Meta-Lambda Calculus and Linguistic Monads

Daisuke Bekki and Moe Masuko

Abstract *Meta-lambda calculus* (MLC) is a two-level typed lambda calculus with meta-level types and terms. MLC has been adopted in the analyses of natural language semantics and pragmatics by means of *monads* and *monadic translation* (Bekki 2009; Bekki and Asai 2010), however, the soundness of the equational theory in Bekki (2009) has not been fully proven with respect to the categorical semantics in Bekki (2009). In this article, we introduce a revised syntax and an equational theory of MLC with base-level/meta-level substitution and $\alpha/\beta/\eta$-conversions, and prove their soundness with respect to a revised categorical semantics of MLC.

Keywords Meta-lambda calculus · Type theoretic semantics · Category theory · Monads · Non-determinism · Ostension · Continuations

---

Our sincere thanks to the participants at LENLS5 and LENLS6, especially Eric McCready, Robert van Rooij, Kei Yoshimoto, Masahiko Sato, and the late Norry Ogata, and also to the participants at FLOPS2010, especially Olivier Danvy, Ian Zerney, Chung-chieh Shan, Oleg Kiselyov, Yukiyoshi Kameyama for their valuable comments. Our special thanks go to Kenichi Asai for many discussions in our research group. We also thank the anonymous reviewers for their insightful comments. Daisuke Bekki is partially supported by a Grant-in-Aid for Young Scientists (A) (No. 22680013) from the Ministry of Education, Science, Sports and Culture

D. Bekki (✉) · M. Masuko
Ochanomizu University, Graduate School of Humanities and Sciences, 2-1-1 Ohtsuka, Tokyo, Bunkyo-ku 112-8610, Japan
e-mail: bekki@is.ocha.ac.jp

D. Bekki
National Institute of Informatics, 2-1-2 Hitotsubashi, Tokyo, Chiyoda-ku 101-8430, Japan

D. Bekki
CREST, Japan Science and Technology Agency, 4-1-8 Honcho, Saitama, Kawaguchi 332-0012, Japan

M. Masuko
Research Fellow of the Japan Society for the Promotion of Science, Tokyo, Japan
e-mail: masuko.moe@is.ocha.ac.jp

E. McCready et al. (eds.), *Formal Approaches to Semantics and Pragmatics*, Studies in Linguistics and Philosophy, DOI: 10.1007/978-94-017-8813-7_3,

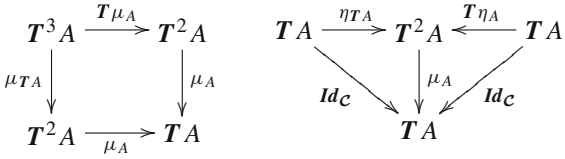
1 Introduction

1.1 Monads in Category Theory and Programming Language

The notion of monads originates in homological algebra and category theory: a monad in a category $\mathcal{C}$ is a triple $\langle \boldsymbol{T}, \eta, \mu \rangle$ that consists of a functor $\boldsymbol{T} : \mathcal{C} \longrightarrow \mathcal{C}$ and two natural transformations:

$$\eta : \boldsymbol{Id}_{\mathcal{C}} \stackrel{\cdot}{\longrightarrow} \boldsymbol{T}, \qquad \mu : \boldsymbol{T}^2 \stackrel{\cdot}{\longrightarrow} \boldsymbol{T}$$

such that the following diagrams commute for any object A in $\mathcal{C}$.



Lambek (1980) established categorical semantics of simply-typed lambda calculi (hereafter STLC), showing that STLC are equivalent to Cartesian closed categories (CCC), in which STLC terms are interpreted as morphisms.

These studies converged to the *monadic* categorical semantics of STLC in Moggi's seminal work (Moggi 1989), where each lambda term is interpreted as a morphism in the Kleisli category generated by a certain monad. This setting is intended to uniformly encapsulate "impure" aspects of functional programming languages, such as side-effects, exceptions and continuations, within the enhanced data types specified by the monad, and hide them within "pure" structures of STLC. The method requires, however, some tangled notions such as tensorial strength (Kock 1970) for the definition of lambda abstraction, evaluation and products.

This complexity motivated Wadler (1990) to propose a simplified model, known as *monad comprehension*, which generalizes the notion of list comprehension. Results from this study were incorporated into the programming language Haskell, and this showed that the monadic analyses can treat a wider range of computational concepts than those enumerated in Moggi (1989), such as state readers, array update, non-determinism, inputs/outputs, and even parsers and interpreters.

1.2 Monads in Linguistics

It was first suggested in Shan (2001) that the notion of monad can be imported to the field of natural language semantics, where various semantic, pragmatic, or computational phenomena such as non-determinism, focus, intensionality, variable binding, continuation and quantification, can be represented as monads, just as the "impure" aspects in programming languages. A few works such as Ogata (2008) and Unger (2011) have pursued this view recently.

The general idea is that a monad expands the data structure of semantic representations in such a way that certain phenomena become representable. Many theories in formal semantics have employed their own extended type-theoretical languages in order to explain certain linguistic phenomena, however, monads achieve this in a uniform way: for semantic representations, we use a simple language of STLC, except that it may contain *control operators* in the sense of Danvy and Filinski (1990). Those control operators are interpreted by translating the simple language (called a *direct-style*) into the expanded language (called a *monadic-style*) by the map defined by the given monad.

This enterprise, encapsulation of "impure" aspects of computation by monads, seems to be an attractive prospect, especially given the lack of standard formal models for the interfaces between semantics/pragmatics or semantics/computation. On the other hand, monadic analyses, as they have drifted among different fields, seem to have become gradually dissociated from the original monad concept.

In Bekki (2009) we aim to restore the relation between monadic analyses and categorical monads. In other words, we aim to combine the recent advances in monadic analyses with the categorical semantics of STLC along the lines of Lambek (1980). This is realized through *Meta-Lambda Calculus* (henceforth MLC; Bekki (2009)) (for details, see Sects. 2, 3 and 5) and a framework of *monadic translations* (for details, see Sect. 4). In our subsequent works, we have proposed a unified analysis for phenomena including non-deterninism and contextual parameters (Bekki 2009), focus, *only* and inverse scope (Bekki and Asai (2010)), and conjoined nominals in Japanese (Hayashishita and Bekki 2011) by means of MLC and corresponding monads, which are otherwise problematic for a pure type-theoretical treatment.

1.3 Meta-Lambda Calculus

MLC is a two-level typed lambda calculus with finite products that has *meta-level types* and *meta-level terms*. Each meta-level type represents a *judgment* of a base-level term, and each meta-level lambda abstraction and product corresponds to a function relating judgments and a tuple of judgments.

While a judgment in STLC *á la* Curry is in the form of (1a), it is represented in MLC in the form of (1b), where M is a *meta-level term* and $\Gamma \vdash \alpha$ is a *meta-level type*.

(1) a. $\Gamma \vdash M : \alpha$
b. $\Vdash M : (\Gamma \vdash \alpha)$

A judgement of a term of functional type such as (2a) in STLC is represented as shown in (2b) in MLC.

(2) a. $\Gamma \vdash M : \alpha \to \beta$
b. $\Vdash M : (\Gamma \vdash \alpha \to \beta)$

This setting of MLC enables us to represent a *meta*-function type, which is not representable in STLC, as follows.

(3) $\Vdash M : (\Gamma \vdash \alpha) \Rightarrow (\Gamma' \vdash \alpha')$

As for β-conversion, a pair consisting of a base-level abstraction (notation: $\lambda x.M$) and an application (notation: MN) and a pair consisting of a meta-level abstraction (notation: $\zeta X.M$) and an application (notation: $M \natural N$) behave in a parallel way:

$$\begin{aligned}(\lambda x.x)c &=_{\beta} \quad c\\ (\zeta X.X) \natural c &=_{M\beta} \; c\end{aligned}$$

where β and $M\beta$ signify base-level and meta-level β-conversion, respectively.

The difference between base-level/meta-level terms emerges when free/bound variables are considered. In the following base-level β-conversion, y is not allowed as a substitute for x within the scope of λy.

$$\begin{aligned}(\lambda x.\lambda y.x)y &=_{\beta} (\lambda y.x)[y/x]\\ &=_{\alpha} (\lambda z.x)[y/x]\\ &\equiv \lambda z.x[y/x]\\ &\equiv \lambda z.y\end{aligned}$$

However, in the following meta-level β-conversion, y *is* allowed as a substitute for X within the scope of λy since X is a meta-level variable.

$$\begin{aligned}(\zeta X.\lambda y.X) \natural y &=_{M\beta} (\lambda y.X)[y/X]\\ &\equiv \quad \lambda y.X[y/X]\\ &\equiv \quad \lambda y.y\end{aligned}$$

This occurs because the "input" of the meta-level abstraction $(\zeta X.\lambda y.X)$ is not a simple value y but rather a base-level judgment $y : \alpha \vdash y : \alpha$ (namely, $\Vdash y : (y : \alpha \vdash \alpha)$ in MLC), which is illustrated in the following type inference diagram for the term $(\zeta X.\lambda y.X) \natural y$.

(4)

$$\dfrac{\dfrac{\dfrac{\dfrac{}{X : (y : \alpha \vdash \alpha) \Vdash X : (y : \alpha \vdash \alpha)}(MVAR)}{X : (y : \alpha \vdash \alpha) \Vdash \lambda y.X : (\vdash \alpha \to \alpha)}(LAM)}{\Vdash \zeta X.\lambda y.X : (y : \alpha \vdash \alpha) \Rightarrow (\vdash \alpha \to \alpha)}(MLAM) \qquad \dfrac{}{\Vdash y : (y : \alpha \vdash \alpha)}(VAR)}{\Vdash (\zeta X.\lambda y.X) \natural y : (\vdash \alpha \to \alpha)}(MAPP)$$

At first glance, this behavior of MLC might seem peculiar enough to make one suspect whether it is a reliable calculus in the first place. Thus, it is of paramount importance for us to ensure that MLC is a sound formal system.

This article presents a revised syntax of MLC and an equational theory including base-level/meta-level substitutions and $\alpha/\beta/\eta$-conversions, together with a revised categorical semantics with respect to which we give a proof that the equational theory is sound. It gives us a firm foundation in order to pursue linguistic analyses based

on various monads and monadic translations defined by means of MLC, which were presented in our previous work (Bekki 2009; Bekki and Asai 2010; Hayashishita and Bekki 2011).

1.4 Remarks on Previous Formulations of MLC

Several remarks should be made on the different formulations of MLC that we adopted in our previous works. We first presented the framework of MLC in Bekki (2009) which contains a syntax of MLC and an equational theory with meta-level β-conversion and base-level β-conversion (that we term "normal" β-conversion), which we proved to be sound with respect to a categorical semantics of MLC.

At that time, substitution was treated as a distinct syntactic structure, and in terms such as $M[L/x]$, M was implicitly assumed to be a term of base-level type following the definition of the categorical semantics.

One problem was that this setting did not provide a way to calculate base-level substitutions in a syntactic way. For example, suppose that $M : \tau_1 \Rightarrow \tau_2$ and $N : \tau_1$, where τ_2 is a base-level type. Then, $M \natural N : \tau_2$ is a term of a base-level type, so $(M \natural N)[L/x] : \tau_2$ must be defined, and we want to calculate it by using an equation, such as $(M \natural N)[L/x] = (M[L/x]) \natural (N[L/x]) : \tau_2$. However, $M[L/x]$ is not defined in Bekki (2009) since M is of meta-level type $\tau_1 \Rightarrow \tau_2$, and base-level substitution is defined only for a term of a base-level type.

This problem led us to define substitution by a set of rewriting rules in Bekki and Asai (2010), and not as an independent syntactic structure as in Bekki (2009). On the other hand, this means that base-level/meta-level β-conversions are *no longer sound* with respect to the categorical semantics in Bekki (2009) since β-conversions refer to substitutions.

This problem was fixed in Masuko and Bekki (2011) by extending the categorical semantics of substitution so that it can interpret the form $M[L/x]$, where M is a term of any type.

Another problem that was fixed in Masuko and Bekki (2011) is the relation between base-level/meta-level terms and base-level/meta-level types. In Bekki (2009), base-level terms are assumed to be of base-level types, and meta-level terms are assumed to be of meta-level types, while in fact this is not the case. It is true that a base-level term is always of base-level type, but it also means that it is of meta-level type since any base-level type is a meta-level type.

A meta-level term may be of base-level type or meta-level type. For example, in the judgement $X : (x : \alpha \vdash \alpha) \Vdash X : (x : \alpha \vdash \alpha)$, meta-level variable X is of base-level type $x : \alpha \vdash \alpha$. If $M : \tau \Rightarrow (x : \alpha \vdash \alpha)$ and $N : \tau$, then meta-level application $M \natural N$ is of base-level type $(x : \alpha \vdash \alpha)$. Meta-level abstractions and meta-level products must be of meta-level type. Thus, the correspondence between base-level/meta-level terms and base-level/meta-level types is not as straightforward as assumed in Bekki (2009).

The notational variance of MLC is shown in Table 1.

Table 1 Notational variance of MLC

	Bekki (2009)	Bekki and Asai (2010)	This article
Meta-level functional type	$\tau_1 \mapsto \tau_2$	$\tau_1 \Rightarrow \tau_2$	$\tau_1 \Rightarrow \tau_2$
Base-level functional type	$\tau_1 \tau_2$	$\tau_1 \rightarrow \tau_2$	$\tau_1 \rightarrow \tau_2$
Meta-level functional application	$M\lceil N \rceil$	MN or $M \triangleleft N$	$M \not\downarrow N$

2 Syntax of MLC

2.1 Base-level/Meta-level Types

The syntax of MLC is specified by the following definitions.

Definition 1 *(Alphabet for MLC)* An *alphabet* for MLC is a sextuple $\langle \mathcal{GT}, \mathcal{C}on, \mathcal{M}con, \mathcal{V}ar, \mathcal{M}var, \mathfrak{S} : \mathcal{M}var \rightarrow \mathcal{P}ow(\mathcal{V}ar) \rangle$, the elements of which respectively represent a finite collection of ground types, base-level constant symbols, *meta-level* constant symbols, base-level variables, *meta-level* variables and an assignment function of *free base-level variables* for each meta-level variable.

