \vdash coh_a : & (i : S^{\cong}[M_1, M_2]) \\ & \to coe_A\ i\ a[M_1] = a[M_2] \end{aligned}$$

Conceptually, this amounts to say that every term a determines a natural transformation from the constant $\mathbb{J}$-presheaf 1 to the presheaf A.

The proof of this lemma is omitted here for lack of space, and it can be found in Appendix A.

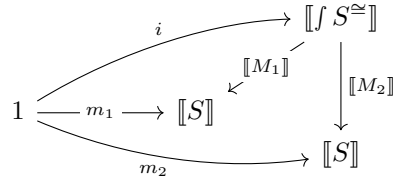
Remark 25. The constructions $\langle S^{\cong}, coe, coh\rangle$ in the preceding proof are essentially the same idea as *observational equality* [2, 4], but at one homotopy level higher. Altenkirch et al. [2] constructed a universe of sets (with extensional principles such as function extensionality and uniqueness of identity proofs) in intensional type theory with proof-irrelevant propositions, while here we constructed groupoids from sets. But the idea of defining equalities or isomorphisms *structurally for each type former* remains the same.

Given an signature S, again, we will reuse S as the name of the iterative Σ-type of the fields of S, and denote the signature $(M_1 : S, M_2 : S, i : S^{\cong})$ by $\int S^{\cong}$. In the category LFSIG of LF-contexts, there are two projections morphisms

$$M_1, M_2 : \int S^{\cong} \longrightarrow S$$

which are interpreted by $[\![-]\!]$ (Section IV-B) in the presheaf category $\mathrm{PR}\,\mathscr{C}$ for every LCCC $\mathscr{C}$ as two morphisms $[\![M_1]\!], [\![M_2]\!] : [\![\int S^{\cong}]\!] \to [\![S]\!]$.

Definition 26. Given a signature S and an LCCC $\mathscr{C}$, an *isomorphism* of diagrammatic models $m_1, m_2 : 1 \to [\![S]\!]$ in $\mathscr{C}$ is a morphism $i : 1 \to [\![\int S^{\cong}]\!]$ in $\mathrm{PR}\,\mathscr{C}$ such that $m_1 = [\![M_1]\!] \cdot i$ and $m_2 = [\![M_2]\!] \cdot i$:



The groupoid of diagrammatic models and isomorphisms, induced by the internal groupoid structure on $S^{\cong}$, is denoted as S-MOD$(\mathscr{C})$.

Theorem 27. *For every LCCC $\mathscr{C}$ and signature S, the mappings F and D between diagrammatic and functorial models from 21 and 22 extend to an equivalence:*

$$F : S\text{-MOD}(\mathscr{C}) \cong \mathrm{LCCC}_{\cong}(\mathrm{JDG}\,S, \mathscr{C}) : D$$

where $\mathrm{LCCC}_{\cong}(\mathrm{JDG}\,S, \mathscr{C})$ is the groupoid of (1) functors $\mathrm{JDG}\,S \to \mathscr{C}$ that preserve locally cartesian closed structures and (2) natural isomorphisms.

The proof of is omitted here for lack of space, and it can be found in Appendix A.

VI. DISCUSSION

In this paper we have developed the categorical semantics of the logical framework LCCLF. Our development follows the tradition of categorical logic pioneered by Lawvere [44]: a syntactic presentation S of a theory generates a *classifying category* C with certain structures, and structure-preserving functors $C \to D$ are equivalent to models of S in D.

Our development is presented in the traditional set-theoretic language, but we did not rely on anything specific to material set theories or the axiom of choice, as long as concepts such as LCCCs are all defined as categories with chosen structures rather than just mere existence of structures. Therefore the development in this paper is valid internally in any elementary topos with an NNO and universes. This in particular includes the effective topos EFF [53, 34], which is useful for studying type theories with impredicative polymorphisms.

Unlike in the untyped and simply typed settings [13, 35], the definition of diagrammatic models of LCCLF is significantly harder than that of functorial models because of type dependency. The payoff of our effort of bridging these two sides is that functorial semantics unlocks the opportunity for using *abstract* categorical tools, while diagrammatic semantics allows us to use *concrete* type-theoretic internal languages of categories to define models of type theories.

The development of LCCLF in this paper is by no means the end of the story. Firstly, it is worthwhile to mechanise LCCLF in a proof assistant. Secondly, signatures of LCCLF in this paper are finitary in two aspects: there can only be finitely many declarations, and every operation has finitely many operands. It is worthwhile to relax both restrictions, so that, for example, type theories with countably many universes or infinitary products can be accommodated.

References

[1] J. Adamek, J. Rosicky, E. M. Vitale, and F. W. Lawvere, *Algebraic Theories*. Cambridge: Cambridge University Press, 2010. [Online]. Available: http://ebooks.cambridge.org/ref/id/CBO9780511760754

[2] T. Altenkirch, C. McBride, and W. Swierstra, "Observational equality, now!" in *Proceedings of the 2007 Workshop on Programming Languages Meets Program Verification*, ser. PLPV '07. Association for Computing Machinery, 2007, p. 57–68.

