A Linear-Logical Reconstruction of Intuitionistic Modal Logic S4

Yosuke Fukuda
Graduate School of Informatics, Kyoto University, Japan
http://www.fos.kuis.kyoto-u.ac.jp/~yfukuda
yfukuda@fos.kuis.kyoto-u.ac.jp

Akira Yoshimizu
French Institute for Research in Computer Science and Automation (INRIA), France
http://www.cs.unibo.it/~akira.yoshimizu
akira.yoshimizu@inria.fr

---- **Abstract** ----

We propose a *modal linear logic* to reformulate intuitionistic modal logic S4 (IS4) in terms of linear logic, establishing an S4-version of Girard translation from IS4 to it. While the Girard translation from intuitionistic logic to linear logic is well-known, its extension to modal logic is non-trivial since a naive combination of the S4 modality and the exponential modality causes an undesirable interaction between the two modalities. To solve the problem, we introduce an extension of intuitionistic multiplicative exponential linear logic with a modality combining the S4 modality and the exponential modality, and show that it admits a sound translation from IS4. Through the Curry–Howard correspondence we further obtain a Geometry of Interaction Machine semantics of the modal λ-calculus by Pfenning and Davies for staged computation.

1 Introduction

Linear logic discovered by Girard [7] is, as he wrote, not an alternative logic but should be regarded as an "extension" of usual logics. Whereas usual logics such as classical logic and intuitionistic logic admit the structural rules of weakening and contraction, linear logic does not allow to use the rules freely, but it reintroduces them in a controlled manner by using the exponential modality '!' (and its dual '?'). Usual logics are then reconstructed in terms of linear logic with the power of the exponential modalities, via the Girard translation.

In this paper, we aim to extend the framework of linear-logical reconstruction to the $(\Box, \supset)$-fragment of intuitionistic modal logic S4 (IS4) by establishing what we call "modal linear logic" and an S4-version of Girard translation from IS4 into it. However, the crux to give a faithful translation is that a naive combination of the $\Box$-modality and the !-modality causes an undesirable interaction between the inference rules of the two modalities. To solve

Syntactic category	**Inference rule**
Formulae $A, B, C ::= p \mid A \multimap B \mid \,!A$	$\dfrac{}{A \vdash A}\ \text{Ax} \quad \dfrac{\Gamma \vdash A \quad \Gamma', A \vdash B}{\Gamma, \Gamma' \vdash B}\ \text{Cut} \quad \dfrac{!\Gamma \vdash A}{!\Gamma \vdash\, !A}\ !\text{R}$

$$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\ \multimap\text{R} \quad \dfrac{\Gamma \vdash A \quad \Gamma', B \vdash C}{\Gamma, \Gamma', A \multimap B \vdash C}\ \multimap\text{L} \quad \dfrac{\Gamma, A \vdash B}{\Gamma, !A \vdash B}\ !\text{L} \quad \dfrac{\Gamma \vdash B}{\Gamma, !A \vdash B}\ !\text{W} \quad \dfrac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B}\ !\text{C}$$

■ **Figure 1** Definition of IMELL.

$$\dfrac{\ulcorner A \urcorner}{} \qquad \ulcorner p \urcorner \stackrel{\text{def}}{=} p, \qquad \ulcorner A \supset B \urcorner \stackrel{\text{def}}{=} (!\ulcorner A \urcorner) \multimap \ulcorner B \urcorner \qquad\qquad \dfrac{\ulcorner \Gamma \urcorner}{} \qquad \ulcorner \Gamma \urcorner \stackrel{\text{def}}{=} \{\ulcorner A \urcorner \mid A \in \Gamma\}$$

■ **Figure 2** Definition of the Girard translation from intuitionistic logic.

the problem, we define the modal linear logic as an extension of intuitionistic multiplicative exponential linear logic with a modality '⊡' (pronounced by "bangbox") that integrates '□' and '!', and show that it admits a faithful translation from IS4.

As an application, we consider a computational interpretation of the modal linear logic. A typed λ-calculus that we will define corresponds to a natural deduction for the modal linear logic through the Curry–Howard correspondence, and it can be seen as a reconstruction of the modal λ-calculus by Pfenning and Davies [18, 5] for the so-called staged computation. Thanks to our linear-logical reconstruction, we can further obtain a Geometry of Interaction Machine (GoIM) for the modal λ-calculus.

The remainder of this paper is organized as follows. In Section 2 we review some formalizations of linear logic and IS4. In Section 3 we explain a linear-logical reconstruction of IS4. First, we discuss how a naive combination of linear logic and modal logic fails to obtain a faithful translation. Then, we propose a modal linear logic with the ⊡-modality that admits a faithful translation from IS4. In Section 4 we give a computational interpretation of modal linear logic through a typed λ-calculus. In Section 5 we provide an axiomatization of modal linear logic by a Hilbert-style deductive system. In Section 6 we obtain a GoIM of our typed λ-calculus as an application of our linear-logical reconstruction. In Sections 7 and 8 we discuss related work and conclude our work, respectively.

2   Preliminaries

We recall several systems of linear logic and modal logic. In this paper, we consider the minimal setting to give an S4-version of Girard translation and its computational interpretation. Thus, every system we will use only contain an implication and a modality as operators.

2.1   Intuitionistic MELL and its Girard translation

Figure 1 shows the standard definition of the $(!, \multimap)$-fragment of *intuitionistic multiplicative exponential linear logic*, which we refer to as IMELL. A formula is either a propositional variable, a linear implication, or an exponential modality. We let p range over the set of propositional variables, and A, B, C range over formulae. A *context* Γ is defined to be a multiset of formulae, and hence the exchange rule is assumed as a meta-level operation. A *judgment* consists of a context and a formula, written as $\Gamma \vdash A$. As a convention, we often write $\Gamma \vdash A$ to mean that the judgment is derivable (and we assume similar conventions throughout this paper). The notation $!\Gamma$ in the rule $!$R denotes the multiset $\{!A \mid A \in \Gamma\}$.

Syntactic category	Inference rule

$$\text{Formulae} \quad A, B, C ::= p \mid A \supset B \mid \Box A$$

$$\dfrac{}{A \vdash A} \ \text{Ax} \qquad \dfrac{\Gamma \vdash A \quad \Gamma', A \vdash B}{\Gamma, \Gamma' \vdash B} \ \text{Cut} \qquad \dfrac{\Box\Gamma \vdash A}{\Box\Gamma \vdash \Box A} \ \Box\text{R}$$

$$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \supset B} \supset\text{R} \qquad \dfrac{\Gamma \vdash A \quad \Gamma', B \vdash C}{\Gamma, \Gamma', A \supset B \vdash C} \supset\text{L} \qquad \dfrac{\Gamma, A \vdash B}{\Gamma, \Box A \vdash B} \Box\text{L} \qquad \dfrac{\Gamma \vdash B}{\Gamma, A \vdash B} \text{W} \qquad \dfrac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B} \text{C}$$

Figure 3 Definition of $\text{LJ}^\Box$.

Figure 2 defines the Girard translation[1] from the $\supset$-fragment of intuitionistic propositional logic IL. For an IL-formula A, $\lceil A \rceil$ will be an IMELL-formula; and $\lceil \Gamma \rceil$ a multiset of IMELL-formulae. Then, we can show that the Girard translation from IL to IMELL is sound.

▶ **Theorem 1** (Soudness of the translation). *If* $\Gamma \vdash A$ *in* IL*, then* $!\lceil \Gamma \rceil \vdash \lceil A \rceil$ *in* IMELL*.*

2.2 Intuitionistic S4

We review a formalization of the $(\Box, \supset)$-fragment of intuitionistic propositional modal logic S4 (IS4). In what follows, we use a sequent calculus for the logic, called $\text{LJ}^\Box$. The calculus $\text{LJ}^\Box$ used here is defined in a standard manner in the literature (e.g. it can be seen as the IS4-fragment of **G1s** for classical modal logic S4 by Troelstra and Schwichtenberg [23]).

Figure 3 shows the definition of $\text{LJ}^\Box$. A formula is either a propositional variable, an intuitionistic implication, or a box modality. A *context* and a *judgment* are defined similarly in IMELL. The notation $\Box\Gamma$ in the rule $\Box$R denotes the multiset $\{\Box A \mid A \in \Gamma\}$.

▶ **Remark 2**. It is worth noting that the !-exponential in IMELL and the $\Box$-modality in $\text{LJ}^\Box$ have similar structures. To see this, let us imagine the rules $\Box$R and $\Box$L replacing the symbol '$\Box$' with '!'. The results will be exactly the same as !R and !L. In fact, the !-exponential satisfies the S4 axiomata in IMELL, which is the reason we also call it as a modality.

2.3 Typed λ-calculus of the intuitionistic S4

We review the modal λ-calculus developed by Pfenning and Davies [18, 5], which we call $\lambda^\Box$. The system $\lambda^\Box$ is essentially the same calculus as $\lambda_e^{\to\Box}$ in [5], although some syntax are changed to fit our notation in this paper. $\lambda^\Box$ is known to correspond to a natural deduction system for IS4, as is shown in [18].

Figure 4 shows the definition of $\lambda^\Box$. The set of types corresponds to that of formulae of IS4. We let x range over the set of term variables, and M, N, L range over the set of terms. The first three terms are as in the simply-typed λ-calculus. The terms $\Box M$ and $\textbf{let } \Box x = M \textbf{ in } N$ is used to represent a constructor and a destructor for types $\Box A$, respectively. The variable x in $\lambda x : A.M$ and $\textbf{let } \Box x = M \textbf{ in } N$ is supposed to be *bound* in the usual sense and the *scope* of the biding is M and N, respectively. The set of *free* (i.e., unbound) variables in M is denoted by $\text{FV}(M)$. We write the *capture-avoiding substitution* $M[x := N]$ to denote the result of replacing N for every free occurrence of x in M.

A *(type) context* is defined to be the set of pairs of a term variable x_i and a type A_i such that all the variables are distinct, which is written as $x_1 : A_1, \cdots, x_n : A_n$ and is denoted by Γ, Δ, Σ, etc. Then, a *(type) judgment* is defined, in the so-called *dual-context* style, to consists of two contexts, a term, and a type, written as $\Delta; \Gamma \vdash M : A$.

---

[1] This is known to be the *call-by-name* Girard translation (cf. [12]) and we only follow this version in later discussions. However, we conjecture that our work can apply to other versions.

Syntactic category	Reduction rule
Types $A, B, C ::= p \mid A \supset B \mid \Box A$	$(\beta \supset)$ $(\lambda x : A.M) N \qquad \rightsquigarrow M[x := N]$
Terms $M, N, L ::= x \mid \lambda x : A.M \mid M\, N$	$(\beta\Box)$ $\mathbf{let}\,\Box x = \Box N \mathbf{\ in\ } M \rightsquigarrow M[x := N]$
$\qquad\qquad \mid \Box M \mid \mathbf{let}\,\Box x = M \mathbf{\ in\ } N$	

Typing rule

$$\frac{}{\Delta; \Gamma, x : A \vdash x : A}\ \text{Ax} \qquad\qquad \frac{}{\Delta, x : A; \Gamma \vdash x : A}\ \Box\text{Ax}$$

$$\frac{\Delta; \Gamma, x : A \vdash M : B}{\Delta; \Gamma \vdash (\lambda x : A.M) : A \supset B}\ \supset\text{I} \qquad \frac{\Delta; \Gamma \vdash M : A \supset B \qquad \Delta; \Gamma \vdash N : A}{\Delta; \Gamma \vdash M\, N : B}\ \supset\text{E}$$

$$\frac{\Delta; \emptyset \vdash M : A}{\Delta; \Gamma \vdash \Box M : \Box A}\ \Box\text{I} \qquad \frac{\Delta; \Gamma \vdash M : \Box A \qquad \Delta, x : A; \Gamma \vdash N : B}{\Delta; \Gamma \vdash \mathbf{let}\,\Box x = M \mathbf{\ in\ } N : B}\ \Box\text{E}$$

🟨 **Figure 4** Definition of $\lambda^\Box$.

The intuition behind the judgment $\Delta; \Gamma \vdash M : A$ is that the context Δ is intended to implicitly represent assumptions for types of form $\Box A$, while the context Γ is used to represent ordinary assumptions as in the simply-typed λ-calculus.

The typing rules are summarized as follows. Ax, $\supset$I, and $\supset$E are all standard, although they are defined in the dual-context style. $\Box$Ax is another variable rule, which can be seen as what to formalize the modal axiom T (i.e., $\vdash \Box A \supset A$) from the logical viewpoint. $\Box$I is a rule for the constructor of $\Box A$, which corresponds to the necessitation rule for the $\Box$-modality. Similarly, $\Box$E is for the destructor of $\Box A$, which corresponds to the elimination rule.

The *reduction* $\rightsquigarrow$ is defined to be the least compatible relation on terms generated by $(\beta \supset)$ and $(\beta\Box)$. The *multistep reduction* $\rightsquigarrow^+$ is defined to be the transitive closure of $\rightsquigarrow$.

3   Linear-logical reconstruction

3.1   Naive attempt at the linear-logical reconstruction

It is natural for a "linear-logical reconstruction" of IS4 to define a system that has both properties of linear logic and modal logic, so as to be a target system for an S4-version of Girard translation. However, a naive combination of linear logic and modal logic is not suitable to establish a faithful translation.