Definition 2 *(Base-level types)* A *collection of base-level types* (notation: $\mathcal{T}yp$) for an alphabet $\langle \mathcal{GT}, \mathcal{C}on, \mathcal{M}con, \mathcal{V}ar, \mathcal{M}var, \mathfrak{S} \rangle$ is recursively defined by the following BNF grammar (where $\gamma \in \mathcal{GT}$).

$$\mathcal{T}yp := \gamma \mid \mathit{unit} \mid \mathcal{T}yp \times \cdots \times \mathcal{T}yp \mid \mathcal{T}yp \rightarrow \mathcal{T}yp$$

Definition 3 *(Base-level contexts)* A *base-level context* is a finite list of pairs that are members of $\mathcal{V}ar \times \mathcal{T}yp$ (notation: $\Gamma = x_1 : \alpha_1, \ldots, x_n : \alpha_n$).

Definition 4 *(Meta-level types)* A *collection of meta-level types* (notation: $\mathcal{M}typ$) for an alphabet $\langle \mathcal{GT}, \mathcal{C}on, \mathcal{M}con, \mathcal{V}ar, \mathcal{M}var, \mathfrak{S} \rangle$ is recursively defined by the following BNF grammar (where Γ is a base-level context and $\alpha \in \mathcal{T}yp$).

$$\mathcal{M}typ \ := \ \Gamma \vdash \alpha \mid \mathit{munit} \mid \mathcal{M}typ \otimes \cdots \otimes \mathcal{M}typ \mid \mathcal{M}typ \Rightarrow \mathcal{M}typ$$

Definition 5 *(Meta-level contexts)* A *meta-level context* is a finite list of pairs that are members of $\mathcal{M}var \times \mathcal{M}typ$ (notation: $\Delta = X_1 : \tau_1, \ldots, X_n : \tau_n$).

2.2 Raw Terms

Definition 6 *(Raw terms)* A collection of *raw terms* (notation: Λ) for an alphabet $\langle \mathcal{GT}, \mathcal{C}on, \mathcal{M}con, \mathcal{V}ar, \mathcal{M}var, \mathfrak{S} \rangle$ is recursively defined by the following BNF notation, where $x \in \mathcal{V}ar$, $c \in \mathcal{C}on$, $X \in \mathcal{M}var$, and $C \in \mathcal{M}con$.[1]

[1] This definition of raw terms is a slightly revised version of that in Bekki (2009) and Bekki and Asai (2010).

$$
\begin{array}{rcl}
\overline{\mathcal{X}} & ::= & X \mid \overline{\mathcal{X}}[\Lambda/x] \\
\overline{\mathcal{C}} & ::= & C \mid \overline{\mathcal{C}}[\Lambda/x] \\
\Lambda & ::= & x \mid c \mid \langle\,\rangle \mid \langle \Lambda, \ldots, \Lambda \rangle \mid \pi_i(\Lambda) \mid \lambda x.\Lambda \mid \Lambda\Lambda \mid (\Lambda) \\
 & \mid & \overline{\mathcal{X}} \mid \overline{\mathcal{C}} \mid \langle\langle\,\rangle\rangle \mid \langle\langle \Lambda, \ldots, \Lambda \rangle\rangle \mid p_i(\Lambda) \mid \zeta X.\Lambda \mid \Lambda \lightning \Lambda \mid
\end{array}
$$

where $1 \leq i \leq$ (*length of* Λ).[2]

In Definition 6, $\overline{\mathcal{X}}$ and $\overline{\mathcal{C}}$ are sets of *meta-level variables with substitutions* and *meta-level constants with substitutions*, respectively.

Among MLC terms, those of the forms listed in the first row in the definition of Λ are called *base-level terms*, and those in the second row are called *meta-level terms*. Each row lists a form of a base-level/meta-level variable, a constant, a unit, a finite product, a projection, a lambda abstraction, and a function application, respectively. (Λ) is a bracketed term. The concept of size of terms is defined as follows:

Definition 7 *(Size of Terms)*

$$
\begin{array}{ll}
size(x \mid c \mid \langle\,\rangle \mid \overline{\mathcal{X}} \mid \overline{\mathcal{C}} \mid \langle\langle\,\rangle\rangle) & \stackrel{def}{\equiv} 1 \\
size(\langle M_1, \ldots, M_n \rangle \mid \langle\langle M_1, \ldots, M_n \rangle\rangle) & \stackrel{def}{\equiv} max(size(M_1), \ldots, size(M_n)) + 1 \\
size(\pi_i(M) \mid \lambda x.M \mid p_i(M) \mid \zeta X.M) & \stackrel{def}{\equiv} size(M) + 1 \\
size(M_1 M_2 \mid M_1 \lightning M_2) & \stackrel{def}{\equiv} max(size(M_1), size(M_2)) + 1
\end{array}
$$

2.3 Free Base-level/Meta-level Variables

The sets of *free base-level variables* and *free meta-level variables* are defined respectively by the following two sets of rules.

Definition 8 *(Free Base-level Variables)*

$$
\begin{array}{rcl}
fv(x) & \stackrel{def}{\equiv} & \{x\} \\
fv(c) & \stackrel{def}{\equiv} & \{\} \\
fv(\langle\,\rangle) & \stackrel{def}{\equiv} & \{\} \\
fv(\langle M_1, \ldots, M_n \rangle) & \stackrel{def}{\equiv} & \bigcup_{1 \leq i \leq n} fv(M_i) \\
fv(\pi_i(M)) & \stackrel{def}{\equiv} & fv(M) \\
fv(\lambda x.M) & \stackrel{def}{\equiv} & fv(M) - \{x\} \\
fv(M_1 M_2) & \stackrel{def}{\equiv} & fv(M_1) \cup fv(M_2)
\end{array}
\qquad
\begin{array}{rcl}
fv(X) & \stackrel{def}{\equiv} & \mathfrak{S}(X) \\
fv(\overline{\mathcal{X}}[M/x]) & \stackrel{def}{\equiv} & (fv(\overline{\mathcal{X}}) - \{x\}) \cup fv(M) \\
fv(C) & \stackrel{def}{\equiv} & \{\} \\
fv(\overline{\mathcal{C}}[M/x]) & \stackrel{def}{\equiv} & (fv(\overline{\mathcal{C}}) - \{x\}) \cup fv(M) \\
fv(\langle\langle\,\rangle\rangle) & \stackrel{def}{\equiv} & \{\} \\
fv(\langle\langle M_1, \ldots, M_n \rangle\rangle) & \stackrel{def}{\equiv} & \bigcup_{1 \leq i \leq n} fv(M_i) \\
fv(p_i(M)) & \stackrel{def}{\equiv} & fv(M) \\
fv(\zeta X.M) & \stackrel{def}{\equiv} & fv(M) \\
fv(M_1 \lightning M_2) & \stackrel{def}{\equiv} & fv(M_1)
\end{array}
$$

[2] The *length* of a base-level unit $\langle\rangle$ and a meta-level unit $\langle\langle\rangle\rangle$ is defined to be 0, and a base-level finite product of the form $\langle M_1, \ldots, M_n \rangle$ and a meta-level finite product of the form $\langle\langle M_1, \ldots, M_n \rangle\rangle$ is defined to be n.

Definition 9 *(Free Meta-level Variables)*

$$
\begin{array}{rcl}
fmv(x) & \stackrel{def}{\equiv} & \{\} \\
fmv(c) & \stackrel{def}{\equiv} & \{\} \\
fmv(\langle\,\rangle) & \stackrel{def}{\equiv} & \{\} \\
fmv(\langle M_1, \ldots, M_n \rangle) & \stackrel{def}{\equiv} & \bigcup_{1 \leq i \leq n} fmv(M_i) \\
fmv(\pi_i(M)) & \stackrel{def}{\equiv} & fmv(M) \\
fmv(\lambda x.M) & \stackrel{def}{\equiv} & fmv(M) \\
fmv(M_1 M_2) & \stackrel{def}{\equiv} & fmv(M_1) \cup fmv(M_2)
\end{array}
\qquad
\begin{array}{rcl}
fmv(X) & \stackrel{def}{\equiv} & \{X\} \\
fmv(\overline{\mathcal{X}}[M/x]) & \stackrel{def}{\equiv} & fmv(\overline{\mathcal{X}}) \cup fmv(M) \\
fmv(C) & \stackrel{def}{\equiv} & \{\} \\
fmv(\overline{\mathcal{C}}[M/x]) & \stackrel{def}{\equiv} & fmv(\overline{\mathcal{C}}) \cup fmv(M) \\
fmv(\langle\!\langle\,\rangle\!\rangle) & \stackrel{def}{\equiv} & \{\} \\
fmv(\langle\!\langle M_1, \ldots, M_n \rangle\!\rangle) & \stackrel{def}{\equiv} & \bigcup_{1 \leq i \leq n} fmv(M_i) \\
fmv(p_i(M)) & \stackrel{def}{\equiv} & fmv(M) \\
fmv(\zeta\, X.M) & \stackrel{def}{\equiv} & fmv(M) - \{X\} \\
fmv(M_1 \,\natural\, M_2) & \stackrel{def}{\equiv} & fmv(M_1) \cup fmv(M_2)
\end{array}
$$

2.4 Judgment

A *judgment* in MLC is of the following form:

$$\Delta \Vdash M : \tau$$

where Δ is a meta-level context, M is a raw term, and τ is a meta-level type recursively derived by the following set of rules.

Definition 10 *(Base-level Structural Rules)*

$$\textbf{\textit{Weakening}} \quad (w)\frac{\Delta \Vdash M : (\Gamma \vdash \alpha_1) \qquad x \notin fv(M)}{\Delta \Vdash M : (\Gamma, x : \alpha_2 \vdash \alpha_1)}$$

$$\textbf{\textit{Exchange}} \quad (e)\frac{\Delta \Vdash M : (\Gamma,\ x : \alpha_1,\ y : \alpha_2,\ \Gamma' \vdash \alpha)}{\Delta \Vdash M : (\Gamma,\ y : \alpha_2,\ x : \alpha_1,\ \Gamma' \vdash \alpha)}$$

$$\textbf{\textit{Contraction}} \quad (c)\frac{\Delta \Vdash M : (\Gamma, x : \alpha_2, x : \alpha_2 \vdash \alpha_1)}{\Delta \Vdash M : (\Gamma, x : \alpha_2 \vdash \alpha_1)}$$

Definition 11 *(Meta-level Structural Rules)*

$$\textbf{\textit{Weakening}} \quad (Mw)\frac{\Delta \Vdash M : \tau_1 \qquad X \notin fmv(M)}{\Delta, X : \tau_2 \Vdash M : \tau_1}$$

$$\textbf{\textit{Exchange}} \quad (Me)\frac{\Delta,\ X : \tau_1,\ Y : \tau_2,\ \Delta' \Vdash M : \tau}{\Delta,\ Y : \tau_2,\ X : \tau_1,\ \Delta' \Vdash M : \tau}$$

$$\textbf{\textit{Contraction}} \quad (Mc)\frac{\Delta, X : \tau_2, X : \tau_2 \Vdash M : \tau_1}{\Delta, X : \tau_2 \Vdash M : \tau_1}$$

Definition 12 *(Typing Rules for Base-level Terms)*

Variables $(VAR)\dfrac{}{\Vdash x : (x : \alpha \vdash \alpha)}$

Constants $(CON)\dfrac{}{\Vdash c : (\vdash \alpha)}$

Products $(UNIT)\dfrac{}{\Vdash \langle\ \rangle : (\Gamma \vdash unit)}$

$$(PRD)\frac{\Delta \Vdash M_1 : (\Gamma \vdash \alpha_1)\ \cdots\ \Delta \Vdash M_n : (\Gamma \vdash \alpha_n)}{\Delta \Vdash \langle M_1, \ldots, M_n\rangle : (\Gamma \vdash \alpha_1 \times \cdots \times \alpha_n)}\,(1 \leq n)$$

Projections $$(PJ)\frac{\Delta \Vdash M : (\Gamma \vdash \alpha_1 \times \cdots \times \alpha_n)}{\Delta \Vdash \pi_i(M) : (\Gamma \vdash \alpha_i)}\,(1 \leq i \leq n)$$

Abstractions $$(LAM)\frac{\Delta \Vdash M : (\Gamma, x : \alpha_2 \vdash \alpha_1)}{\Delta \Vdash \lambda x.M : (\Gamma \vdash \alpha_2 \to \alpha_1)}$$

Applications $$(APP)\frac{\Delta \Vdash M_1 : (\Gamma \vdash \alpha_2 \to \alpha_1) \qquad \Delta \Vdash M_2 : (\Gamma \vdash \alpha_2)}{\Delta \Vdash M_1 M_2 : (\Gamma \vdash \alpha_1)}$$

Definition 13 *(Typing Rules for Meta-level Terms)*

Variables $(MVAR)\dfrac{}{X : \tau \Vdash X : \tau}$

$$(\overline{MVAR})\frac{\Delta \Vdash \overline{\mathcal{X}} : S_{\Gamma,\, x:\alpha}(\tau) \quad \Delta \Vdash M : (\Gamma \vdash \alpha)}{\Delta \Vdash \overline{\mathcal{X}}[M/x] : \tau}$$

Constants $(MCON)\dfrac{}{\Vdash C : \tau}$

$$(\overline{MCON})\frac{\Delta \Vdash \overline{\mathcal{C}} : S_{\Gamma,\, x:\alpha}(\tau) \quad \Delta \Vdash M : (\Gamma \vdash \alpha)}{\Delta \Vdash \overline{\mathcal{C}}[M/x] : \tau}$$

Products $(MUNIT)\dfrac{}{\Vdash \langle\!\langle\ \rangle\!\rangle : munit}$

$$(MPRD)\frac{\Delta \Vdash M_1 : \tau_1\ \cdots\ \Delta \Vdash M_n : \tau_n}{\Delta \Vdash \langle\!\langle M_1, \ldots, M_n\rangle\!\rangle : \tau_1 \otimes \cdots \otimes \tau_n}\,(1 \leq n)$$

Projections $$(MPJ)\frac{\Delta \Vdash M : \tau_1 \otimes \cdots \otimes \tau_n}{\Delta \Vdash p_i(M) : \tau_i}\,(1 \leq i \leq n)$$

Abstractions $$(MLAM)\frac{\Delta, X : \tau_2 \Vdash M : \tau_1}{\Delta \Vdash \zeta X.M : \tau_2 \Rightarrow \tau_1}$$

Applications $$(MAPP)\frac{\Delta \Vdash M_1 : \tau_2 \Rightarrow \tau_1 \quad \Delta \Vdash M_2 : \tau_2}{\Delta \Vdash M_1 \natural M_2 : \tau_1}$$

In the substitution rules, $S_{\Gamma,\, x:\alpha}$ (where $x \in \mathcal{V}ar, \alpha \in \mathcal{T}yp$ and Γ is a base-level context) is a type transformer defined as follows:

Definition 14 *(Type Transformer $S_{\Gamma,\, x:\alpha}$)*

$$S_{\Gamma,\, x:\alpha}(\Gamma \vdash \alpha') \stackrel{def}{\equiv} (\Gamma, x : \alpha \vdash \alpha')$$

$$S_{\Gamma,\, x:\alpha}(\tau_2 \Rightarrow \tau_1) \stackrel{def}{\equiv} \tau_2 \Rightarrow S_{\Gamma,\, x:\alpha}(\tau_1)$$

$$S_{\Gamma,\, x:\alpha}(\mathit{munit}) \stackrel{def}{\equiv} \mathit{munit}$$

$$S_{\Gamma,\, x:\alpha}(\tau_1 \otimes \cdots \otimes \tau_n) \stackrel{def}{\equiv} S_{\Gamma,\, x:\alpha}(\tau_1) \otimes \cdots \otimes S_{\Gamma,\, x:\alpha}(\tau_n)$$

2.5 Substitution

Definition 15 *(Substitution of Base-level Variables)*

$$x[L/x] \stackrel{def}{\equiv} L$$
$$y[L/x] \stackrel{def}{\equiv} y \quad \textit{where } y \not\equiv x$$
$$c[L/x] \stackrel{def}{\equiv} c$$
$$\langle\,\rangle[L/x] \stackrel{def}{\equiv} \langle\,\rangle \qquad \langle\!\langle\,\rangle\!\rangle[L/x] \stackrel{def}{\equiv} \langle\!\langle\,\rangle\!\rangle$$
$$\langle M_1, \ldots, M_n\rangle[L/x] \stackrel{def}{\equiv} \langle M_1[L/x], \ldots, M_n[L/x]\rangle \qquad \langle\!\langle M_1, \ldots, M_n\rangle\!\rangle[L/x] \stackrel{def}{\equiv} \langle\!\langle M_1[L/x], \ldots, M_n[L/x]\rangle\!\rangle$$
$$(\pi_i(M))[L/x] \stackrel{def}{\equiv} \pi_i(M[L/x]) \qquad (p_i(M))[L/x] \stackrel{def}{\equiv} p_i(M[L/x])$$
$$(\lambda x.M)[L/x] \stackrel{def}{\equiv} \lambda x.M$$
$$(\lambda y.M)[L/x] \stackrel{def}{\equiv} \lambda y.M[L/x] \quad \textit{where } y \not\equiv x \wedge (x \notin fv(M) \vee y \notin fv(L)) \qquad (\zeta X.M)[L/x] \stackrel{def}{\equiv} \zeta X.M[L/x] \quad \textit{where } x \notin fv(M) \vee X \notin fmv(L)$$
$$(\lambda y.M)[L/x] \stackrel{def}{\equiv} \lambda z.M[z/y][L/x] \quad \textit{where } y \not\equiv x \wedge x \in fv(M) \wedge y \in fv(L) \wedge z \notin fv(M) \cup fv(L) \qquad (\zeta X.M)[L/x] \stackrel{def}{\equiv} \zeta Z.M[Z/X][L/x] \quad \textit{where } x \in fv(M) \wedge X \in fmv(L) \wedge Z \notin fmv(M) \cup fmv(L)$$
$$(M_1 M_2)[L/x] \stackrel{def}{\equiv} (M_1[L/x])(M_2[L/x]) \qquad (M_1 \natural M_2)[L/x] \stackrel{def}{\equiv} (M_1[L/x]) \natural M_2$$

Definition 16 *(Substitution of Meta-level Variables)*

$$x[L/X] \stackrel{def}{\equiv} x \qquad X[L/X] \stackrel{def}{\equiv} L$$
$$Y[L/X] \stackrel{def}{\equiv} Y \quad \textit{where } Y \not\equiv X$$
$$(\overline{\mathcal{X}}[M/x])[L/X] \stackrel{def}{\equiv} (\overline{\mathcal{X}}[L/X])[(M[L/X])/x]$$
$$c[L/X] \stackrel{def}{\equiv} c \qquad C[L/X] \stackrel{def}{\equiv} C$$
$$(\overline{\mathcal{C}}[M/x])[L/X] \stackrel{def}{\equiv} (\overline{\mathcal{C}}[L/X])[(M[L/X])/x]$$
$$\langle\,\rangle[L/X] \stackrel{def}{\equiv} \langle\,\rangle \qquad \langle\!\langle\,\rangle\!\rangle[L/X] \stackrel{def}{\equiv} \langle\!\langle\,\rangle\!\rangle$$
$$\langle M_1, \ldots, M_n\rangle[L/X] \stackrel{def}{\equiv} \langle M_1[L/X], \ldots, M_n[L/X]\rangle \qquad \langle\!\langle M_1, \ldots, M_n\rangle\!\rangle[L/X] \stackrel{def}{\equiv} \langle\!\langle M_1[L/X], \ldots, M_n[L/X]\rangle\!\rangle$$
$$(\pi_i(M))[L/X] \stackrel{def}{\equiv} \pi_i(M[L/X]) \qquad (p_i(M))[L/X] \stackrel{def}{\equiv} p_i(M[L/X])$$
$$(\zeta X.M)[L/X] \stackrel{def}{\equiv} \zeta X.M$$
$$(\lambda x.M)[L/X] \stackrel{def}{\equiv} \lambda x.M[L/X] \qquad (\zeta Y.M)[L/X] \stackrel{def}{\equiv} \zeta Y.M[L/X] \quad \textit{where } X \not\equiv Y \wedge (X \notin fmv(M) \vee Y \notin fmv(L))$$
$$(\zeta Y.M)[L/X] \stackrel{def}{\equiv} \zeta Z.M[Z/Y][L/X] \quad \textit{where } X \not\equiv Y \wedge X \in fmv(M) \wedge Y \in fmv(L) \wedge Z \notin fmv(M) \cup fmv(L)$$
$$(M_1 M_2)[L/X] \stackrel{def}{\equiv} (M_1[L/X])(M_2[L/X]) \qquad (M_1 \natural M_2)[L/X] \stackrel{def}{\equiv} (M_1[L/X]) \natural (M_2[L/X])$$

3 Theory of MLC

Axiom 1 (Equivalence)

$$(=R)\frac{\Delta \Vdash M : \tau}{\Delta \Vdash M = M : \tau} \qquad (=S)\frac{\Delta \Vdash M = N : \tau}{\Delta \Vdash N = M : \tau}$$

$$(=T)\frac{\Delta \Vdash L = M : \tau \quad \Delta \Vdash M = N : \tau}{\Delta \Vdash L = N : \tau}$$

Axiom 2 (Replacement)

$$(=\lambda)\frac{\Delta \Vdash M = N : (\Gamma, x : \alpha_2 \vdash \alpha_1)}{\Delta \Vdash \lambda x.M = \lambda x.N : (\Gamma \vdash \alpha_2 \to \alpha_1)}$$

$$(=F)\frac{\Delta \Vdash M : (\Gamma \vdash \alpha_2 \to \alpha_1) \qquad \Delta \Vdash N_1 = N_2 : (\Gamma \vdash \alpha_2)}{\Delta \Vdash MN_1 = MN_2 : (\Gamma \vdash \alpha_1)}$$

$$(=A)\frac{\Delta \Vdash M_1 = M_2 : (\Gamma \vdash \alpha_2 \to \alpha_1) \qquad \Delta \Vdash N : (\Gamma \vdash \alpha_2)}{\Delta \Vdash M_1 N = M_2 N : (\Gamma \vdash \alpha_1)}$$

$$(=\langle\rangle)\frac{\Delta \Vdash M_1 = N_1 : (\Gamma \vdash \alpha_1) \ \cdots \ \Delta \Vdash M_n = N_n : (\Gamma \vdash \alpha_n)}{\Delta \Vdash \langle M_1, \ldots, M_n\rangle = \langle N_1, \ldots, N_n\rangle : (\Gamma \vdash \alpha_1 \times \cdots \times \alpha_n)}\,(1 \le n)$$

$$(=\pi)\frac{\Delta \Vdash M = N : (\Gamma \vdash \alpha_1 \times \cdots \times \alpha_n)}{\Delta \Vdash \pi_i(M) = \pi_i(N) : (\Gamma \vdash \alpha_i)}\,(1 \le i \le n)$$

Axiom 3 (Meta-level Replacement)

$$(=M\lambda)\frac{\Delta, X : \tau_2 \Vdash M = N : \tau_1}{\Delta \Vdash \zeta X.M = \zeta X.N : \tau_2 \Rightarrow \tau_1}$$

$$(=MF)\frac{\Delta \Vdash M : \tau_2 \Rightarrow \tau_1 \qquad \Delta \Vdash N_1 = N_2 : \tau_2}{\Delta \Vdash M \not\downarrow N_1 = M \not\downarrow N_2 : \tau_1}$$

$$(=MA)\frac{\Delta \Vdash M_1 = M_2 : \tau_2 \Rightarrow \tau_1 \qquad \Delta \Vdash N : \tau_2}{\Delta \Vdash M_1 \not\downarrow N = M_2 \not\downarrow N : \tau_1}$$

$$(=\langle\!\langle\rangle\!\rangle)\frac{\Delta \Vdash M_1 = N_1 : \tau_1 \ \cdots \ \Delta \Vdash M_n = N_n : \tau_n}{\Delta \Vdash \langle\!\langle M_1, \ldots, M_n\rangle\!\rangle = \langle\!\langle N_1, \ldots, N_n\rangle\!\rangle : \tau_1 \otimes \cdots \otimes \tau_n}\,(1 \le n)$$

$$(=p)\frac{\Delta \Vdash M = N : \tau_1 \otimes \cdots \otimes \tau_n}{\Delta \Vdash p_i(M) = p_i(N) : \tau_i}\,(1 \le i \le n)$$

Axiom 4 (Base-level Product Equations)

$$(\pi 0)\frac{\Delta \Vdash M : (\Gamma \vdash unit)}{\Delta \Vdash M = \langle\,\rangle : (\Gamma \vdash unit)}$$

$$(\pi 1)\frac{\Delta \Vdash M : (\Gamma \vdash \alpha_1 \times \cdots \times \alpha_n)}{\Delta \Vdash \langle \pi_1(M), \cdots, \pi_n(M)\rangle = M : (\Gamma \vdash \alpha_1 \times \cdots \times \alpha_n)}\ (1 \leq n)$$

$$(\pi 2)\frac{\Delta \Vdash M_1 : (\Gamma \vdash \alpha_1)\ \cdots\ \Delta \Vdash M_n : (\Gamma \vdash \alpha_n)}{\Delta \Vdash \pi_i(\langle M_1, \ldots, M_n\rangle) = M_i : (\Gamma \vdash \alpha_i)}\ (1 \leq i \leq n)$$

Axiom 5 (Meta-level Product Equations)

$$(p0)\frac{\Delta \Vdash M : munit}{\Delta \Vdash M = \langle\!\langle\,\rangle\!\rangle : munit}$$

$$(p1)\frac{\Delta \Vdash M : \tau_1 \otimes \cdots \otimes \tau_n}{\Delta \Vdash \langle\!\langle p_1(M), \ldots, p_n(M)\rangle\!\rangle = M : \tau_1 \otimes \cdots \otimes \tau_n}\ (1 \leq n)$$

$$(p2)\frac{\Delta \Vdash M_1 : \tau_1\ \cdots\ \Delta \Vdash M_n : \tau_n}{\Delta \Vdash p_i(\langle\!\langle M_1, \ldots, M_n\rangle\!\rangle) = M_i : \tau_i}\ (1 \leq i \leq n)$$

Axiom 6 (Base-level/Meta-level Conversions)

$$(\alpha)\frac{\Delta \Vdash M : (\Gamma, x : \alpha_2 \vdash \alpha_1) \qquad y \notin fv(M)}{\Delta \Vdash \lambda x.M = \lambda y.M[y/x] : (\Gamma \vdash \alpha_2 \to \alpha_1)} \qquad (M\alpha)\frac{\Delta, X : \tau_2 \Vdash M : \tau_1 \qquad Y \notin fmv(M)}{\Delta \Vdash \zeta X.M = \zeta Y.M[Y/X] : \tau_2 \Rightarrow \tau_1}$$

$$(\beta)\frac{\Delta \Vdash M_1 : (\Gamma, x : \alpha_2 \vdash \alpha_1) \qquad \Delta \Vdash M_2 : (\Gamma \vdash \alpha_2)}{\Delta \Vdash (\lambda x.M_1)M_2 = M_1[M_2/x] : (\Gamma \vdash \alpha_1)} \qquad (M\beta)\frac{\Delta, X : \tau_2 \Vdash M_1 : \tau_1 \qquad \Delta \Vdash M_2 : \tau_2}{\Delta \Vdash (\zeta X.M_1)\natural M_2 = M_1[M_2/X] : \tau_1}$$

$$(\eta)\frac{\Delta \Vdash M : (\Gamma \vdash \alpha_2 \to \alpha_1) \qquad x \notin fv(M)}{\Delta \Vdash \lambda x.Mx = M : (\Gamma \vdash \alpha_2 \to \alpha_1)} \qquad (M\eta)\frac{\Delta \Vdash M : \tau_2 \Rightarrow \tau_1 \qquad X \notin fmv(M)}{\Delta \Vdash \zeta X.M\natural X = M : \tau_2 \Rightarrow \tau_1}$$

These conversions are proved to be sound in the categorical semantics introduced in the next section. For the proof of soundness, see Sect. 6 and Sect. 7.

4 Linguistic Monad and Monadic Translations

MLC provides a natural way for representing monads (called *internal monad* in Bekki (2009) in contrast with monads in category theory), and define such translation in which internal monads serves as parameters.