[3] T. Altenkirch, P. Capriotti, G. Dijkstra, N. Kraus, and F. Nordvall Forsberg, "Quotient inductive-inductive types," in *Foundations of Software Science and Computation Structures*, C. Baier and U. Dal Lago, Eds. Springer International Publishing, 2018, pp. 293–310.

[4] T. Altenkirch, S. Boulier, A. Kaposi, and N. Tabareau, *Setoid Type Theory – A Syntactic Translation*. Springer International Publishing, 2019, p. 155–196.

[5] C. Angiuli and D. Gratzer, "Principles of dependent type theory," 2024, draft. [Online]. Available: https://www.danielgratzer.com/papers/type-theory-book.pdf

[6] S. Awodey, "Natural models of homotopy type theory," *Mathematical Structures in Computer Science*, vol. 28, no. 2, p. 241–286, 2018.

[7] J. Bénabou, *Problèmes dans les topos: d'après le cours de Questions spéciales de mathématique: rapport no 34, mars 1973, Seminaires de mathématique pure*. Université catholique de Louvain, 1973.

[8] ——, "Fibrations petites et localement petites," *C. R. Acad. Sci. Paris*, vol. 281, pp. 897–900, 1975. [Online]. Available: https://gallica.bnf.fr/ark:/12148/bpt6k6228235m/f171.image

[9] J. Cartmell, "Generalised algebraic theories and contextual categories," *Annals of Pure and Applied Logic*, vol. 32, pp. 209–243, 1986.

[10] S. Castellan, P. Clairambault, and P. Dybjer, *Categories with Families: Unityped, Simply Typed, and Dependently Typed*. Springer International Publishing, 2021, pp. 135–180.

[11] P. Clairambault and P. Dybjer, "The biequivalence of locally cartesian closed categories and Martin-Löf type theories," *Mathematical Structures in Computer Science*, vol. 24, no. 6, 2014.

[12] P. Cohn, *Universal Algebra*, ser. Mathematics and Its Applications. Springer Dordrecht, 1981.

[13] R. L. Crole, *Categories for Types*. Cambridge University Press, 1994.

[14] P.-L. Curien, R. Garner, and M. Hofmann, "Revisiting the categorical interpretation of dependent type theory," *Theoretical Computer Science*, vol. 546, p. 99–119, 2014.

[15] M. de Boer, "A proof and formalization of the initiality conjecture of dependent type theory," p. 94, 2020, licentiate defense over Zoom.

[16] P. Dybjer, *Internal type theory*. Springer Berlin Heidelberg, 1996, p. 120–134.

[17] M. Fiore, "Discrete generalised polynomial functors," 2012, talk given at ICALP 2012. [Online]. Available: https://www.cl.cam.ac.uk/~mpf23/talks/ICALP2012.pdf

[18] M. Fiore and C.-K. Hur, "Second-order equational logic," in *Computer Science Logic*, A. Dawar and H. Veith, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 320–335.

[19] M. Fiore and O. Mahmoud, "Second-order algebraic theories," in *Mathematical Foundations of Computer Science 2010*, P. Hliněný and A. Kučera, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 368–380.

[20] M. Fiore and D. Szamozvancev, "Formal metatheory of second-order abstract syntax," *Proc. ACM Program. Lang.*, vol. 6, no. POPL, 2022.

[21] J.-Y. Girard, "Interprétation fonctionelle et élimination des coupures de l'arithmétique d'ordre supérieur," Thèse d'État, Université Paris VII, 1972.

[22] J. Giraud, "Cohomologie non abélienne," *C. R. Acad. Sci. Paris*, vol. 260, pp. 2666–2668, 1965.

[23] D. Gratzer, "Syntax and semantics of modal type theory," Ph.D. dissertation, Aarhus University, 2023. [Online]. Available: https://pure.au.dk/portal/en/publications/syntax-and-semantics-of-modal-type-theory

[24] D. Gratzer and J. Sterling, "Syntactic categories for dependent type theory: sketching and adequacy," 2021. [Online]. Available: https://arxiv.org/abs/2012.10783

[25] D. Gratzer, M. Shulman, and J. Sterling, "Strict universes for Grothendieck topoi," 2022. [Online]. Available: https://arxiv.org/abs/2202.12012

[26] H. Grodin, Y. Niu, J. Sterling, and R. Harper, "Decalf: A directed, effectful cost-aware logical framework," *Proceedings of the ACM on Programming Languages*, vol. 8, no. POPL, p. 273–301, 2024.

[27] R. Harper, "An equational logical framework for type theories," 2021. [Online]. Available: https://arxiv.org/abs/2106.01484

[28] R. Harper, F. Honsell, and G. Plotkin, "A framework for defining logics," *Journal of the ACM*, vol. 40, no. 1, p. 143–184, 1993.