Let us consider what happens if we adopt a naive system. The simplest way to define a target system for the S4-version of Girard translation is to make an extension of IMELL with the $\Box$-modality. Suppose that a deductive system IMELL$^\Box$ is such a calculus, that is, the formulae of IMELL$^\Box$ are defined by the following grammar:

$$A, B ::= p \mid A \multimap B \mid !A \mid \Box A$$

with the inference rules being those of IMELL, along with the rules $\Box$R and $\Box$L of LJ$^\Box$.

As in the case of Girard translation from IL to IMELL, we have to establish the following theorem for some translation $\lceil - \rceil$:

> If $\Gamma \vdash A$ is derivable in LJ$^\Box$, then so is $!\lceil \Gamma \rceil \vdash \lceil A \rceil$ in IMELL$^\Box$.

but, if we extend our previous translation $\lceil - \rceil$ from IL to IMELL with $\lceil \Box A \rceil \stackrel{\text{def}}{=} \Box\lceil A \rceil$, we get stuck in the case of $\Box$R. This is because we need to establish the inference $\Box'$ in Figure 5, which means that we have to be able to obtain a derivation of form $!\lceil \Box\Gamma \rceil \vdash \Box\lceil A \rceil$ from that of $!\lceil \Box\Gamma \rceil \vdash \lceil A \rceil$ in IMELL$^\Box$.

Figure 5 Translation for the case of □R.

Figure 6 Valid inference in LJ$^\square$.

Figure 7 Invalid inference in IMELL$^\square$.

Figure 8 Definition of IMELL$^\square$.

However, the inference $\square'$ is invalid in IMELL$^\square$ in general, because there exists a counterexample. First, the inference shown in Figure 6 is valid, and the judgment $\square(p \supset q), \square p \vdash \square q$ is indeed derivable in LJ$^\square$. However, the corresponding inference via $\lceil - \rceil$ is invalid as Figure 7 shows. In the figure, the judgments correspond to those in Figure 6 via $\lceil - \rceil$, but the inference $\square$R in Figure 7 is invalid in IMELL$^\square$ due to the side-condition of $\square$R. Even worse, we can see that the judgment $!\square(!p \multimap q), !\square p \vdash \square q$ is itself underivable in IMELL$^{\square 2}$.

Moreover, one may think the other cases that we extend the original translation $\lceil - \rceil$ from IL to IMELL with $\lceil \square A \rceil \overset{\text{def}}{=} !\square \lceil A \rceil$ or $\lceil \square A \rceil \overset{\text{def}}{=} \square! \lceil A \rceil$ will work to obtain a faithful translation. However, the judgment $\square p \vdash \square \square p$ will be a counter-example in either case.

All in all, the problem of the naive combination formulated as IMELL$^\square$ intuitively came from an undesirable interaction between the right rules of the two modalities:

$$\frac{!\Gamma \vdash A}{!\Gamma \vdash !A} \text{ !R} \qquad\qquad \frac{\square \Gamma \vdash A}{\square \Gamma \vdash \square A} \text{ □R}$$

Each of these rules has a side-condition: the conclusion $!A$ in !R must be derived from the modalized context $!\Gamma$, and similarly for $\square A$ in $\square$R. This makes it hard to obtain a faithful S4-version of Girard translation for this naive extension.

3.2 Modal linear logic

We propose a *modal linear logic* to give a faithful S4-version of Girard translation from IS4.

First of all, the problem we have identified essentially came from the fact that there is no relationship between '!' and '□', and hence the side-conditions of !R and □R do not hold when we intuitively expect them to hold. Thus, we introduce a modality, '⊡' combining '!' and '□', to solve this problem.

---

[2] Precisely speaking, this can be shown as a consequence of the cut-elimination theorem of IMELL$^\square$, and the theorem was shown in the authors' previous work [6].

$$\frac{\lceil A \rceil}{\lceil p \rceil \overset{\text{def}}{=} p, \ \lceil A \supset B \rceil \overset{\text{def}}{=} (!\lceil A \rceil) \multimap \lceil B \rceil, \ \lceil \Box A \rceil \overset{\text{def}}{=} \boxdot \lceil A \rceil} \quad \Bigg| \quad \frac{\lceil \Gamma \rceil}{\lceil \Gamma \rceil \overset{\text{def}}{=} \{(x : \lceil A \rceil) \mid (x : A) \in \Gamma\}}$$

Figure 9 Definition of the S4-version of Girard translation.

Our modal linear logic, which is called IMELL$^{\boxdot}$, is defined by a sequent calculus which is given in Figure 8. As we mentioned, the formulae are defined as an extension of those of IMELL with the $\boxdot$-modality. A point is that the !-modality is still there with the $\boxdot$-modality.

The $\boxdot$-modality is defined so as to have properties of both '!' and '$\Box$', but '!' still behaves similarly to IMELL. Therefore, all the intuitions of the inference rules except !R and $\boxdot$R should be clear. The rules !R and $\boxdot$R reflect the "strength" between the modalities '!' and '$\boxdot$'. Indeed, '!' and '$\boxdot$' satisfy the S4 axiomata and '$\boxdot$' is stronger than '!'.

▶ **Example 3.** The following hold:
1. $\vdash !A \multimap A$ and $\vdash \boxdot A \multimap A$
2. $\vdash !(A \multimap B) \multimap !A \multimap !B$ and $\vdash \boxdot(A \multimap B) \multimap \boxdot A \multimap \boxdot B$
3. $\vdash !A \multimap !!A$ and $\vdash \boxdot A \multimap \boxdot\boxdot A$
4. $\vdash \boxdot A \multimap !A$ but $\nvdash !A \multimap \boxdot A$

▶ Remark 4. In Example 3, the first three represent the so-called S4 axiomata: *T*, *K*, and *4*. The last one represents the strength of the two modalities. Actually, assuming the !-modality and the $\boxdot$-modality to satisfy the S4 axiomata and the "strength" axiom $\vdash \boxdot A \multimap !A$ is enough to characterize our modal linear logic (see Section 5 for more details).

The cut-elimination theorem for IMELL$^{\boxdot}$ is shown similarly to the case of IMELL, and hence IMELL$^{\boxdot}$ is consistent. The addition of '$\boxdot$' causes no problems in the proof.

▶ **Definition 5** (Cut-degree and degree). *For an application of* Cut *in a proof, its* cut-degree *is defined to be the number of logical connectives in the cut-formula. The* degree *of a proof is defined to be the maximal cut-degree of the proof (and* 0 *if there is no application of* Cut*).*

▶ **Theorem 6** (Cut-elimination). *The rule* Cut *in* IMELL$^{\boxdot}$ *is admissible, i.e., if* $\Gamma \vdash A$ *is derivable, then there is a derivation of the same judgment without any applications of* Cut*.*

Proof. We follow the proof for propositional linear logic by Lincoln et al. [9]. To show the admissibility of Cut, we consider the admissibility of the following cut rules:

$$\frac{\Gamma \vdash !A \qquad \Gamma', (!A)^n \vdash B}{\Gamma, \Gamma' \vdash B} \ !\text{Cut} \qquad\qquad \frac{\Gamma \vdash \boxdot A \qquad \Gamma', (\boxdot A)^n \vdash B}{\Gamma, \Gamma' \vdash B} \ \boxdot\text{Cut}$$

where $(C)^n$ denotes the multiset that has n occurrences of C and n is assumed to be positive as a side-condition; and Γ' in !Cut (resp. in $\boxdot$Cut) is supposed to contain no formulae of form $!A$ (resp. $\boxdot A$). The *cut-degrees* of !Cut and $\boxdot$Cut are defined similarly to that of Cut.

Then, all the three rules (Cut, !Cut, $\boxdot$Cut) are shown to be admissible by simultaneous induction on the lexicographic complexity $\langle \delta, h \rangle$, where δ is the degree of the assumed derivation and h is its height. See the appendix for details of the proof. ◀

▶ **Corollary 7** (Consistency). IMELL$^{\boxdot}$ *is consistent, i.e., there exists an underivable judgment.*

Then, we can define an S4-version of Girard translation as in Figure 9, and it can be justified by the following theorem, which is readily shown by induction on the derivation.

▶ **Theorem 8** (Soundness). *If* $\Box\Delta, \Gamma \vdash A$ *in* LJ$^{\Box}$*, then* $\boxdot\lceil\Delta\rceil, !\lceil\Gamma\rceil \vdash \lceil A \rceil$ *in* IMELL$^{\boxdot}$*.*

Syntactic category	Reduction rule
Types $A, B, C ::= p \mid A \multimap B \mid !A \mid \boxdot A$	$(\beta \multimap)$ $(\lambda x : A.M)\, N \qquad \rightsquigarrow M[x := N]$
Terms $M, N, L ::= x \mid \lambda x : A.M \mid M\, N \mid !M \mid \boxdot M$	$(\beta !)$ $\quad \mathbf{let}\, !x =!N\, \mathbf{in}\, M \quad \rightsquigarrow M[x := N]$
$\mid \mathbf{let}\, !x = M\, \mathbf{in}\, N \mid \mathbf{let}\, \boxdot x = M\, \mathbf{in}\, N$	$(\beta\, \boxdot)$ $\mathbf{let}\, \boxdot x =\boxdot N\, \mathbf{in}\, M \rightsquigarrow M[x := N]$

Typing rule

$$\frac{}{\Delta; \Gamma; x : A \vdash x : A}\ \mathrm{LinAx} \qquad \frac{}{\Delta; \Gamma, x : A; \emptyset \vdash x : A}\ !\mathrm{Ax} \qquad \frac{}{\Delta, x : A; \Gamma; \emptyset \vdash x : A}\ \boxdot\mathrm{Ax}$$

$$\frac{\Delta; \Gamma; \Sigma, x : A \vdash M : B}{\Delta; \Gamma; \Sigma \vdash \lambda x : A.M : A \multimap B}\ \multimap\mathrm{I} \qquad \frac{\Delta; \Gamma; \Sigma \vdash M : A \multimap B \qquad \Delta; \Gamma; \Sigma' \vdash N : A}{\Delta; \Gamma; \Sigma, \Sigma' \vdash M\, N : B}\ \multimap\mathrm{E}$$

$$\frac{\Delta; \Gamma; \emptyset \vdash M : A}{\Delta; \Gamma; \emptyset \vdash !M\, :!A}\ !\mathrm{I} \qquad \frac{\Delta; \Gamma; \Sigma \vdash M\, :!A \qquad \Delta, \Gamma, x : A; \Sigma' \vdash N : B}{\Delta; \Gamma; \Sigma, \Sigma' \vdash \mathbf{let}\, !x = M\, \mathbf{in}\, N : B}\ !\mathrm{E}$$

$$\frac{\Delta; \emptyset; \emptyset \vdash M : A}{\Delta; \Gamma; \emptyset \vdash \boxdot M\, :\boxdot A}\ \boxdot\mathrm{I} \qquad \frac{\Delta; \Gamma; \Sigma \vdash M\, :\boxdot A \qquad \Delta, x : A; \Gamma; \Sigma' \vdash N : B}{\Delta; \Gamma; \Sigma, \Sigma' \vdash \mathbf{let}\, \boxdot x = M\, \mathbf{in}\, N : B}\ \boxdot\mathrm{E}$$

■ **Figure 10** Definition of $\lambda^{\boxdot}$.

4 Curry–Howard correspondence

In this section, we give a computational interpretation for our modal linear logic through the Curry–Howard correspondence and establish the corresponding S4-version of Girard translation for the modal linear logic in terms of typed λ-calculus.

4.1 Typed λ-calculus for the intuitionistic modal linear logic

We introduce $\lambda^{\boxdot}$ (pronounced by "lambda bangbox") that is a typed λ-calculus corresponding to the modal linear logic under the Curry–Howard correspondence. The calculus $\lambda^{\boxdot}$ can be seen as an integration of $\lambda^{\square}$ of Pfenning and Davies and the linear λ-calculus for dual intuitionistic linear logic of Barber [2]. The rules of $\lambda^{\boxdot}$ are designed considering the "necessity" of modal logic and the "linearity" of linear logic, and formally defined as in Figure 10.

The structure of types are exactly the same as that of formulae in IMELL$^{\boxdot}$. Terms are defined as an extension of the simply-typed λ-calculus with the following: the terms $!M$ and $\mathbf{let}\, !x = M\, \mathbf{in}\, N$, which are a constructor and a destructor for types $!A$, respectively; and the terms $\boxdot M$ and $\mathbf{let}\, \boxdot x = M\, \mathbf{in}\, N$, which are those for types $\boxdot A$ similarly. Note that the variable x in $\mathbf{let}\, !x = M\, \mathbf{in}\, N$ and $\mathbf{let}\, \boxdot x = M\, \mathbf{in}\, N$ is supposed to be *bound*.

A *(type) context* is defined by the same way as $\lambda^{\square}$ and a *(type) judgment* consists of three contexts, a term and a type, written as $\Delta; \Gamma; \Sigma \vdash M : A$. These three contexts of a judgment $\Delta; \Gamma; \Sigma \vdash M : A$ have the following intuitive meaning: (1) Δ implicitly represents a context for modalized types of form $\boxdot A$; (2) Γ implicitly represents a context for modalized types of form $\square A$; (3) Σ represents an ordinary context but its elements must be used linearly.

The intuitive meanings of the typing rules are as follows. Each of the first three rules is a variable rule depending on the context's kind. It is allowed for the Δ-part and the Γ-part to weaken the antecedent in these rules, but is not for the Σ-part since it must satisfy the linearity condition. The rules $\multimap \mathrm{I}$ and $\multimap \mathrm{E}$ are for the type $\multimap$, and again, the $\multimap \mathrm{E}$ is designed to satisfy the linearity. The remaining rules are for types $!A$ and $\boxdot A$.

