Higher-Order Categorical Substructural Logic: Expanding the Horizon of Tripos Theory

Yoshihiro Maruyama(✉)

Research School of Computer Science,
The Australian National University,
Canberra, Australia
yoshihiro.maruyama@anu.edu.au

Abstract. Higher-order intuitionistic logic categorically corresponds to toposes or triposes; here we address what are toposes or triposes for higher-order substructural logics. Full Lambek calculus gives a framework to uniformly represent different logical systems as extensions of it. Here we define higher-order Full Lambek calculus, which boils down to higher-order intuitionistic logic when equipped with all the structural rules, and give categorical semantics for (any extension of) it in terms of triposes or higher-order Lawvere hyperdoctrines, which were originally conceived for intuitionistic logic, and yet are flexible enough to be adapted for substructural logics. Relativising the completeness result thus obtained to different axioms, we can obtain tripos-theoretical completeness theorems for a broad variety of higher-order logics. The framework thus developed, moreover, allows us to obtain tripos-theoretical Girard and Kolmogorov translation theorems for higher-order logics.

1 Introduction

Propositional logic corresponds to a class of algebras; for example, the algebras of classical intuitionistic logic are Heyting algebras. What are, then, the algebras of predicate logic? There is seemingly no agreed concept of algebras of predicate logic. Cylindric algebras [11] give a candidate for it. It is not very clear how far and how uniformly cylindric algebraic semantics can be extended so as to treat different sorts of logical systems, especially substructural logics (linear, relevant, fuzzy, etc.). Lawvere's hyperdoctrines [18] give another concept of algebras of predicate logic, and may be seen as a categorical extension of cylindric algebras (see, e.g., Jacobs [13], which gives a fibrational understanding of cylindric algebras; fibrations and hyperdoctrines as indexed categories are connected with each other via the Grothendieck construction). From an algebraic point of view, a hyperdoctrine is a fibred algebra, i.e., an algebra indexed by a category:

$$P : \mathbf{C}^{\mathrm{op}} \to \mathbf{Alg}.$$

Alg is a category of algebras of propositional logic (e.g., Heyting algebras or BI-algebras as in Biering et al. [2]). There are logical conditions to express quantifiers

U. Fahrenberg et al. (Eds.): RAMiCS 2020, LNCS 12062, pp. 187–203, 2020.
https://doi.org/10.1007/978-3-030-43520-2_12

and others as we shall detail below. The intuitive meaning of the base category $\mathbf{C}$ is the category of types (aka. sorts) or domains of discourse, and then $P(C)$ is the algebra of predicates on a type C. And P is called a predicate functor. Roughly, if a propositional logic L is complete with respect to a variety $\mathbf{Alg}_L$, then the corresponding fibred algebras $P : \mathbf{C}^{\mathrm{op}} \rightarrow \mathbf{Alg}_L$ yield complete semantics for the predicate logic that extends L. This may be called completeness lifting: the completeness of propositional logic with respect to $\mathbf{Alg}$ lifts to the completeness of predicate logic with respect to $P : \mathbf{C}^{\mathrm{op}} \rightarrow \mathbf{Alg}$. While this completeness lifting is demonstrated for first-order logic in [21], in the present paper, we demonstrate completeness lifting for higher-order logic of different sorts.

In order to represent different logical systems in a uniform setting, we rely upon the framework of substructural logics over Full Lambek calculus FL and their algebras (see, e.g., Galatos-Jipsen-Kowalski-Ono [8]); FL algebras (defined below) play the rôle of **Alg** above. Diverse logical systems can be represented as axiomatic extensions of FL, including classical, intuitionistic, fuzzy, relevant, paraconsistent, and linear logics. In this field, there are vital developments of the correspondence between cut elimination and algebraic completion (see Ciabattoni-Galatos-Terui [3], which focus upon the propositional case, but might possibly be extended to the first-order and higher-order cases via the framework of substructural hyperdctrines). In this paper we think of higher-order Full Lambek calculus, which boils down to higher-order intuitionistic logic (as in Lambek-Scott [17]) when equipped with all the structural rules, and give hyperdoctrine semantics for (any extension of) it. Lawvere's hyperdoctrines were originally for intuitionistic logic; yet they are flexible enough so as to be adapted for a variety of substructural logics as we shall see below. Note that, whilst toposes are impredicative, triposes can have their type theories predicative (e.g., Martin-Löf); the two-level structure of triposes allows more flexibility than toposes do.

There is a tight connection between toposes and higher-order hyperdoctrines, which are also called triposes (for triposes, see, e.g., Hyland-Johnstone-Pitts [12] and Pitts [26]; there are actually several non-equivalent definitions of triposes; we simply call higher-order hyperdoctrines triposes). Indeed, toposes and triposes correspond to each other via the two functors of taking subobject hyperdoctrines and of the tripos-to-topos construction (see, e.g., Coumans [4] and Frey [10]); note that the subobject functor Sub of a topos plays the rôle of a predicate functor P above. Both toposes and (intuitionistic) higher-order hyperdoctrines give complete semantics for higher-order intuitionistic logic; the completeness result of this paper generalises this classic result quite vastly in terms of higher-order substructural hyperdoctrines or triposes. The contributions of this paper may be summarised succinctly as follows: (i) higher-order completeness via Full Lambek triposes, which can be instantiated for a broad variety of logical systems; (ii) tripos-theoretical Girard's ! translation and Kolmogorov's $\neg\neg$ translation theorems for higher-order logic, in which the internal language of triposes is at work. As illustrated by the translation theorems, the general framework of the present paper allows us to compare different categorical logics within the one setting (many categorical logics have only been developed locally so far; there has been no global framework to compare them in the same setting).

The rest of the paper is organised as follows. We first present the syntax of Higher-order Full Lambek calculus HoFL, which obtains by adapting higher-order intuitionistic logic to Full Lambek calculus FL. And we introduce the concept of Full Lambek tripos (FL tripos for short; aka. higher-order FL hyperdoctrine; for brevity we use the former terminology), thereby obtaining the higher-order completeness theorem for HoFL. Finally, our general framework thus developed is applied, via the internal language of FL triposes, to the categorical analysis of Girard's and Kolmogorov's translation for higher-order logics.

2 Higher-Order Full Lambek Calculus

In this section we introduce Higher-order Full Lambek calculus HoFL, which extends quantified FL as in Ono [23,24] so that HoFL equipped with all the structural rules boils down to higher-order intuitionistic logic, the logic of toposes (see Lambek-Scott [17], Jacobs [13], or Johnstone [15]). Our presentation of HoFL, especially its type-theoretic part, follows the style of Pitts [25]; thus we write, e.g., "$t : \sigma\ [\Gamma]$" and "$\varphi\ [\Gamma]$", rather than "$\Gamma \vdash t : \sigma$" and "$\Gamma \vdash \varphi$", respectively, where t is a term of type σ in context Γ, and φ is a formula in context Γ.

