

# Modal type theory as based on the intuitionistic modal logic $IEL^-$

Daniel Rogozin<sup>1</sup>

<sup>1</sup>Moscow State University, Moscow, Russia

<sup>1</sup>Serokell OÜ, Tallinn, Estonia

## Abstract

Modal intuitionistic epistemic logic  $IEL^-$  was proposed by S. Artemov and T. Protopopescu as the formal foundation for the intuitionistic theory of knowledge. We construct a modal simply typed lambda-calculus which is Curry-Howard isomorphic to  $IEL^-$  as the formal theory of calculations with applicative functors in functional programming.

## 1 Introduction

Modal intuitionistic epistemic logic  $IEL$  was proposed by S. Artemov and T. Protopopescu [3].  $IEL$  provides the epistemology and the theory of knowledge as based on BHK-semantics of intuitionistic logic.  $IEL^-$  is a variant of  $IEL$ , that corresponds to intuitionistic belief. Informally,  $\Box A$  denotes that the agent believes in  $A$ .

Intuitionistic epistemic logic  $IEL^-$  is defined by following axioms and derivation rules:

**Definition 1.** *Intuitionistic epistemic logic  $IEL$ :*

- 1) *IPC axioms;*
  - 2)  $\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$  (*normality*);
  - 3)  $A \rightarrow \Box A$  (*co-reflection*);
- Rule: MP.*

V. Krupski and A. Yatmanov provided the sequential calculus for  $IEL$  and proved that this calculus is PSPACE-complete [14].

Functional programming languages such as Haskell, Idris or Purescript have special type classes <sup>1</sup> for calculations with type operators like `Functor` and `Applicative` <sup>2</sup>:

```
class Functor f where
  fmap :: (a -> b) -> f a -> f b

class Functor f => Applicative f where
  pure :: a -> f a
  (<*>) :: f (a -> b) -> f a -> f b
```

---

<sup>1</sup>Type class in Haskell is a general interface for special group of datatypes.

<sup>2</sup>Reader may read more about these types in the Haskell standard library documentation.

By *container* (or *computational context*) type we mean some type operator  $f$ , where  $f$  is a “function” from  $*$  to  $*$ : type operator takes a simple type (kind  $*$ ) and returns another simple type of kind  $*$ . For more detailed description of the type system with kinds used in Haskell see [24].

Applicative functor allows to generalize the action of a functor for functions with arbitrary number of arguments, for instance:

```
liftA2 :: Applicative f => (a -> b -> c) -> f a -> f b -> f c
liftA2 f x y = ((pure f) <*> x) <*> y
```

It's not difficult to see that modal axioms in  $IEL^-$  and types of the methods of Applicative class in Haskell-like languages (which is described below) are syntactically similar and we are going to show that this coincidence has a non-trivial computational meaning.

We investigate the relationship between intuitionistic epistemic logic  $IEL^-$  and applicative programming with side-effects by constructing the type system (which is called  $\lambda_{IEL^-}$ ) which is Curry-Howard isomorphic to  $IEL^-$ . So we will consider  $IEL^-$  modality as an arbitrary applicative functor and we prove that the obtained type system is sound and complete for an applicative functor on cartesian closed category (using the categorical definition proposed by Paterson [21]).

$\lambda_{IEL^-}$  consists of the rules for simply typed lambda-calculus and special typing rules for lifting types into the applicative functor  $\Box$ . We assume that our type system will axiomatize the simplest case of computation with effects with one container. We provide a proof-theoretical view at this kind of computations in functional programming and prove strong normalization and confluence.

## 2 Typed lambda-calculus based on $IEL^-$

The first is to define the natural deduction calculus for  $NIEL^-$ :

**Definition 2.** *Natural deduction  $NIEL^-$  for  $IEL^-$  is an extension of intuitionistic natural deduction calculus with additional inference rules for modality:*

$$\begin{array}{c}
\frac{}{\Gamma, A \vdash A} \mathbf{ax} \\
\\
\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \rightarrow_I \qquad \frac{\Gamma \vdash A \rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B} \rightarrow_E \\
\\
\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \wedge_I \qquad \frac{\Gamma \vdash A_1 \wedge A_2}{\Gamma \vdash A_i} \wedge_E, i = 1, 2 \\
\\
\frac{\Gamma \vdash A}{\Gamma \vdash \Box A} \Box_{I1} \qquad \frac{\Gamma \vdash \Box A_1, \dots, \Gamma \vdash \Box A_n \quad A_1, \dots, A_n \vdash B}{\Gamma \vdash \Box B} \Box_{I2}
\end{array}$$

Without loss of generality we will consider the calculus without inference rule for disjunction and negation. Obtained calculus is equivalent to Hilbert-style  $IEL^-$  without axioms for negation and disjunction.

The first modal rule allows to derive co-reflexion. The second modal rule is a counterpart of  $\Box_I$  rule in natural deduction calculus for constructive K (see [13]).

We will denote  $\Gamma \vdash \Box A_1, \dots, \Gamma \vdash \Box A_n$  and  $A_1, \dots, A_n \vdash B$  as  $\Gamma \vdash \Box \vec{A}$  and  $\vec{A} \vdash B$  for brevity.

**Lemma 1.**  $\Gamma \vdash_{NIEL^-} A \Rightarrow IEL^- \vdash \bigwedge \Gamma \rightarrow A$ .

*Proof.* Induction on the derivation.

Let us consider cases with modality.

- 1) If  $\Gamma \vdash_{NIEL^-} A$ , then  $IEL^- \vdash \bigwedge \Gamma \rightarrow \Box A$ .
  - (1)  $\bigwedge \Gamma \rightarrow A$  assumption
  - (2)  $A \rightarrow \Box A$  co-reflection
  - (3)  $(\bigwedge \Gamma \rightarrow A) \rightarrow ((A \rightarrow \Box A) \rightarrow (\bigwedge \Gamma \rightarrow \Box A))$  IPC theorem
  - (4)  $(A \rightarrow \Box A) \rightarrow (\bigwedge \Gamma \rightarrow \Box A)$  from (1), (3) and MP
  - (5)  $\bigwedge \Gamma \rightarrow \Box A$  from (2), (4) and MP
- 2) If  $\Gamma \vdash_{NIEL^-} \Box \vec{A}$  and  $\vec{A} \vdash B$ , then  $IEL^- \vdash \bigwedge \Gamma \rightarrow \Box B$ .
  - (1)  $\bigwedge \Gamma \rightarrow \Box A_1, \dots, \bigwedge \Gamma \rightarrow \Box A_n$  assumption
  - (2)  $\bigwedge \Gamma \rightarrow \bigwedge_{i=1}^n \Box A_i$  IEL<sup>-</sup> theorem
  - (3)  $\bigwedge_{i=1}^n \Box A_i \rightarrow \Box \bigwedge_{i=1}^n A_i$  IEL<sup>-</sup> theorem
  - (4)  $\bigwedge \Gamma \rightarrow \Box \bigwedge_{i=1}^n A_i$  from (2), (3) and transitivity
  - (5)  $\bigwedge_{i=1}^n A_i \rightarrow B$  assumption
  - (6)  $(\bigwedge_{i=1}^n A_i \rightarrow B) \rightarrow \Box(\bigwedge_{i=1}^n A_i \rightarrow B)$  co-reflection
  - (7)  $\Box(\bigwedge_{i=1}^n A_i \rightarrow B)$  from (5), (6) and MP
  - (8)  $\Box \bigwedge_{i=1}^n A_i \rightarrow \Box B$  from (7) and normality
  - (9)  $\bigwedge \Gamma \rightarrow \Box B$  from (4), (8) and transitivity

□

**Lemma 2.** *If  $IEL^- \vdash A$ , then  $NIEL^- \vdash A$ .*

*Proof.* Straightforward derivation of modal axioms in  $NIEL^-$ . We shall consider these derivations using terms below. □

At the next step we build the typed lambda-calculus based on the  $NIEL^-$ . by proof-assigning in rules.

At first, we define lambda-terms and types for this lambda-calculus.

**Definition 3.** *The set of terms:*

*Let  $\mathbb{V}$  be the set of variables. The set  $\Lambda_\Box$  of terms is defined by the grammar:*

$$\Lambda_\Box ::= \mathbb{V} \mid (\lambda \mathbb{V}. \Lambda_\Box) \mid (\Lambda_\Box \Lambda_\Box) \mid \langle \Lambda_\Box, \Lambda_\Box \rangle \mid (\pi_1 \Lambda_\Box) \mid (\pi_2 \Lambda_\Box) \mid (\mathbf{box} \Lambda_\Box) \mid (\mathbf{let} \mathbf{box} \mathbb{V}^* = \Lambda_\Box^* \mathbf{in} \Lambda_\Box)$$

Where  $\mathbb{V}^*$  and  $\Lambda_{\square}^*$  denote the set of finite sequences of variables  $\bigcup_{i=0}^{\infty} \mathbb{V}^i$  and the set of finite sequences of terms  $\bigcup_{i=0}^{\infty} \Lambda_{\square}^i$ . Note that the sequence of variables  $\vec{x}$  and the sequence of terms  $\vec{M}$  should have the same length. Otherwise, term is not well-formed.