The *reduction* $\rightsquigarrow$ is defined to be the least compatible relation on terms generated by $(\beta \multimap)$, $(\beta !)$, and $(\beta\, \boxdot)$. The *multistep reduction* $\rightsquigarrow^+$ is defined as in the case of $\lambda^{\square}$.

Then, we can show the subject reduction and the strong normalization of $\lambda^{\boxdot}$ as follows.

▶ **Lemma 9** (Substitution).
1. *If $\Delta; \Gamma; \Sigma, x : A \vdash M : B$ and $\Delta; \Gamma; \Sigma' \vdash N : A$, then $\Delta; \Gamma; \Sigma, \Sigma' \vdash M[x := N] : B$;*
2. *If $\Delta; \Gamma, x : A; \Sigma \vdash M : B$ and $\Delta; \Gamma; \emptyset \vdash N : A$, then $\Delta; \Gamma; \Sigma \vdash M[x := N] : B$;*
3. *If $\Delta, x : A; \Gamma; \Sigma \vdash M : B$ and $\Delta; \emptyset; \emptyset \vdash N : A$, then $\Delta; \Gamma; \Sigma \vdash M[x := N] : B$.*

▶ **Theorem 10** (Subject reduction). *If $\Delta; \Gamma; \Sigma \vdash M : A$ and $M \leadsto N$, then $\Delta; \Gamma; \Sigma \vdash N : A$.*

Proof. By induction on the derivation of $\Delta; \Gamma; \Sigma \vdash M : A$ together with Lemma 9.   ◀

▶ **Theorem 11** (Strong normalization). *For well-typed term M, there are no infinite reduction sequences starting from M.*

Proof. By embedding to a typed λ-calculus of the $(!, \multimap)$-fragment of dual intuitionistic linear logic, named $\lambda^{!,\multimap}$, which is shown to be strongly normalizing by Ohta and Hasegawa [16].

The details are in the appendix, but the intuition is described as follows. First, for every well-typed term M, we define the term $(M)^{\ddagger}$ by replacing the occurrences of $\Box N$ and **let** $\Box x = N$ **in** L in M with $!(N)^{\ddagger}$ and **let** $!x = (N)^{\ddagger}$ **in** $(L)^{\ddagger}$, respectively. Then, we can show that $(M)^{\ddagger}$ is typable in $\lambda^{!,\multimap}$, because the structure of '$\Box$' collapses to that of '$!$', and that the embedding $(-)^{\ddagger}$ preserves reductions. Therefore, $\lambda^{\Box}$ is strongly normalizing.   ◀

As we mentioned, we can view that $\lambda^{\Box}$ is indeed a typed λ-calculus for the intuitionistic modal linear logic. A natural deduction that corresponds to $\lambda^{\Box}$ is obtained as the "logical-part" of the calculus, and we can show that the natural deduction is equivalent to IMELL$^{\Box}$.

▶ **Definition 12** (Natural deduction). *A natural deduction for modal linear logic, called NJ$^{\Box}$, is defined to be one that is extracted from $\lambda^{\Box}$ by erasing term annotations.*

▶ **Fact 13** (Curry–Howard correspondence). *There is a one-to-one correspondence between NJ$^{\Box}$ and $\lambda^{\Box}$, which preserves provability/typability and proof-normalizability/reducibility.*

▶ **Lemma 14** (Judgmental reflection). *The following hold in NJ$^{\Box}$.*
1. $\Delta; \Gamma; \Sigma, !A \vdash B$ *if and only if* $\Delta; \Gamma, A; \Sigma \vdash B$;
2. $\Delta; \Gamma; \Sigma, \Box A \vdash B$ *if and only if* $\Delta, A; \Gamma; \Sigma \vdash B$.

▶ **Theorem 15** (Equivalence). $\Delta; \Gamma; \Sigma \vdash A$ *in* NJ$^{\Box}$ *if and only if* $\Box\Delta, !\Gamma, \Sigma \vdash A$ *in* IMELL$^{\Box}$.

Proof. By straightforward induction. Lemma 14 is used to show the if-part.   ◀

4.2   Embedding from the modal λ-calculus by Pfenning and Davies

We give a translation from Pfenning and Davies' $\lambda^{\square}$ to our $\lambda^{\Box}$. We also show that the translation preserves the reductions of $\lambda^{\square}$, and thus it can be seen as the S4-version of Girard translation on the level of proofs through the Curry–Howard correspondence.

To give the translation, we introduce two meta λ-terms in $\lambda^{\Box}$ to encode the function space $\supset$ of $\lambda^{\square}$. The simulation of reduction of $(\lambda x : A.M) N$ in $\lambda^{\square}$ can be shown readily.

▶ **Definition 16.** *Let M and N be terms such that $\Delta; \Gamma, x : A; \Sigma \vdash M : B$ and $\Delta; \Gamma; \emptyset \vdash N : A$. Then, $\overline{\lambda} x : A.M$ and $M\overline{@}N$ are defined as the terms $\lambda y : !A.$**let** $!x = y$ **in** M and $M(!N)$, respectively, where y is chosen to be fresh, i.e., it is a variable satisfying $y \notin (\mathrm{FV}(M) \cup \{x\})$.*

▶ **Lemma 17** (Derivable full-function space). *The following rules are derivable in $\lambda^{\Box}$:*

$$\frac{\Delta; \Gamma, x : A; \Sigma \vdash M : B}{\Delta; \Gamma; \Sigma \vdash (\overline{\lambda} x : A.M) : !A \multimap B} \qquad \frac{\Delta; \Gamma; \Sigma \vdash M : !A \multimap B \qquad \Delta; \Gamma; \emptyset \vdash N : A}{\Delta; \Gamma; \Sigma \vdash M\overline{@}N : B}$$

Moreover, it holds that $(\overline{\lambda} x : A.M)\overline{@}N \leadsto^{+} M[x := N]$ in $\lambda^{\Box}$.

$$
\begin{array}{l}
\underline{\lceil A\rceil} \\
\qquad\lceil p\rceil \stackrel{\mathrm{def}}{=} p \\[4pt]
\qquad\lceil A \supset B\rceil \stackrel{\mathrm{def}}{=} !\lceil A\rceil \multimap \lceil B\rceil \\[4pt]
\qquad\lceil \Box A\rceil \stackrel{\mathrm{def}}{=} \boxdot\lceil A\rceil \\[4pt]
\hline
\underline{\lceil \Gamma\rceil} \\
\qquad\lceil\Gamma\rceil \stackrel{\mathrm{def}}{=} \{(x:\lceil A\rceil) \mid (x:A)\in\Gamma\}
\end{array}
\qquad
\begin{array}{l}
\underline{\mathcal{T}\lceil M\rceil} \qquad\qquad \mathcal{T}\lceil x\rceil \stackrel{\mathrm{def}}{=} x \\[4pt]
\qquad \mathcal{T}\lceil \lambda x:A.M\rceil \stackrel{\mathrm{def}}{=} \overline\lambda x:\lceil A\rceil.\mathcal{T}\lceil M\rceil \\[4pt]
\qquad \mathcal{T}\lceil M\,N\rceil \stackrel{\mathrm{def}}{=} \mathcal{T}\lceil M\rceil\overline{@}\mathcal{T}\lceil N\rceil \\[4pt]
\qquad \mathcal{T}\lceil \Box M\rceil \stackrel{\mathrm{def}}{=} \boxdot\mathcal{T}\lceil M\rceil \\[4pt]
\mathcal{T}\lceil \mathbf{let}\,\Box x = M\,\mathbf{in}\,N\rceil \stackrel{\mathrm{def}}{=} \mathbf{let}\,\boxdot x = \mathcal{T}\lceil M\rceil\,\mathbf{in}\,\mathcal{T}\lceil N\rceil
\end{array}
$$

■ **Figure 11** Definitions of the S4-version of Girard translation in term of typed λ-calculus.

Together with the above meta λ-terms $\overline\lambda x:A.M$ and $M\overline{@}N$, we can define the translation from $\lambda^\Box$ into $\lambda^\boxdot$ and show that it preserves typability and reducibility.

▶ **Definition 18** (Translation). *The translation from $\lambda^\Box$ to $\lambda^\boxdot$ is defined to be the triple of the type/context/term translations $\lceil A\rceil$, $\lceil\Gamma\rceil$, and $\mathcal{T}\lceil M\rceil$ defined in Figure 11.*

▶ **Theorem 19** (Embedding). *$\lambda^\Box$ can be embedded into $\lambda^\boxdot$, i.e., the following hold:*
1. *If $\Delta;\Gamma \vdash M:A$ in $\lambda^\Box$, then $\lceil\Delta\rceil;\lceil\Gamma\rceil;\emptyset \vdash \mathcal{T}\lceil M\rceil:\lceil A\rceil$ in $\lambda^\boxdot$.*
2. *If $M \rightsquigarrow M'$ in $\lambda^\Box$, then $\mathcal{T}\lceil M\rceil \rightsquigarrow^+ \mathcal{T}\lceil M'\rceil$ in $\lambda^\boxdot$.*

Proof. By induction on the derivation of $\Delta;\Gamma \vdash M:A$ and $M \rightsquigarrow M'$ in $\lambda^\Box$, respectively. ◀

From the logical point of view, Theorem 19.1 can be seen as another S4-version of Girard translation (in the style of natural deduction) that corresponds to Theorem 8; and Theorem 19.2 gives a justification that the S4-version of Girard translation is correct with respect to the level of proofs, i.e., it preserves proof-normalizations as well as provability.

5 Axiomatization of modal linear logic

We give an axiomatic characterization of the intuitionistic modal linear logic. To do so, we define a typed combinatory logic, called $\mathrm{CL}^\boxdot$, which can be seen as a Hilbert-style deductive system of modal linear logic through the Curry–Howard correspondence. In this section, we only aim to provide the equivalence between $\mathrm{NJ}^\boxdot$ and the Hilbert-style, while $\mathrm{CL}^\boxdot$ satisfies several desirable properties, e.g., the subject reduction and the strong normalizability.

The definition of $\mathrm{CL}^\boxdot$ is given in Figure 12. The set of types has the same structure as that in $\lambda^\boxdot$. A term is either a *variable*, a *combinator*, a necessitated term by '!', or a necessitated term by '$\boxdot$'. The notions of *(type) context* and *(type) judgment* are defined similarly to those of $\lambda^\boxdot$.

Every combinator c has its type as defined in the list in the figure, and is denoted by typeof(c). Then, the typing rules are described as follows: Ax and MP are the standard rules, which logically correspond to an axiom rule of the set of axiomata, and *modus ponens*, respectively. The others are defined by the same way as in $\lambda^\boxdot$.

The *reduction* $\rightsquigarrow$ of combinators is defined to be the least compatible relation on terms generated by the reduction rules listed in the figure.

▶ Remark 20. $\mathrm{CL}^\boxdot$ can be seen as an extension of *linear combinatory algebra* of Abramsky et al. [1] with the $\boxdot$-modality, or equivalently, a linear-logical reconstruction of Pfenning's modally-typed combinatory logic [17]. The combinators $\mathrm{T}^!$, $\mathrm{D}^!$, $4^!$ represent the S4 axiomata for the !-modality, and similarly, $\mathrm{T}^\boxdot$, $\mathrm{D}^\boxdot$, $4^\boxdot$ represent those for the $\boxdot$-modality. E is the only one combinator to characterize the strength between the two modalities.

As we defined $\mathrm{NJ}^\boxdot$ from $\lambda^\boxdot$, we can define the Hilbert-style deductive system (with open assumptions) for the intuitionistic modal linear logic via $\mathrm{CL}^\boxdot$.

Syntactic category

Types $A, B, C ::= p \mid A \multimap B \mid !A \mid \Box A$

Terms $M, N, L ::= x \mid c \mid !M \mid \Box M$

Typing rule

$$\frac{(c \text{ is a combinator})}{\Delta; \Gamma; \emptyset \vdash c : \text{typeof}(c)} \text{Ax} \qquad \frac{\Delta; \Gamma; \Sigma \vdash M : A \multimap B \qquad \Delta; \Gamma; \Sigma' \vdash N : A}{\Delta; \Gamma; \Sigma, \Sigma' \vdash M\,N : B} \text{MP}$$

$$\frac{}{\Delta; \Gamma; x : A \vdash x : A} \text{LinAx} \qquad \frac{}{\Delta; \Gamma, x : A; \emptyset \vdash x : A} !\text{Ax} \qquad \frac{}{\Delta, x : A; \Gamma; \emptyset \vdash x : A} \Box\text{Ax}$$

$$\frac{\Delta; \Gamma; \emptyset \vdash M : A}{\Delta; \Gamma; \emptyset \vdash !M : !A} ! \qquad\qquad \frac{\Delta; \emptyset; \emptyset \vdash M : A}{\Delta; \Gamma; \emptyset \vdash \Box M : \Box A} \Box$$

Combinator	**Reduction**
▪ $\vdash \text{I} : A \multimap A$	$\text{I}\,M \qquad\qquad \rightsquigarrow\ M$
▪ $\vdash \text{B} : (B \multimap C) \multimap (A \multimap B) \multimap A \multimap C$	$\text{B}\,M\,N\,L \qquad \rightsquigarrow\ M\,(N\,L)$
▪ $\vdash \text{C} : (A \multimap B \multimap C) \multimap B \multimap A \multimap C$	$\text{C}\,M\,N\,L \qquad \rightsquigarrow\ M\,L\,N$
▪ $\vdash \text{S}^\delta : (\delta A \multimap B \multimap C) \multimap (\delta A \multimap B) \multimap \delta A \multimap C$	$\text{S}^\delta\,M\,N\,(\delta L) \quad \rightsquigarrow\ M\,(\delta L)\,(N\,(\delta L))$
▪ $\vdash \text{K}^\delta : A \multimap \delta B \multimap A$	$\text{K}^\delta\,M\,(\delta N) \qquad \rightsquigarrow\ M$
▪ $\vdash \text{W}^\delta : (\delta A \multimap \delta A \multimap B) \multimap \delta A \multimap B$	$\text{W}^\delta\,M\,(\delta N) \qquad \rightsquigarrow\ M\,(\delta N)\,(\delta N)$
▪ $\vdash \text{T}^\delta : \delta A \multimap A$	$\text{T}^\delta\,(\delta M) \qquad\quad \rightsquigarrow\ M$
▪ $\vdash \text{D}^\delta : \delta(A \multimap B) \multimap \delta A \multimap \delta B$	$\text{D}^\delta\,(\delta M)\,(\delta N) \rightsquigarrow\ \delta(M\,N)$
▪ $\vdash 4^\delta : \delta A \multimap \delta\delta A$	$4^\delta\,(\delta M) \qquad\qquad \rightsquigarrow\ \delta\delta M$
▪ $\vdash \text{E} : \Box A \multimap !A$	$\text{E}\,\Box M \qquad\qquad \rightsquigarrow\ !M$
where $\delta \in \{!, \Box\}$	where $\delta \in \{!, \Box\}$

▪ **Figure 12** Definition of $\text{CL}^\Box$.