[29] M. Hofmann, "Extensional concepts in intensional type theory," Ph.D. dissertation, University of Edinburgh, 1995. [Online]. Available: https://era.ed.ac.uk/handle/1842/399

[30] ——, "On the interpretation of type theory in locally cartesian closed categories," *Lecture Notes in Computer Science*, vol. 933, p. 427–441, 1995.

[31] ——, "Syntax and semantics of dependent types," in *Semantics and Logics of Computation*, ser. Publications of the Newton Institute, A. Pitts and P. Dybjer, Eds. Cambridge University Press, 1997, p. 79–130.

[32] M. Hofmann and T. Streicher, *The groupoid interpretation of type theory*. Oxford University Press, 1998.

[33] ——, "Lifting Grothendieck universes," 1999, unpublished note. [Online]. Available: https://www2.mathematik.tu-darmstadt.de/~streicher/NOTES/lift.pdf

[34] J. Hyland, *The Effective Topos*. Elsevier, 1982, vol. 110,

p. 165–216.

[35] B. Jacobs, *Categorical Logic and Type Theory*, ser. Studies in Logic and the Foundations of Mathematics. Amsterdam: North Holland, 1999, no. 141.

[36] A. Kaposi and S. Xie, "Second-Order Generalised Algebraic Theories: Signatures and First-Order Semantics," in *9th International Conference on Formal Structures for Computation and Deduction (FSCD 2024)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), J. Rehof, Ed., vol. 299. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024.

[37] A. Kaposi, A. Kovács, and T. Altenkirch, "Constructing quotient inductive-inductive types," *Proc. ACM Program. Lang.*, vol. 3, no. POPL, 2019.

[38] A. Kaposi, A. Kovács, and A. Lafont, "For finitary induction-induction, induction is enough," in *25th International Conference on Types for Proofs and Programs (TYPES 2019)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), M. Bezem and A. Mahboubi, Eds., vol. 175. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020, pp. 6:1–6:30.

[39] C. Kapulkin and Y. Li, "Extensional concepts in intensional type theory, revisited," 2023, doi:https://doi.org/10.48550/arXiv.2310.05706.

[40] K. Kapulkin and P. L. Lumsdaine, "The simplicial model of Univalent Foundations (after Voevodsky)," *Journal of the European Mathematical Society*, vol. 23, no. 6, p. 2071–2126, 2021.

[41] Y. Kinoshita, J. Power, and M. Takeyama, "Sketches," *Journal of Pure and Applied Algebra*, vol. 143, no. 1–3, p. 275–291, 1999.

[42] A. Kovács, "Type-theoretic signatures for algebraic theories and inductive types," Ph.D. dissertation, Eötvös Loránd University, 2023. [Online]. Available: https://arxiv.org/abs/2302.08837

[43] J. Lambek and P. J. Scott, *Introduction to higher order categorical logic*. Cambridge University Press, 1986.

[44] F. Lawvere, "Functorial semantics of algebraic theories," Ph.D. dissertation, Columbia University, 1963. [Online]. Available: http://www.tac.mta.ca/tac/reprints/articles/5/tr5abs.html

[45] P. L. Lumsdaine and M. A. Warren, "The local universes model: An overlooked coherence construction for dependent type theories," *ACM Trans. Comput. Logic*, vol. 16, no. 3, 2015.

[46] P. Martin-Löf, *Intuitionistic type theory*. Bibliopolis Naples, 1984, vol. 6.

[47] C. Maurer, "Universes in topoi," in *Model Theory and Topoi*, F. W. Lawvere, C. Maurer, and G. C. Wraith, Eds. Springer Berlin Heidelberg, 1975, pp. 284–296.

[48] C. Newstead, "Algebraic models of dependent type theory," Ph.D. dissertation, Carnegie Mellon University, 2018.

[49] Y. Niu, J. Sterling, H. Grodin, and R. Harper, "A cost-aware logical framework," *Proc. ACM Program. Lang.*, vol. 6, no. POPL, 2022.

[50] nLab, "slice of presheaves is presheaves on slice," https://ncatlab.org/nlab/show/slice+of+presheaves+is+presheaves+on+slice, 2024, Revision 20.

[51] B. Nordström, K. Petersson, and J. M. Smith, *Programming in Martin-Löf's type theory*. Oxford University Press, 1990, vol. 200.

[52] U. Norell, "Dependently typed programming in Agda," in *Advanced Functional Programming: 6th International School, AFP 2008, Heijen, The Netherlands, May 2008, Revised Lectures*, P. Koopman, R. Plasmeijer, and D. Swierstra, Eds. Springer Berlin Heidelberg, 2009, pp. 230–266.

[53] J. v. Oosten, *Realizability: an introduction to its categorical side*, 1st ed., ser. Studies in logic and the foundations of mathematics. Oxford: Elsevier, 2008.

[54] F. Pfenning, *Logical Frameworks—A Brief Introduction*. Springer Netherlands, 2002, p. 137–166.