**Definition 4.** *The set of types:*

*Let  $\mathbb{T}$  be the set of atomic types. The set  $\mathbb{T}_{\square}$  of types with applicative functor  $\square$  is generated by the grammar:*

$$\mathbb{T}_{\square} ::= \mathbb{T} \mid (\mathbb{T}_{\square} \rightarrow \mathbb{T}_{\square}) \mid (\mathbb{T}_{\square} \times \mathbb{T}_{\square}) \mid (\square \mathbb{T}_{\square}) \quad (1)$$

Context, domain of context and range of context are defined standardly [20][24].

Our typing system is based on the Curry-style typing rules:

**Definition 5.** *Modal typed lambda calculus  $\lambda_{IEL^-}$  based on  $NIEL_{\rightarrow}^-$ :*

$$\begin{array}{c} \frac{}{\Gamma, x : A \vdash x : A} \text{ax} \\[10pt] \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x. M : A \rightarrow B} \rightarrow_i \qquad \frac{\Gamma \vdash M : A \rightarrow B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B} \rightarrow_e \\[10pt] \frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash \langle M, N \rangle : A \times B} \times_i \qquad \frac{\Gamma \vdash M : A_1 \times A_2}{\Gamma \vdash \pi_i M : A_i} \times_e, i = 1, 2 \\[10pt] \frac{\Gamma \vdash M : A}{\Gamma \vdash \mathbf{box} M : \square A} \square_I \qquad \frac{\Gamma \vdash \vec{M} : \square \vec{A} \quad \vec{x} : \vec{A} \vdash N : B}{\Gamma \vdash \mathbf{let box} \vec{x} = \vec{M} \mathbf{in} N : \square B} \text{let}_{\square} \end{array}$$

$\square_I$ -typing rule is the same as  $\bigcirc$ -introduction in monadic metalanguage [22].  $\square_I$  allows to inject an object of type  $A$  into the functor  $\square$ .  $\square_I$  reflects the Haskell method **box** for Applicative class. It plays the same role as the **return** method in Monad class.

$\text{let}_{\square}$  is similar to the  $\square$ -rule in typed lambda calculus for intuitionistic normal modal logic **IK**, which is described in [11].

$\Gamma \vdash \vec{M} : \square \vec{A}$  is a syntax sugar for the sequence  $\Gamma \vdash M_1 : \square A_1, \dots, \Gamma \vdash M_n : \square A_n$  and  $\vec{x} : \vec{A} \vdash N : B$  is a short form for  $x_1 : A_1, \dots, x_n : A_n \vdash N : B$ . **let box**  $\vec{x} = \vec{M}$  **in**  $N$  is a simultaneous local binding in  $N$ . We use this short form instead of **let box**  $x_1, \dots, x_n = M_1, \dots, M_n$  **in**  $N$ .

In fact, our calculus is the extension of typed lambda calculus for **IK** by  $\square_I$ -rule that is appropriate to co-reflection.

Here are some examples of derivation trees:

$$\frac{\frac{x : A \vdash x : A}{x : A \vdash \mathbf{box} x : \square A} \square_I}{\vdash (\lambda x. \mathbf{box} x) : A \rightarrow \square A} \rightarrow_I$$

$$\begin{array}{c}
\frac{f : \Box(A \rightarrow B) \vdash f : \Box(A \rightarrow B) \quad x : \Box A \vdash x : \Box A \quad \frac{g : A \rightarrow B \vdash g : A \rightarrow B \quad y : A \vdash y : A}{g : A \rightarrow B, y : A \vdash gy : B} \rightarrow_e}{\frac{f : \Box(A \rightarrow B), x : \Box A \vdash \mathbf{let\ box\ } g, y = f, x \mathbf{ in\ } gy : \Box B}{f : \Box(A \rightarrow B) \vdash \lambda x. \mathbf{let\ box\ } g, y = f, x \mathbf{ in\ } gy : \Box A \rightarrow \Box B} \rightarrow_I} \mathbf{let}_{\Box} \\
\frac{f : \Box(A \rightarrow B) \vdash \lambda x. \mathbf{let\ box\ } g, y = f, x \mathbf{ in\ } gy : \Box A \rightarrow \Box B}{\vdash \lambda f. \lambda x. \mathbf{let\ box\ } g, y = f, x \mathbf{ in\ } gy : \Box(A \rightarrow B) \rightarrow \Box A \rightarrow \Box B} \rightarrow_I
\end{array}$$

Now we define free variables and substitutions:

**Definition 6.** *The set  $FV(M)$  of free variables for a term  $M$ :*

1.  $FV(x) = \{x\}$ ;
2.  $FV(\lambda x.M) = FV(M) \setminus \{x\}$ ;
3.  $FV(MN) = FV(M) \cup FV(N)$ ;
4.  $FV(\langle M, N \rangle) = FV(M) \cup FV(N)$ ;
5.  $FV(\pi_i M) = FV(M)$ ,  $i = 1, 2$ ;
6.  $FV(\mathbf{box\ } M) = FV(M)$ ;
7.  $FV(\mathbf{let\ box\ } \vec{x} = \vec{M} \mathbf{ in\ } N) = \bigcup_{i=1}^n FV(M_i)$ , where  $n = |\vec{M}|$ .

**Definition 7.** *Substitution:*

1.  $x[x := N] = N$ ,  $x[y := N] = x$ ;
2.  $(MN)[x := N] = M[x := N]N[x := N]$ ;
3.  $(\lambda x.M)[x := N] = \lambda x.M[y := N]$ ,  $y \in FV(M)$ ;
4.  $(M, N)[x := P] = (M[x := P], N[x := P])$ ;
5.  $(\pi_i M)[x := P] = \pi_i(M[x := P])$ ,  $i = 1, 2$ ;
6.  $(\mathbf{box\ } M)[x := P] = \mathbf{box\ } (M[x := P])$ ;
7.  $(\mathbf{let\ box\ } \vec{x} = \vec{M} \mathbf{ in\ } N)[y := P] = \mathbf{let\ box\ } \vec{x} = (\vec{M}[y := P]) \mathbf{ in\ } N$ .

Substitution and free variable for terms of kind **let box** are defined similary to [11].

**Definition 8.**  *$\beta$ -reduction and  $\eta$ -reduction rules for  $\lambda_{IEL^-}$ .*

1.  $(\lambda x.M)N \rightarrow_{\beta} M[x := N]$ ;
2.  $\pi_1 \langle M, N \rangle \rightarrow_{\beta} M$ ;
3.  $\pi_2 \langle M, N \rangle \rightarrow_{\beta} N$ ;
4.  $\mathbf{let\ box\ } \vec{x}, y, \vec{z} = \vec{M}, \mathbf{let\ box\ } \vec{w} = \vec{N} \mathbf{ in\ } Q, \vec{P} \mathbf{ in\ } R \rightarrow_{\beta} \mathbf{let\ box\ } \vec{x}, \vec{w}, \vec{z} = \vec{M}, \vec{N}, \vec{P} \mathbf{ in\ } R[y := Q]$ ;

5.  $\text{let } \mathbf{box } \vec{x} = \mathbf{box } \vec{M} \text{ in } N \rightarrow_\beta \mathbf{box } N[\vec{x} := \vec{M}];$
6.  $\text{let } \mathbf{box } \_ = \_ \text{ in } M \rightarrow_\beta \mathbf{box } M$ , where  $\_$  is an empty sequence of terms;
7.  $\lambda x. fx \rightarrow_\eta f;$
8.  $\langle \pi_1 P, \pi_2 P \rangle \rightarrow_\eta P;$
9.  $\text{let } \mathbf{box } x = M \text{ in } x \rightarrow_\eta M;$

By default we use call-by-name evaluation strategy.

Now we will prove standard lemmas for contexts in type systems<sup>3</sup>:

**Lemma 3.** *Generation for  $\Box_I$ .*

Let  $\Gamma \vdash \mathbf{box } M : \Box A$ , then  $\Gamma \vdash M : A$ ;

*Proof.* Straightforwardly. □

**Lemma 4.** *Basic lemmas.*

1. If  $\Gamma \vdash M : A$  and  $\Gamma \subseteq \Delta$ , then  $\Delta \vdash M : A$ ;
2. If  $\Gamma \vdash M : A$ , then  $\Delta \vdash M : A$ , where  $\Delta = \{x : A \mid (x : A) \in \Gamma \ \& \ x \in FV(M)\}$
3. If  $\Gamma, x : A \vdash M : B$  and  $\Gamma \vdash N : A$ , then  $\Gamma \vdash M[x := N] : B$ .