▶ **Definition 21** (Hilbert-style). *A Hilbert-style deductive system for modal linear logic, called* $\text{HJ}^\Box$, *is defined to be one that is extracted from* $\text{CL}^\Box$ *by erasing term annotations.*

▶ **Fact 22** (Curry–Howard correspondnece). *There is a one-to-one correspondence between* $\text{HJ}^\Box$ *and* $\text{CL}^\Box$, *which preserves provability/typability and proof-normalizability/reducibility.*

The deduction theorem of $\text{HJ}^\Box$ can be obtained as a consequence of the so-called *bracket abstraction* of $\text{CL}^\Box$ through Fact 22, which allows us to show the equivalence between $\text{HJ}^\Box$ and $\text{NJ}^\Box$. Therefore, the modal linear logic is indeed axiomatized by $\text{HJ}^\Box$.

▶ **Theorem 23** (Deduction theorem).
1. *If* $\Delta; \Gamma; \Sigma, x : A \vdash M : B$, *then* $\Delta; \Gamma; \Sigma \vdash (\lambda_* x.M) : (A \multimap B)$;
2. *If* $\Delta; \Gamma, x : A; \Sigma \vdash M : B$, *then* $\Delta; \Gamma; \Sigma \vdash (\lambda_*^! x.M) : (!A \multimap B)$;
3. *If* $\Delta, x : A; \Gamma; \Sigma \vdash M : B$, *then* $\Delta; \Gamma; \Sigma \vdash (\lambda_*^\Box x.M) : (\Box A \multimap B)$.
where $(\lambda_* x.M)$, $(\lambda_*^! x.M)$, $(\lambda_*^\Box x.M)$ *are bracket abstraction operations that take a variable* x *and a* $\text{CL}^\Box$*-term* M *and returns a* $\text{CL}^\Box$*-term, and the definitions are given in the appendix.*

Proof. By induction on the derivation. The proof is just a type-checking of the result of the bracket abstraction operations. ◀

▶ **Theorem 24** (Equivalence). $\Delta; \Gamma; \Sigma \vdash A$ *in* $\text{HJ}^\Box$ *if and only if* $\Delta; \Gamma; \Sigma \vdash A$ *in* $\text{NJ}^\Box$.

Proof. By straightforward induction. We use Theorem 23 and Fact 22 to show the if-part. ◀

▶ **Corollary 25.** $\text{IMELL}^\Box$, $\text{NJ}^\Box$, *and* $\text{HJ}^\Box$ *are equivalent with respect to provability.*

Syntactic category	Inference rule
Formulae $A, B, C ::= p \mid p^{\perp} \mid A \otimes B \mid A \,\mathbin{\rotatebox[origin=c]{180}{\&}}\, B \mid {!}A \mid {?}A \mid \boxdot A \mid \lozenge A$	$\dfrac{}{\vdash A^{\perp}, A}\,\text{Ax}$

$$\dfrac{\vdash \Gamma, A \quad \vdash A^{\perp}, \Gamma'}{\vdash \Gamma, \Gamma'}\,\text{Cut} \quad \dfrac{\vdash \Gamma, A \quad \vdash \Gamma', B}{\vdash \Gamma, \Gamma', A \otimes B}\,\otimes \quad \dfrac{\vdash \Gamma, A, B}{\vdash \Gamma, A \,\mathbin{\rotatebox[origin=c]{180}{\&}}\, B}\,\mathbin{\rotatebox[origin=c]{180}{\&}} \quad \dfrac{\vdash \lozenge\Delta, {?}\Gamma, A}{\vdash \lozenge\Delta, {?}\Gamma, {!}A}\,! \quad \dfrac{\vdash \Gamma, A}{\vdash \Gamma, {?}A}\,?$$

$$\dfrac{\vdash \lozenge\Delta, A}{\vdash \lozenge\Delta, \boxdot A}\,\boxdot \quad \dfrac{\vdash \Gamma, A}{\vdash \Gamma, \lozenge A}\,\lozenge \quad \dfrac{\vdash \Gamma}{\vdash \Gamma, {?}A}\,{?}\text{W} \quad \dfrac{\vdash \Gamma, {?}A, {?}A}{\vdash \Gamma, {?}A}\,{?}\text{C} \quad \dfrac{\vdash \Gamma}{\vdash \Gamma, \lozenge A}\,\lozenge\text{W} \quad \dfrac{\vdash \Gamma, \lozenge A, \lozenge A}{\vdash \Gamma, \lozenge A}\,\lozenge\text{C}$$

Figure 13 Definition of $\text{CMELL}^{\boxdot}$.

6 Geometry of Interaction Machine

In this section, we show a dynamic semantics, called *context semantics*, for the modal linear logic in the style of geometry of interaction machine [10, 11]. As in the usual linear logic, we first define a notion of *proof net* and then define the machine as a token-passing system over those proof nets. Thanks to the simplicity of our logic, the definitions are mostly straightforward extension of those for classical MELL (CMELL).

6.1 Sequent calculus for classical modal linear logic

We define a sequent calculus of classical modal linear logic, called $\text{CMELL}^{\boxdot}$. The reason why we define it in the classical setting is for ease of defining the proof nets in the latter part.

Figure 13 shows the definition of $\text{CMELL}^{\boxdot}$. The set of formulae are defined as an extension of CMELL-formulae with the two modalities '$\boxdot$' and '$\lozenge$'. A *dual formula* of A, written $A^{\perp}$, is defined by the standard dual formulae in CMELL along with $(\boxdot A)^{\perp} \stackrel{\text{def}}{=} \lozenge(A^{\perp})$ and $(\lozenge A)^{\perp} \stackrel{\text{def}}{=} \boxdot(A^{\perp})$. Here, the $\lozenge$-modality is the dual of the $\boxdot$-modality by definition, and it can be seen as an integration of the ?-modality and the $\Diamond$-modality. The *linear implication* $A \multimap B$ is defined as $A^{\perp} \,\mathbin{\rotatebox[origin=c]{180}{\&}}\, B$ as usual. The inference rules are defined as a simple extension of $\text{IMELL}^{\boxdot}$ to the classical setting in the style of "one-sided" sequent.

Then, the cut-elimination theorem for $\text{CMELL}^{\boxdot}$ can be shown similarly to the case of $\text{IMELL}^{\boxdot}$, and we can see that there exists a trivial embedding from $\text{IMELL}^{\boxdot}$ to $\text{CMELL}^{\boxdot}$.

▶ **Theorem 26** (Cut-elimination)**.** *The rule* Cut *in* $\text{CMELL}^{\boxdot}$ *is admissible.*

▶ **Theorem 27** (Embedding)**.** *If* $\Gamma \vdash A$ *in* $\text{IMELL}^{\boxdot}$*, then* $\vdash \Gamma^{\perp}, A$ *in* $\text{CMELL}^{\boxdot}$*.*

6.2 Proof-nets formalization

First, we define *proof structures* for $\text{CMELL}^{\boxdot}$. The *proof nets* are then defined to be those proof structures satisfying a condition called *correctness criterion*. Intuitively, a proof net corresponds to an (equivalence class of) proof in $\text{CMELL}^{\boxdot}$.

▶ **Definition 28.** *A* node *is one of the graph-theoretic node shown in Figure 14 equipped with* $\text{CMELL}^{\boxdot}$ *types on the edges. They are all directed from top to bottom: for example, the* $\mathbin{\rotatebox[origin=c]{180}{\&}}$ *node has two incoming edges and one outgoing edge. A* !-*node (resp.* $\boxdot$-*node) has one outgoing edge typed by* $!A$ *(resp.* $\boxdot A$*) and arbitrarily many (possibly zero) outgoing edges typed by* $?A_i$ *and* $\lozenge B_i$ *(resp.* $\lozenge A_i$*).*
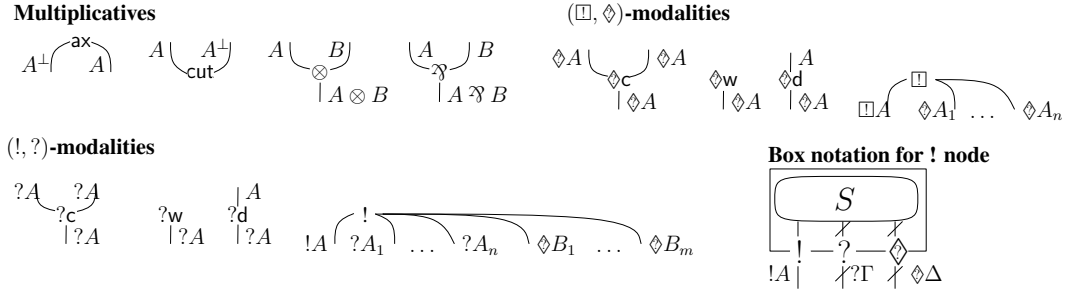
Figure 14 Nodes of proof net and box notation.

A proof structure *is a finite directed graph that satisfies the following conditions:*

- *each edge is with a type that matches the types specified by the nodes (in Figure 14) it is connected to;*
- *some edges may not be connected to any node (called* dangling edges*). Those dangling edges and also the types on those edges are called the* conclusions of the structure*;*
- *the graph is associated with a total map from all the !-nodes and □-nodes in it to proof structures called the* contents *of the* !*/*□*-nodes. The map satisfies that the types of the conclusions of a !-node (resp. □-node) coincide with the conclusions of its content.*

▶ **Remark 29.** Formally, a !-node (resp. a □-node) and its content are distinctive objects and they are not connected as a directed graph. Though, it is convenient to depict them as if the !-node (resp. □-node) represents a "box" filled with its content, as shown at the bottom-right of Figure 14. We also depict multiple edges by an edge with a diagonal line. In what follows, we adopt this "box" notation and multiple edges notation without explicit note.

▶ **Definition 30.** *Given a proof structure S, a* switching path *is an undirected path on S (meaning that the path is allowed to traverse an edge forward or backward) satisfying that on each $\invamp$ node, $?c$ node, and $\Diamond c$ node, the path uses at most one of the premises, and that the path uses any edge at most once.*

▶ **Definition 31.** *The* correctness criterion *is the following condition: given a proof structure S, switching paths of S and all contents of !-nodes, □-nodes in S are all acyclic and connected. A proof structure satisfying the correctness criterion is called a* proof net*.*

As a counterpart of cut-elimination process in CMELL$^\square$, the notion of *reduction* is defined for proof structures (and hence for proof nets): this intuition is made precise by Lemma 34 where $(-)^\bullet$ is the translation from CMELL$^\square$ to proof nets, whose definition is omitted here since it is defined analogously to that of CMELL and CMELL proof net. The lemmata below are naturally obtained by extending the case for CMELL since the □-modality has mostly the same logical structure as the !-modality.