[55] A. Pitts, "Categorical logic," in *Handbook of Logic in Computer Science: Volume 5. Algebraic and Logical Structures*. Oxford University Press, 2001.

[56] G. Plotkin, "LCF considered as a programming language," *Theoretical Computer Science*, vol. 5, no. 3, pp. 223–255, 1977.

[57] J. C. Reynolds, "Towards a theory of type structure," in *Programming Symposium, Proceedings Colloque sur la Programmation, Paris, France, April 9-11, 1974*, ser. Lecture Notes in Computer Science, B. Robinet, Ed., vol. 19. Springer, 1974, pp. 408–423.

[58] R. A. Seely, "Locally cartesian closed categories and type theory," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 95, no. 1, p. 33–48, 1984.

[59] J. Sterling, "Algebraic type theory and universe hierarchies," 2019, doi:10.48550/arXiv.1902.08848.

[60] ——, "First steps in synthetic Tait computability: The objective metatheory of cubical type theory," Ph.D. dissertation, Carnegie Mellon University, 2021, version 1.1, revised May 2022.

[61] J. Sterling and C. Angiuli, "Normalization for cubical type theory," *Proceedings - Symposium on Logic in Computer Science*, vol. 2021-June, p. 1–22, 2021, arXiv: 2101.11479.

[62] J. Sterling and R. Harper, "Sheaf semantics of termination-insensitive noninterference," in *7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), A. P. Felty, Ed., vol. 228. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, pp. 5:1–5:19.

[63] ——, "Logical relations as types: Proof-relevant parametricity for program modules," *Journal of the ACM*, vol. 68, no. 6, 2021.

[64] T. Streicher, "Universes in toposes," *From Sets and Types to Topology and Analysis: Towards practical foundations for constructive mathematics.*, vol. 48, 2005.

[65] ——, *Semantics of Type Theory: Correctness, Completeness and Independence Results*. Boston, MA: Birkhäuser Boston, 1991.

[66] ——, "Fibered categories a la Jean Benabou," 2023. [Online]. Available: https://arxiv.org/abs/1801.02927
[67] T. Uemura, "A general framework for the semantics of type theory," *Mathematical Structures in Computer Science*, vol. 33, no. 3, p. 134–179, 2023.
[68] ——, "Abstract and concrete type theories," Ph.D. dissertation, University of Amsterdam, 2021. [Online]. Available: https://dare.uva.nl/search?identifier=41ff0b60-64d4-4003-8182-c244a9afab3b
[69] V. Voevodsky, "A C-system defined by a universe category," *Theory and Applications of Categories*, vol. 30, pp. 1181–1214, 2015.
[70] ——, "The (Π, λ)-structures on the C-systems defined by universe categories," *Theory and Applications of Categories*, vol. 32, pp. 113–121, 2017.
[71] C. Wells, "Sketches: Outline with references," 1993, available at https://ncatlab.org/nlab/files/Wells_Sketches.pdf.

APPENDIX A
PROOFS OMITTED IN THE MAIN TEXT

The proofs omitted in the main text are listed in this section, with their corresponding statements restated for convenience.

Lemma 28. LCCLF satisfies the following statements:

1) Every LF-signature S determines a judgement of *S-isomorphisms*
$$M_1 : S,\ M_2 : S \vdash S^{\cong} : \mathbb{J}$$
in which we reuse the name S for the iterative Σ-type of the fields of S. Moreover, there are terms inv, id, $comp$ of the following types and they satisfy (judgementally in the LF) the axioms of groupoids:
$$\begin{aligned} M_1, M_2 : S \vdash\ & inv_S\ : S^{\cong}[M_1, M_2] \to S^{\cong}[M_2, M_1] \\ M_1 : S \vdash\ & id_S\ : S^{\cong}[M_1, M_1] \\ M_1, M_2, M_3 : S \vdash\ & comp_S : S^{\cong}[M_1, M_2] \to S^{\cong}[M_2, M_3] \\ & \to S^{\cong}[M_1, M_3] \end{aligned}$$
where square brackets mean substitution.
2) Every judgement $S \vdash A : \mathbb{J}$ over a signature S determines a term coe_A for *coercing* along S-isomorphisms
$$M_1, M_2 : S \vdash coe_A : S^{\cong}[M_1, M_2] \to A[M_1] \to A[M_2]$$
such that coe_A is functorial, i.e., it preserves id_S and $comp_S$ of $S^{\cong}$. Conceptually, this amounts to say that every judgement in S determines a $\mathbb{J}$-valued presheaf over the groupoid $S^{\cong}$ in the LF.
3) Every term $S \vdash a : A$ of a judgement $S \vdash A : \mathbb{J}$ over a signature S determines a term coh_a showing the *coherence* of coercion:
$$\begin{aligned} M_1, M_2 : S \vdash coh_a : & (i : S^{\cong}[M_1, M_2]) \\ & \to coe_A\ i\ a[M_1] = a[M_2] \end{aligned}$$
Conceptually, this amounts to say that every term a determines a natural transformation from the constant $\mathbb{J}$-presheaf 1 to the presheaf A.