*Proof.*

- 1–3) Induction on the derivation of  $\Gamma \vdash M : A$ .
- 

**Theorem 1.** *Subject reduction*

If  $\Gamma \vdash M : A$  and  $M \rightarrow_r N$ , then  $\Gamma \vdash N : A$ .

*Proof.* Induction on the derivation  $\Gamma \vdash M : A$  and on the generation of  $\rightarrow_r$ .

1. If  $\Gamma \vdash \text{let } \mathbf{box } \vec{x}, y, \vec{z} = \vec{M}, \text{let } \mathbf{box } \vec{w} = \vec{N} \text{ in } Q, \vec{P} \text{ in } R : \Box B$ , then  $\Gamma \vdash \text{let } \mathbf{box } \vec{x}, \vec{w}, \vec{z} = \vec{M}, \vec{N}, \vec{P} \text{ in } R[y := Q] : \Box B$  by rule 4).
2. Let  $\Gamma \vdash \text{let } \mathbf{box } x = M \text{ in } x : \Box A$ , then  $\Gamma \vdash M : \Box A$  by rule 9). See [11].
3. The derivation ends in

$$\frac{\Gamma \vdash \mathbf{box } \vec{M} : \Box \vec{A} \quad \vec{x} : \vec{A} \vdash N : B}{\Gamma \vdash \text{let } \mathbf{box } \vec{x} = \mathbf{box } \vec{M} \text{ in } N : \Box B}$$

So  $\Gamma \vdash \vec{M} : \vec{A}$  by Lemma 4 and  $\Gamma \vdash N[\vec{x} := \vec{M}] : B$  by Lemma 4, part 3. Then we can transform this into the following derivation:

$$\frac{\Gamma \vdash N[\vec{x} := \vec{M}] : B}{\Gamma \vdash \mathbf{box } N[\vec{x} := \vec{M}] : \Box B} \Box_I$$

<sup>3</sup>Cases with  $\rightarrow$ -constructor are proven standardly in similar lemmas for simply typed lambda calculus, for example see [20] [24] [23]. We will consider only modal cases

4. The derivation ends in

$$\frac{\vdash M : A}{\vdash \mathbf{let\ box\ } \_ = \_ \mathbf{in\ } M : \Box A}$$

So, if  $\vdash M : A$ , then  $\vdash \mathbf{box\ } M : \Box A$ .

□

**Theorem 2.**

$\rightarrow_r$  is strongly normalizing;

*Proof.*

Let us define the transformation from  $\lambda_{\text{IEL-}}$  into the simple type theory with  $\rightarrow$ ,  $\times$  and natural number type  $\mathbb{N}$  with additional typing and reduction rules<sup>4</sup>:

$$\begin{array}{c} \frac{}{\Gamma \vdash 0 : \mathbb{N}} \mathbf{zero} \\[10pt] \frac{\Gamma \vdash n : \mathbb{N}}{\Gamma \vdash \mathbf{succ\ } n : \mathbb{N}} \mathbf{succ} \qquad \frac{\Gamma \vdash n : \mathbb{N} \quad \Gamma \vdash m : \mathbb{N}}{\Gamma \vdash n + m : \mathbb{N}} \mathbf{add} \end{array}$$

1.  $n + 0 \rightarrow_{\beta} n$ ;
2.  $(n + \mathbf{succ\ } m) \rightarrow_{\beta} \mathbf{succ\ } (n + m)$

The transformation  $|\cdot|$  is defined as follows:

**Definition 9.** *Interpretation of types:*

1.  $A \in \mathbb{T} \Rightarrow |A| = A$ ;
2.  $|A \rightarrow B| = |A| \rightarrow |B|$ ;
3.  $|A \times B| = |A| \times |B|$ ;
4.  $|\Box A| = \mathbb{N} \times |A|$ .

**Definition 10.** *Interpretation of terms:*

1.  $x \in \mathbb{V} \Rightarrow |x| = x$ ;
2.  $|\lambda x.M| = \lambda x.|M|$ ;
3.  $|(MN)| = |M||N|$ ;
4.  $|\langle M, N \rangle| = \langle |M|, |N| \rangle$ ;
5.  $|\pi_i M| = \pi_i |M|$ ,  $i = 1, 2$ ;
6.  $|\mathbf{box\ } M| = \langle 0, |M| \rangle$ ;
7.  $|\mathbf{let\ box\ } \_ = \_ \mathbf{in\ } M| = \langle 0, |M| \rangle$ ;
8.  $|\mathbf{let\ box\ } x = N \mathbf{in\ } M| = \langle \pi_1 |N|, |M| [x := \pi_2 |N|] \rangle$

---

<sup>4</sup>Strong normalization for stronger system was shown here [9]

$$9. |\mathbf{let\ box\ } \vec{x} = \vec{N} \mathbf{ in\ } M| = \langle \sum_{i=1}^n \pi_1 |N|, |M|[\vec{x} := \pi_2 \vec{N}] \rangle$$

Let us consider the interpretation for  $\mathbf{let}_{\square}$  rule:

$$\frac{|\Gamma \vdash \vec{N} : \square \vec{A}| = |\Gamma| \vdash |\vec{N}| : \mathbb{N} \times |\vec{A}| \quad |\vec{x} : \vec{A} \vdash M : B| = \vec{x} : |\vec{A}| \vdash |M| : |B|}{|\Gamma \vdash \mathbf{let\ box\ } \vec{x} = \vec{N} \mathbf{ in\ } M : \square B| = |\Gamma| \vdash \langle \sum_{i=1}^n \pi_1 |N|, |M|[\vec{x} := \pi_2 \vec{N}] \rangle : \mathbb{N} \times |B|} \mathbf{let}_{\square}$$

**Lemma 5.** :

$|M[x := N]| = |M|[x := |N|]$  for any term  $M$ .

*Proof.* Induction on the structure of  $M$ . □

**Lemma 6.**  $M \rightarrow_r N \Rightarrow |M| \rightarrow_{\beta\eta} |N|$

*Proof.* Let us consider cases with  $\beta_{\square}$ ,  $\beta_{\square\mathbf{box}}$  and  $\square id$ .

$$\begin{aligned} 1. |\mathbf{let\ box\ } x = (\mathbf{let\ box\ } y = N \mathbf{ in\ } P) \mathbf{ in\ } M| &= \\ |\mathbf{let\ box\ } y = N \mathbf{ in\ } M[x := P]| &= \\ |\mathbf{let\ box\ } x = (\mathbf{let\ box\ } y = N \mathbf{ in\ } P) \mathbf{ in\ } M| &= \\ \text{Interpretation} & \\ \langle \pi_1 |N|, |M|[x := |P|[y := \pi_2 |N|]] \rangle &= \\ |\mathbf{let\ box\ } y = N \mathbf{ in\ } M[x := P]| &= \\ \langle \pi_1 |N|, |M|[x := |P|][y := \pi_2 |N|] \rangle &\equiv \\ \text{So far as } y \notin FV(M) & \\ \langle \pi_1 |N|, |M|[x := |P|[y := \pi_2 |N|]] \rangle & \end{aligned}$$

$$2. |\mathbf{let\ box\ } \vec{x} = \mathbf{box\ } \vec{N} \mathbf{ in\ } M| = |\mathbf{box\ } M[\vec{x} := \vec{N}]|$$

$$\begin{aligned} |\mathbf{let\ box\ } \vec{x} = \mathbf{box\ } \vec{N} \mathbf{ in\ } M| &= \\ \text{Interpretation} & \\ \langle 0 + \dots + 0, |M|[\vec{x} := |\vec{N}|] \rangle &\rightarrow_{\beta} \\ \text{Multistep reduction for natural numbers} & \\ \langle 0, |M|[\vec{x} := |\vec{N}|] \rangle &= \\ \text{Interpretation} & \\ |\mathbf{box\ } M[\vec{x} := \vec{N}]] & \end{aligned}$$

$$3. |\mathbf{let\ box\ } x = M \mathbf{ in\ } x| = |M|$$

$$\begin{aligned} |\mathbf{let\ box\ } x = M \mathbf{ in\ } x| &= \\ \text{Interpretation} & \\ \langle \pi_1 |M|, x[x := \pi_2 |M|] \rangle &= \\ \text{Substitution} & \\ \langle \pi_1 |M|, \pi_2 |M| \rangle &\rightarrow_{\eta} \\ \eta\text{-reduction for pairs} & \\ |M| & \end{aligned}$$

$$4. |\mathbf{let\ box\ } \_ = \_ \mathbf{ in\ } M| = \langle 0, |M| \rangle = |\mathbf{box\ } M|$$



Hence  $\lambda_{\text{IEL-}}$  sounds for  $\lambda_{\rightarrow, \times, \mathbb{N}}$ , then multistep reduction in  $\lambda_{\text{IEL-}}$  is strongly normalizing, so far as multistep reduction in  $\lambda_{\rightarrow, \times, \mathbb{N}}$  is strongly normalizing.  $\square$

$\square$

**Theorem 3.**

$\rightarrow_r$  is confluent.