▶ **Definition 32.** Reductions *of proof structures are local graph reductions defined by the set of rules depicted in Figure 15.*

▶ **Lemma 33.** *Let $S \to S'$ be a reduction between proof structures. If S is a proof net (i.e., satisfies the correctness criterion), so is S'.*

▶ **Lemma 34.** *Let Π be a proof of $\vdash \Diamond\Delta^\perp, ?\Gamma^\perp, \Sigma^\perp, A$ and suppose that Π reduces to another proof Π'. Then there is a sequence of reductions $(\Pi)^\bullet \to^* (\Pi')^\bullet$ between the proof nets.*
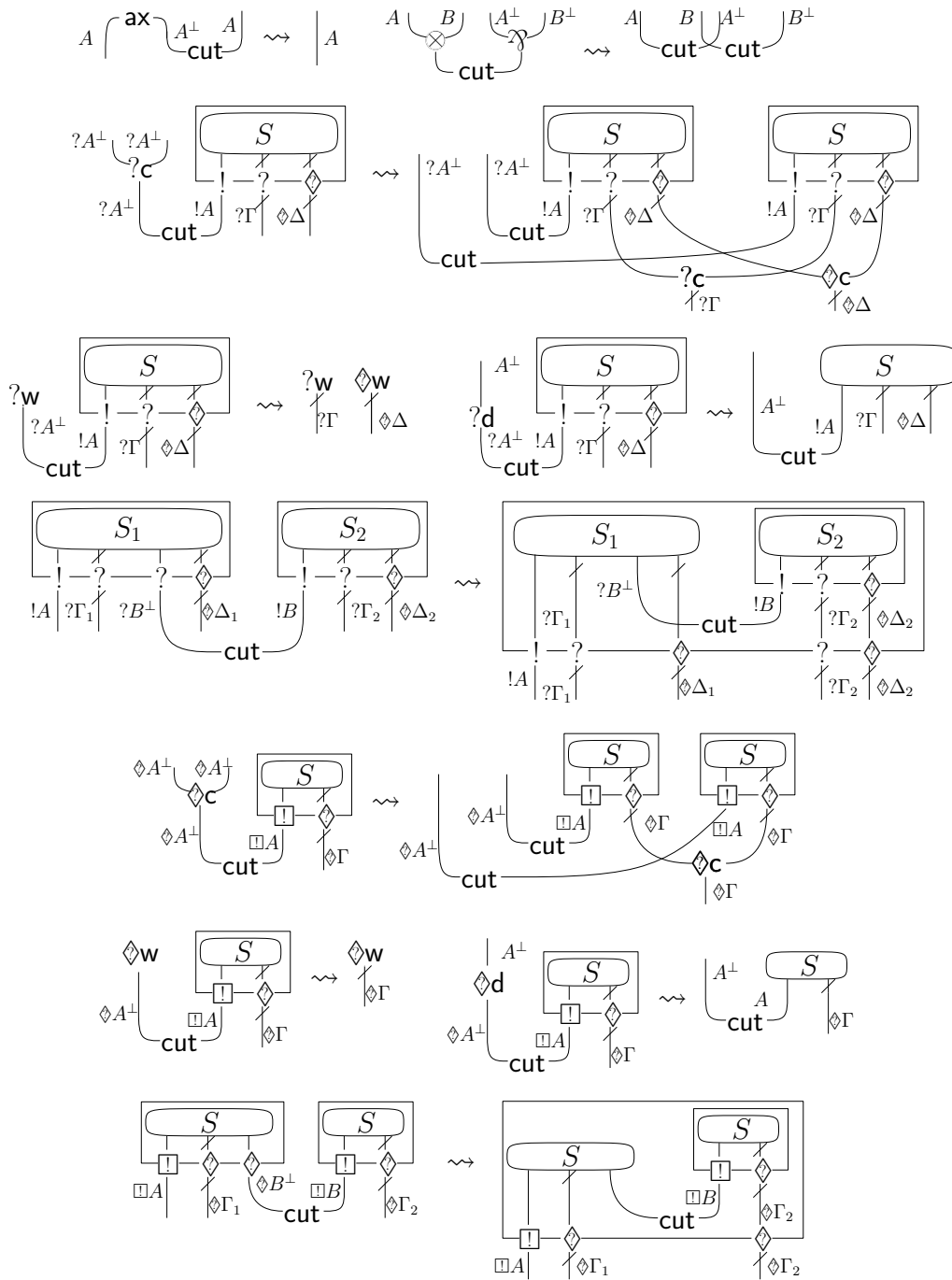
Figure 15 Reduction rules.

6.3   Computational interpretation

▶ **Definition 35.** *A* context *is a triple* $(\mathcal{M}, \mathcal{B}, \mathcal{N})$ *where* $\mathcal{M}, \mathcal{B}, \mathcal{N}$ *are generated by the following grammar:*

$$\mathcal{M} ::= \varepsilon \mid l.\mathcal{M} \mid r.\mathcal{M} \quad \mathcal{B} ::= \varepsilon \mid L.\mathcal{B} \mid R.\mathcal{B} \mid \langle \mathcal{B}, \mathcal{B} \rangle \mid \star \quad \mathcal{N} ::= \varepsilon \mid L'.\mathcal{N} \mid R'.\mathcal{N} \mid \langle \mathcal{N}, \mathcal{N} \rangle \mid \star$$

The intuition of a context is an intermediate state while "evaluating" the proof net (and, by translating into a proof net, a term in $\lambda^{\square}$). The geometry of interaction machine calculates the semantic value of a net by traversing the net from a conclusion to another; to traverse the net in a "right way" (more precisely, in a way invariant under net reduction), the context accumulates the information about the path that is already passed. Then, *how* the net is traversed is defined by the notion of *path* over a proof net as we define below.

▶ **Definition 36.** *The* extended dynamic algebra $\Lambda^{\square *}$ *is a single-sorted* Σ *algebra that contains* $0, 1, p, q, r, r', s, s', t, t', d, d' \colon \Sigma$ *as constants, has an associative operator* $\cdot \colon \Sigma \times \Sigma \to \Sigma$ *and operators* $(-)^* \colon \Sigma \to \Sigma, ! \colon \Sigma \to \Sigma, \square \colon \Sigma \to \Sigma$, *equipped with a formal sum* $+$, *and satisfies the equations below. Hereafter, we write* xy *for* $x \cdot y$ *where* x *and* y *are metavariables over* Σ.

$$
\begin{array}{llll}
0^* = !0 = 0 & 1^* = !1 = 1 & 0x = x0 = 0 & 1x = x1 = x \\
!(x)^* = !(x^*) & (xy)^* = y^* x^* & (x^*)^* = x & !(x)!(y) = !(xy) \\
\square(x)\,\square(y) = \square(xy) & p^* p = q^* q = 1 & q^* p = p^* q = 0 & r^* r = s^* s = 1 \\
s^* r = r^* s = 0 & d^* d = 1 & t^* t = 1 & p'^* p' = q'^* q' = 1 \\
q'^* p' = p'^* q' = 0 & r'^* r' = s'^* s' = 1 & s'^* r' = r'^* s' = 0 & d'^* d' = 1 \\
t'^* t' = 1 & !(x)r = r!(x) & !(x)s = s!(x) & !(x)t = t!!(x) \\
!(x)d = dx & \square(x)r' = r'\,\square(x) & \square(x)s' = s'\,\square(x) & \square(x)t' = t'\,\square\square(x) \\
\square(x)d' = d'x & x + y = y + x & x + 0 = x & (x+y)z = xz + yz \\
z(x+y) = zx + zy & (x+y)^* = x^* + y^* & !(x+y) = !x + !y & \square(x+y) = \square x + \square y
\end{array}
$$

▶ Remark 37. The equations in the definition above are mostly the same as the standard dynamic algebra Λ^* [10, 11] except those equations concerning the symbols with $'$ and the operator $\square$, and their structures are analogous to those for $!$ operator. This again reflects the fact that the logical structure of rules for $\square$ is analogous to that of $!$.

▶ **Definition 38.** *A* label *is an element of* $\Lambda^{\square *}$ *that is associated to edges of proof structures as in Figure 16. Let* S *be a proof structure and* T_S *be the set of edge traversals in the structure.* S *is associated with a function* $w \colon T_S \to \Lambda^{\square *}$ *defined by* $w(e) = l$ *(resp.* l^**) if* e *is a forward (resp. backward) traversal of an edge* e *and* l *is the label of the edge;* $w(e_1 e_2) = w(e_1)w(e_2)$.
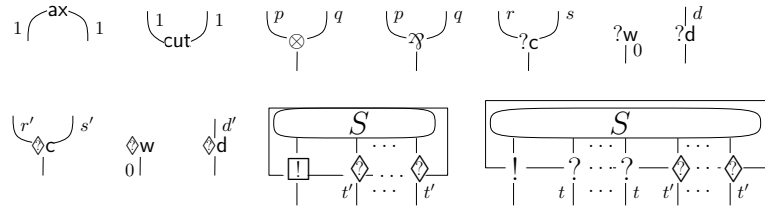
Figure 16 Labels on edges.

▶ **Definition 39.** *A* walk *over a proof structure S is an element of $\Lambda^{\square *}$ that is obtained by concatenating labels along a graph-theoretic path over S such that the graph-theoretic path does not traverse an edge forward (resp. backward) immediately after the same edge backward (resp. forward); and does not traverse a premise of one of $\otimes, \mathbin{⅋}, c$ node and another premise of the same node immediately after that. A* path *is a walk that is not proved to be equal to 0. A path is called* maximal *if it starts and finishes at a conclusion.*

The intuition of the notion of path is that a path is a "correct way" of traversing a proof net, in the sense that any path is preserved before and after a reduction. All the other walks that are not paths will be broken, which is represented by the constant 0 of $\Lambda^{\square *}$. Then, we obtain a *context semantics* from paths in the following way.

▶ **Definition 40.** *Given a monomial path a, its action $[\![a]\!] : \Sigma \rightharpoonup \Sigma$ on contexts is defined as follows. We define $[\![1]\!]$ as the identity mapping on contexts. There is no definition of $[\![0]\!]$. The $[\![f^*]\!]$ is the inverse translation, i.e., $[\![f]\!]^{-1}$. The transformer of the composition of a and b is defined as $[\![ab]\!](m) \stackrel{\text{def}}{=} [\![a]\!]([\![b]\!](m))$. For the other labels, the interpretation are defined as follows where exponential morphisms $!$ and $\square$ are defined by the meta-level pattern matchings:*

$$[\![p]\!](\mathcal{M},\mathcal{B},\mathcal{N}) \stackrel{\text{def}}{=} (l.\mathcal{M},\mathcal{B},\mathcal{N}) \qquad\qquad [\![q]\!](\mathcal{M},\mathcal{B},\mathcal{N}) \stackrel{\text{def}}{=} (r.\mathcal{M},\mathcal{B},\mathcal{N})$$

$$[\![r]\!](\mathcal{M},\mathcal{B},\mathcal{N}) \stackrel{\text{def}}{=} (\mathcal{M},L.\mathcal{B},\mathcal{N}) \qquad\qquad [\![s]\!](\mathcal{M},\mathcal{B},\mathcal{N}) \stackrel{\text{def}}{=} (\mathcal{M},R.\mathcal{B},\mathcal{N})$$

$$[\![t]\!](\mathcal{M},\langle\mathcal{B}_1,\langle\mathcal{B}_2,\mathcal{B}_3\rangle\rangle,\mathcal{N}) \stackrel{\text{def}}{=} (\mathcal{M},\langle\langle\mathcal{B}_1,\mathcal{B}_2\rangle,\mathcal{B}_3\rangle,\mathcal{N}) \quad [\![d]\!](\mathcal{M},\mathcal{B},\mathcal{N}) \stackrel{\text{def}}{=} (\mathcal{M},\star.\mathcal{B},\mathcal{N})$$

$$[\![r']\!](\mathcal{M},\mathcal{B},\mathcal{N}) \stackrel{\text{def}}{=} (\mathcal{M},\mathcal{B},L'.\mathcal{N}) \qquad\qquad [\![s']\!](\mathcal{M},\mathcal{B},\mathcal{N}) \stackrel{\text{def}}{=} (\mathcal{M},\mathcal{B},R'.\mathcal{N})$$

$$[\![t']\!](\mathcal{M},\mathcal{B},\langle\mathcal{N}_1,\langle\mathcal{N}_2,\mathcal{N}_3\rangle\rangle) \stackrel{\text{def}}{=} (\mathcal{M},\mathcal{B},\langle\langle\mathcal{N}_1,\mathcal{N}_2\rangle,\mathcal{N}_3\rangle) \quad [\![d']\!](\mathcal{M},\mathcal{B},\mathcal{N}) \stackrel{\text{def}}{=} (\mathcal{M},\mathcal{B},\star.\mathcal{N})$$

$$[\![!(f)]\!](\mathcal{M},\langle\mathcal{B}_1,\mathcal{B}_2\rangle,\mathcal{N}) \stackrel{\text{def}}{=} \textbf{\textit{let }} (\mathcal{M}',\mathcal{B}'_2,\mathcal{N}') = [\![f]\!](\mathcal{M},\mathcal{B}_2,\mathcal{N}) \textbf{ \textit{in }} (\mathcal{M}',\langle\mathcal{B}_1,\mathcal{B}'_2\rangle,\mathcal{N}')$$

$$[\![\square(f)]\!](\mathcal{M},\mathcal{B},\langle\mathcal{N}_1,\mathcal{N}_2\rangle) \stackrel{\text{def}}{=} \textbf{\textit{let }} (\mathcal{M}',\mathcal{B}',\mathcal{N}'_2) = [\![f]\!](\mathcal{M},\mathcal{B},\mathcal{N}_2) \textbf{ \textit{in }} (\mathcal{M}',\mathcal{B}',\langle\mathcal{N}_1,\mathcal{N}'_2\rangle)$$

Given a path a, its action $[\![a]\!] : \Sigma \rightharpoonup \mathbb{M}(\Sigma)$ is defined by the rules above (regarding the codomain as a multiset) and $[\![a + b]\!](m) = ([\![a]\!](m)) \uplus ([\![b]\!](m))$ where $\uplus$ is the multiset sum.

▶ **Remark 41.** In Mackie's work [10], the multiset in the codomain is not used since the main interest of his work is on terms of a base type: in that setting any proof net corresponding to a term has an execution formula that is monomial. In general, this style of context semantics is slightly degenerated compared to Girard's original version and its successors because the information of "current position" is dropped from the definition of contexts.