HoFL is a so-called "logic over type theory" or "logic-enriched type theory" in Aczel's terms; there is an underlying type theory, upon which logic is built (see, e.g., Jacobs [13]). To begin with, let us give a bird's-eye view of the structure of HoFL. The type theory of HoFL is given by simply typed λ-calculus with finite product types (i.e., 1 and $\times$; these amount to the structure of CCCs, cartesian closed categories), and moreover, with the special, distinguished type

$$\mathrm{Prop}$$

which is a "proposition" type, intended to represent a truth-value object Ω on the categorical side. The logic of HoFL is given by Full Lambek calculus FL. The Prop type plays the key rôle of reflecting the logical or propositional structure into the type or term structure: every formula or proposition φ may be seen as a term of type Prop. This is essentially what the subobject classifier Ω of a topos $\mathbf{E}$ is required to satisfy, that is,

$$\mathrm{Sub}_{\mathbf{E}}(\text{-}) \simeq \mathrm{Hom}_{\mathbf{E}}(\text{-}, \Omega).$$

Spelling out the meaning of this axiom in logical terms, we have got

$$\mathrm{Pred}(\sigma) \simeq \mathrm{Term}(\sigma, \mathrm{Prop})$$

which means the structure of predicates on each type σ (or context Γ in general) is isomorphic to the structure of terms from σ to Prop. The logical meaning of Ω may thus be summarised by a sort of reflection principle, namely the reflection of the propositional structure into the type structure, which may also be called the "propositions-as-terms" or "propositions-as-functions" correspondence, arguably lying at the heart of higher-order categorical logic, for Ω would presumably be the raison d'être of higher-order categorical logic (toposes are CCCs with Ω).

The power type $P\sigma$ of a given type σ can be defined in the present framework as $\sigma \rightarrow \mathrm{Prop}$; the comprehension term $\{x : \sigma \mid \varphi\} : P\sigma$ and the membership predicate $s \in t : \mathrm{Prop}$ are definable via λ-abstraction or currying (categorically, transposing) and λ-application (categorically, evaluation), respectively. That is, $\{x : \sigma \mid \varphi\}$ may be defined as $\lambda x : \sigma.\ \varphi$ where φ is seen as a term of type Prop, and also $s \in t$ may be defined as ts where $t : \sigma \rightarrow \mathrm{Prop}$ and $s : \sigma$. These definable operations allow us to express set-theoretical reasoning in higher-order logic. There is, of course, some freedom on the choice of primitives, just as toposes can be defined in terms of either subobject classifiers or power objects. All this is to facilitate an intuitive understanding of the essential features of higher-order logic; we give a formal account below.

The syntactic details of HoFL are as follows. HoFL is equipped with the following logical connectives of Full Lambek calculus:

$$\otimes, \wedge, \vee, \backslash, /, 1, 0, \top, \bot, \forall, \exists.$$

The non-commutativity of HoFL gives rise to two kinds of implication ($\backslash$ and $/$). We have basic variables and types, denoted by letters like x and σ, respectively. And as usual $x : \sigma$ is a formal expression to say that a variable x is of type σ. Note that every variable must be typed in HoFL, unlike untyped FL. A context is a finite list of typings of variables: $x_1 : \sigma_1, ..., x_n : \sigma_n$ which is often abbreviated as Γ. Formulae and terms are then defined within specific contexts. There are relation symbols and function symbols, both in context: $R(x_1, ..., x_n)\ [x_1 : \sigma_1, ..., x_n : \sigma_n]$ is a formal expression to say that R is a relation symbol with variables $x_1, ..., x_n$ of types $\sigma_1, ..., \sigma_n$ respectively; and also $f : \tau\ [x_1 : \sigma_1, ..., x_n : \sigma_n]$ is a formal expression to say that f is a function symbol with its domain (the product of) $\sigma_1, ..., \sigma_n$ and with its codomain τ.

The type constructors of HoFL are product ×, function space →, and the proposition type Prop, which is a nullary type constructor. The term constructors of × and → are as usual: pairing $\langle \text{-}, \text{-} \rangle$ and (first and second) projections π_1, π_2 for product ×, and λ-abstraction and λ-application for function space →. The term constructors of Prop are all the logical connectives of Full Lambek calculus as listed above, the relation symbols taken to be of type Prop and thus working as generators of the terms of type Prop. Formulae in context, $\varphi\ [\Gamma]$, and terms in context, $t : \tau\ [\Gamma]$, are then defined in the usual, inductive manner (our terminology and notation mostly follow Pitts [25]; we are extending his framework so as to encompass higher-order substructural logics). Finally, sequents in contexts are defined as:

$$\Phi \vdash \varphi\ [\Gamma]$$

where Γ is a context, Φ is a finite list of formulae $\varphi_1, ..., \varphi_n$, and all the formulae involved are in context Γ.

So far we have not touched upon any axiom (or inference rule) involved. In the following, we first give axioms for terms, and then for sequents. The axioms for × and → are as usual (see, e.g., Pitts [25]). The axiom for Prop is as follows:

$$\frac{\varphi \vdash \psi\ [\varGamma] \quad \psi \vdash \varphi\ [\varGamma]}{\varphi = \psi : \mathrm{Prop}\ [\varGamma]}\ (prop)$$

This axiom relates the structure of propositions to that of terms, thus guaranteeing the aforementioned "propositions-as-functions" correspondence for higher-order categorical logic. There are several standard rules for contexts and substitution, which are the same as those in Pitts [25] (we do not repeat them here, referring to the Sect. 2 of Pitts [25] for the details). We now turn to inference rules for sequents. We first have the identity and cut rules as follows:

$$\frac{}{\varphi \vdash \varphi\ [\varGamma]}\ (id) \qquad \frac{\varPhi_1 \vdash \varphi\ [\varGamma] \quad \varPhi_2, \varphi, \varPhi_3 \vdash \psi\ [\varGamma]}{\varPhi_2, \varPhi_1, \varPhi_3 \vdash \psi\ [\varGamma]}\ (cut)$$

where ψ may be empty; this applies to the following L (Left) rules as well. Note that HoFL has no structural rule other than the cut rule. The rules governing the use of the logical connectives are as follows.

$$\frac{\varPhi, \varphi, \psi, \varPsi \vdash \chi\ [\varGamma]}{\varPhi, \varphi \otimes \psi, \varPsi \vdash \chi\ [\varGamma]}\ (\otimes L) \qquad \frac{\varPhi \vdash \varphi\ [\varGamma] \quad \varPsi \vdash \psi\ [\varGamma]}{\varPhi, \varPsi \vdash \varphi \otimes \psi\ [\varGamma]}\ (\otimes R)$$

$$\frac{\varPhi, \varphi, \varPsi \vdash \chi\ [\varGamma]}{\varPhi, \varphi \wedge \psi, \varPsi \vdash \chi\ [\varGamma]}\ (\wedge L_1) \qquad \frac{\varPhi, \varphi, \varPsi \vdash \chi\ [\varGamma]}{\varPhi, \psi \wedge \varphi, \varPsi \vdash \chi\ [\varGamma]}\ (\wedge L_2)$$

$$\frac{\varPhi \vdash \varphi\ [\varGamma] \quad \varPhi \vdash \psi\ [\varGamma]}{\varPhi \vdash \varphi \wedge \psi\ [\varGamma]}\ (\wedge R)$$

$$\frac{\varPhi, \varphi, \varPsi \vdash \chi\ [\varGamma] \quad \varPhi, \psi, \varPsi \vdash \chi\ [\varGamma]}{\varPhi, \varphi \vee \psi, \varPsi \vdash \chi\ [\varGamma]}\ (\vee L)$$

$$\frac{\varPhi \vdash \varphi\ [\varGamma]}{\varPhi \vdash \varphi \vee \psi\ [\varGamma]}\ (\vee R_1) \qquad \frac{\varPhi \vdash \varphi\ [\varGamma]}{\varPhi \vdash \psi \vee \varphi\ [\varGamma]}\ (\vee R_2)$$

$$\frac{\varPhi \vdash \varphi\ [\varGamma] \quad \varPsi_1, \psi, \varPsi_2 \vdash \chi\ [\varGamma]}{\varPsi_1, \varPhi, \varphi \backslash \psi, \varPsi_2 \vdash \chi\ [\varGamma]}\ (\backslash L) \qquad \frac{\varphi, \varPhi \vdash \psi\ [\varGamma]}{\varPhi \vdash \varphi \backslash \psi\ [\varGamma]}\ (\backslash R)$$