*Proof.*

By Newman's lemma [24], if relation is strongly normalizing and locally confluent, then this relation is confluent.

It is sufficient to show that  $\rightarrow_r$  is locally confluent.

**Lemma 7. Local confluence**

If  $M \rightarrow_r N$  and  $M \rightarrow_r Q$ , then there exists some term  $P$ , such that  $N \rightarrow_r P$  and  $Q \rightarrow_r P$ .

*Proof.* Let us consider the following critical pairs and show that they are joinable:

1.

$$\begin{array}{ccc}
 \text{let box } x = (\text{let box } \vec{y} = \text{box } \vec{N} \text{ in } P) \text{ in } M & & \\
 \downarrow \beta\Box & \searrow \beta\Box\text{box} & \\
 \text{let box } \vec{y} = \text{box } \vec{N} \text{ in } M[x := P] & & \text{let box } x = \text{box } P[\vec{y} := \vec{N}] \text{ in } M \\
 \\ 
 \text{let box } \vec{y} = \text{box } \vec{N} \text{ in } M[x := P] & \rightarrow_{\beta\Box\text{box}} & \text{box } M[x := P][\vec{y} := \vec{N}] \\
 \text{let box } x = \text{box } P[\vec{y} := \vec{N}] \text{ in } M & \rightarrow_{\beta\Box\text{box}} & \text{box } M[x := P[\vec{y} := \vec{N}]] \equiv \\
 \text{So far as } x \notin \vec{y} & & \text{box } M[x := P][\vec{y} := \vec{N}]
 \end{array}$$

2.

$$\begin{array}{ccc}
 \text{let box } x = (\text{let box } \_ = \_ \text{ in } N) \text{ in } M & & \\
 \downarrow \beta\Box & \searrow \beta\Box\_ & \\
 \text{let box } \_ = \_ \text{ in } M[x := N] & & \text{let box } x = \text{box } N \text{ in } M \\
 \\ 
 \text{let box } \_ = \_ \text{ in } M[x := N] & \rightarrow_{\beta\Box\_} & \text{box } (M[x := N]) \\
 \text{let box } x = \text{box } N \text{ in } M & \rightarrow_{\beta\Box\text{box}} & \text{box } (M[x := N])
 \end{array}$$

$\square$

Also we may consider four critical pairs which are considered in confluence proof for lambda-calculus as based on **IK** [11].  $\square$

**Theorem 4.**

*Normal form in call-by-name  $\lambda_{IEL-}$  has the subformula property: if  $M$  is in normal form, then its all subterms are in normal form too.*

*Proof.*

By induction on the structure of  $M$ . Case with **let**  $\mathbf{box} \vec{x} = \vec{M}$  **in**  $N$  was considered by Kakutani [11] [12].

If  $\mathbf{box} M$  is in normal form, so  $M$  is in normal form and its subterms are in normal form too by hypothesis.

Thus if  $\mathbf{box} M$  is in normal form, then all its subterms are in normal form too.  $\square$

### 3 Categorical semantics

**Definition 11.** *Monoidal category*

*Monoidal category is a category  $\mathcal{C}$  with the following additional structure:*

1. A bifunctor  $\otimes : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$  called *tensor product*;
2. Identity object  $\mathbf{1}$ ;
3. A natural isomorphism called *associator*:  $\alpha_{A,B,C} : (A \otimes B) \otimes C \cong A \otimes (B \otimes C)$ ;
4. A natural isomorphism called *left unitor*:  $L_A : \mathbf{1} \otimes A \cong A$ ;
5. A natural isomorphism called *right unitor*  $R_A : A \otimes \mathbf{1} \cong A$ ;
6. A coherence condition called *MacLane pentagon*, i.e. the following diagram commutes:

$$\begin{array}{ccc}
 & (A \otimes (B \otimes C)) \otimes D & \\
 \alpha_{A,B,C} \otimes id_D \nearrow & & \searrow \alpha_{A,B \otimes C,D} \\
 ((A \otimes B) \otimes C) \otimes D & & A \otimes ((B \otimes C) \otimes D) \\
 \alpha_{A \otimes B,C,D} \downarrow & & \downarrow id_A \otimes \alpha_{B,C,D} \\
 (A \otimes B) \otimes (C \otimes D) & \xrightarrow{\alpha_{A,B,C \otimes D}} & A \otimes (B \otimes (C \otimes D))
 \end{array}$$

7. A coherence condition called *triangle identity*:

$$\begin{array}{ccc}
 (A \otimes \mathbf{1}) \otimes B & \xrightarrow{\alpha_{A,\mathbf{1},B}} & A \otimes (\mathbf{1} \otimes B) \\
 \searrow R_A \otimes id_B & & \swarrow id_A \otimes L_B \\
 & A \otimes B &
 \end{array}$$

**Definition 12.** *Cartesian closed category*

*Cartesian closed category is a category with a terminal object, finite products and exponentiation.*

Note that, any cartesian closed category is the special case of a monoidal category, where tensor is a product and identity object is a terminal object.

**Definition 13.** *Monoidal functor*

Let  $\langle \mathcal{C}, \otimes_1, \mathbb{1}_{\mathcal{C}} \rangle$  and  $\langle \mathcal{D}, \otimes_2, \mathbb{1}_{\mathcal{D}} \rangle$  are monoidal categories.

A monoidal functor  $\mathcal{F} : \langle \mathcal{C}, \otimes_1, \mathbb{1}_{\mathcal{C}} \rangle \rightarrow \langle \mathcal{D}, \otimes_2, \mathbb{1}_{\mathcal{D}} \rangle$  is a functor  $\mathcal{F} : \mathcal{C} \rightarrow \mathcal{D}$  with natural transformations:

1.  $u : \mathbb{1}_{\mathcal{D}} \rightarrow \mathcal{F}\mathbb{1}_{\mathcal{C}}$ ;
2.  $A \otimes B : \mathcal{F}A \otimes_{\mathcal{D}} \mathcal{F}B \rightarrow \mathcal{F}(A \otimes_{\mathcal{C}} B)$ .

and coherence maps:

- *Associativity:*

$$\begin{array}{ccc}
 (\mathcal{F}A \otimes_{\mathcal{D}} \mathcal{F}B) \otimes_{\mathcal{D}} \mathcal{F}C & \xrightarrow{\alpha_{\mathcal{F}A, \mathcal{F}B, \mathcal{F}C}^{\mathcal{D}}} & \mathcal{F}A \otimes_{\mathcal{D}} (\mathcal{F}B \otimes_{\mathcal{D}} \mathcal{F}C) \\
 \downarrow (A \otimes B) \otimes_{\mathcal{D}} id_{\mathcal{F}B} & & \downarrow id_{\mathcal{F}A} \otimes_{\mathcal{D}} B \otimes C \\
 \mathcal{F}(A \otimes_{\mathcal{C}} B) \otimes_{\mathcal{D}} \mathcal{F}C & & \mathcal{F}A \otimes_{\mathcal{D}} \mathcal{F}(B \otimes_{\mathcal{C}} C) \\
 \downarrow (A \otimes_{\mathcal{C}} B) \otimes C & & \downarrow A \otimes (B \otimes_{\mathcal{C}} C) \\
 \mathcal{F}((A \otimes_{\mathcal{C}} B) \otimes_{\mathcal{C}} C) & \xrightarrow{\mathcal{F}(\alpha_{A, B, C}^{\mathcal{C}})} & \mathcal{F}(A \otimes_{\mathcal{C}} (B \otimes_{\mathcal{C}} C))
 \end{array}$$

- *Left unitality:*

$$\begin{array}{ccc}
 \mathbb{1}_{\mathcal{D}} \otimes_{\mathcal{D}} \mathcal{F}A & \xrightarrow{u \otimes id_{\mathcal{F}A}} & \mathcal{F}\mathbb{1}_{\mathcal{C}} \otimes_{\mathcal{D}} \mathcal{F}A \\
 \downarrow L_{\mathcal{F}A}^{\mathcal{D}} & & \downarrow \mathbb{1}_{\mathcal{C}} \otimes A \\
 \mathcal{F}A & \xleftarrow{\mathcal{F}(L_A^{\mathcal{C}})} & \mathcal{F}(\mathbb{1}_{\mathcal{C}} \otimes_{\mathcal{C}} A)
 \end{array}$$