▶ **Definition 42.** *Let S be a closed proof net and χ be the set of maximal paths between conclusions of S. The* execution formula *is defined by $\mathcal{EX}(S) = \Sigma_{\phi \in \chi}\phi$ where the RHS is the sum of all paths in χ. The* context semantics *of S is defined to be $[\![\mathcal{EX}(S)]\!] : \Sigma \rightharpoonup \mathbb{M}(\Sigma)$.*

▶ **Definition 43.** *Let M be a closed well-typed term in $\lambda^{\square}$. The* context semantics *of M is defined to be $[\![(M)^\dagger]\!]$, where $(-)^\dagger$ is a straightforward translation from $\lambda^{\square}$-terms to proof nets, defined by constructing proof nets from $\lambda^{\square}$-derivations as in Figure 17.*

▶ **Lemma 44.** *Let S be a closed proof net and S' be its normal form. Then $[\![S]\!] = [\![S']\!]$.*

The lemma is proved through two auxiliary lemmata below.

▶ **Lemma 45.** *Let ϕ be a path from a conclusion of a closed net S ending at a node a. Let $(\mathcal{M}',\mathcal{B}',\mathcal{N}') = [\![\phi]\!](\mathcal{M},\varepsilon,\varepsilon)$. The height of $\mathcal{B}'$ (resp. $\mathcal{N}'$) matches with the number of exponential (resp. necessitation) boxes containing the node a.*
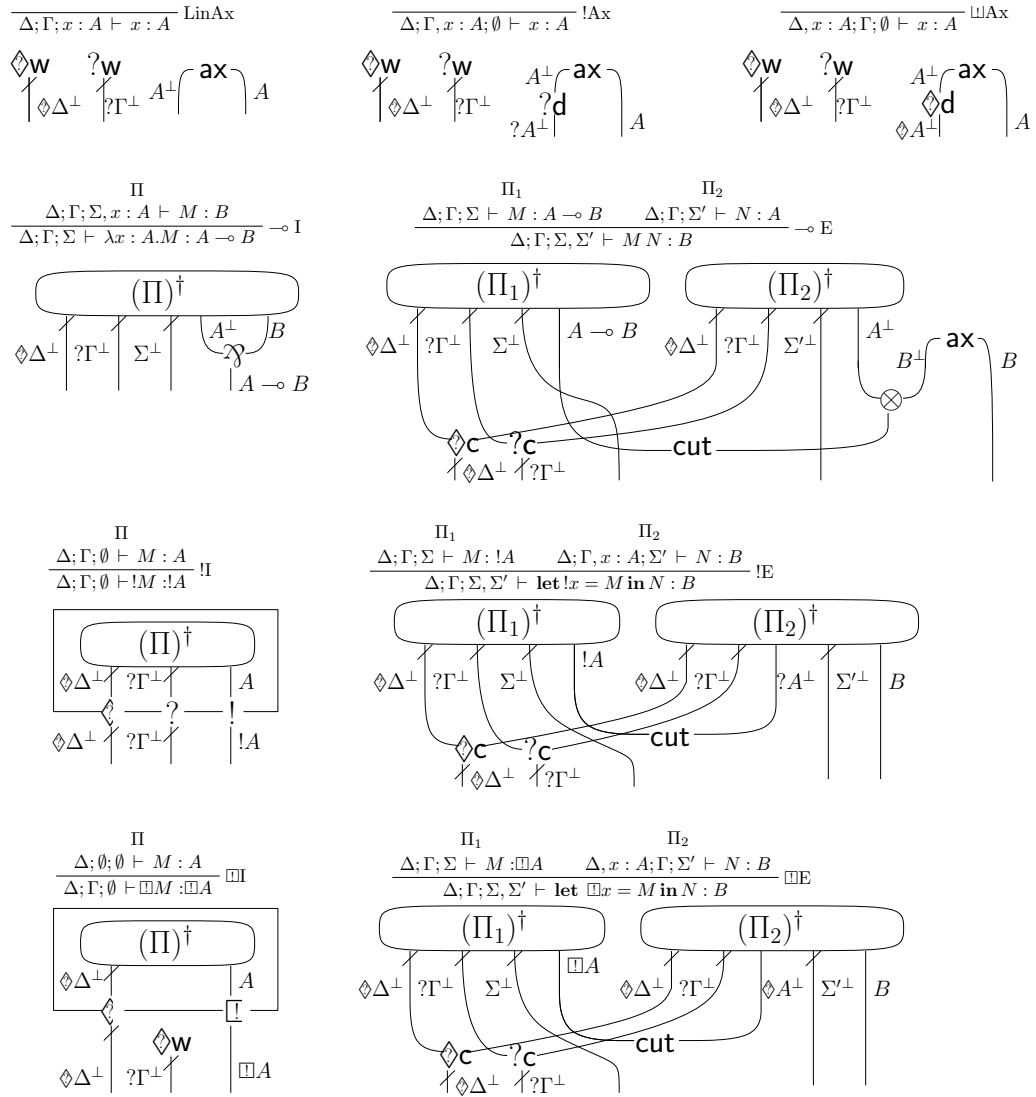
Figure 17 Translation from $\lambda^{\square}$ to CMELL$^{\square}$ proof nets.

Proof. By spectating the rules of actions above: the height of stacks only changes at doors of a box. ◀

▶ **Lemma 46.** *Let ϕ be a path inside a box of a closed net S. $[\![\phi]\!](\mathcal{M}, \sigma.\mathcal{B}, \tau.\mathcal{N})$ is in the form $(\mathcal{M}', \sigma'.\mathcal{B}, \tau.\mathcal{N})$.*

Proof. Again, by spectating the rules of actions. ◀

▶ **Theorem 47.** *If a closed term M in $\lambda^{\square}$ is typable and $M \rightsquigarrow M'$, then $[\![(M)^{\dagger}]\!] = [\![(M')^{\dagger}]\!]$.*

▶ Remark 48. This notion of context semantics inherently captures the "dynamics" of computation, and indeed Mackie exploited [10, 11] the character to implement a compiler, in the level of machine code, for PCF. In this paper we do not cover such a concrete compiler, but the definition of $[\![-]\!]$ can be seen as "context transformers" of virtual machine that is mathematically rigorous enough to model the computation of $\lambda^{\square}$ (and hence of $\lambda^{\square}$).

7 Related work

7.1 Linear-logical reconstruction of modal logic

The work on translations from modal logic to linear logic goes back to Martini and Masini [13]. They proposed a translation from classical S4 (CS4) to full propositional linear logic by means of the Grisin–Ono translation. However, their work only discusses provability.

The most similar work to ours is a "linear analysis" of CS4 by Schellinx [20], in which Girard translation from CS4 with respect to proofs is proposed. He uses a bi-colored linear logic, a subsystem of multicolored linear logic by Danos et al. [4], called **2-LL**, for a target calculus of the translation. It has two pairs of exponentials $\langle \underset{0}{!}, \underset{0}{?} \rangle$ and $\langle \underset{1}{!}, \underset{1}{?} \rangle$, called *subexponentials* following the terminology by Nigam and Miller [14], with the following rules:

$$\frac{\underset{1}{!}\Gamma, \underset{0}{!}\Gamma' \vdash A, \underset{1}{?}\Delta, \underset{0}{?}\Delta'}{\underset{1}{!}\Gamma, \underset{0}{!}\Gamma' \vdash \underset{0}{!}A, \underset{1}{?}\Delta, \underset{0}{?}\Delta'} \; \underset{0}{!}\text{R} \qquad\qquad \frac{\underset{1}{!}\Gamma \vdash A, \underset{1}{?}\Delta}{\underset{1}{!}\Gamma \vdash \underset{1}{!}A, \underset{1}{?}\Delta} \; \underset{1}{!}\text{R}$$

These rules have, while they are defined in the classical setting, essentially the same structure to what we defined as !R and $\square$R for IMELL$^\square$, respectively.

To mention the difference between the results of Schellinx and ours, his work has investigated only in terms of proof theory. Neither a typed λ-calculus nor a Geometry of Interaction interpretation was given. However, even so, he already gave a reduction-preserving Girard translation for the sequent calculi of CS4 and **2-LL**, and his *linear decoration* (cf. [20, 4]) allows us to obtain the cut-elimination theorem for CS4 as a corollary of that of **2-LL**. Thus, it should be interesting to investigate a relationship between his work and ours.

Furthermore, there also exist two uniform logical frameworks that can encode various logics including IS4 and CS4. One is the work by Nigam et al. [15] which based on Nigam and Miller's linear-logical framework with subexponentials and on the notion of focusing by Andreoli. The other work is *adjoint logic* by Pruiksma et al. [19] which based on, again, subexponentials, and the so-called LNL model for intuitionistic linear logic by Benton. While our present work is still far from the two works, it seems fruitful to take our discussion into their frameworks to give linear-logical computational interpretations for various logics.

7.2 Computation of modal logic and its relation to metaprogramming

Computational interpretations of modal logic have been considered not only for intuitionistic S4 but also for various logics, including the modal logics K, T, K4, and GL, and a few constructive temporal logics (cf. the survey by Kavvos in [8]). This field of modal logics is known to be connected to "metaprogramming" in the theory of programming languages and has been substantially studied. One of the studies is *(multi-)staged computation* (cf. [22]), which is a programming paradigm that supports Lisp-like *quasi-quote*, *unquote*, and *eval*. The work of $\lambda^\square$ by Davies and Pfenning [5] is actually one of logical investigations of it.

Furthermore, the multi-stage programming is not a mere theory but has "real" implementations such as MetaML [22] and MetaOCaml (cf. a survey in [3]) in the style of functional programming languages. Some core calculi of these implementations are formalized as type systems (e.g. [21, 3]) and investigated from the logical point of view (e.g. [24]).

8 Conclusion

We have presented a linear-logical reconstruction of the intuitionistic modal logic S4, by establishing the modal linear logic with the $\square$-modality and the S4-version of Girard

translation from IS4. The translation from IS4 to the modal linear logic is shown to be correct with respect to the level of proofs, through the Curry–Howard correspondence.

While the proof-level Girard translation for modal logic is already proposed by Schellinx, our typed λ-calculus $\lambda^\boxdot$ and its Geometry of Interaction Machine (GoIM) are novel. Also, the significance of our formalization is its simplicity. All we need to establish the linear-logical reconstruction of modal logic is the $\boxdot$-modality, an integration of !-modality and $\Box$-modality, that gives the structure of modal logic into linear logic. Thanks to the simplicity, our λ-calculus and the GoIM can be obtained as simple extensions of existing works.

As a further direction, we plan to enrich our framework to cover other modal logics such as K, T, and K4, following the work of contextual modal calculi by Kavvos [8]. Moreover, reinvestigating of the modal-logical foundation for multi-stage programming by Tsukada and Igarashi [24] via our methods and extending Mackie's GoIM for PCF [11] to the modal-logical setting seem to be interesting from the viewpoint of programming languages.

Lastly, we have also left a semantical study for modal linear logic with respect to the validity. At the present stage, we think that we could give a sound-and-complete characterization of modal linear logic by an integration of Kripke semantics of modal logic and phase semantics of linear logic, but details will be studied in a future paper.

References

1 Samson Abramsky, Esfandiar Haghverdi, and Philip Scott. Geometry of Interaction and Linear Combinatory Algebras. *Mathematical Structures in Computer Science*, 12(5):625–665, 2002. `doi:10.1017/S0960129502003730`.

2 Andrew Barber. Dual intuitionistic linear logic, 1996. Technical report LFCS-96-347. URL: `http://www.lfcs.inf.ed.ac.uk/reports/96/ECS-LFCS-96-347`.

3 Cristiano Calcagno, Eugenio Moggi, and Walid Taha. ML-like inference for classifiers. In *Proceedings of ESOP 2004*, pages 79–93, 2004. `doi:10.1007/978-3-540-24725-8_7`.

4 Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. The Structure of Exponentials: Uncovering the Dynamics of Linear Logic Proofs. In G. Gottlob, A. Leitsch, and D. Mundici, editors, *Computational Logic and Proof Theory*, pages 159–171. Springer-Verlag, 1993. `doi:10.1007/BFb0022564`.

5 Rowan Davies and Frank Pfenning. A Modal Analysis of Staged Computation. *Journal of the ACM*, 48(3):555–604, 2001. `doi:10.1145/382780.382785`.

6 Yosuke Fukuda and Akira Yoshimizu. A Higher-arity Sequent Calculus for Modal Linear Logic. In *RIMS Kôkyûroku 2083: Symposium on Proof Theory and Proving*, pages 76–87, 2018. URL: `http://www.kurims.kyoto-u.ac.jp/~kyodo/kokyuroku/contents/pdf/2083-07.pdf`.

7 Jean-Yves Girard. Linear Logic. *Theoretical Compututer Science*, 50(1):1–102, 1987. `doi:10.1016/0304-3975(87)90045-4`.

8 G. A. Kavvos. Dual-Context Calculi for Modal Logic. In *Proceedings of LICS 2017*, pages 1–12, 2017. `doi:10.1109/LICS.2017.8005089`.

9 Patrick Lincoln, John Mitchell, Andre Scedrov, and Natarajan Shankar. Decision problems for propositional linear logic. *Annals of Pure and Applied Logic*, 56(1):239–311, 1992. `doi:10.1016/0168-0072(92)90075-B`.

10 Ian Mackie. *The Geometry of Implementation*. PhD thesis, University of London, 1994.

11 Ian Mackie. The Geometry of Interaction Machine. In *Proceedings of POPL 1995*, pages 198–208, 1995. `doi:10.1145/199448.199483`.

12 John Maraist, Martin Odersky, David N. Turner, and Philip Wadler. Call-by-name, call-by-value, call-by-need and the linear lambda lalculus. *Theoretical Computer Science*, 228(1-2):175–210, 1999. `doi:10.1016/S0304-3975(98)00358-2`.

13 Simone Martini and Andrea Masini. A Modal View of Linear Logic. *Journal of Symbolic Logic*, 59(3):888–899, 1994. `doi:10.2307/2275915`.