Proof of Lemma 24. Following Section IV-C, it is sufficient to show these statements only for standard signatures and standard judgements, since these statements can be transported along isomorphisms of LF-signatures. We show these statements simultaneously by induction on the structure of standard signatures, standard judgements, and terms.

Part 1. We start with defining a groupoid $S^{\cong}$ for every signature S, with the inductive hypothesis that all the claims above about signatures, judgements and terms are already shown for structurally smaller cases.

Case 1.1. If S is the empty signature, we let $S^{\cong}$ be the unit type 1 together with the trivial groupoid structure.

Case 1.2. If $S = (T, a : A)$ for some $T \vdash A : \mathbb{J}$, in the context of $M_1, M_2 : S$, we will write $M_i.T$ and $M_i.a$ for the first and second projections of $M_i : S$, and we let $M_1, M_2 : S \vdash S^{\cong} : \mathbb{J}$ be the judgement
$$\Sigma(i : T^{\cong}[M_1.T, M_2.T]).\ \big(coe_A[M_1.T, M_2.T]\ i\ M_1.a = M_2.a\big).$$
The groupoid structure on this $S^{\cong}$ is defined by that of $T^{\cong}$ and the assumption that coe_A preserves the structure id_T and $comp_T$ of $T^{\cong}$. Conceptually, this definition of $S^{\cong}$ is the category of elements for the presheaf determined by $T \vdash A : \mathbb{J}$.

Case 1.3. If $S = (T, B : A \to \mathbb{J})$ for some $T \vdash A : \mathbb{J}$, in the context of $M_1, M_2 : S$, we will write $A_1, A_2 : \mathbb{J}$ for $A[M_1]$ and $A[M_2]$ respectively and similarly B_1 and B_2 for $B[M_1]$ and $B[M_2]$ respectively. We define $M_1, M_2 : S \vdash S^{\cong} : \mathbb{J}$ to be
$$\begin{aligned} & \Sigma(i : T^{\cong}[M_1.T, M_2.T]). \\ & \quad (a_1 : A_1) \to (B_1\ a_1 \cong B_2\ (coe_A\ i\ a_1)) \end{aligned}$$
The groupoid structure on this $S^{\cong}$ is defined using that of $T^{\cong}$ and the evident groupoid structure on isomorphisms of judgements in $\mathbb{J}$.

Part 2. Now we construct coe_A for every possibility of standard judgements.

Case 2.1. If the judgement is $S \vdash B\ a : \mathbb{J}$ for some variable $B : A \to \mathbb{J}$ in the context S and $S \vdash a : A$, we need to define
$$M_1, M_2 : S \vdash coe_{B\ a} : S^{\cong} \to B_1\ a_1 \to B_2\ a_2$$
where a_i and B_i are $a[M_i]$ and $B[M_i]$ as usual. Let $S = (T, B : A \to \mathbb{J}, S')$. In the context of $M_1, M_2 : S$, given any $i : S^{\cong}$, by the definition of $(T, B : A \to \mathbb{J})^{\cong}$ in Case 1.3, we can project out from $i : S^{\cong}$ to an element
$$p_i : \Sigma(j : T^{\cong}).\ (a_1 : A_1) \to (B_1\ a_1 \cong B_2\ (coe_A\ j\ a_1))$$
Now given any $b_1 : B_1\ a_1$, using $(\pi_2\ p_i\ a_1).fwd$, we get an element of type $B_2\ (coe_A\ (\pi_1\ p_i)\ a_1)$. Use the coherence $coh_a\ i : coe_A\ (\pi_1\ p_i)\ a_1 = a_2$, we get an element of $B_2\ a_2$.

Case 2.2. The case for the unit judgement is simple. There is a unique choice of
$$M_1, M_2 : S \vdash coe_1 : S^{\cong}[M_1, M_2] \to 1 \to 1$$
which is functorial because 1 has a unique element.

Case 2.3. For the case $S \vdash \Sigma\ A\ B : \mathbb{J}$ for some $S \vdash A : \mathbb{J}$ and $S.A \vdash B : \mathbb{J}$, we need to define a term that coerce elements of Σ-types along isomorphisms:

$$M_1, M_2 : S \vdash coe_{\Sigma AB} : S^{\cong}[M_1, M_2] \to \Sigma\ A_1\ B_1 \to \Sigma\ A_2\ B_2$$

where A_i and B_i stand for $A[M_i]$ and $(M_1, M_2 : S, a_i : A_i \vdash B[M_i, a_i])$ respectively. By the inductive hypotheses, we can use terms

$$M_1, M_2 : S \vdash coe_A : S^{\cong}[M_1, M_2] \to A_1 \to A_2$$
$$M_1', M_2' : S.A \vdash coe_B : (S.A)^{\cong}[M_1', M_2'] \to B[M_1'] \to B[M_2']$$