- *Right unitality:*

$$\begin{array}{ccc}
 \mathcal{F}A \otimes_{\mathcal{D}} \mathbb{1}_{\mathcal{D}} & \xrightarrow{id_{\mathcal{F}A} \otimes u} & \mathcal{F}A \otimes_{\mathcal{D}} \mathcal{F}\mathbb{1}_{\mathcal{C}} \\
 \downarrow R_{\mathcal{F}A}^{\mathcal{D}} & & \downarrow A \otimes \mathbb{1}_{\mathcal{C}} \\
 \mathcal{F}A & \xleftarrow{\mathcal{F}(R_A^{\mathcal{C}})} & \mathcal{F}(A \otimes_{\mathcal{C}} \mathbb{1}_{\mathcal{C}})
 \end{array}$$

**Definition 14.** *Applicative functor*

An applicative functor is a triple  $\langle \mathcal{C}, \mathcal{F}, \eta \rangle$ , where  $\mathcal{C}$  is a monoidal category,  $\mathcal{F}$  is a monoidal endofunctor and  $\eta : Id_{\mathcal{C}} \Rightarrow \mathcal{F}$  is a natural transformation (similar to unit in monad), such that:

1.  $u = \eta_{\mathbb{1}}$ ;
2.  $A \otimes B \circ (\eta_A \otimes \eta_B) = \eta_{A \otimes B}$ , i.e. the following diagram commutes:

$$\begin{array}{ccc}
 A \otimes B & \xrightarrow{\eta_A \otimes \eta_B} & \mathcal{F}A \otimes \mathcal{F}B \\
 & \searrow \eta_{A \otimes B} & \downarrow A \otimes B \\
 & & \mathcal{F}(A \otimes B)
 \end{array}$$

By default we will consider an arbitrary monoidal functor on cartesian closed category below.

### 3.1 Soundness and completeness

**Definition 15.** *Semantical translation from  $\lambda_{IEL^-}$  to some cartesian closed category  $\mathcal{C}$  with an applicative functor  $\langle \mathcal{C}, \sqsubseteq, \eta \rangle$ :*

1. *Interpretation for types:*

- $\llbracket A \rrbracket := \hat{A}, A \in \mathbb{T}$ , where  $\hat{A}$  is an object of  $\mathcal{C}$  obtained by some given assignment;
- $\llbracket A \rightarrow B \rrbracket := \llbracket B \rrbracket^{\llbracket A \rrbracket}$ ;
- $\llbracket A \times B \rrbracket := \llbracket A \rrbracket \times \llbracket B \rrbracket$ .

2. *Interpretation for modal types:*

$$(a) \llbracket \Box A \rrbracket = \sqsubseteq \llbracket A \rrbracket;$$

3. *Interpretation for contexts:*

- (a)  $\llbracket \quad \rrbracket = \mathbb{1}$ , where  $\mathbb{1}$  is a terminal object of a given ccc;
- (b)  $\llbracket \Gamma, x : A \rrbracket = \llbracket \Gamma \rrbracket \times \llbracket A \rrbracket$

4. *Interpretation for typing assignment:*

$$(a) \llbracket \Gamma \vdash M : A \rrbracket := \llbracket M \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket.$$

5. *Interpretation for typing rules:*

$$\begin{array}{c} \frac{}{\llbracket \Gamma, x : A \vdash x : A \rrbracket = \pi_2 : \llbracket \Gamma \rrbracket \times \llbracket A \rrbracket \rightarrow \llbracket A \rrbracket} \\[10pt] \frac{\llbracket \Gamma, x : A \vdash M : B \rrbracket = \llbracket M \rrbracket : \llbracket \Gamma \rrbracket \times \llbracket A \rrbracket \rightarrow \llbracket B \rrbracket}{\llbracket \Gamma \vdash (\lambda x.M) : A \rightarrow B \rrbracket = \Lambda(\llbracket M \rrbracket) : \llbracket \Gamma \rrbracket \rightarrow \llbracket B \rrbracket^{\llbracket A \rrbracket}} \\[10pt] \frac{\llbracket \Gamma \vdash M : A \rightarrow B \rrbracket = \llbracket M \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket B \rrbracket^{\llbracket A \rrbracket} \quad \llbracket \Gamma \vdash N : A \rrbracket = \llbracket N \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket}{\llbracket \Gamma \vdash (MN) : B \rrbracket = \llbracket \Gamma \rrbracket \xrightarrow{\langle \llbracket M \rrbracket, \llbracket N \rrbracket \rangle} \llbracket B \rrbracket^{\llbracket A \rrbracket} \times \llbracket A \rrbracket \xrightarrow{\epsilon_{\llbracket A \rrbracket, \llbracket B \rrbracket}} \llbracket B \rrbracket}} \\[10pt] \frac{\llbracket \Gamma \vdash M : A \rrbracket = \llbracket M \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket \quad \llbracket \Gamma \vdash N : B \rrbracket = \llbracket N \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket B \rrbracket}{\llbracket \Gamma \vdash \langle M, N \rangle : A \times B \rrbracket = \langle \llbracket M \rrbracket, \llbracket N \rrbracket \rangle : \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket \times \llbracket B \rrbracket}} \\[10pt] \frac{\llbracket \Gamma \vdash M : A_1 \times A_2 \rrbracket = \llbracket M \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket A_1 \rrbracket \times \llbracket A_2 \rrbracket}{\llbracket \Gamma \vdash \pi_i M : A_i \rrbracket = \llbracket \Gamma \rrbracket \xrightarrow{\llbracket M \rrbracket} \llbracket A_1 \rrbracket \times \llbracket A_2 \rrbracket \xrightarrow{\pi_i} \llbracket A_i \rrbracket} \quad i = 1, 2 \\[10pt] \frac{\llbracket \Gamma \vdash M : A \rrbracket = \llbracket M \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket}{\llbracket \Gamma \vdash \mathbf{box} M : \Box A \rrbracket := \llbracket \Gamma \rrbracket \xrightarrow{\llbracket M \rrbracket} \llbracket A \rrbracket \xrightarrow{\eta_{\llbracket A \rrbracket}} \sqsubseteq \llbracket A \rrbracket}} \\[10pt] \frac{\llbracket \Gamma \vdash \vec{M} : \Box \vec{A} \rrbracket = \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle : \llbracket \Gamma \rrbracket \rightarrow \prod_{i=1}^n \sqsubseteq \llbracket A_i \rrbracket \quad \llbracket \vec{x} : \vec{A} \vdash N : B \rrbracket = \llbracket N \rrbracket : \prod_{i=1}^n \llbracket A_i \rrbracket \rightarrow \llbracket B \rrbracket}{\llbracket \Gamma \vdash \mathbf{let} \mathbf{box} \vec{x} = \vec{M} \mathbf{in} M : \Box B \rrbracket = \sqsubseteq(\llbracket N \rrbracket) \circ (\llbracket A_1 \rrbracket \otimes \dots \otimes \llbracket A_n \rrbracket) \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle : \llbracket \Gamma \rrbracket \rightarrow \sqsubseteq \llbracket B \rrbracket}} \end{array}$$

Interpretation for **let box**-rule is similar to interpretation for  $\Box$ -rule in term calculus for intuitionistic **K** [7].

**Theorem 5.** *Soundness*

Let  $\Gamma \vdash M : A$  and  $M =_r N$ , then  $\llbracket \Gamma \vdash M : A \rrbracket = \llbracket \Gamma \vdash N : A \rrbracket$

*Proof.*

**Lemma 8.**

$$\llbracket M[x_1 := M_1, \dots, x_n := M_n] \rrbracket = \llbracket M \rrbracket \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle.$$

*Proof.*

1)

$$\begin{aligned} & \llbracket \Gamma \vdash (\mathbf{box} M)[\vec{x} := \vec{M}] : \Box A \rrbracket = \\ & \quad \text{By substitution definition} \\ & \llbracket \Gamma \vdash \mathbf{box} (M[\vec{x} := \vec{M}]) : \Box A \rrbracket \\ & \quad \text{Interpretation for } \mathbf{box} \\ & \eta_{\llbracket A \rrbracket} \circ \llbracket (M[\vec{x} := \vec{M}]) \rrbracket \\ & \quad \text{Assumption} \\ & \eta_{\llbracket A \rrbracket} \circ (\llbracket M \rrbracket \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle) = \\ & \quad \text{Associativity of composition} \\ & (\eta_{\llbracket A \rrbracket} \circ \llbracket M \rrbracket) \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle = \\ & \quad \text{Interpretation for } \mathbf{box} \\ & \llbracket \Gamma \vdash \mathbf{box} M : \Box A \rrbracket \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle = \end{aligned}$$