14 Vivek Nigam and Dale Miller. Algorithmic Specifications in Linear Logic with Subexponentials. In *Proceedings of PPDP 2009*, pages 129–140, 2009. `doi:10.1145/1599410.1599427`.

15 Vivek Nigam, Elaine Pimentel, and Giselle Reis. An extended framework for specifying and reasoning about proof systems. *Journal of Logic and Computation*, 26(2):539–576, 2016. `doi:10.1093/logcom/exu029`.

16 Yo Ohta and Masahito Hasegawa. A Terminating and Confluent Linear Lambda Calculus. In *Proceedings of RTA 2006*, pages 166–180, 2006. `doi:10.1007/11805618_13`.

17 Frank Pfenning. Lecture Notes on Combinatory Modal Logic, 2010. Lecture note. URL: `http://www.cs.cmu.edu/~fp/courses/15816-s10/lectures/09-combinators.pdf`.

18 Frank Pfenning and Rowan Davies. A judgmental reconstruction of modal logic. *Mathematical Structures in Computer Science*, 11(4):511–540, 2001. `doi:10.1017/S0960129501003322`.

19 Klaas Pruiksma, William Chargin, Frank Pfenning, and Jason Reed. Adjoint Logic, 2018. Manuscript. URL: `http://www.cs.cmu.edu/~fp/papers/adjoint18b.pdf`.

20 Harold Schellinx. A Linear Approach to Modal Proof Theory. In *Proof Theory of Modal Logic*, pages 33–43. Springer, 1996. `doi:10.1007/978-94-017-2798-3_3`.

21 Walid Taha and Michael Florentin Nielsen. Environment Classifiers. In *Proceedings of POPL 2003*, pages 26–37, 2003. `doi:10.1145/640128.604134`.

22 Walid Taha and Tim Sheard. MetaML and multi-stage programming with explicit annotations. *Theoretical Computer Science*, 248(1-2):211–242, 2000. `doi:10.1016/S0304-3975(00)00053-0`.

23 A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, 1996.

24 Takeshi Tsukada and Atsushi Igarashi. A Logical Foundation for Environment Classifiers. *Logical Methods in Computer Science*, 6(4:8):1–43, 2010. `doi:10.2168/LMCS-6(4:8)2010`.

A Appendix

A.1 Cut-elimination theorem for the intuitionistic modal linear logic

In this section, we give a complete proof of the cut-elimination theorem of IMELL$^\Box$.

▶ **Theorem 49** (Cut-elimination). *The rule* Cut *in* IMELL$^\Box$ *is admissible, i.e., if* $\Gamma \vdash A$ *is derivable, then there is a derivation of the same judgment without any applications of* Cut.

Proof. As we mentioned in the body, we will show the following rules are admissible.

$$\frac{\Gamma \vdash \,!A \qquad \Gamma', (!A)^n \vdash B}{\Gamma, \Gamma' \vdash B} \ !\text{Cut} \qquad\qquad \frac{\Gamma \vdash \Box A \qquad \Gamma', (\Box A)^n \vdash B}{\Gamma, \Gamma' \vdash B} \ \Box\text{Cut}$$

The admissibility of Cut, !Cut, $\Box$Cut are shown by simultaneous induction on the derivation of $\Gamma \vdash A$ with the lexicographic complexity $\langle \delta, h \rangle$, where δ is the degree of the assumed derivation and h is its height. Therefore, it is enough to show that for every application of cuts, one of the following hold: (1) it can be reduced to a cut with a smaller cut-degree; (2) it can be reduced to a cut with a smaller height; (3) it can be eliminated immediately.

In what follows, we will explain the admissibility of each cut rule separately although the actual proofs are done simultaneously.

- The admissibility of Cut. We show that every application of the rule Cut whose cut-degree is maximal is eliminable. Thus, consider an application of Cut in the derivation:

$$\frac{\overset{\Pi_0}{\Gamma \vdash A} \qquad \overset{\Pi_1}{\Gamma', A \vdash B}}{\Gamma, \Gamma' \vdash B} \ \text{Cut}$$

such that its cut-degree is maximal and its height is minimal (comparing to the other applications whose cut-degree is maximal). The proof proceeds by case analysis on Π_0.

- Π_0 ends with Cut. In this case the derivation is as follows:

$$
\cfrac{\cfrac{\vdots \quad\quad \vdots}{\cfrac{\Gamma_0 \vdash C \quad \Gamma_1, C \vdash A}{\Gamma_0, \Gamma_1 \vdash A}\ \text{Cut}} \quad \cfrac{\Pi_1}{\Gamma', A \vdash B}}{\Gamma_0, \Gamma_1, \Gamma' \vdash B}\ \text{Cut}
$$

Since the bottom application of Cut was chosen to have the maximal cut-degree and the minimum height, the cut-degree of the above is less than that of the bottom. Therefore, the derivation can be translated to the following:

$$
\cfrac{\cfrac{\vdots}{\Gamma_0 \vdash C} \quad \cfrac{\cfrac{\vdots \quad\quad\quad \Pi_1}{\Gamma_1, C \vdash A \quad \Gamma', A \vdash B}}{\Gamma_1, C, \Gamma' \vdash B}\ \text{I.H.}}{\Gamma_0, \Gamma_1, \Gamma' \vdash B}\ \text{I.H.}
$$

- Π_0 ends with $\multimap$R. In this case, the derivation is as follows:

$$
\cfrac{\cfrac{\cfrac{\vdots}{\Gamma, A_0 \vdash A_1}}{\Gamma \vdash A_0 \multimap A_1}\ \multimap\text{R} \quad \cfrac{\Pi_1}{\Gamma', A_0 \multimap A_1 \vdash B}}{\Gamma, \Gamma' \vdash B}\ \text{Cut}
$$

for some A_0 and A_1 such that $A \equiv A_0 \multimap A_1$. If the last step in Π_1 is Ax, then the result is obtained as Π_0. If the last step in Π_1 is $\multimap$L, the derivation is as follows:

$$
\cfrac{\cfrac{\cfrac{\vdots}{\Gamma, A_0 \vdash A_1}}{\Gamma \vdash A_0 \multimap A_1}\ \multimap\text{R} \quad \cfrac{\cfrac{\vdots \quad\quad \vdots}{\Gamma' \vdash A_0 \quad \Gamma'', A_1 \vdash B}}{\Gamma', \Gamma'', A_0 \multimap A_1 \vdash B}\ \multimap\text{L}}{\Gamma, \Gamma', \Gamma'' \vdash B}\ \text{Cut}
$$

which is translated to the following:

$$
\cfrac{\cfrac{\cfrac{\vdots \quad\quad \vdots}{\Gamma' \vdash A_0 \quad \Gamma, A_0 \vdash A_1}}{\Gamma, \Gamma' \vdash A_1}\ \text{I.H.} \quad \cfrac{\vdots}{\Gamma'', A_1 \vdash B}}{\Gamma, \Gamma', \Gamma'' \vdash B}\ \text{I.H.}
$$

since the cut-degrees of A_0 and A_1 are less than that of A. The other cases can be shown by simple commutative conversions.

- Π_0 ends with !R. This case is dealt as a special case of the case !R in !Cut.
- Π_0 ends with $\Box$R. This case is dealt as a special case of the case $\Box$R in $\Box$Cut.
- Π_0 ends with the other rules. Easy.

- The admissibility of !Cut. As in the case of Cut, consider an application of !Cut:

$$
\cfrac{\cfrac{\Pi_0}{\Gamma \vdash !A} \quad \cfrac{\Pi_1}{\Gamma', (!A)^n \vdash B}}{\Gamma, \Gamma' \vdash B}\ \text{!Cut}
$$

such that its cut-degree is maximal and its height is minimal. By case analysis on Π_0.

- Π_0 ends with Ax. In this case the cut-elimination is done as follows:

$$
\cfrac{\cfrac{}{!A \vdash !A} \quad \cfrac{\Pi_1}{\Gamma', (!A)^n \vdash B}}{!A, \Gamma' \vdash B}\ \text{!Cut} \qquad \overset{\text{Cut elim.}}{\Longrightarrow} \qquad \cfrac{\cfrac{\Pi_1}{\Gamma', (!A)^n \vdash B}}{\Gamma', !A \vdash B}\ \text{!C}
$$

- Π_0 ends with !R. In this case the derivation is as follows:

$$\cfrac{\cfrac{\vdots}{\cfrac{\boxdot\Gamma_0, !\Gamma_1 \vdash A}{\boxdot\Gamma_0, !\Gamma_1 \vdash !A}\ !\text{R}} \quad \cfrac{\Pi_1}{\Gamma', (!A)^n \vdash B}}{\boxdot\Gamma_0, !\Gamma_1, \Gamma' \vdash B}\ !\text{Cut}$$

Due to the side-condition of !R, we have to do case analysis on Π_1 further as follows.

  * Π_1 ends with Cut. In this case the derivation is as follows:

$$\cfrac{\cfrac{\Pi_0}{\boxdot\Gamma_0, !\Gamma_1 \vdash !A} \quad \cfrac{\cfrac{\vdots}{\Gamma', (!A)^k \vdash C} \quad \cfrac{\vdots}{\Gamma'', (!A)^l, C \vdash B}}{\Gamma', \Gamma'', (!A)^n \vdash B}\ \text{Cut}}{\boxdot\Gamma_0, !\Gamma_1, \Gamma', \Gamma'' \vdash B}\ !\text{Cut}$$

where $n = k + l$. We only deal with the case of $k > 0$ and $l > 0$, and the other cases are easy. Then, the derivation can be translated to the following:

$$\cfrac{\cfrac{\cfrac{\Pi_0}{\boxdot\Gamma_0, !\Gamma_1 \vdash !A} \quad \cfrac{\vdots}{\Gamma', (!A)^k \vdash C}}{\boxdot\Gamma_0, !\Gamma_1, \Gamma' \vdash C}\ \text{I.H.} \quad \cfrac{\cfrac{\Pi_0}{\boxdot\Gamma_0, !\Gamma_1 \vdash !A} \quad \cfrac{\vdots}{\Gamma'', (!A)^l, C \vdash B}}{\boxdot\Gamma_0, !\Gamma_1, \Gamma'', C \vdash B}\ \text{I.H.}}{\cfrac{(\boxdot\Gamma_0)^2, (!\Gamma_1)^2, \Gamma', \Gamma'' \vdash B}{\boxdot\Gamma_0, !\Gamma_1, \Gamma', \Gamma'' \vdash B}\ !\text{C}, \boxdot\text{C}}\ \text{I.H.}$$

since the cut-degree of !Cut is less than that of Cut from the assumption.

  * Π_1 ends with !L. If the formula introduced by !L is not the cut-formula, then it is easy. For the other case, the derivation is as follows:

$$\cfrac{\cfrac{\cfrac{\vdots}{\boxdot\Gamma_0, !\Gamma_1 \vdash A}}{\boxdot\Gamma_0, !\Gamma_1 \vdash !A}\ !\text{R} \quad \cfrac{\cfrac{\vdots}{\Gamma', (!A)^{n-1}, A \vdash B}}{\Gamma', (!A)^n \vdash B}\ !\text{L}}{\boxdot\Gamma_0, !\Gamma_1, \Gamma' \vdash B}\ !\text{Cut}$$

which is translated to the following:

$$\cfrac{\cfrac{\vdots}{\boxdot\Gamma_0, !\Gamma_1 \vdash A} \quad \cfrac{\cfrac{\Pi_0}{\boxdot\Gamma_0, !\Gamma_1 \vdash !A} \quad \cfrac{\vdots}{\Gamma', (!A)^{n-1}, A \vdash B}}{\boxdot\Gamma_0, !\Gamma_1, \Gamma', A \vdash B}\ \text{I.H.}}{\cfrac{(\boxdot\Gamma_0)^2, (!\Gamma_1)^2, \Gamma' \vdash B}{\boxdot\Gamma_0, !\Gamma_1, \Gamma' \vdash B}\ !\text{C}, \boxdot\text{C}}\ \text{I.H.}$$

  * Π_1 ends with !C. If the formula introduced by !C is not the cut-formula, then it is easy. For the other case, the cut-elimination is done as follows:

$$\cfrac{\cfrac{\Pi_0}{\boxdot\Gamma_0, !\Gamma_1 \vdash !A} \quad \cfrac{\cfrac{\vdots}{\Gamma', (!A)^{n+1} \vdash B}}{\Gamma', (!A)^n \vdash B}\ !\text{C}}{\boxdot\Gamma_0, !\Gamma_1, \Gamma' \vdash B}\ !\text{Cut} \quad \overset{\text{Cut elim.}}{\Longrightarrow} \quad \cfrac{\cfrac{\Pi_0}{\boxdot\Gamma_0, !\Gamma_1 \vdash !A} \quad \cfrac{\vdots}{\Gamma', (!A)^{n+1} \vdash B}}{\boxdot\Gamma_0, !\Gamma_1, \Gamma' \vdash B}\ \text{I.H.}$$

Note that the whole proof has not been proceeding by induction on n, and hence the number of occurrences of $!A$ does not matter in this case.