We can use coe_A to coerce the first component of $\Sigma\, A\, B$:

$$coe_{\Sigma AB}\ i\ (a_1, b_1) = (coe_A\ i\ a_1,\ ?0 : B[M_2, coe_A\ i\ a_1]\ )$$

Now to use coe_B to fill out the hole $?0$, we recall that the judgement $(S.A)^{\cong}$ of $(S.A)$-isomorphisms is defined earlier in Case 1.2 to be

$$M_1', M_2' : S.A \vdash \Sigma(i : S^{\cong}[M_1'.S, M_2'.S]).\ \big(coe_A\ i\ (\pi_2\ M_1') = \pi_2\ M_2'\big) : \mathbb{J}$$

Thus we fill out the hole $?0$ by putting

$$?0 = coe_B[(M_1, a_1), (M_2, coe_A\ i\ a_1)]\ (i, \mathit{refl})\ b_1$$

and the resulting coercion term is

$$coe_{\Sigma AB}\ i\ (a_1, b_1) = (coe_A\ i\ a_1,\ coe_B[(M_1, a_1), (M_2, coe_A\ i\ a_1)]\ (i, \mathit{refl})\ b_1).$$

whose functoriality follows from that of coe_A and coe_B.

Case 2.4. The case of $\Pi\ A\ B$ is similar to the one of $\Sigma\ A\ B$ above, except that we need to use the backward direction of an isomorphism to coerce A_2 to A_1 in order to coerce a function $f : (a_1 : A_1) \to B_1$ to a function $(a_2 : A_2) \to B_2$:

$$coe_{\Pi AB}\ i\ f = \lambda(a_2 : A_2).\ coe_B[\sigma]\ \ (i, \mathit{refl})\ \ (f\ (coe_A\ (inv_S\ i)\ a_2))$$

where the substitution σ is $[(M_1, coe_A\ (inv_S\ i)\ a_2),\ (M_2, a_2)]$. This definition is well typed because coe_A is functorial, so

$$coe_A\ i\ (coe_A\ (inv_S\ i)\ a_2) = a_2$$

and thus (i, refl) is indeed an element of $(S.A)^{\cong}[\sigma]$.

Case 2.5. For the case $S \vdash a = b : \mathbb{J}$, where $S \vdash A : \mathbb{J}$ and $S \vdash a, b : A$, we need

$$M_1, M_2 : S \vdash coe_{a=b} : S^{\cong}[M_1, M_2] \to (a_1 = b_1) \to (a_2 = b_2)$$

where a_i and b_i stand for $a[M_i]$ and $b[M_i]$ as usual. Given $i : S^{\cong}[M_1, M_2]$ and $a_1 = b_1$, we have $coe_A\ i\ a_1 = coe_A\ i\ b_1$, and the inductive hypotheses give us

$$M_1, M_2 : S \vdash coh_a\ i : (coe_A\ i\ a_1) = a_2$$
$$M_1, M_2 : S \vdash coh_b\ i : (coe_A\ i\ b_1) = b_2$$

Hence we have $a_2 = b_2$ as required.

Part 3. Finally, we need to verify that every term $S \vdash a : A$ satisfies the coherence condition:

$$M_1, M_2 : S \vdash coh_a : (i : S^{\cong}[M_1, M_2]) \to coe_A\ i\ a[M_1] = a[M_2]$$

We omit the details here because it is relatively routine: for the case where a is a variable in the S, the coherence is guaranteed by the definition of $(T, a : A)^{\cong}$ earlier in Case 1.2; the case for other term formers follow from the inductive hypotheses of subterms and the $\beta\eta$-equalities of the connectives. □

Theorem 27. *For every LCCC $\mathscr{C}$ and signature S, the mappings F and D between diagrammatic and functorial models from 21 and 22 extend to an equivalence:*

$$F : S\text{-}\textsc{Mod}(\mathscr{C}) \cong \textsc{LCCC}_{\cong}(\textsc{Jdg}\, S, \mathscr{C}) : D$$

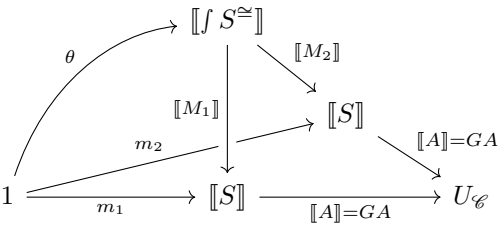
where $\textsc{LCCC}_{\cong}(\textsc{Jdg}\, S, \mathscr{C})$ is the groupoid of (1) functors $\textsc{Jdg}\, S \to \mathscr{C}$ that preserve locally cartesian closed structures and (2) natural isomorphisms.

Proof sketch of Theorem 27. So far F and D are merely functions between diagrammatic and functorial models, so we first need to extend them to functors and then show that they form an equivalence.