Definition 17 *(Internal Monad)* An *internal monad* is a triple $\langle \boldsymbol{T}, \eta, \mu\rangle$ (where $\boldsymbol{T}, \eta, \mu$ are MLC terms) that satisfies the following four sets of equations, where $\circ$ is a composition operator: $f \circ g \stackrel{def}{\equiv} \zeta X.f\natural(g\natural X)$

$\boldsymbol{T}$ conditions: $\boldsymbol{T} \natural (\zeta X.X) = \zeta X.X \quad (\boldsymbol{T} \natural g) \circ (\boldsymbol{T} \natural f) = \boldsymbol{T} \natural (g \circ f)$
η and μ conditions: $(\boldsymbol{T} \natural f) \circ \eta = \eta \circ f \quad (\boldsymbol{T} \natural f) \circ \mu = \mu \circ (\boldsymbol{T} \natural (\boldsymbol{T} \natural f))$
Square identity: $\mu \circ (\boldsymbol{T} \natural \mu) = \mu \circ \mu$
Triangular identity: $\mu \circ \eta = \zeta X.X \quad \mu \circ (\boldsymbol{T} \natural \eta) = \zeta X.X$

The four sets of conditions above are sufficient for an internal monad to correspond to a categorical monad (Bekki 2009). By specifying an internal monad, a monadic translation is defined as follows:

Definition 18 *(Translation with Internal Monad: call-by-value* (Bekki 2009)

$$
\begin{aligned}
[\![x]\!]_{\boldsymbol{T}} &\stackrel{def}{\equiv} \eta \natural x \\
[\![c]\!]_{\boldsymbol{T}} &\stackrel{def}{\equiv} \eta \natural c \\
[\![\lambda x.M]\!]_{\boldsymbol{T}} &\stackrel{def}{\equiv} \boldsymbol{T} \natural (\zeta X.\lambda x.X) \natural [\![M]\!]_{\boldsymbol{T}} \\
[\![MN]\!]_{\boldsymbol{T}} &\stackrel{def}{\equiv} \mu \natural ((\boldsymbol{T} \natural (\zeta X.((\boldsymbol{T} \natural \zeta Y.(X \natural Y)) \natural [\![N]\!]_{\boldsymbol{T}}))) \natural [\![M]\!]_{\boldsymbol{T}}) \\
[\![\langle\,\rangle]\!]_{\boldsymbol{T}} &\stackrel{def}{\equiv} \eta \natural \langle\,\rangle \\
[\![\langle M, N \rangle]\!]_{\boldsymbol{T}} &\stackrel{def}{\equiv} \mu \natural ((\boldsymbol{T} \natural (\zeta Y.((\boldsymbol{T} \natural \zeta X.\langle X, Y \rangle) \natural [\![M]\!]_{\boldsymbol{T}}))) \natural [\![N]\!]_{\boldsymbol{T}}) \\
[\![\pi_i(M)]\!]_{\boldsymbol{T}} &\stackrel{def}{\equiv} \boldsymbol{T} \natural (\zeta X.\pi_i(X)) \qquad \textit{where } i = 1, 2
\end{aligned}
$$

Bekki (2009) establishes a link between a monadic translation (by means of an internal monad) and a categorical monad. In other words, an internal monad *is* a monad in a corresponding category called Δ-indexed category.[3]

4.1 Non-determinism

The sentence (5) is ambiguous with respect to at least two factors: the antecedent of the pronoun 'he' and the lexical meaning of 'a suit' (clothing or a legal action).

(5) He brought a suit.

Suppose that there are currently two possible antecedents for the subject pronoun, say 'John' and 'Bill'. Then, in the context of standard natural language processing, a parser is expected to spell out the following set of semantic representations for the input sentence (5).

(6) $\{\ brought'(suit_1)(j'), brought'(suit_2)(j'),$
$brought'(suit_1)(b'), (brought')(suit_2)(b')\ \}$

[3] See Theorem 9 in Bekki (2009).

But this kind of 'duplication' of output trees is known to bring about a combinatorial explosion in parsing complexity, which has motivated the pursuit of an 'information packing' strategy. The monadic treatment of such non-deterministic information has been discussed in Shan (2001) and Bekki (2009), for which the internal monad for non-determinism is defined as follows in Bekki (2009).

Definition 19 *(Internal Monad for Non-determinism)*

$$\begin{aligned} \boldsymbol{T}_{nd} &\stackrel{def}{\equiv} \zeta F.\zeta X.\{F\not\downarrow x \mid x \in X\} \\ \eta_{nd} &\stackrel{def}{\equiv} \zeta X.\{X\} \\ \mu_{nd} &\stackrel{def}{\equiv} \zeta X.\bigcup X \end{aligned}$$

The internal monad specified in Definition 19 defines a translation $[\![-]\!]_{nd}$ as in (7) via Definition 18.

(7) Translation to Non-deterministic Monads:

$$\begin{aligned} [\![x]\!]_{nd} &= \{x\} \\ [\![c]\!]_{nd} &= \{c\} \\ [\![\lambda x.M]\!]_{nd} &= \left\{\lambda x.m \;\middle|\; m \in [\![M]\!]_{nd}\right\} \\ [\![MN]\!]_{nd} &= \left\{mn \;\middle|\; m \in [\![M]\!]_{nd} \wedge n \in [\![N]\!]_{nd}\right\} \\ [\![\langle\,\rangle]\!]_{nd} &= \{\langle\,\rangle\} \\ [\![\langle M, N\rangle]\!]_{nd} &= \left\{\langle m, n\rangle \;\middle|\; m \in [\![M]\!]_{nd} \wedge n \in [\![N]\!]_{nd}\right\} \\ [\![\pi_i(M)]\!]_{nd} &= \left\{\pi_i(m) \;\middle|\; m \in [\![M]\!]_{nd}\right\} \end{aligned}$$

Then, each ambiguity due to the antecedent of 'he' and the lexical ambiguity of 'a suit' can be lexically represented in the following way.

(8)
$$\begin{aligned} [\![he']\!]_{nd} &\stackrel{def}{\equiv} \left\{j', b'\right\} \\ [\![suit']\!]_{nd} &\stackrel{def}{\equiv} \{suit_1, suit_2\} \end{aligned}$$

Using these expressions, the set of representations (6) can be packed into the single representation (9).

(9) $(brought'(suit'))(he')$

The non-deterministic aspects in the sentence (5) are successfully encapsulated within (9). The following equations show that the monadic translation of (9) is equivalent to (6).

(10)
$$\begin{aligned} & [\![(brought'(suit'))]\!]_{nd} \\ &= \left\{mn \;\middle|\; m \in [\![brought']\!]_{nd} \wedge n \in [\![suit']\!]_{nd}\right\} \\ &= \left\{mn \;\middle|\; m \in \left\{brought'\right\} \wedge n \in \{suit_1, suit_2\}\right\} \end{aligned}$$

$$= \{brought'(suit_1), brought'(suit_2)\}$$

$$\begin{aligned}
&[\![(brought'(suit'))(he')]\!]_{nd} \\
&= \{mn \mid m \in [\![brought'(suit')]\!]_{nd} \wedge n \in [\![he']\!]_{nd}\} \\
&= \{mn \mid m \in \{brought'(suit_1), brought'(suit_2)\} \wedge n \in \{j', b'\}\} \\
&= \{brought'(suit_1)(j'), brought(suit_2)(j'), \\
&\quad brought'(suit_1)(b'), brought(suit_2)(b')\}
\end{aligned}$$

Generally, a non-deterministic monad is called a *powerset monad* and has applications other than the representation of non-deterministic information. In Hayashishita and Bekki (2011), we defined a *disjunctive monad*, which is isomorphic to the non-deterministic monad, and this allows for the interpretation of various types of conjoined nominals in Japanese.

4.2 Contextual Parameters

Contextual parameters such as speaker/hearer, topic, and point of view, can be analyzed by means of the internal monad for contextual parameters.

Definition 20 *(Internal Monad for Contextual Parameters)*

$$\begin{aligned}
\boldsymbol{T}_{cp} &\stackrel{def}{\equiv} \zeta F.\zeta X.\lambda h.\langle F(\pi_1(x)), \pi_2(x)\rangle[Xh/x] \\
\eta_{cp} &\stackrel{def}{\equiv} \zeta X.\lambda h.\langle X, h\rangle \\
\mu_{cp} &\stackrel{def}{\equiv} \zeta X.\lambda h.(\pi_1(x))(\pi_2(x))[Xh/x]
\end{aligned}$$

The internal monad specified in Definition 20 defines a translation $[\![-]\!]_{cp}$ as in (11) via Definition 18.

(11) Translation to Contextual Parameter Monads:

$$\begin{aligned}
[\![x]\!]_{cp} &= \lambda h.\langle x, h\rangle \\
[\![c]\!]_{cp} &= \lambda h.\langle c, h\rangle \\
[\![\lambda x.M]\!]_{cp} &= \lambda h.\langle \lambda x.\pi_1(m), \pi_2(m)\rangle[[\![M]\!]_{cp}h/m] \\
[\![MN]\!]_{cp} &= \lambda h.\langle \pi_1(m)\pi_1(n), \pi_2(n)\rangle[[\![N]\!]_{cp}(\pi_2(m))/n][[\![M]\!]_{cp}h/m] \\
[\![\langle\,\rangle]\!]_{cp} &= \lambda h.\langle\langle\,\rangle, h\rangle \\
[\![\langle M, N\rangle]\!]_{cp} &= \lambda h.\langle\langle \pi_1(m), \pi_1(n)\rangle, \pi_2(n)\rangle[[\![N]\!]_{cp}(\pi_2(m))/n][[\![M]\!]_{cp}h/m] \\
[\![\pi_i(M)]\!]_{cp} &= \lambda h.\langle \pi_i(\pi_1(m)), \pi_2(m)\rangle[[\![M]\!]_{cp}h/m]
\end{aligned}$$

In this case, contextual parameters can be easily referenced in semantic representations and can also be changed.

A contextual monad provides control operators to represent a contextual parameter such as a hearer. By using $set\_hearer(x)$, we can set the current hearer to x, even in the middle of a sentence. By using $hearer()$, we can refer to a current hearer.

(12) $$[\![set\_hearer(x)]\!]_{cp} \stackrel{def}{\equiv} \lambda h.\langle \top, x \rangle$$
$$[\![hearer()]\!]_{cp} \stackrel{def}{\equiv} \lambda h.\langle h, h \rangle$$

For example, the semantic representation of the sentence (13) can be simply stated as (14).

(13) (Pointing to John) You passed, (pointing to Mary) and you passed.

(14) $set\_hearer(j') \wedge passed'(hearer()) \wedge set\_hearer(m') \wedge passed'(hearer())$

When (14) is translated by $[\![-]\!]_{cp}$, each occurrence of 'you' successfully refers to the intended individual, although the two representations for 'you passed' in (14) are exactly the same. Suppose that $A \wedge B = \wedge(\langle A, B \rangle)$.

(15)
$$\begin{aligned}
[\![\wedge]\!]_{cp} &= \lambda h.\langle \wedge, h \rangle \\
[\![passed']\!]_{cp} &= \lambda h.\langle passed', h \rangle \\
[\![passed'(hearer())]\!]_{cp} &= \lambda h.\langle passed'(h), h \rangle \\
[\![\langle set_hearer(j'), passed'(hearer()) \rangle]\!]_{cp} &= \lambda h.\langle \langle \top, passed'(j') \rangle, j' \rangle \\
[\![set_hearer(j') \wedge passed'(hearer())]\!]_{cp} &= \lambda h.\langle \top \wedge passed'(j'), j' \rangle \\
&= \lambda h.\langle passed'(j'), j' \rangle
\end{aligned}$$

Therefore, the following result is obtained.

(16) $[\![set\_hearer(j') \wedge passed'(hearer()) \wedge set\_hearer(m') \wedge passed'(hearer())]\!]_{cp}$
$= \lambda h.\langle passed'(j') \wedge passed'(m'), m' \rangle$

4.3 Continuations

The effect of covert movements, involved in focus movements and inverse scope, can be simulated in the type-theoretical framework by the internal monad for delimited continuations (Bekki and Asai 2010).

Definition 21 *(Internal Monad for Deliminted Continuations)*

$$\begin{aligned}
\boldsymbol{T}_c &\stackrel{def}{\equiv} \zeta F.\zeta X.\zeta \kappa.X \nmid (\zeta V.(\kappa \nmid (F \nmid V))) \\
\eta_c &\stackrel{def}{\equiv} \zeta X.\zeta \kappa.(\kappa \nmid X) \\
\mu_c &\stackrel{def}{\equiv} \zeta X.\zeta \kappa.(X \nmid (\zeta V.V \nmid \kappa))
\end{aligned}$$

The internal monad specified in Definition 21 defines a translation $[\![-]\!]_c$ as in (17) via Definition 18.

(17) Translation to Continuation Monad:

$$\begin{aligned}
[\![x]\!]_c &= \zeta\kappa.\kappa \diagup x \\
[\![c]\!]_c &= \zeta\kappa.\kappa \diagup c \\
[\![\lambda x.M]\!]_c &= \zeta\kappa.[\![M]\!]_c \diagup (\zeta v.\kappa \diagup (\lambda x.v)) \\
[\![MN]\!]_c &= \zeta\kappa.[\![M]\!]_c \diagup (\zeta m.[\![N]\!]_c \diagup (\zeta n.\kappa \diagup (m \diagup n))) \\
[\![\langle\,\rangle]\!]_c &= \zeta\kappa.\kappa \diagup \langle\,\rangle \\
[\![\langle M, N\rangle]\!]_c &= \zeta\kappa.[\![M]\!]_c \diagup (\zeta m.[\![N]\!]_c \diagup (\zeta n.\kappa \diagup \langle m, n\rangle)) \\
[\![\pi_i(M)]\!]_c &= \zeta\kappa.[\![M]\!]_c \diagup (\zeta m.(\kappa \diagup \pi_i(m)))
\end{aligned}$$

In light of this setting, the operators for *delimited continuations*, such as *shift* and *reset* in Danvy and Filinski (1990), are defined in the following way.

(18)
$$\begin{aligned}
[\![shift\ \kappa.M]\!]_c &\stackrel{def}{\equiv} \zeta\kappa.([\![M]\!]_c \diagup (\zeta x.x)) \\
[\![reset(M)]\!]_c &\stackrel{def}{\equiv} \zeta\kappa.\kappa \diagup ([\![M]\!]_c \diagup (\zeta x.x))
\end{aligned}$$

Although the definition above may look too simple as compared with that in Danvy and Filinski (1990), it has the same effects, as indicated by the following computations (see Bekki and Asai (2010) for the details).

(19)
$$\begin{aligned}
[\![1 + reset(10 + shift\, f.(f(f(100))))]\!] &= \zeta\kappa.(\kappa \diagup 121) \\
[\![1 + reset(10 + shift\, f.(100))]\!] &= \zeta\kappa.(\kappa \diagup 101) \\
[\![1 + reset(10 + shift\, f.(f(100) + f(1000)))]\!] &= \zeta\kappa.(\kappa \diagup 1121)
\end{aligned}$$

5 Categorical Semantics of MLC

An *interpretation* of MLC terms for an alphabet $\langle \mathcal{GT}, \mathcal{C}on, \mathcal{M}con, \mathcal{V}ar, \mathcal{M}var, \mathfrak{S}\rangle$ is specified by a quadruple as follows:

$$\mathcal{M} = \langle \mathcal{C}, valT, valC, valMC\rangle$$

where $\mathcal{C}$ is a Cartesian closed category with small hom-sets, $valT$ is a function that sends each $\gamma \in \mathcal{GT}$ to an object in $\mathcal{C}$, and $valC$ and $valMC$ are functions that send each $c \in \mathcal{C}on$ and each $C \in \mathcal{M}con$ to a global element in $\mathcal{C}$ and $\mathcal{S}et$, respectively.