2)

$$\begin{aligned} & \llbracket \Gamma \vdash (\mathbf{let box} \vec{x} = \vec{M} \mathbf{in} N)[\vec{y} := \vec{P}] : \Box B \rrbracket = \\ & \quad \text{Substitution} \\ & \llbracket \Gamma \vdash \mathbf{let box} \vec{x} = (\vec{M}[\vec{y} := \vec{P}]) \mathbf{in} N : \Box B \rrbracket = \\ & \quad \text{Interpretation for } \mathbf{let}_{\Box} \\ & \Box(\llbracket N \rrbracket) \circ (\llbracket A_1 \rrbracket \otimes \dots \otimes \llbracket A_n \rrbracket) \circ \llbracket \Gamma \vdash (\vec{M}[\vec{y} := \vec{P}]) : \Box \vec{A} \rrbracket = \\ & \quad \text{Induction hypothesis} \\ & \Box(\llbracket N \rrbracket) \circ (\llbracket A_1 \rrbracket \otimes \dots \otimes \llbracket A_n \rrbracket) \circ (\llbracket \vec{M} \rrbracket \circ \langle \llbracket P_1 \rrbracket, \dots, \llbracket P_n \rrbracket \rangle) = \\ & \quad \text{Associativity of composition} \\ & (\Box(\llbracket N \rrbracket) \circ (\llbracket A_1 \rrbracket \otimes \dots \otimes \llbracket A_n \rrbracket) \circ \llbracket \vec{M} \rrbracket) \circ \langle \llbracket P_1 \rrbracket, \dots, \llbracket P_n \rrbracket \rangle = \\ & \quad \text{Interpretation} \\ & \llbracket \Gamma \vdash \mathbf{let box} \vec{x} = \vec{M} \mathbf{in} N : \Box B \rrbracket \circ \langle \llbracket P_1 \rrbracket, \dots, \llbracket P_n \rrbracket \rangle \end{aligned}$$

□

**Lemma 9.**

Let  $\Gamma \vdash M : A$  and  $M \rightarrow_r N$ , then  $\llbracket \Gamma \vdash M : A \rrbracket = \llbracket \Gamma \vdash N : A \rrbracket$ ;

*Proof.*

Cases with  $\beta$ -reductions for  $\mathbf{let}_{\Box}$  are shown in [12]. Let us consider cases with **box**.

$$1) \llbracket \Gamma \vdash \mathbf{let box} \vec{x} = \mathbf{box} \vec{M} \mathbf{in} N : \Box B \rrbracket = \llbracket \Gamma \vdash \mathbf{box} N[\vec{x} := \vec{M}] : \Box B \rrbracket$$

$$\begin{aligned}
& \llbracket \Gamma \vdash \text{let } \vec{\mathbf{box}} \vec{x} = \vec{\mathbf{box}} \vec{M} \text{ in } N : \Box B \rrbracket = \\
& \quad \text{By interpretation} \\
& \Box(\llbracket N \rrbracket) \circ (\llbracket A_1 \rrbracket \otimes \cdots \otimes \llbracket A_n \rrbracket) \circ \langle \eta_{\llbracket A_1 \rrbracket} \circ \llbracket M_1 \rrbracket, \dots, \eta_{\llbracket A_n \rrbracket} \circ \llbracket M_n \rrbracket \rangle = \\
& \quad \text{By the property of a pair of morphisms} \\
& \Box(\llbracket N \rrbracket) \circ (\llbracket A_1 \rrbracket \otimes \cdots \otimes \llbracket A_n \rrbracket) \circ (\eta_{\llbracket A_1 \rrbracket} \times \cdots \times \eta_{\llbracket A_n \rrbracket}) \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle = \\
& \quad \text{Associativity of composition} \\
& \Box(\llbracket N \rrbracket) \circ ((\llbracket A_1 \rrbracket \otimes \cdots \otimes \llbracket A_n \rrbracket) \circ (\eta_{\llbracket A_1 \rrbracket} \times \cdots \times \eta_{\llbracket A_n \rrbracket})) \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle = \\
& \quad \text{By the definition of an applicative functor} \\
& \Box(\llbracket N \rrbracket) \circ \eta_{\llbracket A_1 \rrbracket \times \cdots \times \llbracket A_n \rrbracket} \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle = \\
& \quad \text{Naturality of } \eta \\
& \eta_{\llbracket B \rrbracket} \circ \llbracket N \rrbracket \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle = \\
& \quad \text{Associativity of composition} \\
& \eta_{\llbracket B \rrbracket} \circ (\llbracket N \rrbracket \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle) = \\
& \quad \text{Substitution lemma} \\
& \eta_{\llbracket B \rrbracket} \circ \llbracket \Gamma \vdash N[\vec{x} := \vec{M}] : \Box B \rrbracket \\
& \quad \text{By interpretation} \\
& \llbracket \Gamma \vdash \vec{\mathbf{box}} (N[\vec{x} := \vec{M}]) : \Box B \rrbracket
\end{aligned}$$

2)  $\llbracket \vdash \text{let } \vec{\mathbf{box}} \_ = \_ \text{ in } M : \Box A \rrbracket = \llbracket \vdash \vec{\mathbf{box}} M : \Box A \rrbracket$   
 $\llbracket \vdash \text{let } \vec{\mathbf{box}} \_ = \_ \text{ in } M : \Box A \rrbracket =$   
 $\quad \text{By interpretation}$   
 $\Box(\llbracket M \rrbracket) \circ u_{\mathbb{1}} =$   
 $\quad \text{By the definition of an applicative functor}$   
 $\Box(\llbracket M \rrbracket) \circ \eta_{\mathbb{1}} =$   $\square$   
 $\quad \text{By naturality of } \eta$   
 $\eta_{\llbracket A \rrbracket} \circ \llbracket M \rrbracket =$   
 $\quad \text{By interpretation}$   
 $\llbracket \vdash \vec{\mathbf{box}} M : \Box A \rrbracket$

$\square$

**Theorem 6. Completeness**

Let  $\llbracket \Gamma \vdash M : A \rrbracket = \llbracket \Gamma \vdash N : A \rrbracket$ , then  $M =_r N$ .

*Proof.*

We will consider term model for simply typed lambda calculus  $\times$  and  $\rightarrow$  standardly described in [15]:

**Definition 16. Equivalence on term pairs:**

Let us define relation  $\sim_{A,B} \subseteq (\mathbb{V} \times \Lambda_{\Box})^2$ , such that:

$$(x, M) \sim_{A,B} (y, N) \Leftrightarrow x : A \vdash M : B \ \& \ y : A \vdash N : A \ \& \ M =_r N[y := x];$$

We will denote equivalence class as  $[x, M]_{A,B} = \{(y, N) \mid (x, M) \sim_{A,B} (y, N)\}$  (we will drop indices below).

**Definition 17. Category  $\mathcal{C}(\lambda)$ :**

1.  $Ob_{\mathcal{C}} = \{\hat{A} \mid A \in \mathbb{T}\} \cup \{\mathbb{1}\};$
2.  $Hom_{\mathcal{C}(\lambda)}(\hat{A}, \hat{B}) = \{[x, M] \mid x : A \vdash_{\lambda_{IEL-}} M : B\};$
3. Let  $[x, M] \in Hom_{\mathcal{C}(\lambda)}(\hat{A}, \hat{B})$  and  $[y, N] \in Hom_{\mathcal{C}(\lambda)}(\hat{B}, \hat{C})$ , then  $[y, M] \circ [x, M] = [x, N[y := M]];$

4. Identity morphism  $id_{\hat{A}} = [x, x] \in Hom_{\mathcal{C}(\lambda)(\hat{A}, \hat{A})}$ ;
5.  $\mathbf{1}$  is a terminal object;
6.  $\widehat{A \times B} = \hat{A} \times \hat{B}$ ;
7. Canonical projection is defined as  $[x, \pi_i x] \in Hom_{\mathcal{C}(\lambda)}(\hat{A}_1 \times \hat{A}_2, \hat{A}_i)$  for  $i = 1, 2$ ;
8.  $\widehat{A \rightarrow B} = \hat{B}^{\hat{A}}$ ;
9. Evaluation arrow  $\epsilon_{\hat{A}, \hat{B}} = [x, (\pi_1 x)(\pi_2 x)] \in Hom_{\mathcal{C}(\lambda)}(\hat{B}^{\hat{A}} \times \hat{A}, \hat{B})$ .

We define endofunctor  $\square$  on  $\mathcal{C}(\lambda)$  and natural transformation  $\eta$  from  $id_{\mathcal{C}(\lambda)}$  to this endofunctor. It is sufficient to show  $\square$  and  $\eta$  form the relevant structure on  $\mathcal{C}(\lambda)$ .