Part 1. Let us start with extending F to a functor. Let $m_1, m_2 : 1 \to [\![S]\!]$ be two diagrammatic models and $\theta : 1 \to [\![\int S^{\cong}]\!]$ be an isomorphism between them. By Definition 21, $F(m_i) : \textsc{Jdg}\, S \to \mathscr{C}$ is the composite

$$\textsc{Jdg}\, S \xrightarrow{G} [U_{\mathscr{C}}]_{[\![S]\!]} \xrightarrow{m_i^*} [U_{\mathscr{C}}]_1 \xrightarrow{\cong} \mathscr{C},$$

where G maps every $S \vdash A : \mathbb{J}$ to its interpretation $[\![A]\!] : [\![S]\!] \to U_{\mathscr{C}}$, viewed as an object of the fiber category $[U_{\mathscr{C}}]_{[\![S]\!]}$. To define $F(\theta)$, it is sufficient to define a natural isomorphism $m_1^* G \cong m_2^* G$; by the definition of diagrammatic models, $m_i = [\![M_i]\!] \cdot \theta$, so it is furthermore sufficient to define a natural isomorphism $[\![M_1]\!]^* G \cong [\![M_2]\!]^* G : \textsc{Jdg}\, S \to [U_{\mathscr{C}}]_{[\![\int S^{\cong}]\!]}$. The situation is as follows:



To define the required natural transformation at every judgement $S \vdash A : \mathbb{J}$, we use the coerce morphism of the judgement A from Lemma 24 in the uncurried form:

$$M_1, M_2 : S, i : S^{\cong}[M_1, M_2] \vdash \overline{coe_A} : A[M_1] \to A[M_2].$$

The interpretation of this morphism in $\textsc{Pr}\,\mathscr{C}$ is a morphism

$$[\![\overline{coe_A}]\!] : [\![M_1]\!]^*[\![A]\!] \to [\![M_2]\!]^*[\![A]\!]$$

in the fiber category $[U_{\mathscr{C}}]_{[\![\int S^{\cong}]\!]}$ that we are looking for. The invertibility of this map follows from the invertibility inv_S of $S^{\cong}$ in Lemma 24.

Naturality of the family of morphisms defined above comes from the coherence terms in Lemma 24: given a morphism $f : A \to B$ in $\textsc{Jdg}\, S$, it induces a term $S \vdash \lambda f : A \to B$ in the LF. Now using the coherence $coh_{\lambda f}$ from Lemma 24, we have a commutative diagram in the category $\textsc{Jdg}(\int S^{\cong})$:

$$\begin{array}{ccc} A[M_1] & \xrightarrow{\overline{coe_A}} & A[M_2] \\ {\scriptstyle f[M_1]}\downarrow & & \downarrow{\scriptstyle f[M_2]} \\ B[M_1] & \xrightarrow[\overline{coe_B}]{} & B[M_2] \end{array}$$

whose interpretation in $[U_{\mathscr{C}}]_{[\![\int S^{\cong}]\!]}$ is exactly the needed naturality square.

Part 2. To extend D to a functor $\textsc{Lccc}_{\cong}(\textsc{Jdg}\, S, \mathscr{C}) \to S\text{-}\textsc{Mod}(\mathscr{C})$, given two LCC-functors $N_1, N_2 : \textsc{Jdg}\, S \to \mathscr{C}$ and a natural isomorphism $\sigma : N_1 \cong N_2$, we would like to define $D(\sigma) : 1 \to [\![\int S^{\cong}]\!]$ in $\textsc{Pr}\, \mathscr{C}$ that lies over $D(N_1)$ and $D(N_2)$. Following the structure of the proofs of Lemma 22 and Lemma 24, we define $D(\sigma)$ by induction on the structure of standard signatures S, and maintain the invariant that the coercion $\int S^{\cong} \vdash \overline{coe_A} : A[M_1] \to A[M_2]$ for every judgement A is mapped by the functor $F(D(\sigma)) : \textsc{Jdg}(\int S^{\cong}) \to \mathscr{C}$ to exactly the component of the natural isomorphism $\sigma_A : N_1 A \to N_2 A$ at A, modulo the isomorphisms $N_i A \cong F(D(N_i))$ in Lemma 22. There is no need to impose any invariant for the coherence terms coh_a in this proof, since in LCCCs two morphisms can be equal in at most one way. This is a rather tedious inductive proof so we only provide a sketch here.

Case 2.1. The case for the empty signature is trivial, as $F(\sigma)$ is unique.

Case 2.2. For $S = (T, a : A)$ where $T \vdash A : \mathbb{J}$, according to Case 1.2 of the proof of Lemma 24, we need a global element of the interpretation of the type

$$\begin{array}{l} \Sigma(i : T^{\cong}[M_1.T, M_2.T]). \\ \quad coe_A[M_1.T, M_2.T]\ i\ M_1.a = M_2.a. \end{array}$$

The first component is obtained from the induction hypothesis for the signature T, and the second component is obtained from the invariant that coercion morphisms are interpreted as the components of the natural isomorphism.