5.1 Interpretation of Bas-level/Meta-level Types

Given a quadruple, a *base-level/meta-level type interpretation* $[\![-]\!]$ is defined as follows.

Definition 22 *(Interpretation of Base-level/Meta-level Types)* $[\![-]\!]$ maps each member of $\mathcal{T}yp$ to an object in $\mathcal{C}$ and $\mathcal{M}typ$ to an object in $\mathcal{S}et$ via the following rules:

$$[\![\gamma]\!] = valT(\gamma) \qquad [\![\Gamma \vdash \alpha]\!] = \mathcal{C}([\![\Gamma]\!], [\![\alpha]\!])$$
$$[\![\alpha_2 \to \alpha_1]\!] = [\![\alpha_1]\!]^{[\![\alpha_2]\!]} \qquad [\![\tau_2 \Rightarrow \tau_1]\!] = [\![\tau_1]\!]^{[\![\tau_2]\!]}$$
$$[\![unit]\!] = 1 \qquad [\![munit]\!] = *$$
$$[\![\alpha_1 \times \cdots \times \alpha_n]\!] = [\![\alpha_1]\!] \times \cdots \times [\![\alpha_n]\!] \qquad [\![\tau_1 \otimes \cdots \otimes \tau_n]\!] = [\![\tau_1]\!] \times \cdots \times [\![\tau_n]\!]$$

where $\gamma \in \mathcal{GT}$, 1 is a selected terminal object in $\mathcal{C}$, and $*$ is a selected terminal object in $\mathcal{S}$et, namely, a singleton set.

Definition 23 *(Interpretation of Base-level/Meta-level Contexts)* Suppose that $\Gamma = x_1 : \alpha_1, \ldots, x_n : \alpha_n$ and $\Delta = X_1 : \tau_1, \ldots, X_n : \tau_n$. Then, $[\![\Gamma]\!] = [\![\alpha_1]\!] \times \cdots \times [\![\alpha_n]\!]$ and $[\![\Delta]\!] = [\![\tau_1]\!] \times \cdots \times [\![\tau_n]\!]$. When $n = 0$, $[\![\Gamma]\!] = 1$ and $[\![\Delta]\!] = *$.

In contrast to the standard categorical semantics[4] of STLC where the interpretation $[\![-]\!]$ of a lambda term of type α under a base-level context Γ is a morphism: $[\![\Gamma]\!] \longrightarrow [\![\alpha]\!]$ in a Cartesian closed category $\mathcal{C}$, an MLC term that is of base-level type α under a base-level context Γ is interpreted under a meta-level context Δ as a morphism $[\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha]\!])$ in $\mathcal{S}et$.

5.2 Interpretation of Structural Rules

Definition 24 *(Base-level Structural Rules)* Let $|\Gamma| = n$ and $|\Gamma'| = n'$.

$$(w)\ \frac{[\![\Delta \Vdash M : (\Gamma \vdash \alpha_1)]\!] = m : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha_1]\!]) \quad x \notin fv(M)}{[\![\Delta \Vdash M : (\Gamma, x : \alpha_2 \vdash \alpha_1)]\!] = \mathcal{C}(\langle \pi_1, \ldots, \pi_n \rangle, [\![\alpha_1]\!]) \circ m : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!] \times [\![\alpha_2]\!], [\![\alpha_1]\!])}$$

$$(e)\ \frac{[\![\Delta \Vdash M : (\Gamma,\ x : \alpha_1,\ y : \alpha_2,\ \Gamma' \vdash \alpha)]\!] = m : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!] \times [\![\alpha_1]\!] \times [\![\alpha_2]\!] \times [\![\Gamma']\!], [\![\alpha]\!])}{\begin{array}{l}[\![\Delta \Vdash M : (\Gamma,\ y : \alpha_2,\ x : \alpha_1,\ \Gamma' \vdash \alpha)]\!] = \mathcal{C}(\langle \pi_1, \ldots, \pi_n, \pi_{n+2}, \pi_{n+1}, \pi_{n+3}, \ldots, \pi_{n+n'+2} \rangle, [\![\alpha]\!]) \circ m \\ \qquad : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!] \times [\![\alpha_2]\!] \times [\![\alpha_1]\!] \times [\![\Gamma']\!], [\![\alpha]\!])\end{array}}$$

$$(c)\ \frac{[\![\Delta \Vdash M : (\Gamma, x : \alpha_2, x : \alpha_2 \vdash \alpha_1)]\!] = m : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!] \times [\![\alpha_2]\!] \times [\![\alpha_2]\!], [\![\alpha_1]\!])}{[\![\Delta \Vdash M : (\Gamma, x : \alpha_2 \vdash \alpha_1)]\!] = \mathcal{C}(\langle \pi_1, \ldots, \pi_n, \pi_{n+1}, \pi_{n+1} \rangle, [\![\alpha_1]\!]) \circ m : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!] \times [\![\alpha_2]\!], [\![\alpha_1]\!])}$$

In case $n = 0$ (namely, $[\![\Gamma]\!] = 1$), $\langle \pi_1, \ldots, \pi_n \rangle = !_{[\![\alpha_2]\!]}$.

[4] See, for example, Lambek (1980), Lambek and Scott (1986), and Crole (1993).

Definition 25 *(Meta-level Structural Rules)* Let $|\Delta| = d$ and $|\Delta'| = d'$.

$$(Mw)\ \frac{[\![\Delta \Vdash M : \tau_1]\!] = m : [\![\Delta]\!] \longrightarrow [\![\tau_1]\!] \qquad X \notin fmv(M)}{[\![\Delta,\ X : \tau_2 \Vdash M : \tau_1]\!] = m \circ \langle \pi_1, \ldots, \pi_d \rangle : [\![\Delta]\!] \times [\![\tau_2]\!] \longrightarrow [\![\tau_1]\!]}$$

$$(Me)\ \frac{[\![\Delta,\ X : \tau_1,\ Y : \tau_2,\ \Delta' \Vdash M : \tau]\!] = m : [\![\Delta]\!] \times [\![\tau_1]\!] \times [\![\tau_2]\!] \times [\![\Delta']\!] \longrightarrow [\![\tau]\!]}{\begin{array}{l}[\![\Delta,\ Y : \tau_2,\ X : \tau_1,\ \Delta' \Vdash M : \tau]\!] = m \circ \langle \pi_1, \ldots, \pi_d, \pi_{d+2}, \pi_{d+1}, \pi_{d+3}, \ldots, \pi_{d+d'+2} \rangle \\ \qquad\qquad : [\![\Delta]\!] \times [\![\tau_2]\!] \times [\![\tau_1]\!] \times [\![\Delta']\!] \longrightarrow [\![\tau]\!]\end{array}}$$

$$(Mc)\ \frac{[\![\Delta,\ X : \tau_2,\ X : \tau_2 \Vdash M : \tau_1]\!] = m : [\![\Delta]\!] \times [\![\tau_2]\!] \times [\![\tau_2]\!] \longrightarrow [\![\tau_1]\!]}{[\![\Delta,\ X : \tau_2 \Vdash M]\!] = m \circ \langle \pi_1, \ldots, \pi_d, \pi_{d+1}, \pi_{d+1} \rangle : [\![\Delta]\!] \times [\![\tau_2]\!] \longrightarrow [\![\tau_1]\!]}$$

In case $d = 0$ (namely, $[\![\Delta]\!] = *$), $\langle \pi_1, \ldots, \pi_d \rangle = !_{[\![\tau_2]\!]}$.

5.3 Interpretation of Base-level Terms

There is a bijection between the two hom-sets below, which is natural in B.[5]

(20)

$$\mathcal{C}(A, B) \underset{\overline{tp}:\ g \longmapsto g(*)}{\overset{tp:\ f \longmapsto (* \mapsto f)}{\rightleftarrows}} \mathcal{S}et(*, \mathcal{C}(A, B))$$

This bijection amounts to a following natural isomorphism:

$$\mathcal{C}(A, -) \cong \mathcal{S}et(*, \mathcal{C}(A, -))$$

Therefore, elements of MLC with no meta-level variables (namely, Δ is empty and $[\![\Delta]\!] = *$) are in a one-to-one correspondence with elements of STLC. The covariant functor $\mathcal{C}(A, -)$ used here is a *Yoneda functor* $Y(A)$.[6]

The interpretation of (base-level) variables and constant symbols utilizes the abovementioned natural isomorphism. Any interpretation of a STLC term M of type

---

[5] See MacLane (1997), p. 60 for a proof of Proposition 2.

[6] See MacLane (1997), p. 34. $Y([\![\Gamma]\!]) = \mathcal{C}([\![\Gamma]\!], -) : \mathcal{C} \longrightarrow \mathcal{S}et$ maps a morphism $f : A \longrightarrow B$ in $\mathcal{C}$ to the morphism $\mathcal{C}([\![\Gamma]\!], f) : \mathcal{C}([\![\Gamma]\!], A) \longrightarrow \mathcal{C}([\![\Gamma]\!], B)$ in $\mathcal{S}et$. $Y([\![\Gamma]\!])(f) = \mathcal{C}([\![\Gamma]\!], f)$ is (Footnote 6 continued)
also written as f_* and called "composition with f on the left" or "the map induced by f." It then maps a morphism $a : [\![\Gamma]\!] \longrightarrow A$ in $\mathcal{C}([\![\Gamma]\!], A)$ to $f \circ a : [\![\Gamma]\!] \longrightarrow B$ in $\mathcal{C}([\![\Gamma]\!], B)$. The two morphisms $\mathcal{C}([\![\Gamma]\!], f)$ and $\mathcal{C}([\![\Gamma]\!], g)$ induced by two composable morphisms $f : A \longrightarrow B$ and $g : B \longrightarrow C$ are also composable in $\mathcal{S}et$, as indicated in the following diagram.

$$\begin{array}{ccccc} [\![\Gamma]\!] & & & & \\ {\scriptstyle a}\downarrow & & & & \\ A & \xrightarrow{\ f\ } & B & \xrightarrow{\ g\ } & C \end{array}$$

α by the standard categorical semantics, which is also an element of $\mathcal{C}([\![\Gamma]\!], [\![\alpha]\!])$, is mapped to the corresponding element in $\mathcal{S}et(*, \mathcal{C}([\![\Gamma]\!], [\![\alpha]\!]))$ via the (component $tp_{[\![\alpha]\!]}$ of) natural transformation tp.

Definition 26 *(Base-level Variables)*

$$\overline{[\![\Vdash x : (x : \alpha \vdash \alpha)]\!] = tp_{[\![\alpha]\!]}(id_{[\![\alpha]\!]}) : * \longrightarrow \mathcal{C}([\![\alpha]\!], [\![\alpha]\!])}$$

Definition 27 *(Base-level Constant Symbols)*

$$\overline{[\![\Vdash c : (\vdash \alpha)]\!] = tp_{[\![\alpha]\!]}(valC(c)) : * \longrightarrow \mathcal{C}(1, [\![\alpha]\!])}$$

Since a projection π_{n+1} in $\mathcal{C}$ is a morphism $[\![\Gamma]\!] \times [\![\alpha]\!] \times [\![\Gamma']\!] \longrightarrow [\![\alpha]\!]$, its transpose $tp_{[\![\alpha]\!]}(\pi_{n+1})$ is a morphism $* \longrightarrow \mathcal{C}([\![\Gamma]\!] \times [\![\alpha]\!] \times [\![\Gamma']\!], [\![\alpha]\!])$ in $\mathcal{S}et$.

Theorem 7 **(Base-level Variables and Constants)** Suppose that $|\Gamma| = n$.

$$\overline{[\![\Delta \Vdash x : (\Gamma, x : \alpha, \Gamma' \vdash \alpha)]\!] = tp_{[\![\alpha]\!]}(\pi_{n+1}) \circ !_{[\![\Delta]\!]} : [\![\Delta]\!] \longrightarrow * \longrightarrow \mathcal{C}([\![\Gamma]\!] \times [\![\alpha]\!] \times [\![\Gamma']\!], [\![\alpha]\!])}$$

$$\overline{[\![\Delta \Vdash c : (\Gamma \vdash \alpha)]\!] = tp_{[\![\alpha]\!]}(valC(c) \circ !_{[\![\Gamma]\!]}) \circ !_{[\![\Delta]\!]} : [\![\Delta]\!] \longrightarrow * \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha]\!])}$$

Proof By Definition 26, Definition 27, Base-level weakening and exchange, and Meta-level weakening.

The following pair of maps $\ltimes$ and $\rtimes$ is a bijection, for a Yoneda functor $\mathcal{C}(C, -)$ preserves all finite limits.[7]

$$\mathcal{C}(C, A) \times \mathcal{C}(C, B) \underset{\rtimes : h \longmapsto (\pi_1 \circ h, \pi_2 \circ h)}{\overset{\ltimes : (f,g) \longmapsto \langle f, g \rangle}{\rightleftarrows}} \mathcal{C}(C, A \times B) \tag{21}$$

Since this bijection is natural in C, it amounts to the following natural isomorphism that consists of the components $\ltimes_C$ and $\rtimes_C$.

$$\mathcal{C}(-, A) \times \mathcal{C}(-, B) \cong \mathcal{C}(-, A \times B)$$

This natural isomorphism extends to all finite products. We use the same symbol for its components $\ltimes_C$ and $\rtimes_C$ when no confusion seems to arise.

$$\mathcal{C}(-, A_1) \times \cdots \times \mathcal{C}(-, A_n) \cong \mathcal{C}(-, A_1 \times \cdots \times A_n)$$

[7] See MacLane (1997), p.116 for "Theorem 1" and its proof.