**Definition 18.** Let us define an endofunctor  $\square : \mathcal{C}(\lambda) \rightarrow \mathcal{C}(\lambda)$ , such that for all  $[x, M] \in Hom_{\mathcal{C}(\lambda)}(\hat{A}, \hat{B})$ ,  $\square([x, M]) = [y, \text{let } \mathbf{box } x = y \text{ in } M] \in Hom_{\mathcal{C}(\lambda)}(\square \hat{A}, \square \hat{B})$ .

**Lemma 10.** *Functoriality*

1.  $\square(g \circ f) = \square g \circ \square f$ ;
2.  $\square(id_{\hat{A}}) = id_{\square \hat{A}}$ .

*Proof.* Easy checking using reduction rules. □

**Definition 19.** Let us define natural transformations:

1.  $\eta : Id_{\mathcal{C}(\lambda)} \Rightarrow \square$ , s. t.  $\forall \hat{A} \in Ob_{\mathcal{C}(\lambda)}$ ,  $\eta_{\hat{A}} = [x, \mathbf{box } x] \in Hom_{\mathcal{C}(\lambda)}(\hat{A}, \square \hat{A})$ ;
2.  $\hat{A} \otimes \hat{B} : \square \hat{A} \times \square \hat{B} \rightarrow \square(\hat{A} \times \hat{B})$ , s. t.  $\forall \hat{A}, \hat{B} \in Ob_{\mathcal{C}(\lambda)}$ ,  $\hat{A} \otimes \hat{B} = [p, \text{let } \mathbf{box } x, y = \pi_1 p, \pi_2 p \text{ in } \langle x, y \rangle] \in Hom_{\mathcal{C}(\lambda)}(\square \hat{A} \times \square \hat{B}, \square(\hat{A} \times \hat{B}))$ .

Implementation for  $\otimes$  in our term model is the instance of  $\text{let}_{\square}$ -rule:

$$\frac{\frac{p : \square A \times \square B \vdash p : \square A \times \square B}{p : \square A \times \square B \vdash \pi_1 p : \square A} \times_E \quad \frac{p : \square A \times \square B \vdash p : \square A \times \square B}{p : \square A \times \square B \vdash \pi_2 p : \square B} \times_E \quad \frac{x : A \vdash x : A \quad y : B \vdash y : B}{x : A, y : B \vdash \langle x, y \rangle : A \times B} \times_I}{p : \square A \times \square B \vdash \text{let } \mathbf{box } x, y = \pi_1 p, \pi_2 p \text{ in } \langle x, y \rangle : \square(A \times B)} \text{let}_{\square}$$

**Lemma 11.**

$\square$  is a monoidal endofunctor

*Proof.*

See [11] □

**Lemma 12.** *Naturality and coherence for  $\eta$ :*

1.  $\square f \circ \eta_A = \eta_B \circ f$ ;
2.  $(\hat{A} \otimes \hat{B}) \circ (\eta_A \times \eta_B) = \eta_{\hat{A} \times \hat{B}}$ ;

*Proof.*

$$1. \quad \Box f \circ \eta_{\hat{A}} = \eta_{\hat{B}} \circ f$$

$$\begin{aligned} \eta_{\hat{B}} \circ f &= \\ [y, \mathbf{box} y] \circ [x, M] &= \\ \text{Composition} & \\ [x, \mathbf{box} y[y := M]] &= \\ \text{By substitution} & \\ [x, \mathbf{box} M] & \end{aligned}$$

On the other hand:

$$\begin{aligned} \Box f \circ \eta_{\hat{A}} &= \\ [z, \mathbf{let} \mathbf{box} x = z \mathbf{in} M] \circ [x, \mathbf{box} x] &= \\ \text{By the definition of composition} & \\ [x, \mathbf{let} \mathbf{box} x = z \mathbf{in} M[z := \mathbf{box} x]] &= \\ \text{By substitution} & \\ [x, \mathbf{let} \mathbf{box} x = \mathbf{box} x \mathbf{in} M] &= \\ \beta\text{-reduction rule} & \\ [x, \mathbf{box} M[x := x]] &= \\ \text{By substitution} & \\ [x, \mathbf{box} M] & \end{aligned}$$

$$2. \quad (\hat{A} \otimes \hat{B}) \circ (\eta_A \times \eta_B) = \eta_{\hat{A} \times \hat{B}}$$

$$\begin{aligned} (\hat{A} \otimes \hat{B}) \circ (\eta_A \times \eta_B) &= \\ [q, \mathbf{let} \mathbf{box} x, y = \pi_1 q, \pi_2 q \mathbf{in} \langle x, y \rangle] \circ [p, \langle \mathbf{box} (\pi_1 p), \mathbf{box} (\pi_2 p) \rangle] &= \\ \text{Composition} & \\ [p, \mathbf{let} \mathbf{box} x, y = \pi_1 q, \pi_2 q \mathbf{in} \langle x, y \rangle [q := \langle \mathbf{box} (\pi_1 p), \mathbf{box} (\pi_2 p) \rangle]] &= \\ \text{By substitution} & \\ [p, \mathbf{let} \mathbf{box} x, y = \pi_1 (\langle \mathbf{box} (\pi_1 p), \mathbf{box} (\pi_2 p) \rangle), \pi_2 (\langle \mathbf{box} (\pi_1 p), \mathbf{box} (\pi_2 p) \rangle) \mathbf{in} \langle x, y \rangle] &= \\ \text{Reduction rules} & \\ [p, \mathbf{let} \mathbf{box} x, y = \mathbf{box} (\pi_1 p), \mathbf{box} (\pi_2 p) \mathbf{in} \langle x, y \rangle] &= \\ \text{Reduction rule} & \\ [p, \mathbf{box} (\langle x, y \rangle [x := \pi_1 p, y := \pi_2 p])] &= \\ \text{Substitution} & \\ [p, \mathbf{box} \langle \pi_1 p, \pi_2 p \rangle] &= \\ \eta\text{-reduction} & \\ [p, \mathbf{box} p] &= \\ \text{By definition} & \\ \eta_{\hat{A} \times \hat{B}} & \end{aligned}$$

□

**Definition 20.**

$$u_{\mathbb{1}} = [\blacksquare, \mathbf{let} \mathbf{box} \_ = \_ \mathbf{in} \blacksquare] \in Hom_{\mathcal{C}(\lambda)}(\mathbb{1}, \Box \mathbb{1}).$$

**Lemma 13.**

$$u_{\mathbb{1}} = \eta_{\mathbb{1}}$$

*Proof.* Immediately. □

**Lemma 14.**  $\langle \mathcal{C}(\lambda), \Box, \eta \rangle$  is an applicative functor



*Proof.* Immediately follows from previous lemmas in the section.  $\square$

$\square$

## 4 Relation with Moggi's monadic metalanguage

**Definition 21.** *Monadic metalanguage*

*Monadic metalanguage is a simply typed lambda calculus with additional typing rules [19]:*

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \mathbf{val} M : \bigcirc A} \bigcirc_I \qquad \frac{\Gamma \vdash M : \bigcirc A \quad \Gamma, x : A \vdash N : \bigcirc B}{\Gamma \vdash \mathbf{let val} x = M \mathbf{in} N : \bigcirc B} \bigcirc_E$$

**Definition 22.** *Reduction rules for monadic metalanguage*

1.  $\mathbf{let val} x = \mathbf{val} M \mathbf{in} N \rightarrow_\beta N[x := M];$
2.  $\mathbf{let val} x = (\mathbf{let val} y = N \mathbf{in} P) \mathbf{in} M \rightarrow_\beta \mathbf{let val} y = N \mathbf{in} (\mathbf{let val} x = P \mathbf{in} M);$
3.  $\mathbf{let val} x = M \mathbf{in val} x \rightarrow_\eta M.$

Let us define translation  $\ulcorner \cdot \urcorner$  from  $\lambda_{\text{IEL-}}$  into monadic metalanguage. Without loss of generality we will consider  $\lambda_{\text{IEL-}}$  with  $\rightarrow$  and  $\Box$ :

**Definition 23.** *Translation for types*

1.  $\ulcorner A \urcorner = A$  for an atomic  $A$ ;
2.  $\ulcorner A \rightarrow B \urcorner = \ulcorner A \urcorner \rightarrow \ulcorner B \urcorner$ ;
3.  $\ulcorner \Box A \urcorner = \bigcirc \ulcorner A \urcorner$

**Definition 24.** *Translation for terms*

1.  $\ulcorner x \urcorner = x$  for  $x \in \mathbb{V}$ ;
2.  $\ulcorner \lambda x. M \urcorner = \lambda x. \ulcorner M \urcorner$ ;
3.  $\ulcorner MN \urcorner = \ulcorner M \urcorner \ulcorner N \urcorner$ ;
4.  $\ulcorner \mathbf{box} M \urcorner = \mathbf{val} \ulcorner M \urcorner$
5.  $\ulcorner \mathbf{let box} \_ = \_ \mathbf{in} N \urcorner = \mathbf{val} \ulcorner N \urcorner$ ;
6.  $\ulcorner \mathbf{let box} \vec{x} = \vec{M} \mathbf{in} N \urcorner = \mathbf{let val} \vec{x} = \ulcorner \vec{M} \urcorner \mathbf{in val} \ulcorner N \urcorner$ ;

Where  $\ulcorner \Gamma \urcorner \vdash \mathbf{let val} \vec{x} = \ulcorner \vec{M} \urcorner \mathbf{in val} \ulcorner N \urcorner$  denotes  $\mathbf{let val} x_1 = \ulcorner M_1 \urcorner \mathbf{in} (\dots \mathbf{in} (\mathbf{let val} x_n = \ulcorner M_n \urcorner \mathbf{in val} \ulcorner N \urcorner) \dots)$ .