$$\frac{\varPhi \vdash \varphi\ [\varGamma] \quad \varPsi_1, \psi, \varPsi_2 \vdash \chi\ [\varGamma]}{\varPsi_1, \psi / \varphi, \varPhi, \varPsi_2 \vdash \chi\ [\varGamma]}\ (/L) \qquad \frac{\varPhi, \varphi \vdash \psi\ [\varGamma]}{\varPhi \vdash \psi / \varphi\ [\varGamma]}\ (/R)$$

$$\frac{\varPsi_1, \varPsi_2 \vdash \varphi\ [\varGamma]}{\varPsi_1, 1, \varPsi_2 \vdash \varphi\ [\varGamma]}\ (1L) \qquad \frac{}{\vdash 1\ [\varGamma]}\ (1R)$$

$$\frac{}{0 \vdash\ \ [\varGamma]}\ (0L) \qquad \frac{\varPhi \vdash\ \ [\varGamma]}{\varPhi \vdash 0\ [\varGamma]}\ (0R)$$

$$\frac{}{\varPhi \vdash \top\ [\varGamma]}\ (\top R) \qquad \frac{}{\varPhi_1, \bot, \varPhi_2 \vdash \varphi\ [\varGamma]}\ (\bot L)$$

$$\frac{\varPhi_1, \varphi, \varPhi_2 \vdash \psi\ [x : \sigma, \varGamma]}{\varPhi_1, \forall_\sigma x \varphi, \varPhi_2 \vdash \psi\ [x : \sigma, \varGamma]}\ (\forall L) \qquad \frac{\varPhi \vdash \varphi\ [x : \sigma, \varGamma]}{\varPhi \vdash \forall_\sigma x \varphi\ [\varGamma]}\ (\forall R)$$

$$\frac{\varPhi_1, \varphi, \varPhi_2 \vdash \psi \ [x:\sigma, \varGamma]}{\varPhi_1, \exists_\sigma x\varphi, \varPhi_2 \vdash \psi \ [\varGamma]} \ (\exists L) \qquad \frac{\varPhi \vdash \varphi \ [x:\sigma, \varGamma]}{\varPhi \vdash \exists_\sigma x\varphi \ [x:\sigma, \varGamma]} \ (\exists R)$$

There are eigenvariable conditions on the quantification rules: x must not appear as a free variable in the bottom sequents of the $\forall R$ and $\exists L$ rules. We write $\forall x$ and $\exists x$ when the type of x is obvious. These are all of the rules of HoFL; the provability of sequents in context is defined in the usual way. The essential difference from the first-order case is the existence of function and truth value types; they are what make the logic higher-order, enabling set-theoretical reasoning.

For a collection X of axiom schemata (which we often simply call axioms), let us denote by HoFL_X the axiomatic extension of HoFL via X. In particular, we can recover higher-order intuitionistic logic as HoFL_{ecw}, i.e., by adding to HoFL the exchange, weakening, and contraction rules (as axiom schemata).

Lemma 1. *The following sequents-in-context are deducible in* HoFL*:*

- *(i)* $\varphi \otimes (\exists x\psi) \vdash \exists x(\varphi \otimes \psi) \ [\varGamma]$ *and* $\exists x(\varphi \otimes \psi) \vdash \varphi \otimes (\exists x\psi) \ [\varGamma]$*;*
- *(ii)* $(\exists x\psi) \otimes \varphi \vdash \exists x(\psi \otimes \varphi) \ [\varGamma]$ *and* $\exists x(\psi \otimes \varphi) \vdash (\exists x\psi) \otimes \varphi \ [\varGamma]$

where it is supposed that φ does not contain x as a free variable, and $\varGamma$ contains type declarations on those free variables that appear in φ and $\exists x\psi$.

As explained in [21], typed logic allows domains of discourse to be empty; they must be non-empty in the Tarski semantics. A type σ can be interpreted as an initial object in a category. We need no ad hoc condition on domains of discourse if we work with typed logic. This is due to Joyal as noted in Marquis and Reyes [19]. Proof-theoretically, the following is not deducible in HoFL: $\forall x\varphi \vdash \exists x\varphi \ [\ ]$. Still the following is deducible: $\forall x\varphi \vdash \exists x\varphi \ [x:\sigma, \varGamma]$. That is, we can prove the sequent above when a type σ is inhabited (see [21] for more details).

3 Full Lambek Tripos

The algebras of propositional FL are FL algebras, the definition of which is reviewed below. The algebras of first-order FL are arguably FL hyperdoctrines; note that complete FL algebras only give us completeness in the presence of the *ad hoc* condition of so-called safe valuations (cf. [24]), and yet FL hyperdoctrines allow us to prove completeness without any such *ad hoc* condition, and at the same time, to recover the complete FL algebra semantics as a special, set-theoretical instance of the FL hyperdoctrine semantics (in a nutshell, the condition of safe valuations is only necessary to show completeness with respect to the restricted class of FL hyperdoctrines with the category of sets their base categories). In this section we define FL triposes, which are arguably the (fibred) algebras of higher-order FL, and prove higher-order completeness, again without any ad hoc condition such as safe valuations or Henkin-style restrictions on quantification (set-theoretical semantics is only complete under this condition).

Definition 2. *$(A, \otimes, \wedge, \vee, \backslash, /, 1, 0, \top, \bot)$ is an* FL *algebra iff the following hold:*

- *$(A, \otimes, 1)$ is a monoid; 0 is a distinguished element of A;*
- *$(A, \wedge, \vee, \top, \bot)$ is a bounded lattice;*
- *for any $a \in A$, $a\backslash(\text{-}) : A \to A$ is a right adjoint of $a \otimes (\text{-}) : A \to A$: $a \otimes b \leq c$ iff $b \leq a\backslash c$ for any $a, b, c \in A$;*
- *for any $b \in A$, $(\text{-})/b : A \to A$ is a right adjoint of $(\text{-}) \otimes b : A \to A$: $a \otimes b \leq c$ iff $a \leq c/b$. for any $a, b, c \in A$.*

A homomorphism of FL *algebras is required to preserve all the operations of* FL *algebras. Let* **FL** *denote the category of* FL *algebras and their homomorphisms.*

FL is an algebraic category (namely, a category monadic over the category of sets; see [1]), and then an axiomatic extension FL_X of FL corresponds to an algebraic subcategory of **FL**, which shall be denoted $\mathbf{FL}_X$. Note that algebraic categories are called varieties or equational classes in universal algebra.