Definition 28 *(Base-level Products)*

$$\frac{\begin{array}{c}\overline{[\![\Vdash \langle\,\rangle : (\Gamma \vdash unit)]\!] = tp_1(!_{[\![\Gamma]\!]}) : * \longrightarrow \mathcal{C}([\![\Gamma]\!], 1)} \\ [\![\Delta \Vdash M_1 : (\Gamma \vdash \alpha_1)]\!] = m_1 : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha_1]\!]) \\ \vdots \\ [\![\Delta \Vdash M_n : (\Gamma \vdash \alpha_n)]\!] = m_n : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha_n]\!])\end{array}}{\begin{array}{l}[\![\Delta \Vdash \langle M_1, \ldots, M_n\rangle : (\Gamma \vdash \alpha_1 \times \cdots \times \alpha_n)]\!] = \ltimes_{[\![\Gamma]\!]} \circ \langle m_1, \ldots, m_n\rangle \\ \quad : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha_1]\!]) \times \cdots \times \mathcal{C}([\![\Gamma]\!], [\![\alpha_n]\!]) \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha_1]\!] \times \cdots \times [\![\alpha_n]\!])\end{array}} (1 \leq n)$$

$$\frac{[\![\Delta \Vdash M : (\Gamma \vdash \alpha_1 \times \cdots \times \alpha_n)]\!] = m : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha_1]\!] \times \cdots \times [\![\alpha_n]\!])}{[\![\Delta \Vdash \pi_i(M) : (\Gamma \vdash \alpha_i)]\!] = \mathcal{C}([\![\Gamma]\!], \pi_i) \circ m : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha_i]\!])} (1 \leq i \leq n)$$

We assume that $\mathcal{C}$ is Cartesian closed and there is a bijection between the following two hom-sets, which is natural in C.

$$\mathcal{C}(C \times A, B) \underset{\hat{\_} : g \longmapsto ev \circ (g \times id_A)}{\overset{\lambda : f \longmapsto \lambda(f)}{\rightleftarrows}} \mathcal{C}(C, B^A) \tag{22}$$

This bijection amounts to the following natural isomorphism, whose components are λ_C and $\hat{\_}_C$ that are morphisms in $\mathcal{S}et$.

$$\mathcal{C}(- \times A, B) \cong \mathcal{C}(-, B^A)$$

Definition 29 *(Base-level Lambda Abstractions)*

$$\frac{[\![\Delta \Vdash M : (\Gamma, x : \alpha_2 \vdash \alpha_1)]\!] = m : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!] \times [\![\alpha_2]\!], [\![\alpha_1]\!])}{[\![\Delta \Vdash \lambda x.M : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!] = \lambda_{[\![\Gamma]\!]} \circ m : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha_1]\!]^{[\![\alpha_2]\!]})}$$

Definition 30 *(Base-level Function Applications)*

$$\frac{\begin{array}{c}[\![\Delta \Vdash M_1 : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!] = m_1 : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha_1]\!]^{[\![\alpha_2]\!]}) \\ [\![\Delta \Vdash M_2 : (\Gamma \vdash \alpha_2)]\!] = m_2 : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha_2]\!])\end{array}}{\begin{array}{l}[\![\Delta \Vdash M_1 M_2 : (\Gamma \vdash \alpha_1)]\!] = \mathcal{C}([\![\Gamma]\!], ev) \circ \ltimes_{[\![\Gamma]\!]} \circ \langle m_1, m_2\rangle \\ : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha_1]\!]^{[\![\alpha_2]\!]}) \times \mathcal{C}([\![\Gamma]\!], [\![\alpha_2]\!]) \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha_1]\!]^{[\![\alpha_2]\!]} \times [\![\alpha_2]\!]) \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha_1]\!])\end{array}}$$

Definition 31 *(Base-level Substitution)*

$$\frac{\Delta \Vdash M : S_{\Gamma,\, x:\alpha}(\tau) \qquad [\![\Delta \Vdash L : (\Gamma \vdash \alpha)]\!] = l : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha]\!])}{[\![\Delta \Vdash M[L/x] : \tau]\!] = \left\| \Delta \Vdash M : S_{\Gamma,\, x:\alpha}(\tau) \right\|_l^{\Gamma, x:\alpha} : [\![\Delta]\!] \longrightarrow [\![\tau]\!]}$$

$$where\ M \in \overline{\mathcal{X}} \cup \overline{\mathcal{C}}$$

The interpretation of base-level substitution given in Definition 31 is based on the following natural transformation, which is natural in C.

$$\mathcal{C}(C \times A, B) \times \mathcal{C}(C, A) \stackrel{\cdot}{\longrightarrow} \mathcal{C}(C, B)$$

In this case, the substitution map is a component of this natural transformation, which is defined as follows, where $m_1 : C \times A \longrightarrow B$ and $m_2 : C \longrightarrow A$.

$$\mathcal{S}ub_C : (m_1, m_2) \longmapsto m_1 \circ \langle id_C, m_2 \rangle$$

By means of the substitution map $\mathcal{S}ub$, we can define the following set of translation rules, which is utilized in Definition 31.

Definition 32 *(Substitution Translation Rule)* Suppose that $|\Delta| = d$. For any base-level variable x of type α and any morphism $m : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha]\!])$, $\|-\|_m^{\Gamma, x:\alpha}$ is a map from a meta-level term $\Delta \Vdash M : \tau$ (of one of the following forms) to a morphism $[\![\Delta]\!] \longrightarrow [\![S_{\Gamma,\, x:\alpha}(\tau)]\!]$, defined as follows[8]:

$$\begin{aligned}
\|\Delta \Vdash M : (\Gamma, x : \alpha \vdash \alpha')\|_m^{\Gamma,x:\alpha} &\stackrel{def}{\equiv} \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle [\![\Delta \Vdash M : (\Gamma, x : \alpha \vdash \alpha')]\!], m \rangle \\
\|\Delta \Vdash M : \tau_2 \Rightarrow \tau_1\|_m^{\Gamma,x:\alpha} &\stackrel{def}{\equiv} \lambda(\|\Delta, X : \tau_2 \Vdash M \,\natural\, X : \tau_1\|^{\Gamma,x:\alpha}_{m \circ \langle \pi_1, \ldots, \pi_d \rangle}) \\
\|\Delta \Vdash M : munit\|_m^{\Gamma,x:\alpha} &\stackrel{def}{\equiv} !_{[\![\Delta]\!]} \\
\|\Delta \Vdash M : \tau_1 \otimes \cdots \otimes \tau_n\|_m^{\Gamma,x:\alpha} &\stackrel{def}{\equiv} \langle \|\Delta \Vdash p_1(M) : \tau_1\|_m^{\Gamma,x:\alpha}, \ldots, \|\Delta \Vdash p_n(M) : \tau_n\|_m^{\Gamma,x:\alpha} \rangle
\end{aligned}$$

5.4 Interpretation of Meta-level Terms

The interpretation of meta-level terms in MLC is similar to the interpretation of (base-level) terms of STLC.

Definition 33 *(Meta-level Variables)*

$$\overline{[\![X : \tau \Vdash X : \tau]\!] = id_{[\![\tau]\!]} : [\![\tau]\!] \longrightarrow [\![\tau]\!]}$$

Definition 34 *(Meta-level Constant Symbols)*

$$\overline{[\![\Vdash C : \tau]\!] = valMC(C) : * \longrightarrow [\![\tau]\!]}$$

[8] For the definition of $S_{\Gamma,\, x:\alpha}$, see Definition 14.

Theorem 8 (Meta-level Variables and Constants) Suppose that $|\Delta| = d$.

$$\frac{}{[\![\Delta, X : \tau, \Delta' \Vdash X : \tau]\!] = \pi_{d+1} : [\![\Delta]\!] \times [\![\tau]\!] \times [\![\Delta']\!] \longrightarrow [\![\tau]\!]}$$

$$\frac{}{[\![\Delta \Vdash C : \tau]\!] = valMC(C) \circ !_{\Delta} : [\![\Delta]\!] \longrightarrow * \longrightarrow [\![\tau]\!]}$$

Proof By Definition 33, Definition 34, and the meta-level weakening and exchange.

In other words, a meta-level variable X_i is interpreted simply as the projection π_i in $\mathcal{S}et$, which selects the i-th member of Δ and returns its value.

Definition 35 *(Meta-level Products)*

$$\frac{}{[\![\Vdash \langle\!\langle\, \rangle\!\rangle : munit]\!] = id_* : * \longrightarrow *}$$

$$\frac{\begin{array}{c}[\![\Delta \Vdash M_1 : \tau_1]\!] = m_1 : [\![\Delta]\!] \longrightarrow [\![\tau_1]\!] \\ \vdots \\ [\![\Delta \Vdash M_n : \tau_n]\!] = m_n : [\![\Delta]\!] \longrightarrow [\![\tau_n]\!]\end{array}}{[\![\Delta \Vdash \langle\!\langle M_1, \ldots, M_n \rangle\!\rangle : \tau_1 \otimes \cdots \otimes \tau_n]\!] = \langle m_1, \ldots, m_n \rangle : [\![\Delta]\!] \longrightarrow [\![\tau_1]\!] \times \cdots \times [\![\tau_n]\!]} (1 \leq n)$$

$$\frac{[\![\Delta \Vdash M : \tau_1 \otimes \cdots \otimes \tau_n]\!] = m : [\![\Delta]\!] \longrightarrow [\![\tau_1]\!] \times \cdots \times [\![\tau_n]\!]}{[\![\Delta \Vdash p_i(M) : \tau_i]\!] = \pi_i \circ m : [\![\Delta]\!] \longrightarrow [\![\tau_i]\!]} (1 \leq i \leq n)$$

Meta-level products roughly correspond to products in $\mathcal{S}et$, whose interpretation is similar to the interpretation of (base-level) products in STLC.

Definition 36 *(Meta-level Lambda Abstractions)*

$$\frac{[\![\Delta, X : \tau_2 \Vdash M : \tau_1]\!] = m : [\![\Delta]\!] \times [\![\tau_2]\!] \longrightarrow [\![\tau_1]\!]}{[\![\Delta \Vdash \zeta X.M : \tau_2 \Rightarrow \tau_1]\!] = \lambda(m) : [\![\Delta]\!] \longrightarrow [\![\tau_1]\!]^{[\![\tau_2]\!]}}$$

Definition 37 *(Meta-level Function Applications)*

$$\frac{\begin{array}{c}[\![\Delta \Vdash M_1 : \tau_2 \Rightarrow \tau_1]\!] = m_1 : [\![\Delta]\!] \longrightarrow [\![\tau_1]\!]^{[\![\tau_2]\!]} \\ [\![\Delta \Vdash M_2 : \tau_2]\!] = m_2 : [\![\Delta]\!] \longrightarrow [\![\tau_2]\!]\end{array}}{[\![\Delta \Vdash M_1 \,\zeta\, M_2 : \tau_1]\!] = ev \circ \langle m_1, m_2 \rangle : [\![\Delta]\!] \longrightarrow [\![\tau_1]\!]}$$

6 Soundness of Base-level Substitution and Conversions

In the following two sections, we prove the soundness of base-level/meta-level substitution and conversions.[9]

The domain of interpretation of substitution given in Definition 31 is not limited to meta-level variables and meta-level constants, but is extended to encompass any other terms as well.

Lemma 1 (Base-level Substitution Lemma)

$$\frac{\Delta \Vdash M : S_{\Gamma,\, x:\alpha}(\tau) \qquad [\![\Delta \Vdash L : (\Gamma \vdash \alpha)]\!] = l : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha]\!])}{[\![\Delta \Vdash M[L/x] : \tau]\!] = \left\| \Delta \Vdash M : S_{\Gamma,\, x:\alpha}(\tau) \right\|_{l}^{\Gamma, x:\alpha} : [\![\Delta]\!] \longrightarrow [\![\tau]\!]}$$

By induction on the size and type of the term M. Suppose that $size(M) = s$, $|\Delta| = d$, $|\Gamma| = n$, $h_1, \ldots, h_d \in [\![\Delta]\!]$, and the morphism l maps $h_1, \ldots, h_d$ to $f : [\![\Gamma]\!] \longrightarrow [\![\alpha]\!]$.

First, let us consider all possible cases for $s = 1$ ($M = x \mid y \mid c \mid \langle\,\rangle \mid \overline{\mathcal{X}} \mid \overline{\mathcal{C}} \mid \langle\langle\,\rangle\rangle$).

– If $M = x$ and $\tau = (\Gamma \vdash \alpha)$, the left side of the equation is $[\![\Delta \Vdash L : (\Gamma \vdash \alpha)]\!] = l$. On the other hand,

$$\begin{aligned} &\|\Delta \Vdash x : (\Gamma, x : \alpha \vdash \alpha)\|_{l}^{\Gamma, x:\alpha} \\ &= \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle [\![\Delta \Vdash x : (\Gamma, x : \alpha \vdash \alpha)]\!], l \rangle \\ &= \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle tp_{[\![\alpha]\!]}(\pi_{n+1}) \circ !_{[\![\Delta]\!]}, l \rangle \text{ — } (\dagger) \end{aligned}$$

Recall that tp and $\mathcal{S}ub$ map morphisms in the following way:

$$\begin{aligned} tp_{[\![\alpha]\!]}(\pi_{n+1}) \circ !_{[\![\Delta]\!]} &: h_1, \ldots, h_d \longmapsto \pi_{n+1} \\ \mathcal{S}ub_{[\![\Gamma]\!]} &: (\pi_{n+1}, f) \longmapsto \pi_{n+1} \circ \langle id_{[\![\Gamma]\!]}, f \rangle \end{aligned}$$

Thus, l and ($\dagger$) are identical morphisms since $\pi_{n+1} \circ \langle id_{[\![\Gamma]\!]}, f \rangle = f$.

– If $M = y$ and $\tau = (\Gamma \vdash \alpha')$, the left side of the equation is $[\![\Delta \Vdash y : (\Gamma \vdash \alpha')]\!]$. On the other hand,

$$\begin{aligned} &\left\| \Delta \Vdash y : (\Gamma, x : \alpha \vdash \alpha') \right\|_{l}^{\Gamma, x:\alpha} \\ &= \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle [\![\Delta \Vdash y : (\Gamma, x : \alpha \vdash \alpha')]\!], l \rangle \\ &= \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle \mathcal{C}(\langle \pi_1, \ldots, \pi_n \rangle, [\![\alpha']\!]) \circ [\![\Delta \Vdash y : (\Gamma \vdash \alpha')]\!], l \rangle \text{ — } (\dagger) \end{aligned}$$

Suppose that the morphism $[\![\Delta \Vdash y : (\Gamma \vdash \alpha')]\!]$ maps $h_1, \ldots, h_d$ to $g : [\![\Gamma]\!] \longrightarrow [\![\alpha']\!]$. Then, ($\dagger$) maps $h_1, \ldots, h_d$ to

$$\begin{aligned} g \circ \langle \pi_1, \ldots, \pi_n \rangle \circ \langle id_{[\![\Gamma]\!]}, f \rangle &= g \circ \langle \pi_1 \circ \langle id_{[\![\Gamma]\!]}, f \rangle, \ldots, \pi_n \circ \langle id_{[\![\Gamma]\!]}, f \rangle \rangle \\ &= g \circ \langle \pi_1, \ldots, \pi_n \rangle \\ &= g \end{aligned}$$

[9] The proof is the revised version of what we presented in Masuko and Bekki (2011).