**Definition 25.** *Interpretation for modal rules*

$$\frac{\ulcorner \Gamma \urcorner \vdash M : A = \ulcorner \Gamma \urcorner \vdash \ulcorner M \urcorner : \ulcorner A \urcorner}{\ulcorner \Gamma \urcorner \vdash \mathbf{box} M : \Box A = \ulcorner \Gamma \urcorner \vdash \mathbf{val} \ulcorner M \urcorner : \bigcirc \ulcorner A \urcorner}$$

$$\frac{\frac{\Gamma \vdash \vec{M} : \Box \vec{A} = \Gamma' \vdash \vec{M}' : \bigcirc \vec{A}' \quad \frac{\vec{x} : \vec{A} \vdash N : B = \vec{x} : \vec{A}' \vdash N' : B'}{\vec{x} : \vec{A}' \vdash \text{val } N' : \bigcirc B'}}{\Gamma \vdash \text{let box } \vec{x} = \vec{M} \text{ in } N : \Box B = \Gamma' \vdash \text{let val } \vec{x} = \vec{M}' \text{ in val } N' : \bigcirc B'}$$

**Lemma 15.**

$$\ulcorner M[x := N] \urcorner = \ulcorner M' \urcorner[x := \ulcorner N \urcorner]$$

*Proof.* Induction on the structure of  $M$ . □

**Lemma 16.**

If  $M =_r N$ , then  $\ulcorner M \urcorner =_{\beta\eta} \ulcorner N \urcorner$ .

*Proof.*

$$\begin{aligned} 1) & \quad \ulcorner \text{let box } x = (\text{let box } \vec{y} = \vec{N} \text{ in } P) \text{ in } M \urcorner = \\ & \quad \ulcorner \text{let val } x = (\text{let val } \vec{y} = \ulcorner \vec{N} \urcorner \text{ in val } \ulcorner P \urcorner) \text{ in val } \ulcorner M \urcorner \rightarrow_{\beta} \\ & \quad \ulcorner \text{let val } \vec{y} = \ulcorner \vec{N} \urcorner \text{ in } (\text{let val } x = \ulcorner P \urcorner \text{ in val } \ulcorner M \urcorner) \rightarrow_{\beta} \\ & \quad \ulcorner \text{let val } \vec{y} = \ulcorner \vec{N} \urcorner \text{ in val } \ulcorner M \urcorner[x := \ulcorner P \urcorner] \urcorner = \\ & \quad \ulcorner \text{let box } \vec{y} = \vec{N} \text{ in } M[x := P] \urcorner \\ 2) & \quad \ulcorner \text{let box } \vec{x} = \text{box } \vec{N} \text{ in } M \urcorner = \\ & \quad \ulcorner \text{let val } \vec{x} = \text{val } \ulcorner \vec{N} \urcorner \text{ in val } \ulcorner M \urcorner \rightarrow_{\beta} \\ & \quad \ulcorner \text{val } \ulcorner M \urcorner[\vec{x} := \ulcorner \vec{N} \urcorner] \urcorner = \\ & \quad \ulcorner \text{box } M[\vec{x} := \vec{N}] \urcorner \\ 3) & \quad \ulcorner \text{let box } x = M \text{ in } x \urcorner = \\ & \quad \ulcorner \text{let val } x = \ulcorner M \urcorner \text{ in val } x \rightarrow_{\eta} \\ & \quad \ulcorner M \urcorner \end{aligned}$$

□

**Theorem 7.**

If  $\Gamma \vdash M : A$ , then  $\ulcorner \Gamma \urcorner \vdash \ulcorner M \urcorner : \ulcorner A \urcorner$

*Proof.* Follows from lemmas above. □

## 5 Acknowledgment

We are grateful to Neel Krishnaswami, Vladimir Krupski, Valeria de Paiva, Valerii Plisko and Vladimir Vasyukov for consulting and advice.

The research described in this paper was supported by Russian Foundation for Basic Research (grant 16-03-00364).

## References

- [1] T. Abe, Completeness of modal proofs in first-order predicate logic, Computer Software, JSSST Journal. 24 (2007).

- [2] S. Abramsky and N. Tzevelekos, Introduction to Categories and Categorical Logic, in: B. Coecke, ed., *New Structures for Physics. Lecture Notes in Physics*, vol. 813 (Springer-Verlag, Berlin, 2011).
- [3] S. Artemov and T. Protopopescu, Intuitionistic Epistemic Logic, *The Review of Symbolic Logic*. 9 (2016).
- [4] S. Awodey, *Category Theory* (2nd ed.), Oxford University Press, 2010.
- [5] H. P. Barendregt, *The Lambda Calculus — Its Syntax and Semantics*, North-Holland, 1985.
- [6] H.P. Barendregt. Lambda calculi with types, in: S. Abramsky, Dov M. Gabbay, and S. E. Maibaum, eds., *Handbook of logic in computer science* (vol. 2), *Osborne Handbooks Of Logic In Computer Science*, vol. 2. (Oxford University Press, New York, 1993).
- [7] G. Bellin, V. C. V. de Paiva, and E. Ritter. Extended Curry-Howard correspondence for a basic constructive modal logic, *Proceedings of Methods for Modalities*, 2001.
- [8] S. Eilenberg and M. Kelly. Closed categories, *Proc. Conf. Categorical Algebra*, La Jolla, (1965).
- [9] J.-Y. Girard, P. Taylor and Y. Lafont, *Proofs and Types*, Cambridge University Press, 1989.
- [10] J. Roger Hindley, *Basic Simple Type Theory*, Cambridge University Press, 1997.
- [11] Y. Kakutani. A Curry-Howard Correspondence for Intuitionistic Normal Modal Logic. // *Computer Software*, 2008.
- [12] Y. Kakutani, Call-by-Name and Call-by-Value in Normal Modal Logic, in: Shao Z. (eds) *Programming Languages and Systems*, *Lecture Notes in Computer Science*, vol 4807 (2016).
- [13] G. A. Kavvos. The Many Worlds of Modal  $\lambda$ -calculi: I. Curry-Howard for Necessity, Possibility and Time. // [ ] URL: <https://arxiv.org/abs/1605.08106>
- [14] V. N. Krupski and A. Yatmanov. Sequent Calculus for Intuitionistic Epistemic Logic IEL, in: Artemov S., Nerode A. (eds), *Logical Foundations of Computer Science. Lecture Notes in Computer Science*, vol 9537. Springer, Cham (2016).
- [15] J. Lambek and P.J. Scott, *Introduction to Higher Order Categorical Logic*, Cambridge University Press, 1986.
- [16] S. MacLane. *Categories for the Working Mathematician*, *Graduate Texts in Mathematics*, Springer-Verlag, 1998.
- [17] C. McBride and R. Paterson, Applicative programming with effects, *Journal of Functional Programming*, 18 (2008).
- [18] C. McBride and R. Paterson, Functional Pearl. Idioms: applicative programming with effects, *Journal of Functional Programming*, 18 (2005).

- [19] E. Moggi, Notions of computation and monads, *Inf. Comput.*, 93 (1) (1991).
- [20] R. Nederpelt and H. Geuvers, *Type Theory and Formal Proof: An Introduction*. Cambridge University Press, 2014.
- [21] R. Paterson, Constructing applicative functors, in: *Mathematics of Program Construction, Lecture Notes in Computer Science* (Springer, Berlin, 2012).
- [22] F. Pfenning and R. Davies. A judgmental reconstruction of modal logic. *Mathematical Structures in Computer Science*, vol. 11 (2001).
- [23] B. C. Pierce, *Types and Programming Languages*, The MIT Press, 2002.
- [24] M. H. Sorensen and P. Urzyczyn, *Lectures on the Curry-Howard isomorphism*, Elsevier Science, 1998. pp 261.
- [25] A.S. Troelstra and H. Schwichtenberg, *Basic Proof Theory*, Cambridge University Press, 1996.