Definition 3. *An* FL *(Full Lambek) hyperdoctrine is a contravariant functor*

$$P : \mathbf{C}^{\mathrm{op}} \to \mathbf{FL}$$

such that the base category **C** *of P is a category with finite products, and that the following conditions (to express quantifiers) are satisfied:*

- *For any projection $\pi : X \times Y \to Y$ in* **C***, $P(\pi) : P(Y) \to P(X \times Y)$ has a right adjoint, denoted $\forall_\pi : P(X \times Y) \to P(Y)$. And the corresponding Beck-Chevalley condition holds, i.e., the following diagram commutes for any arrow $f : Z \to Y$ in* **C** *($\pi' : X \times Z \to Z$ below denotes a projection):*

$$\begin{tikzcd} P(X \times Y) \arrow[r, "\forall_\pi"] \arrow[d, "P(X\times f)"'] & P(Y) \arrow[d, "P(f)"] \\ P(X \times Z) \arrow[r, "\forall_{\pi'}"'] & P(Z) \end{tikzcd}$$

- *For any projection $\pi : X \times Y \to Y$ in* **C***, $P(\pi) : P(Y) \to P(X \times Y)$ has a left adjoint, denoted $\exists_\pi : P(X \times Y) \to P(Y)$. The corresponding Beck-Chevalley condition holds:*

$$\begin{tikzcd} P(X \times Y) \arrow[r, "\exists_\pi"] \arrow[d, "P(X\times f)"'] & P(Y) \arrow[d, "P(f)"] \\ P(X \times Z) \arrow[r, "\exists_{\pi'}"'] & P(Z) \end{tikzcd}$$

Furthermore, the Frobenius Reciprocity conditions hold: for any projection $\pi : X \times Y \to Y$ in **C***, any $a \in P(Y)$, and any $b \in P(X \times Y)$,*

$$a \otimes (\exists_\pi b) = \exists_\pi (P(\pi)(a) \otimes b)$$
$$(\exists_\pi b) \otimes a = \exists_\pi (b \otimes P(\pi)(a)).$$

The logical reading of the Beck-Chevalley conditions above is that substitution commutes with quantification.

Now, FL triposes are defined as FL hyperdoctrines with their base categories CCCs, and with truth-value objects Ω (i.e., representability via $\Omega \in \mathbf{C}$):

Definition 4. *An* FL *(Full Lambek) tripos, or higher-order* FL *hyperdoctrine, is an* FL *hyperdoctrine* $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{FL}$ *such that:*

- *The base category* $\mathbf{C}$ *is a CCC (Cartesian Closed Category);*
- *There is an object* $\Omega \in \mathbf{C}$ *such that*

$$P \simeq \mathrm{Hom}_{\mathbf{C}}(\text{-}, \Omega).$$

We then call Ω *the truth-value object of the* FL *tripos* P*. Given a set* X *of axioms, an* FL_X *tripos is defined by replacing* $\mathbf{FL}$ *above with* $\mathbf{FL}_X$*.*

For an FL tripos P, each $P(C)$ is called a fibre of the FL tripos P from a fibrational point of view; intuitively, $P(C)$ may be seen as the algebra of propositions on a type or domain of discourse C. Note that it is also possible to define FL triposes in terms of fibrations, even though the present formulation in terms of indexed categories would be categorically less demanding.

FL tripos semantics for HoFL is defined as follows.

Definition 5. *Let* $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{FL}$ *be an* FL *tripos. An interpretation* $[\![\text{-}]\!]$ *of* HoFL *in the* FL *tripos* P *is defined as follows. Types and atomic symbols are interpreted in the following way:*

- *each basic type* σ *is interpreted as an object* $[\![\sigma]\!]$ *in* $\mathbf{C}$*;*
- *product and function types,* $\sigma \times \tau$ *and* $\sigma \to \tau$*, are interpreted, as usual, by categorical product and exponentiation;*
- *each function symbol* $f : \tau\ [\Gamma]$ *is interpreted as an arrow*

$$[\![f : \tau\ [\Gamma]]\!] : [\![\Gamma]\!] \to [\![\sigma]\!]$$

  in $\mathbf{C}$*; if the context* Γ *is* $x_1 : \sigma_1, ..., x_n : \sigma_n$*, then* $[\![\Gamma]\!]$ *denotes* $[\![\sigma_1]\!] \times ... \times [\![\sigma_n]\!]$*;*
- *each relation symbol* $R\ [\Gamma]$ *is interpreted as an element* $[\![R\ [\Gamma]]\!]$ *in the corresponding fibre* $P([\![\Gamma]\!])$ *of the* FL *tripos* P *at* $[\![\Gamma]\!]$*.*

Terms and their equality are interpreted in the following, inductive manner:

- $[\![x : \sigma\ [\Gamma_1, x : \sigma, \Gamma_2]]\!]$ *is defined as the following projection in* $\mathbf{C}$*:*

$$\pi : [\![\Gamma_1]\!] \times [\![\sigma]\!] \times [\![\Gamma_2]\!] \to [\![\sigma]\!].$$

- $[\![f(t_1, ..., t_n) : \tau\ [\Gamma]]\!]$ *is defined as the following arrow in* $\mathbf{C}$*:*

$$[\![f]\!] \circ \langle [\![t_1 : \sigma_1 \ [\Gamma]]\!], ..., [\![t_n : \sigma_n \ [\Gamma]]\!] \rangle$$

where $f : \tau \ [x_1 : \sigma_1, ..., x_n : \sigma_n]$, *and* $t_1 : \sigma_1 \ [\Gamma], ..., t_n : \sigma_n \ [\Gamma]$ *(note also that* $\langle [\![t_1 : \sigma_1 \ [\Gamma]]\!], ..., [\![t_n : \sigma_n \ [\Gamma]]\!] \rangle$ *denotes the product/pairing of arrows in* $\mathbf{C}$*).*

- λ*-abstraction,* λ*-application, projections, and pairing are interpreted, as usual, by categorical transpose, evaluation, projections, and pairing in the base CCC* $\mathbf{C}$*, respectively;*

Formulae are interpreted in the following, inductive manner:

- $[\![R(t_1, ..., t_n) \ [\Gamma]]\!]$ *is defined as*

$$P(\langle [\![t_1 : \sigma_1[\Gamma]]\!], ..., [\![t_n : \sigma_n[\Gamma]]\!] \rangle)([\![R \ [x : \sigma_1, ..., x_n : \sigma_n]]\!])$$

  where R *is a relation symbol in context* $x_1 : \sigma_1, ..., x_n : \sigma_n$.
- $[\![\varphi \otimes \psi \ [\Gamma]]\!]$ *is defined as* $[\![\varphi \ [\Gamma]]\!] \otimes [\![\psi \ [\Gamma]]\!]$*. The other binary connectives* $\wedge, \vee, \backslash, /$ *are interpreted in a similar way.* $[\![1 \ [\Gamma]]\!]$ *is defined as the monoidal unit of* $P([\![\Gamma]\!])$*. The other constants* $0, \top, \bot$ *are interpreted in a similar way.*
- $[\![\forall x\varphi \ [\Gamma]]\!]$ *is defined as* $\forall_\pi([\![\varphi \ [x : \sigma, \Gamma]]\!])$ *where* $\pi : [\![\sigma]\!] \times [\![\Gamma]\!] \rightarrow [\![\Gamma]\!]$ *is a projection in* $\mathbf{C}$*, and* φ *is a formula in context* $[x : \sigma, \Gamma]$*. Similarly,* $[\![\exists x\varphi \ [\Gamma]]\!]$ *is defined as* $\exists_\pi([\![\varphi \ [x : \sigma, \Gamma]]\!])$*.*

Prop *and its terms are then interpreted as follows:*

- Prop *is interpreted as the truth-value object* Ω *of the* FL *tripos* P*:*

$$[\![\mathrm{Prop}]\!] = \Omega;$$

- *each formula* $\varphi : \mathrm{Prop} \ [\Gamma]$*, regarded as a term of type* Prop*, is interpreted as the element of* $\mathrm{Hom}_{\mathbf{C}}([\![\Gamma]\!], \Omega)$ *which corresponds to* $[\![\varphi \ [\Gamma]]\!] \in P([\![\Gamma]\!])$ *in the defining isomorphism* $P \simeq \mathrm{Hom}_{\mathbf{C}}(\text{-}, \Omega)$ *of the* FL *tripos* P*; in a nutshell,* $[\![\varphi : \mathrm{Prop} \ [\Gamma]]\!]$*'s and* $[\![\varphi \ [\Gamma]]\!]$*'s are linked via the isomorphism.*

Finally, the validity of sequents in context is defined as follows:

- $\varphi_1, ..., \varphi_n \vdash \psi \ [\Gamma]$ *is valid in an interpretation* $[\![\text{-}]\!]$ *in an* FL *tripos* P *iff the following holds in* $P([\![\Gamma]\!])$*:*

$$[\![\varphi_1 \ [\Gamma]]\!] \otimes ... \otimes [\![\varphi_n \ [\Gamma]]\!] \leq [\![\psi \ [\Gamma]]\!].$$

  In case the right-hand side of a sequent is empty, $\varphi_1, ..., \varphi_n \vdash [\Gamma]$ *is valid in* $[\![\text{-}]\!]$ *iff* $[\![\varphi_1 \ [\Gamma]]\!] \otimes ... \otimes [\![\varphi_n \ [\Gamma]]\!] \leq 0$ *in* $P([\![\Gamma]\!])$*. In case the left-hand side of a sequent is empty,* $\vdash \varphi \ [\Gamma]$ *is valid in* $[\![\text{-}]\!]$ *iff* $1 \leq [\![\varphi[\Gamma]]\!]$ *in* $P([\![\Gamma]\!])$*. When* Φ *consists of* $\varphi_1, ..., \varphi_n$*, let* $[\![\Phi \ [\Gamma]]\!]$ *denote* $[\![\varphi_1 \ [\Gamma]]\!] \otimes ... \otimes [\![\varphi_n \ [\Gamma]]\!]$*.*

An interpretation of HoFL_X *in an* FL_X *tripos is defined by replacing* **FL** *and* HoFL *above with* $\mathbf{FL}_X$ *and* HoFL_X*, respectively.*

The categorical conception of interpretation encompasses set-theoretical interpretations and forcing-style model constructions. First of all, interpreting logic in the **2**-valued tripos $\mathrm{Hom}_{\mathbf{Set}}(\text{-}, \mathbf{2})$ (where **2** is the two-element Boolean algebra) is precisely equivalent to the standard Tarski semantics. Yet there is a vast generalisation of this: given a quantale Ω, the representable functor

$$\mathrm{Hom}_{\mathbf{Set}}(\text{-}, \Omega) : \mathbf{Set}^{\mathrm{op}} \to \mathbf{FL}$$

forms an FL tripos, which gives rise to a universe of quantale-valued sets via the generalised tripos-to-topos construction as in [21]; if Ω is a locale in particular (i.e., complete Heyting algebra), it is known that $\mathrm{Hom}_{\mathbf{Set}}(\text{-}, \Omega)$ yields $\mathbf{Sh}(\Omega)$ (i.e., the sheaf topos on Ω). This sort of FL tripos models of set theory could hopefully be applied to solve consistency problems for substructural set theories (especially, Cantor-Łukasiewicz set theory).

Note that the base category of an FL tripos is used to interpret the type theory of HoFL, and the value category is used to interpret the logic part of HoFL. In the following, we first prove soundness and then completeness.

Proposition 6. *If $\Phi \vdash \psi$ $[\Gamma]$ is provable in* HoFL *(resp.* HoFL$_X$*), then it is valid in any interpretation in any* FL *(resp.* FL$_X$*) tripos.*

Proof. Let P be an FL or FL$_X$ tripos, and $[\![\text{-}]\!]$ an interpretation in P. Soundness for the first-order part can be proven in essentially the same way as in [21]; due to space limitations, we do not repeat it, and focus upon Prop, which is the most distinctive part of higher-order logic. So let us prove that the rule for the Prop type preserves validity. Suppose that

$$[\![\varphi\ [\Gamma]]\!] \leq [\![\psi\ [\Gamma]]\!]$$

and that

$$[\![\psi\ [\Gamma]]\!] \leq [\![\varphi\ [\Gamma]]\!].$$

It then follows that

$$[\![\varphi\ [\Gamma]]\!] = [\![\psi\ [\Gamma]]\!].$$

Note that this is a "propositional" equality, i.e., an equality in the fibre $P([\![\Gamma]\!])$ of propositions on $[\![\Gamma]\!]$. Since we have the following isomorphism

$$P([\![\Gamma]\!]) \simeq \mathrm{Hom}_{\mathbf{C}}([\![\Gamma]\!], [\![\mathrm{Prop}]\!])$$

the equality above, together with the definition of the interpretation of terms of type Prop, tells us that

$$[\![\varphi : \mathrm{Prop}\ [\Gamma]]\!] = [\![\psi : \mathrm{Prop}\ [\Gamma]]\!].$$

Note that this is a "functional" equality, i.e., an equality in $\mathrm{Hom}_{\mathbf{C}}([\![\Gamma]\!], [\![\mathrm{Prop}]\!])$. Thus, the propositional equality implies the functional equality (via the isomorphism above), and this is exactly what it is for the Prop rule to preserve validity. □

For the sake of a completeness proof, let us introduce the syntactic tripos construction (for logic over type theory), which is the combination of the syntactic category construction (for type theory) and the Lindenbaum-Tarski algebra construction (for propositional logic):

Definition 7. *The syntactic tripos of* HoFL *is defined as follows. Let us first define the syntactic base category* **C***: an object is a context Γ (up to α-equivalence); an arrow from Γ to Γ' is a list of terms (up to equality on terms)*

$$[t_1, ..., t_n]$$

where $t_1 : \sigma_1\ [\Gamma], ..., t_n : \sigma_n\ [\Gamma]$ and Γ' is supposed to be $x_1 : \sigma_1, ..., x_n : \sigma_n$. Composition is defined via substitution. The syntactic tripos $P_{\mathrm{HoFL}} : \mathbf{C}^{\mathrm{op}} \to \mathbf{FL}$ is then defined as follows. Given an object Γ in **C**, *let* Form_Γ *denote the set of formulas in context Γ, and then define*

$$P_{\mathrm{HoFL}}(\Gamma) = \mathrm{Form}_\Gamma / \sim$$

where $\sim$ is an equivalence relation on Form_Γ *defined as follows: for $\varphi, \psi \in$* Form_Γ, *$\varphi \sim \psi$ iff $\varphi \vdash \psi\ [\Gamma]$ and $\psi \vdash \varphi\ [\Gamma]$ are provable in* HoFL. *The arrow part of P_{HoFL} is defined as follows. Let $[t_1, ..., t_n] : \Gamma \to \Gamma'$ be an arrow in* **C** *where Γ' is $x_1 : \sigma_1, ..., x_n : \sigma_n$. Then we define $P_{\mathrm{HoFL}}([t_1, ..., t_n]) : P_{\mathrm{HoFL}}(\Gamma') \to P_{\mathrm{HoFL}}(\Gamma)$ by*

$$P_{\mathrm{HoFL}}([t_1, ..., t_n])(\varphi\ [\Gamma']) = \varphi[t_1/x_1, ..., t_n/x_n]\ [\Gamma]$$

where it is supposed that $t_1 : \sigma_1\ [\Gamma], ..., t_n : \sigma_n\ [\Gamma]$, and that φ is a formula in context $x_1 : \sigma_1, ..., x_n : \sigma_n$. The syntactic tripos P_{HoFL_X} of HoFL_X *is defined just by replacing* **FL** *and* HoFL *above with* $\mathbf{FL}_X$ *and* HoFL_X, *respectively.*

The syntactic tripos of higher-order logic is the fibrational analogue of the Lindenbaum-Tarski algebra of propositional logic; each fibre $P_{\mathrm{HoFL}}(\Gamma)$ of the syntactic tripos P_{HoFL} is the Lindenbaum-Tarski algebra of formulae in context Γ. The syntactic tripos of HoFL has the universal mapping property that inherits from the syntactic base category of the underlying type theory of HoFL, and also from the fibre-wise Lindenbaum-Tarski algebras of the logic part of HoFL. We of course have to verify that the syntactic tripos P_{HoFL} indeed carries an FL tripos structure; this is the crucial part of the completeness proof.