  * Π_1 ends with the other rules. Easy.
- Π_0 ends with the other rules. Easy.
- The admissibility of $\boxdot$Cut. Similar to the case of !Cut. ◀

Syntactic category

Types $\quad A, B, C ::= p \mid A \multimap B \mid !A$

Terms $\quad M, N, L ::= x \mid \lambda x : A.M \mid M\,N \mid !M \mid \textbf{let}\,!x = M\,\textbf{in}\,N$

Typing rule

$$\frac{}{\Gamma; x : A \vdash x : A}\,\text{LinAx} \qquad\qquad \frac{}{\Gamma, x : A; \emptyset \vdash x : A}\,\text{!Ax}$$

$$\frac{\Gamma; \Sigma, x : A \vdash M : B}{\Gamma; \Sigma \vdash \lambda x : A.M : A \multimap B}\,\multimap\text{I} \qquad \frac{\Gamma; \Sigma \vdash M : A \multimap B \qquad \Gamma; \Sigma' \vdash N : A}{\Gamma; \Sigma, \Sigma' \vdash M\,N : B}\,\multimap\text{E}$$

$$\frac{\Gamma; \emptyset \vdash M : A}{\Gamma; \emptyset \vdash !M : !A}\,\text{!I} \qquad \frac{\Gamma; \Sigma \vdash M : !A \qquad \Gamma, x : A; \Sigma' \vdash N : B}{\Gamma; \Sigma, \Sigma' \vdash \textbf{let}\,!x = M\,\textbf{in}\,N : B}\,\text{!E}$$

Reduction rule

$$(\lambda x : A.M)\,N \rightsquigarrow M[x := N]$$
$$\lambda x : A.M\,x \rightsquigarrow M$$
$$\textbf{let}\,!x = !M\,\textbf{in}\,N \rightsquigarrow N[x := M]$$
$$\textbf{let}\,!x = M\,\textbf{in}\,C[!x] \rightsquigarrow C[M]$$
$$(\textbf{let}\,!x = M\,\textbf{in}\,N)\,L \rightsquigarrow \textbf{let}\,!x = M\,\textbf{in}\,N\,L$$
$$\textbf{let}\,!y = (\textbf{let}\,!x = M\,\textbf{in}\,N)\,\textbf{in}\,L \rightsquigarrow \textbf{let}\,!x = M\,\textbf{in}\,\textbf{let}\,!y = N\,\textbf{in}\,L$$
$$\lambda y : A.(\textbf{let}\,!x = M\,\textbf{in}\,N) \rightsquigarrow \textbf{let}\,!x = M\,\textbf{in}\,\lambda y : A.N \qquad\qquad (\text{if } y \notin \text{FV}(M))$$
$$L\,(\textbf{let}\,!x = M\,\textbf{in}\,N) \rightsquigarrow \textbf{let}\,!x = M\,\textbf{in}\,L\,N$$
$$\lambda z : A.(\textbf{let}\,!y = M\,\textbf{in}\,\textbf{let}\,!x = L\,\textbf{in}\,N) \rightsquigarrow \textbf{let}\,!x = L\,\textbf{in}\,\lambda z : A.(\textbf{let}\,!y = M\,\textbf{in}\,N) \quad (\text{if } y \notin \text{FV}(L))$$

where $C[-]$ is a linear context defined by the following grammar:

$$C ::= [-] \mid \lambda x : A.C \mid C\,M \mid M\,C \mid \textbf{let}\,!x = C\,\textbf{in}\,M \mid \textbf{let}\,!x = M\,\textbf{in}\,C$$

Figure 18 Definition of $\lambda^{!,\multimap}$ (some syntax are changed to fit the present paper's notation).

A.2 Strong normalizability of the typed λ-calculus for modal linear logic

We complete the proof of the strong normalization theorem for $\lambda^{\square}$. As we mentioned, this is done by an embedding to a typed λ-calculus for the $(!, \multimap)$-fragment of dual intuitionistic linear logic, studied by Ohta and Hasegawa [16], and shown to be strongly normalizing.

The calculus of Ohta and Hasegawa, named $\lambda^{!,\multimap}$ here, is given in Figure 18. The syntax and the typing rules can be read in the same way as (the $(!, \multimap)$-fragment of) $\lambda^{\square}$. There are somewhat many reduction rules in contrast to those of $\lambda^{\square}$, but these are due to the purpose of Ohta and Hasegawa to consider η-rules and commutative conversions. The different sets of reduction rules do not cause any problems to prove the strong normalizability of $\lambda^{\square}$.

▶ **Definition 50** (Embedding). *An* embedding *from $\lambda^{\square}$ to $\lambda^{!,\multimap}$ is defined to be the triple of the translations $(A)^{\ddagger}$, $(\Gamma)^{\ddagger}$, and $(M)^{\ddagger}$ given in Figure 19.*

▶ **Lemma 51** (Preservation of typing and reduction).
1. *If $\Delta; \Gamma; \Sigma \vdash M : A$ in $\lambda^{\square}$, then $(\Delta, \Gamma)^{\ddagger}; (\Sigma)^{\ddagger} \vdash (M)^{\ddagger} : (A)^{\ddagger}$ in $\lambda^{!,\multimap}$.*
2. *If $M \rightsquigarrow N$ in $\lambda^{\square}$, then $(M)^{\ddagger} \rightsquigarrow (N)^{\ddagger}$ in $\lambda^{!,\multimap}$.*

Proof. By induction on $\Delta; \Gamma; \Sigma \vdash M : A$ and $M \rightsquigarrow N$, respectively. ◀

$$
\begin{aligned}
\underline{(A)^\ddagger} \\
(p)^\ddagger &\overset{\text{def}}{=} p \\
(A \multimap B)^\ddagger &\overset{\text{def}}{=} (A)^\ddagger \multimap (B)^\ddagger \\
(!A)^\ddagger &\overset{\text{def}}{=} !(A)^\ddagger \\
(\square A)^\ddagger &\overset{\text{def}}{=} !(A)^\ddagger
\end{aligned}
$$

$$
\begin{aligned}
\underline{(\Gamma)^\ddagger} \\
(\Gamma)^\ddagger &\overset{\text{def}}{=} \{(x : (A)^\ddagger) \mid (x : A) \in \Gamma\}
\end{aligned}
$$

$$
\begin{aligned}
\underline{(M)^\ddagger} \qquad (x)^\ddagger &\overset{\text{def}}{=} x \\
(\lambda x : A.M)^\ddagger &\overset{\text{def}}{=} \lambda x : (A)^\ddagger.(M)^\ddagger \\
(M\,N)^\ddagger &\overset{\text{def}}{=} (M)^\ddagger\,(N)^\ddagger \\
(!M)^\ddagger &\overset{\text{def}}{=} !(M)^\ddagger \\
(\mathbf{let}\,!x = M\,\mathbf{in}\,N)^\ddagger &\overset{\text{def}}{=} \mathbf{let}\,!x = (M)^\ddagger\,\mathbf{in}\,(N)^\ddagger \\
(\square M)^\ddagger &\overset{\text{def}}{=} !(M)^\ddagger \\
(\mathbf{let}\,\square x = M\,\mathbf{in}\,N)^\ddagger &\overset{\text{def}}{=} \mathbf{let}\,!x = (M)^\ddagger\,\mathbf{in}\,(N)^\ddagger
\end{aligned}
$$

Figure 19 Definition of the embeddings $(A)^\ddagger$, $(\Gamma)^\ddagger$, and $(M)^\ddagger$.

$$
\begin{aligned}
\underline{\lambda_* x.M} \\
\lambda_* x.x &\overset{\text{def}}{=} \mathrm{I} \\
\lambda_* x.(M\,N) &\overset{\text{def}}{=} \mathrm{C}\,(\lambda_* x.M)\,N \quad \text{if } x \in \mathrm{FV}(M) \\
\lambda_* x.(M\,N) &\overset{\text{def}}{=} \mathrm{B}\,M\,(\lambda_* x.N) \quad \text{if } x \in \mathrm{FV}(N)
\end{aligned}
$$

$$
\begin{aligned}
\underline{\lambda_*^! x.M} \\
\lambda_*^! x.x &\overset{\text{def}}{=} \mathrm{T}^! \\
\lambda_*^! x.M &\overset{\text{def}}{=} \mathrm{K}^!\,M && \text{if } (a) \\
\lambda_*^! x.(M\,N) &\overset{\text{def}}{=} \mathrm{C}\,(\lambda_*^! x.M)\,N && \text{if } (b) \\
\lambda_*^! x.(M\,N) &\overset{\text{def}}{=} \mathrm{B}\,M\,(\lambda_*^! x.N) && \text{if } (c) \\
\lambda_*^! x.(M\,N) &\overset{\text{def}}{=} \mathrm{S}^!\,(\lambda_*^! x.M)\,(\lambda_*^! x.N) && \text{if } (d) \\
\lambda_*^! x.(!M) &\overset{\text{def}}{=} \mathrm{B}\,(\mathrm{D}^!\,!(\lambda_* x.M))\,4^!
\end{aligned}
$$

$$
\begin{aligned}
\underline{\lambda_*^\square x.M} \\
\lambda_*^\square x.x &\overset{\text{def}}{=} \mathrm{T}^\square \\
\lambda_*^\square x.M &\overset{\text{def}}{=} \mathrm{K}^\square M && \text{if } (a) \\
\lambda_*^\square x.(M\,N) &\overset{\text{def}}{=} \mathrm{C}\,(\lambda_*^\square x.M)\,N && \text{if } (b) \\
\lambda_*^\square x.(M\,N) &\overset{\text{def}}{=} \mathrm{B}\,M\,(\lambda_*^\square x.N) && \text{if } (c) \\
\lambda_*^\square x.(M\,N) &\overset{\text{def}}{=} \mathrm{S}^\square(\lambda_*^\square x.M)\,(\lambda_*^\square x.N) && \text{if } (d) \\
\lambda_*^\square x.(!M) &\overset{\text{def}}{=} \mathrm{B}\,(\mathrm{D}^!\,(!(\lambda_*^\square x.M)))\,(\mathrm{B}\,\mathrm{E}\,4^\square) \\
\lambda_*^\square x.(\square M) &\overset{\text{def}}{=} \mathrm{B}\,(\mathrm{D}^\square(\square(\lambda_*^\square x.M)))\,4^\square
\end{aligned}
$$

where $(a), (b), (c), (d)$ means the conditions $(x \notin \mathrm{FV}(M))$, $(x \in \mathrm{FV}(M)$ and $x \notin \mathrm{FV}(N))$, $(x \notin \mathrm{FV}(M)$ and $x \in \mathrm{FV}(N))$, $(x \in \mathrm{FV}(M)$ and $x \in \mathrm{FV}(N))$, respectively.

Figure 20 Definitions of $(\lambda_* x.M)$, $(\lambda_*^! x.M)$, and $(\lambda_*^\square x.M)$ for bracket abstraction.

▶ **Theorem 52** (Strong normalization). *In $\lambda^\square$, there are no infinite reduction sequences starting from M for all well-typed term M.*

Proof. Suppose that there exists an infinite reduction sequence starting from M in $\lambda^\square$. Then, the term $(M)^\ddagger$ is well-typed in $\lambda^{!,\multimap}$ and yields an infinite reduction sequence in $\lambda^{!,\multimap}$ by Lemma 51. However, this contradicts the strong normalizability of $\lambda^{!,\multimap}$. ◀

A.3 Bracket abstraction algorithm

We show the definition of bracket abstraction operators in this section.

▶ **Definition 53** (Bracket abstraction). *Let M be a term M of $\mathrm{CL}^\square$ such that $\Delta; \Gamma; \Sigma \vdash M : A$ and $x \in \mathrm{FV}(M)$ for some $\Delta, \Gamma, \Sigma, A$ and x. Then, the* bracket abstraction of M *with respect*

to x *is defined to be either one of the following, depending the variable kind of* x:

$$(\lambda_* x.M) \qquad \textit{if } x \in \mathrm{dom}(\Sigma)^3;$$
$$(\lambda_*^! x.M) \qquad \textit{if } x \in \mathrm{dom}(\Gamma);$$
$$(\lambda_*^\square x.M) \qquad \textit{if } x \in \mathrm{dom}(\Delta),$$

where each one of $(\lambda_* x.M)$, $(\lambda_*^! x.M)$, *and* $(\lambda_*^\square x.M)$ *is the meta-level* bracket abstraction *operation given in Figure 20, which takes the pair of* x *and* M, *and yields a* $\mathrm{CL}^\square$-*term.*

▶ **Remark 54.** As in the case of standard bracket abstraction algorithm, the intuition behind the operations $(\lambda_* x.M)$, $(\lambda_*^! x.M)$, and $(\lambda_*^! x.M)$ is that they are defined so as to mimic the λ-abstraction operation in the framework of combinatory logic. For instance, the denotation of $(\lambda_* x.M)$ is a $\mathrm{CL}^\square$-term that represents a function with the parameter x, that is, it is a term that satisfies that $(\lambda_* x.M)\, N \rightsquigarrow^+ M[x := N]$ in $\mathrm{CL}^\square$, for all $\mathrm{CL}^\square$-terms N.

▶ **Remark 55.** There are no definitions for some cases in $(\lambda_* x.M)$ and $(\lambda_*^! x.M)$, e.g, the case that $(\lambda_* x.(M\, N))$ such that $x \in \mathrm{FV}(M)$ and $x \in \mathrm{FV}(N)$, and the case that $(\lambda_*^! x.(\square M))$. This is because that these are actually unnecessary due to the linearity condition and the side condition of the rule $\square I$. Moreover, the well-definedness of the bracket abstraction operations can be shown by induction on M, and in reality, the proof of the deduction theorem can be seen as what justifies it. The intentions that $(\lambda_* x.M)\, N \rightsquigarrow^+ M[x := N]$, etc. can also be shown by easy calculation.

---

3 $\mathrm{dom}(\Gamma)$ is defined to be the set $\{x \mid (x : A) \in \Gamma\}$ for all type contexts Γ.