Therefore, $[\![\Delta \Vdash y : (\Gamma \vdash \alpha')]\!]$ and (†) are identical morphisms.

The same is true for $M = c$ and $M = \langle\,\rangle$.

– If $M = \overline{\mathcal{X}}$ or $M = \overline{\mathcal{C}}$, by Definition 31.
– If $M = \langle\langle\,\rangle\rangle$, the equation holds obviously since the morphism $!_{[\![\Delta]\!]}$ is unique.

Next, suppose the equation holds for a term whose size is less than or equals to s and show that the equation also holds for a term whose size is $s+1$ ($M = \langle M_1, \ldots, M_k\rangle \mid \pi_i(M) \mid \lambda x.M \mid M_1 M_2 \mid \langle\langle M_1, \ldots, M_k\rangle\rangle \mid p_i(M) \mid \zeta X.M \mid M_1 \not\downarrow M_2$).

By induction on the structure of the type τ.

First, let us consider the cases for $\tau = (\Gamma \vdash \alpha')$. In this case, the right side of the equation is

$$\left\|\Delta \Vdash M : (\Gamma, x : \alpha \vdash \alpha')\right\|_l^{\Gamma, x:\alpha} = \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle [\![\Delta \Vdash M : (\Gamma, x : \alpha \vdash \alpha')]\!], l\rangle$$

- If $M = \langle\langle M_1, \ldots, M_k\rangle\rangle$ or $\zeta X.M$, then the equation holds vacuously.
- If $M = \langle M_1, \ldots, M_k\rangle$ and $\alpha' = \alpha_1 \times \cdots \times \alpha_k$, the left side of the equation is

$$\begin{aligned}
&[\![\Delta \Vdash \langle M_1, \ldots, M_k\rangle[L/x] : (\Gamma \vdash \alpha_1 \times \cdots \times \alpha_k)]\!] \\
&= [\![\Delta \Vdash \langle M_1[L/x], \ldots, M_k[L/x]\rangle : (\Gamma \vdash \alpha_1 \times \cdots \times \alpha_k)]\!] \\
&= \ltimes_{[\![\Gamma]\!]} \circ \langle [\![\Delta \Vdash M_1[L/x] : (\Gamma \vdash \alpha_1)]\!], \ldots, [\![\Delta \Vdash M_k[L/x] : (\Gamma \vdash \alpha_k)]\!]\rangle \\
&= \ltimes_{[\![\Gamma]\!]} \circ \langle \|\Delta \Vdash M_1 : (\Gamma, x : \alpha \vdash \alpha_1)\|_l^{\Gamma, x:\alpha}, \ldots, \|\Delta \Vdash M_k : (\Gamma, x : \alpha \vdash \alpha_k)\|_l^{\Gamma, x:\alpha}\rangle \\
&= \ltimes_{[\![\Gamma]\!]} \circ \langle \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle [\![\Delta \Vdash M_1 : (\Gamma, x : \alpha \vdash \alpha_1)]\!], l\rangle, \ldots, \\
&\qquad \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle [\![\Delta \Vdash M_k : (\Gamma, x : \alpha \vdash \alpha_k)]\!], l\rangle\rangle \text{ — (†)}
\end{aligned}$$

On the other hand,

$$\begin{aligned}
&\mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle [\![\Delta \Vdash \langle M_1, \ldots, M_k\rangle : (\Gamma, x : \alpha \vdash \alpha_1 \times \cdots \times \alpha_k)]\!], l\rangle \\
&= \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle \ltimes_{[\![\Gamma]\!] \times [\![\alpha]\!]} \circ \langle [\![\Delta \Vdash M_1 : (\Gamma, x : \alpha \vdash \alpha_1)]\!], \ldots, \\
&\qquad [\![\Delta \Vdash M_k : (\Gamma, x : \alpha \vdash \alpha_k)]\!]\rangle, l\rangle \text{ — (‡)}
\end{aligned}$$

Suppose that each morphism $[\![\Delta \Vdash M_i : (\Gamma, x : \alpha \vdash \alpha_i)]\!]$ maps $h_1, \ldots, h_d$ to $g_i : [\![\Gamma]\!] \times [\![\alpha]\!] \longrightarrow [\![\alpha_i]\!]$ for $(1 \le i \le k)$. Then,

$$\begin{aligned}
(\dagger) &: h_1, \ldots, h_d \longmapsto \langle g_1 \circ \langle id_{[\![\Gamma]\!]}, f\rangle, \ldots, g_k \circ \langle id_{[\![\Gamma]\!]}, f\rangle\rangle \\
(\ddagger) &: h_1, \ldots, h_d \longmapsto \langle g_1, \ldots, g_k\rangle \circ \langle id_{[\![\Gamma]\!]}, f\rangle
\end{aligned}$$

Therefore, (†) and (‡) are identical morphisms.

- If $M = \pi_i(M)$, then the left side is

$$\begin{aligned}
&[\![\Delta \Vdash (\pi_i(M))[L/x] : (\Gamma \vdash \alpha')]\!] = [\![\Delta \Vdash \pi_i(M[L/x]) : (\Gamma \vdash \alpha')]\!] \\
&= \mathcal{C}([\![\Gamma]\!], \pi_i) \circ [\![\Delta \Vdash M[L/x] : (\Gamma \vdash \alpha_1 \times \cdots \times \alpha' \times \cdots \times \alpha_k)]\!] \\
&= \mathcal{C}([\![\Gamma]\!], \pi_i) \circ \left\|\Delta \Vdash M : (\Gamma, x : \alpha \vdash \alpha_1 \times \cdots \times \alpha' \times \cdots \times \alpha_k)\right\|_l^{\Gamma, x:\alpha} \\
&= \mathcal{C}([\![\Gamma]\!], \pi_i) \circ \\
&\qquad \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle [\![\Delta \Vdash M : (\Gamma, x : \alpha \vdash \alpha_1 \times \cdots \times \alpha' \times \cdots \times \alpha_k)]\!], l\rangle \text{ — (†)}
\end{aligned}$$

On the other hand, the following equation holds.

$$\begin{aligned}
&\mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle [\![\Delta \Vdash \pi_i(M) : (\Gamma, x : \alpha \vdash \alpha')]\!], l\rangle \\
&= \mathcal{S}ub_{[\![\Gamma]\!]} \circ \\
&\qquad \langle \mathcal{C}([\![\Gamma]\!], \pi_i) \circ [\![\Delta \Vdash M : (\Gamma, x : \alpha \vdash \alpha_1 \times \cdots \times \alpha' \times \cdots \times \alpha_k)]\!], l\rangle \text{ — (‡)}
\end{aligned}$$

Suppose that $[\![\Delta \Vdash M : (\Gamma, x : \alpha \vdash \alpha_1 \times \cdots \times \alpha' \times \cdots \times \alpha_k)]\!]$ maps $h_1, \ldots, h_d$ to $g : [\![\Gamma]\!] \times [\![\alpha]\!] \longrightarrow [\![\alpha_1]\!] \times \cdots \times [\![\alpha']\!] \times \cdots \times [\![\alpha_k]\!]$. Then, since both (†) and (‡) map the d-tuple to $\pi_i \circ g \circ \langle id_{[\![\Gamma]\!]}, f \rangle$, (†) and (‡) are identical morphisms.

- If $M = \lambda x.M$, the case is the same as that for y.
- If $M = \lambda y.M$ $(x \not\equiv y \wedge (x \notin fv(M) \vee y \notin fv(L)))$ and $\alpha' = \alpha_2 \to \alpha_1$, let m be a morphism $[\![\Delta \Vdash M : (\Gamma, x : \alpha, y : \alpha_2 \vdash \alpha_1)]\!]$. Then, the following equation holds.

$$
\begin{aligned}
&[\![\Delta \Vdash (\lambda y.M)[L/x] : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!] \;=\; [\![\Delta \Vdash \lambda y.M[L/x] : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!] \\
&= \lambda_{[\![\Gamma]\!]} \circ [\![\Delta \Vdash M[L/x] : (\Gamma, y : \alpha_2 \vdash \alpha_1)]\!] \\
&= \lambda_{[\![\Gamma]\!]} \circ \|\Delta \Vdash M : (\Gamma, y : \alpha_2, x : \alpha \vdash \alpha_1)\|^{\Gamma, y:\alpha_2, x:\alpha}_{\mathcal{C}(\langle \pi_1, \ldots, \pi_n \rangle, [\![\alpha]\!]) \circ l} \\
&= \lambda_{[\![\Gamma]\!]} \circ \mathcal{S}ub_{[\![\Gamma]\!] \times [\![\alpha_2]\!]} \circ \langle [\![\Delta \Vdash M : (\Gamma, y : \alpha_2, x : \alpha \vdash \alpha_1)]\!], \mathcal{C}(\langle \pi_1, \ldots, \pi_n \rangle, [\![\alpha]\!]) \circ l \rangle \\
&= \lambda_{[\![\Gamma]\!]} \circ \mathcal{S}ub_{[\![\Gamma]\!] \times [\![\alpha_2]\!]} \circ \\
&\qquad \langle \mathcal{C}(\langle \pi_1, \ldots, \pi_n, \pi_{n+2}, \pi_{n+1} \rangle, [\![\alpha_1]\!]) \circ m, \mathcal{C}(\langle \pi_1, \ldots, \pi_n \rangle, [\![\alpha]\!]) \circ l \rangle \text{ — } (\dagger)
\end{aligned}
$$

On the other hand,

$$
\begin{aligned}
&\mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle [\![\Delta \Vdash \lambda y.M : (\Gamma, x : \alpha \vdash \alpha_2 \to \alpha_1)]\!], l \rangle \\
&= \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle \lambda_{[\![\Gamma]\!] \times [\![\alpha]\!]} \circ m, l \rangle \text{ — } (\ddagger)
\end{aligned}
$$

Suppose that m maps $h_1, \ldots, h_d$ to $g : [\![\Gamma]\!] \times [\![\alpha]\!] \times [\![\alpha_2]\!] \longrightarrow [\![\alpha_1]\!]$. Then, (†) maps the d-tuple to

$$
\begin{aligned}
&\lambda(g \circ \langle \pi_1, \ldots, \pi_n, \pi_{n+2}, \pi_{n+1} \rangle \circ \langle id_{[\![\Gamma]\!] \times [\![\alpha_2]\!]}, f \circ \langle \pi_1, \ldots, \pi_n \rangle \rangle) \\
&= \lambda(g \circ \langle \pi_1, \ldots, \pi_n, f \circ \langle \pi_1, \ldots, \pi_n \rangle, \pi_{n+1} \rangle)
\end{aligned}
$$

and (‡) maps to $\lambda(g) \circ \langle id_{[\![\Gamma]\!]}, f \rangle$.
By the uniqueness of the transpose, the equality of (†) and (‡) is proved by the following equation:

$$
\begin{aligned}
&ev \circ ((\lambda(g) \circ \langle id_{[\![\Gamma]\!]}, f \rangle) \times id_{[\![\alpha_2]\!]}) \\
&= ev \circ (\lambda(g) \times id_{[\![\alpha_2]\!]}) \circ (\langle id_{[\![\Gamma]\!]}, f \rangle \times id_{[\![\alpha_2]\!]}) \\
&= g \circ (\langle id_{[\![\Gamma]\!]}, f \rangle \times id_{[\![\alpha_2]\!]}) \\
&= g \circ \langle \pi_1, \ldots, \pi_n, f \circ \langle \pi_1, \ldots, \pi_n \rangle, \pi_{n+1} \rangle
\end{aligned}
$$

- If $M = \lambda y.M$ $(x \not\equiv y \wedge x \in fv(M) \wedge y \in fv(L))$ and $\alpha' = \alpha_2 \to \alpha_1$, let m be a morphism $[\![\Delta \Vdash M : (\Gamma, x : \alpha, y : \alpha_2 \vdash \alpha_1)]\!]$. Then, the left side of the equation is

$$
\begin{aligned}
&[\![\Delta \Vdash (\lambda y.M)[L/x] : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!] \\
&= [\![\Delta \Vdash \lambda z.M[z/y][L/x] : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!] \\
&= \lambda_{[\![\Gamma]\!]} \circ [\![\Delta \Vdash M[z/y][L/x] : (\Gamma, z : \alpha_2 \vdash \alpha_1)]\!] \\
&= \lambda_{[\![\Gamma]\!]} \circ \|\Delta \Vdash M[z/y] : (\Gamma, z : \alpha_2, x : \alpha \vdash \alpha_1)\|^{\Gamma, z:\alpha_2, x:\alpha}_{\mathcal{C}(\langle \pi_1, \ldots, \pi_n \rangle, [\![\alpha_1]\!]) \circ l} \\
&= \lambda_{[\![\Gamma]\!]} \circ \mathcal{S}ub_{[\![\Gamma]\!] \times [\![\alpha_2]\!]} \circ \\
&\qquad \langle [\![\Delta \Vdash M[z/y] : (\Gamma, z : \alpha_2, x : \alpha \vdash \alpha_1)]\!], \mathcal{C}(\langle \pi_1, \ldots, \pi_n \rangle, [\![\alpha_1]\!]) \circ l \rangle \\
&= \lambda_{[\![\Gamma]\!]} \circ \mathcal{S}ub_{[\![\Gamma]\!] \times [\![\alpha_2]\!]} \circ \\
&\qquad \langle \|\Delta \Vdash M : (\Gamma, z : \alpha_2, x : \alpha, y : \alpha_2 \vdash \alpha_1)\|^{\Gamma, z:\alpha_2, x:\alpha, y:\alpha_2}_{tp_{[\![\alpha_2]\!]}(\pi_{n+1}) \circ !_{[\![\Delta]\!]}}, \\
&\qquad \mathcal{C}(\langle \pi_1, \ldots, \pi_n \rangle, [\![\alpha_1]\!]) \circ l \rangle \\
&= \lambda_{[\![\Gamma]\!]} \circ \mathcal{S}ub_{[\![\Gamma]\!] \times [\![\alpha_2]\!]} \circ \\
&\qquad \langle \mathcal{S}ub_{[\![\Gamma]\!] \times [\![\alpha_2]\!] \times [\![\alpha]\!]} \circ \langle [\![\Delta \Vdash M : (\Gamma, z : \alpha_2, x : \alpha, y : \alpha_2 \vdash \alpha_1)]\!], tp_{[\![\alpha_2]\!]}(\pi_{n+1}) \circ !_{[\![\Delta]\!]} \rangle, \\
&\qquad \mathcal{C}(\langle \pi_1, \ldots, \pi_n \rangle, [\![\alpha_1]\!]) \circ l \rangle \\
&= \lambda_{[\![\Gamma]\!]} \circ \mathcal{S}ub_{[\![\Gamma]\!] \times [\![\alpha_2]\!]} \circ \\
&\qquad \langle \mathcal{S}ub_{[\![\Gamma]\!] \times [\![\alpha_2]\!] \times [\![\alpha]\!]} \circ \langle \mathcal{C}(\langle \pi_1, \ldots, \pi_n, \pi_{n+2}, \pi_{n+3} \rangle, [\![\alpha_1]\!]) \circ m, tp_{[\![\alpha_2]\!]}(\pi_{n+1}) \circ !_{[\![\Delta]\!]} \rangle, \\
&\qquad \mathcal{C}(\langle \pi_1, \ldots, \pi_n \rangle, [\![\alpha_1]\!]) \circ l \rangle \text{ — } (\dagger)
\end{aligned}
$$