Lemma 8. *The syntactic tripos $P_{\mathrm{HoFL}} : \mathbf{C}^{\mathrm{op}} \to \mathbf{FL}$ (resp. $\mathbf{FL}_X$) defined above is an* FL *(resp.* FL_X*) tripos. In particular, the base category is a CCC, and there is a truth-value object $\Omega \in \mathbf{C}$ such that*

$$P_{\mathrm{HoFL}} \simeq \mathrm{Hom}_{\mathbf{C}}(\text{-}, \Omega).$$

Proof. The existence of products and exponentials in **C** is guaranteed by the existence of product types and function space types in the type theory of HoFL. Substitution commutes with all the logical connectives. This means that $P([t_1, ..., t_n])$ defined above is a homomorphism; so P is a contravariant functor.

P has quantifier structures as follows. Let $\pi : \Gamma \times \Gamma' \to \Gamma'$ denote the projection in **C** defined above, and consider $P(\pi)$, which has right and left adjoints in the following way. Recall Γ is $x : \sigma_1, ..., x_n : \sigma_n$. Let $\varphi \in P(\Gamma \times \Gamma')$; we identify φ with the equivalence class to which φ belongs. Define $\forall_\pi : P(\Gamma \times \Gamma') \to P(\Gamma')$ by

$$\forall_\pi(\varphi) = \forall x_1 ... \forall x_n \varphi.$$

We also define $\exists_\pi : P(\Gamma \times \Gamma') \rightarrow P(\Gamma')$ by $\exists_\pi(\varphi) = \exists x_1 ... \exists x_n \varphi$. Then, $\forall_\pi$ and $\exists_\pi$ give the right and left adjoints of $P(\pi)$, respectively.

We can verify the Beck-Chevalley condition for $\forall$ as follows. Let $\varphi \in P(\Gamma \times \Gamma')$, $\pi : \Gamma \times \Gamma' \rightarrow \Gamma'$ a projection in $\mathbf{C}$, and $\pi' : \Gamma \times \Gamma'' \rightarrow \Gamma''$ another projection in $\mathbf{C}$ for objects $\Gamma, \Gamma', \Gamma''$ in $\mathbf{C}$. Then,

$$P([t_1, ..., t_n]) \circ \forall_\pi(\varphi) = (\forall x_1 ... \forall x_n \varphi)[t_1/y_1, ..., t_n/y_m]$$

where Γ is supposed to be $x_1 : \sigma_1, ..., x_n : \sigma_n$, Γ' is $y_1 : \tau_1, ..., y_m : \tau_m$, and $t_1 : \tau_1\ [\Gamma''], ..., t_m : \tau_m\ [\Gamma'']$. Likewise we have $\forall_{\pi'} \circ P([t_1, ..., t_n])(\varphi) = \forall x_1 ... \forall x_n (\varphi[t_1/y_1, ..., t_n/y_m])$. The Beck-Chevalley condition for $\forall$ thus follows. The Beck-Chevalley condition for $\exists$ can be verified in a similar way. The two Frobenius Reciprocity conditions for $\exists$ follow immediately from Lemma 1.

In the following we prove the existence of a truth-value object Ω. Let

$$\Omega = x : \mathrm{Prop}.$$

Note that, since the objects of the base category are contexts rather than types, we cannot take Ω to be Prop *per se*; yet x : Prop practically means the same thing as Prop, thanks to α-equivalence required. We now have to show that for each context Γ,

$$P(\Gamma) \simeq \mathrm{Hom}_{\mathbf{C}}(\Gamma, x : \mathrm{Prop})$$

and this correspondence yields a natural transformation. The required isomorphism is given by mapping

$$\varphi\ [\Gamma] \in P(\Gamma)$$

to

$$\varphi : \mathrm{Prop}\ [\Gamma] \in \mathrm{Hom}_{\mathbf{C}}(\Gamma, x : \mathrm{Prop}).$$

Note that φ above is actually an equivalence class, and yet the above mapping is well defined, and also that φ : Prop $[\Gamma]$ is actually a list consisting of a single term φ : Prop $[\Gamma]$. This mapping is an isomorphism by the definition of terms of type Prop. Let us denote the above mapping by

$$\mathrm{PaF}_\Gamma : P(\Gamma) \rightarrow \mathrm{Hom}_{\mathbf{C}}(\Gamma, x : \mathrm{Prop})$$

with the idea of "Propositions-as-Functions" in mind. The naturality of this correspondence then means that the following diagram commutes for any arrow $[t_1, ..., t_n] : \Gamma' \rightarrow \Gamma$ in $\mathbf{C}$:

$$\begin{tikzcd}
P(\Gamma) \arrow[r, "\mathrm{PaF}_\Gamma"] \arrow[d, "P([t_1,...,t_n])"'] & \mathrm{Hom}_{\mathbf{C}}(\Gamma, x : \mathrm{Prop}) \arrow[d, "\mathrm{Hom}_{\mathbf{C}}([t_1,...,t_n], x:\mathrm{Prop})"] \\
P(\Gamma') \arrow[r, "\mathrm{PaF}_{\Gamma'}"'] & \mathrm{Hom}_{\mathbf{C}}(\Gamma', x : \mathrm{Prop})
\end{tikzcd}$$

By the following calculation:

$$\begin{aligned}\mathrm{Hom}_{\mathbf{C}}([t_1,...,t_n],\mathrm{Prop})\circ\mathrm{PaF}_{\Gamma}(\varphi\ [\Gamma]) &= \mathrm{Hom}_{\mathbf{C}}([t_1,...,t_n],\mathrm{Prop})(\varphi:\mathrm{Prop}\ [\Gamma])\\ &= \varphi[t_1/x_1,...,t_n/x_n]:\mathrm{Prop}\ [\Gamma']\\ &= \mathrm{PaF}_{\Gamma'}(\varphi[t_1/x_1,...,t_n/x_n]\ [\Gamma'])\\ &= \mathrm{PaF}_{\Gamma'}\circ P([t_1,...,t_n])(\varphi\ [\Gamma])\end{aligned}$$

we obtain the commutativity of the diagram and hence the naturality of the "propositions-as-functions" correspondence. □

It is straightforward to see that if $\Phi \vdash \psi\ [\Gamma]$ is valid in the canonical interpretation in the syntactic tripos P_{HoFL} (resp. P_{HoFL_X}), then it is provable in HoFL (resp. HoFL_X). And this immediately gives us completeness via the standard counter-model argument. Hence the higher-order completeness theorem:

Theorem 9. *$\Phi \vdash \psi\ [\Gamma]$ is provable in* HoFL *(resp.* HoFL_X*) iff it is valid in any interpretation in any* FL *(resp.* FL_X*) tripos.*

This higher-order completeness theorem can be applied, with a suitable choice of axioms X, for any of classical, intuitionistic, fuzzy, relevant, paraconsistent, and (both commutative and non-commutative) linear logics; higher-order completeness has not been known for these logics except the first two. The concept of (generalised) tripos, therefore, is so broadly applicable as to encompass most logical systems. Modal logics also can readily be incorporated into this framework by working with modal FL rather than plain FL. Coalgebraic dualities for modal logics (see, e.g., [14,16,20,22]) then yield models of modal triposes for them; these modal issues are to be addressed in subsequent papers.