Case 2.3. For $S = (T, B : A \to \mathbb{J})$ where $T \vdash A : \mathbb{J}$, by Case 1.3 of the proof of Lemma 24, we need a global element of the interpretation of the type

$$\begin{array}{l} \Sigma(i : T^{\cong}[M_1.T, M_2.T]). \\ \quad (a_1 : A_1) \to (B_1\ a_1 \cong B_2\ (coe_A\ i\ a_1)). \end{array} \tag{10}$$

The first component is still obtained by induction, and the second component is obtained from the component of $\sigma : N_1 \to N_2$ at the judgement $\Sigma\ A\ B$.

We also need to check that all judgements in a signature S maintain our invariant that the coercion map coe_A is interpreted as the component σ_A. This is again shown by induction on the structure of (standard) judgements, mimicking the structure of the definition of coe_A.

Case 2.a. The case for $S \vdash B\ a : \mathbb{J}$ for some variable $(B : A \to \mathbb{J}) \in S$ follows from the fact that coercion for this case is defined in Case 2.1 of the proof of Lemma 24 to be invoking the forward direction of the second component of (10), which we have defined to be a component of σ.

Case 2.b. The case for the unit type is trivial. The cases for Σ and Γ types uses the fact that N_1 and N_2 preserve LCCC structures and σ is natural, so its components at Σ-types and Π-types behave the same as how the coercion maps of Σ-types and Π-types are defined in the proof of Lemma 24.

Case 2.c. The case for equality types is also trivial, because the interpretation of extensional equality types in LCCCs have at most one element.

Part 3. Next we show that F and D form a pair of equivalence of groupoids. First recall that in Lemma 22, we have already a family of isomorphisms $M \cong F(D(M))$ for all $M : \textsc{Jdg}\, S \to \mathscr{C}$. If we examine the proof of Lemma 22, we can notice that every component of the isomorphism $M \cong F(D(M))$ is defined using the unique morphism into some construct with a universal property. Thus the family of isomorphisms $M \cong F(D(M))$ is necessarily natural in M.

It remains to construct a family of isomorphisms $m \cong D(F(m))$, natural in $m \in S\text{-}\textsc{Mod}(\mathscr{C})$, for all signatures S. Again, this is constructed by induction on the structure of standard signatures S, with the additional invariant that the natural isomorphism $F(D(F(m))) \cong F(m)$ constructed in Lemma 22 coincides with the interpretation of coercion morphisms coe_A in $\mathscr{C}$ for every judgement $S \vdash A : \mathbb{J}$. We will only provide a sketch here.

Case 3.1. For the empty signature, $m \cong D(F(m))$ trivially holds because there is a unique diagrammatic model and there is a unique isomorphism.

Case 3.2. For the signature $S = (T, a : A)$ where $T \vdash A : \mathbb{J}$, by Case 1.2 of Lemma 24, $(M_1, M_2 : S \vdash \int S^{\cong} : \mathbb{J})$ is

$$\begin{array}{l} \Sigma(i : T^{\cong}[M_1.T, M_2.T]). \\ \quad \big(coe_A[M_1.T, M_2.T]\ i\ M_1.a = M_2.a\big). \end{array}$$

We need to construct a global element of $[\![\int S^{\cong}]\!]$ that lies over m and $D(F(m))$. The first component $(i : T^{\cong}[M_1.T, M_2.T])$ can be obtained by the inductive hypothesis for the signature T. The second component $coe_A[M_1.T, M_2.T]\ i\ M_1.a = M_2.a$ follows from the additional invariant that the interpretation of coe_A coincides with the isomorphism $F(D(F(m)))(A) \to F(m)(A)$ in Lemma 22, which was used to define the $(a : A)$ component of $D(F(m))$ in Case 1.2 of Lemma 22.

Case 3.3. For the signature $S = (T, B : A \to \mathbb{J})$ where $T \vdash A : \mathbb{J}$, by Case 1.3 of Lemma 24, $(M_1, M_2 : S \vdash \int S^{\cong} : \mathbb{J})$ is the judgement

$$\begin{array}{l} \Sigma(i : T^{\cong}[M_1.T, M_2.T]). \\ \quad (a_1 : A_1) \to (B_1\ a_1 \cong B_2\ (coe_A\ i\ a_1)). \end{array}$$

The first component is still obtained by induction. To construct the second component, we recall that Case 1.3 of the proof of Lemma 22 defines the component $B : A \to \mathbb{J}$ of the diagrammatic model $D(F(m))$ to be a code of

$$F(m)(\pi_1) : F(m)(\Sigma\ A\ B) \to F(m)(A) \tag{11}$$

in the universe $p : \tilde{U}_{\mathscr{C}} \to U_{\mathscr{C}}$. By the definition of $F(m)$, the B component of m is the classifying map of a morphism isomorphic to (11) in $\textsc{Pr}\,\mathscr{C}/F(m)(A)$, so we have the required $(a_1 : A_1) \to B_1\ a_1 \cong B_2\ (coe_A\ i\ a_1)$. $\square$