4 Girard and Kolmogorov Translation for Triposes

We finally analyse Kolmogorov's double negation $\neg\neg$ translation (Kolmogorov found it earlier than Gödel-Gentzen; see Ferreira and Oliva [7]) and Girard's exponential ! translation from a tripos-theoretical point of view.

Propositional Kolmogorov translation algebraically means that, for any Heyting algebra A, the doubly negated algebra $\neg\neg A$, defined as $\{a \in A \mid \neg\neg a = a\}$, always forms a Boolean algebra. This $\neg\neg$ construction extends to a functor from the category **HA** of Heyting algebras to the category **BA** of Boolean algebras. And then the categorical meaning of first-order Kolmogorov translation is that, for any first-order IL hyperdoctrine $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{HA}$ (where IL denotes intuitionistic logic), the following composed functor

$$\neg\neg \circ P : \mathbf{C}^{\mathrm{op}} \to \mathbf{BA}$$

forms a first-order CL hyperdoctrine (where CL denotes classical logic) as in [21]. Yet this strategy does not extend to the higher-order case: in particular, although the base category does not change in the first-order case, in which types and propositions are separated, it must nevertheless be modified in the higher-order case, in which types and propositions interact via Prop or Ω. Technicalities involved get essentially more complicated in the higher-order case. Still, we can construct from a given IL tripos $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{HA}$ a CL tripos

$$P_{\neg\neg} : \mathbf{C}_{\neg\neg}^{\mathrm{op}} \to \mathbf{BA}.$$

For the sake of the description of $\mathbf{C}_{\neg\neg}$ (and $P_{\neg\neg}$), however, we work within the internal language HoFL_P of the tripos $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{FL}$: in HoFL_P, we have types C and terms f corresponding to objects C and arrows f in $\mathbf{C}$, respectively, and also formulae R on a type $C \in \mathbf{C}$ corresponding to elements $R \in P(C)$.

Now we define the translation on the internal language HoFL_P of the tripos P which allows us to describe the double negation category $\mathbf{C}_{\neg\neg}$ mentioned above. The basic strategy of translation is this: we leave everything in HoFL_P as it is, unless it involves the proposition type Ω of HoFL_P; and if something involves Ω, we always put double negation on it. Formally it goes as follows:

Definition 10. *We recursively define the translation on* HoFL_P *as follows.*

- *If $\varphi : \Omega\ [\Gamma]$ then we put $\neg\neg$ on every sub-formula of φ (do the same for φ seen as formulae).*
- *If $t : \sigma\ [\Gamma, x : \Omega, \Gamma']$ then we replace every occurrence of x in t by $\neg\neg x$.*
- *If $t : \Omega \times \sigma\ [\Gamma]$ then t translates into $\langle \neg\neg\pi_1 t, \pi_2 t\rangle$; if $t : \sigma \times \Omega\ [\Gamma]$ then t translates into $\langle \pi_1 t, \neg\neg\pi_2 t\rangle$.*
- *If $t : \sigma\ [\Gamma, x : \Omega \times \sigma, \Gamma']$ then we replace every occurrence of x in t by $\langle \neg\neg\pi_1 x, \pi_2 x\rangle$; if $t : \sigma\ [\Gamma, x : \sigma \times \Omega, \Gamma']$ then we replace every occurrence of x in t by $\langle \pi_1 x, \neg\neg\pi_2 x\rangle$.*
- *If $t : \sigma \to \Omega\ [\Gamma]$ then t translates into $\lambda x : \sigma.\neg\neg tx$; if $t : \Omega \to \sigma\ [\Gamma]$ then t translates into $\lambda x : \Omega.t\neg\neg x$.*
- *If $t : \sigma\ [\Gamma, x : \sigma \to \Omega, \Gamma']$ then we replace every occurrence of x in t by $\lambda y : \sigma.(\neg\neg x)y$; if $t : \sigma\ [\Gamma, x : \Omega \to \sigma, \Gamma']$ then we replace every occurrence of x in t by $\lambda y : \Omega.x\neg\neg y$.*
- *Finally, if $t : \sigma\ [\Gamma]$ and no Ω appears in it, then t translates into itself.*

The double negation category $\mathbf{C}_{\neg\neg}$ is then defined as follows: the objects of $\mathbf{C}_{\neg\neg}$ are contexts in HoFL_P up to α-equivalence (which are essentially the same as objects in $\mathbf{C}$), and the arrows of $\mathbf{C}_{\neg\neg}$ are the translations of lists of terms in HoFL_P up to equality on terms, with their composition defined via substitution as usual. This intuitively means that those arrows in $\mathbf{C}$ that involve Ω are double negated in $\mathbf{C}_{\neg\neg}$ whilst the other part of $\mathbf{C}_{\neg\neg}$ remains the same as that of $\mathbf{C}$ (to give the rigorous definition of this, we work within the internal language). Then it is not obvious that $\mathbf{C}_{\neg\neg}$ forms a category again, let alone a CCC. Thus:

Lemma 11. $\mathbf{C}_{\neg\neg}$ *defined above forms a category, in particular a CCC.*

Proof. Since everything involving Ω is doubly negated, we have to verify that all of the relevant categorical structures, that is, composition, identity, projection, paring, evaluation, and transpose, preserve or respect double negation. Here we just give several sample proofs to show essential ideas.

Consider the case of composition. We think of single terms for simplicity. The composition of arrows $t : \sigma\ [x : \Omega]$ and $s : \sigma'\ [y : \sigma]$ in $\mathbf{C}_{\neg\neg}$ (which may be seen as $t : \Omega \to \sigma$ and $s : \sigma \to \sigma'$ in terms of the original category $\mathbf{C}$) is defined as $s[t/y] : \sigma'\ [x : \Omega]$, where every occurrence of x in $s[t/y]$ must have been replaced by $\neg\neg x$ (for $s[t/y]$ to be in $\mathbf{C}_{\neg\neg}$); this is true because every occurrence of x in t is replaced by $\neg\neg x$ by the definition of arrows in $\mathbf{C}_{\neg\neg}$. Likewise, the composition of arrows $t : \sigma'\ [x : \sigma]$ and $s : \Omega\ [y : \sigma']$ in $\mathbf{C}_{\neg\neg}$ is defined as $s[t/y] : \Omega\ [x : \sigma]$, where every sub-formula of $s[t/y]$ is doubly negated by the assumption of $s, t \in \mathbf{C}$; and hence $s[t/y] \in \mathbf{C}$. More complex cases can be proven in a similar way.

Consider the case of identity. Think of an identity on Ω, which is given by $\neg\neg x : \Omega\ [x : \Omega]$. Given $t : \Omega\ [y : \sigma']$ in $\mathbf{C}_{\neg\neg}$, $(\neg\neg x) \circ t$ is defined as $(\neg\neg x)[t/x] : \Omega\ [y : \sigma']$, which equals $\neg\neg t : \Omega\ [y : \sigma']$. By $t \in \mathbf{C}_{\neg\neg}$, t can be written as $\neg\neg t'$, and so $\neg\neg t = \neg\neg\neg\neg t' = \neg\neg t' = t$. Hence $(\neg\neg x) \circ t = t$. Likewise, given $t : \sigma'\ [y : \Omega]$ in $\mathbf{C}_{\neg\neg}$, $t \circ \neg\neg x$ is defined as $t[\neg\neg x/y] : \sigma'\ [x : \Omega]$; since every occurrence of y in t is replaced by $\neg\neg y$ because $t \in \mathbf{C}_{\neg\neg}$ and since $\neg\neg\neg\neg$ is equivalent to $\neg\neg$, we have $t[\neg\neg x/y] = t$, whence $t \circ \neg\neg x = t$. More complex cases can be shown in a similar manner.

To show the existence of finite products and exponentials involving Ω (otherwise it is trivial), it is crucial to check that doubly negated projection, pairing, evaluation, and transpose still play their own rôles, just as doubly negated identity still plays the rôle of identity as we have shown above. □

Finally we obtain the following, tripos-theoretical Kolmogorov translation theorem for higher-order logic, which may also be seen as a translation from classical set theory to intuitionistic set theory (since higher-order logic is basically set theory in logical form).

Theorem 12. *Let $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{HA}$ be an* IL *tripos, and $\mathbf{C}_{\neg\neg}$ the double negation category as defined above. Then, $P_{\neg\neg}$ defined as*

$$\mathrm{Hom}_{\mathbf{C}_{\neg\neg}}(\text{-}, \Omega) : \mathbf{C}_{\neg\neg} \to \mathbf{BA}$$

forms a CL *tripos, called the double negation tripos of P.*

Proof. $\mathbf{C}_{\neg\neg}$ is a CCC by the lemma, and $P_{\neg\neg}$ is represented by Ω. This completes the higher-order part of the proof. Concerning the first-order part, the existence of quantifiers follows from this fact: if φ admits the double negation elimination, then $\neg\neg\forall x\varphi$ and $\neg\neg\exists x\varphi$ are equivalent to $\forall x\neg\neg\varphi$ and $\exists x\neg\neg\varphi$, respectively. □

Note that the hyperdoctrinal Kolmogorov translation does not reduce to the construction of toposes via double negation topology because there are more triposes than toposes in the adjunction between them (all toposes come from triposes, but not *vice versa*). Moreover, our hyperdoctrinal method is designed modularly enough to be applicable to Girard's translation as well as Kolmogorov's. Although Glivenko-type theorems have been shown for substructural propositional and first-order logics (see Ferreira-Ono [6] and Galatos-Ono

[9]), no such result is known for higher-order logic (as to the first-order case, [21] is typed and categorical while [6] is single-sorted and proof-theoretical).

An exponential ! on an FL algebra A is defined as a unary operation satisfying: (i) $a \leq b$ implies $!a \leq !b$; (ii) $!!a = !a \leq a$; (iii) $!\top = 1$; (iv) $!a \otimes !b = !(a \wedge b)$ (Coumans, Gehrke, and van Rooijen [5]). We denote by $\mathbf{FL}_c^!$ the category of commutative FL algebras with !, which are algebras for intuitionistic linear logic. $\mathrm{FL}_c^!$ triposes give sound and complete semantics for higher-order intuitionistic linear logic. The Girard category $\mathbf{C}_!$ of an $\mathrm{FL}_c^!$ tripos $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{FL}_c^!$ is defined by replacing double negation in the above definition of $\mathbf{C}_{\neg\neg}$ with Girard's exponential !. The following is the hyperdoctrinal Girard translation theorem for higher-order logic, which can be shown in basically the same way as above; no such higher-order translation has been known so far.

Theorem 13. *Let $P : \mathbf{C}^{\mathrm{op}} \to \mathbf{FL}_c^!$ be an $\mathrm{FL}_c^!$ tripos (for intuitionistic linear logic), and $\mathbf{C}_!$ the Girard category of P. Define*

$$P_! = \mathrm{Hom}_{\mathbf{C}_!}(\text{-}, \Omega) : \mathbf{C}_! \to \mathbf{HA}.$$

Then, $P_!$ forms an IL *tripos (i.e., $\mathrm{FL}_{ecw}^!$ tripos), called the Girard tripos of P.*

Acknowledgements. The author would like to thank the reviewers of the paper for their numerous, substantial comments and suggestions for improvement. The author hereby acknowledges that this work was supported by JST PRESTO (grant code: JPMJPR17G9), JSPS Kakenhi (grant code: 17K14231), and the JSPS Core-to-Core Program "Mathematical Logic and its Applications".

References

1. Adámek, J., Herrlich, H., Strecker, G.E.: Abstract and Concrete Categories. Wiley, Hoboken (1990)
2. Biering, B., Birkedal, L., Torp-Smith, N.: BI-hyperdoctrines, higher-order separation logic, and abstraction. ACM TOPLAS **29**(5), 24 (2007)
3. Ciabattoni, A., Galatos, N., Terui, K.: Algebraic proof theory for substructural logics. Ann. Pure Appl. Logic **163**, 266–290 (2012)
4. Coumans, D.: Canonical extensions in logic - some applications and a generalisation to categories. Ph.D. thesis, Radboud Universiteit Nijmegen (2012)
5. Coumans, D., Gehrke, M., van Rooijen, L.: Relational semantics for full linear logic. J. Appl. Logic **12**, 50–66 (2014)
6. Farahani, H., Ono, H.: Glivenko theorems and negative translations in substructural predicate logics. Arch. Math. Logic **51**, 695–707 (2012)
7. Ferreira, G., Oliva, P.: On the relation between various negative translations. Logic Constr. Comput. **3**, 227–258 (2012)
8. Galatos, N., Jipsen, P., Kowalski, T., Ono, H.: Residuated Lattices: An Algebraic Glimpse at Substructural Logics. Elsevier, Amsterdam (2007)
9. Galatos, N., Ono, H.: Glivenko theorems for substructural logics over FL. J. Symb. Logic **71**, 1353–1384 (2016)
10. Frey, J.: A 2-categorical analysis of the tripos-to-topos construction arXiv:1104.2776

11. Henkin, L., Monk, J.D., Tarski, A.: Cylindric Algebras. North-Holland, Amsterdam (1971)
12. Hyland, M., Johnstone, P.T., Pitts, A.: Tripos theory. Math. Proc. Cambridge Philos. Soc. **88**, 205–232 (1980)
13. Jacobs, B.: Categorical Logic and Type Theory. Elsevier, Amsterdam (1999)
14. Johnstone, P.T.: Stone Spaces. CUP, Cambridge (1982)
15. Johnstone, P.T.: Sketches of an Elephant. OUP, Oxford (2002)
16. Kupke, C., Kurz, A., Venema, Y.: Stone coalgebras. Theoret. Comput. Sci. **327**, 109–134 (2004)
17. Lambek, J., Scott, P.J.: Introduction to Higher-Order Categorical Logic (1986)
18. Lawvere, F.W.: Adjointness in foundations. Dialectica **23**, 281–296 (1969). Reprinted with the author's retrospective commentary. In: Theory and Applications of Categories, vol. 16, pp. 1–16 (2006)
19. Marquis, J.-P., Reyes, G.: The history of categorical logic: 1963–1977. In: Handbook of the History of Logic, vol. 6, pp. 689–800. Elsevier (2011)
20. Maruyama, Y.: Natural duality, modality, and coalgebra. J. Pure Appl. Algebra **216**, 565–580 (2012)
21. Maruyama, Y.: Full lambek hyperdoctrine: categorical semantics for first-order substructural logics. In: Libkin, L., Kohlenbach, U., de Queiroz, R. (eds.) WoLLIC 2013. LNCS, vol. 8071, pp. 211–225. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39992-3_19
22. Maruyama, Y.: Duality theory and categorical universal logic. EPTCS **171**, 100–112 (2014)
23. Ono, H.: Algebraic semantics for predicate logics and their completeness. RIMS Kokyuroku **927**, 88–103 (1995)
24. Ono, H.: Crawley completions of residuated lattices and algebraic completeness of substructural predicate logics. Stud. Logica **100**, 339–359 (2012)
25. Pitts, A.: Categorical logic, Chap. 2. In: Handbook of Logic in Computer Science, vol. 5. OUP (2000)
26. Pitts, A.: Tripos theory in retrospect. Math. Struct. Comput. Sci. **12**, 265–279 (2002)