The right side is the same as the one above: $\mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle \lambda_{[\![\Gamma]\!]\times[\![\alpha]\!]} \circ m, l\rangle$ — (‡)
Suppose that m maps $h_1, \ldots, h_d$ to $g : [\![\Gamma]\!] \times [\![\alpha]\!] \times [\![\alpha_2]\!] \longrightarrow [\![\alpha_1]\!]$. Then, (†) maps the d-tuple to

$$\begin{aligned}
&\lambda(g \circ \langle \pi_1, \ldots, \pi_n, \pi_{n+2}, \pi_{n+3}\rangle \circ \langle id_{[\![\Gamma]\!]\times[\![\alpha_2]\!]\times[\![\alpha]\!]}, \pi_{n+1}\rangle \\
&\qquad \circ \langle id_{[\![\Gamma]\!]\times[\![\alpha_2]\!]}, f \circ \langle \pi_1, \ldots, \pi_n\rangle\rangle) \\
&= \lambda(g \circ \langle \pi_1, \ldots, \pi_n, \pi_{n+2}, \pi_{n+1}\rangle \circ \langle id_{[\![\Gamma]\!]\times[\![\alpha_2]\!]}, f \circ \langle \pi_1, \ldots, \pi_n\rangle\rangle) \\
&= \lambda(g \circ \langle \pi_1, \ldots, \pi_n, f \circ \langle \pi_1, \ldots, \pi_n\rangle, \pi_{n+1}\rangle)
\end{aligned}$$

and (‡) maps to $\lambda(g) \circ \langle id_{[\![\Gamma]\!]}, f\rangle$.
Therefore, following the argument above, (†) = (‡) holds.

- If $M = M_1 M_2$, let $\alpha' = \alpha_1$ and m_1, m_2 be the following morphisms:

$$\begin{aligned}
m_1 &: [\![\Delta \Vdash M_1 : (\Gamma, x : \alpha \vdash \alpha_2 \to \alpha_1)]\!] \\
m_2 &: [\![\Delta \Vdash M_2 : (\Gamma, x : \alpha \vdash \alpha_2)]\!]
\end{aligned}$$

Then, on one hand, the following equation holds.

$$\begin{aligned}
&[\![\Delta \Vdash (M_1 M_2)[L/x] : (\Gamma \vdash \alpha_1)]\!] \\
&= [\![\Delta \Vdash (M_1[L/x])(M_2[L/x]) : (\Gamma \vdash \alpha_1)]\!] \\
&= \mathcal{C}([\![\Gamma]\!], ev) \circ \ltimes_{[\![\Gamma]\!]} \circ \\
&\qquad \langle [\![\Delta \Vdash M_1[L/x] : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!], [\![\Delta \Vdash M_2[L/x] : (\Gamma \vdash \alpha_2)]\!]\rangle \\
&= \mathcal{C}([\![\Gamma]\!], ev) \circ \ltimes_{[\![\Gamma]\!]} \circ \\
&\qquad \langle \|\Delta \Vdash M_1 : (\Gamma, x : \alpha \vdash \alpha_2 \to \alpha_1)\|_l^{\Gamma, x:\alpha}, \|\Delta \Vdash M_2 : (\Gamma, x : \alpha \vdash \alpha_2)\|_l^{\Gamma, x:\alpha}\rangle \\
&= \mathcal{C}([\![\Gamma]\!], ev) \circ \ltimes_{[\![\Gamma]\!]} \circ \langle \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle m_1, l\rangle, \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle m_2, l\rangle\rangle \text{ — (†)}
\end{aligned}$$

On the other hand,

$$\begin{aligned}
&\mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle [\![\Delta \Vdash M_1 M_2 : (\Gamma, x : \alpha \vdash \alpha_1)]\!], l\rangle \\
&= \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle \mathcal{C}([\![\Gamma]\!] \times [\![\alpha]\!], ev) \circ \ltimes_{[\![\Gamma]\!]\times[\![\alpha]\!]} \circ \langle m_1, m_2\rangle, l\rangle \text{ — (‡)}
\end{aligned}$$

Suppose that m_1 and m_2 map $h_1, \ldots, h_d$ to $g_1 : [\![\Gamma]\!] \times [\![\alpha]\!] \longrightarrow [\![\alpha_1]\!]^{[\![\alpha_2]\!]}$ and $g_2 : [\![\Gamma]\!] \times [\![\alpha]\!] \longrightarrow [\![\alpha_2]\!]$, respectively.
Then (†) maps the d-tuple to $ev \circ \langle g_1 \circ \langle id_{[\![\Gamma]\!]}, f\rangle, g_2 \circ \langle id_{[\![\Gamma]\!]}, f\rangle\rangle$ and (‡) maps to $ev \circ \langle g_1, g_2\rangle \circ \langle id_{[\![\Gamma]\!]}, f\rangle = ev \circ \langle g_1 \circ \langle id_{[\![\Gamma]\!]}, f\rangle, g_2 \circ \langle id_{[\![\Gamma]\!]}, f\rangle\rangle$. Therefore, (†) and (‡) are the same morphisms.

- If $M = p_i(M)$, the equation is proved as follows:

$$\begin{aligned}
&[\![\Delta \Vdash (p_i(M))[L/x] : (\Gamma \vdash \alpha')]\!] \\
&= [\![\Delta \Vdash p_i(M[L/x]) : (\Gamma \vdash \alpha')]\!] \\
&= \pi_i \circ [\![\Delta \Vdash M[L/x] : \tau_1 \otimes \cdots \otimes (\Gamma \vdash \alpha') \otimes \cdots \otimes \tau_k]\!] \\
&= \pi_i \circ \|\Delta \Vdash M : S_{\Gamma, x:\alpha}(\tau_1 \otimes \cdots \otimes (\Gamma \vdash \alpha') \otimes \cdots \otimes \tau_k)\|_l^{\Gamma, x:\alpha} \\
&= \pi_i \circ \|\Delta \Vdash M : S_{\Gamma, x:\alpha}(\tau_1) \otimes \cdots \otimes S_{\Gamma, x:\alpha}(\Gamma \vdash \alpha') \otimes \cdots \otimes S_{\Gamma, x:\alpha}(\tau_k)\|_l^{\Gamma, x:\alpha} \\
&= \pi_i \circ \langle \|\Delta \Vdash p_1(M) : S_{\Gamma, x:\alpha}(\tau_1)\|_l^{\Gamma, x:\alpha}, \ldots, \\
&\qquad \|\Delta \Vdash p_i(M) : (\Gamma, x : \alpha \vdash \alpha')\|_l^{\Gamma, x:\alpha}, \ldots, \|\Delta \Vdash p_k(M) : S_{\Gamma, x:\alpha}(\tau_k)\|_l^{\Gamma, x:\alpha}\rangle \\
&= \|\Delta \Vdash p_i(M) : (\Gamma, x : \alpha \vdash \alpha')\|_l^{\Gamma, x:\alpha}
\end{aligned}$$

- If $M = M_1 \not\downarrow M_2$, the equation is proved as follows:

$$
\begin{aligned}
& [\![\Delta \Vdash (M_1 \not\downarrow M_2)[L/x] : (\Gamma \vdash \alpha')]\!] \\
= \;& [\![\Delta \Vdash (M_1[L/x]) \not\downarrow M_2 : (\Gamma \vdash \alpha')]\!] \\
= \;& ev \circ \langle [\![\Delta \Vdash M_1[L/x] : \tau \Rightarrow (\Gamma \vdash \alpha')]\!], [\![\Delta \Vdash M_2 : \tau]\!] \rangle \\
= \;& ev \circ \langle \left\| \Delta \Vdash M_1 : \tau \Rightarrow (\Gamma, x : \alpha \vdash \alpha') \right\|_l^{\Gamma, x:\alpha}, [\![\Delta \Vdash M_2 : \tau]\!] \rangle \\
= \;& ev \circ \langle \lambda(\left\| \Delta, X : \tau \Vdash M_1 \not\downarrow X : (\Gamma, x : \alpha \vdash \alpha') \right\|_{l \circ \langle \pi_1, \ldots, \pi_d \rangle}^{\Gamma, x:\alpha}), [\![\Delta \Vdash M_2 : \tau]\!] \rangle \\
= \;& \left\| \Delta, X : \tau \Vdash M_1 \not\downarrow X : (\Gamma, x : \alpha \vdash \alpha') \right\|_{l \circ \langle \pi_1, \ldots, \pi_d \rangle}^{\Gamma, x:\alpha} \circ \langle id_\Delta, [\![\Delta \Vdash M_2 : \tau]\!] \rangle \\
= \;& \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle [\![\Delta, X : \tau \Vdash M_1 \not\downarrow X : (\Gamma, x : \alpha \vdash \alpha')]\!], l \circ \langle \pi_1, \ldots, \pi_d \rangle \rangle \circ \\
& \langle id_{[\![\Delta]\!]}, [\![\Delta \Vdash M_2 : \tau]\!] \rangle \\
= \;& \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle [\![\Delta, X : \tau \Vdash M_1 \not\downarrow X : (\Gamma, x : \alpha \vdash \alpha')]\!] \circ \langle id_{[\![\Delta]\!]}, [\![\Delta \Vdash M_2 : \tau]\!] \rangle, l \rangle \\
= \;& \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle ev \circ \langle \lambda([\![\Delta, X : \tau \Vdash M_1 \not\downarrow X : (\Gamma, x : \alpha \vdash \alpha')]\!]), [\![\Delta \Vdash M_2 : \tau]\!] \rangle, l \rangle \\
= \;& \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle ev \circ \langle [\![\Delta \Vdash \zeta X.M_1 \not\downarrow X : \tau \Rightarrow (\Gamma, x : \alpha \vdash \alpha')]\!], [\![\Delta \Vdash M_2 : \tau]\!] \rangle, l \rangle \\
=_{M\eta} \;& \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle ev \circ \langle [\![\Delta \Vdash M_1 : \tau \Rightarrow (\Gamma, x : \alpha \vdash \alpha')]\!], [\![\Delta \Vdash M_2 : \tau]\!] \rangle, l \rangle \\
= \;& \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle [\![\Delta \Vdash M_1 \not\downarrow M_2 : (\Gamma, x : \alpha \vdash \alpha')]\!], l \rangle
\end{aligned}
$$

– If $\tau = \tau_2 \Rightarrow \tau_1$, the equation is proved as follows:

$$
\begin{aligned}
& [\![\Delta \Vdash M[L/x] : \tau_2 \Rightarrow \tau_1]\!] \\
=_{M\eta} \;& [\![\Delta \Vdash \zeta X.(M[L/x]) \not\downarrow X : \tau_2 \Rightarrow \tau_1]\!] \\
= \;& \lambda([\![\Delta, X : \tau_2 \Vdash (M[L/x]) \not\downarrow X : \tau_1]\!]) \\
= \;& \lambda([\![\Delta, X : \tau_2 \Vdash (M \not\downarrow X)[L/x] : \tau_1]\!]) \\
= \;& \lambda(\left\| \Delta, X : \tau_2 \Vdash M \not\downarrow X : S_{\Gamma, x:\alpha}(\tau_1) \right\|_{l \circ \langle \pi_1, \ldots, \pi_d \rangle}^{\Gamma, x:\alpha}) \\
= \;& \left\| \Delta \Vdash M : \tau_2 \Rightarrow S_{\Gamma, x:\alpha}(\tau_1) \right\|_l^{\Gamma, x:\alpha} \\
= \;& \left\| \Delta \Vdash M : S_{\Gamma, x:\alpha}(\tau_2 \Rightarrow \tau_1) \right\|_l^{\Gamma, x:\alpha}
\end{aligned}
$$

– If $\tau = \mathit{munit}$, the equation holds obviously since the morphism $!_{[\![\Delta]\!]}$ is unique.
– If $\tau = \tau_1 \otimes \cdots \otimes \tau_k$, the equation is proved as follows:

$$
\begin{aligned}
& [\![\Delta \Vdash M[L/x] : \tau_1 \otimes \cdots \otimes \tau_k]\!] \\
= \;& [\![\Delta \Vdash \langle\langle p_1(M[L/x]), \ldots, p_k(M[L/x]) \rangle\rangle : \tau_1 \otimes \cdots \otimes \tau_k]\!] \\
= \;& \langle [\![\Delta \Vdash p_1(M[L/x]) : \tau_1]\!], \ldots, [\![\Delta \Vdash p_k(M[L/x]) : \tau_k]\!] \rangle \\
= \;& \langle [\![\Delta \Vdash (p_1(M))[L/x] : \tau_1]\!], \ldots, [\![\Delta \Vdash (p_k(M))[L/x] : \tau_k]\!] \rangle \\
= \;& \langle \left\| \Delta \Vdash p_1(M) : S_{\Gamma, x:\alpha}(\tau_1) \right\|_l^{\Gamma, x:\alpha}, \ldots, \left\| \Delta \Vdash p_k(M) : S_{\Gamma, x:\alpha}(\tau_k) \right\|_l^{\Gamma, x:\alpha} \rangle \\
= \;& \left\| \Delta \Vdash M : S_{\Gamma, x:\alpha}(\tau_1) \otimes \cdots \otimes S_{\Gamma, x:\alpha}(\tau_k) \right\|_l^{\Gamma, x:\alpha} \\
= \;& \left\| \Delta \Vdash M : S_{\Gamma, x:\alpha}(\tau_1 \otimes \cdots \otimes \tau_k) \right\|_l^{\Gamma, x:\alpha}
\end{aligned}
$$

Thus, $[\![\Delta \Vdash M[L/x] : \tau]\!] = \left\| \Delta \Vdash M : S_{\Gamma, x:\alpha}(\tau) \right\|_l^{\Gamma, x:\alpha}$ holds.[10]

[10] Using meta-level η-conversion in the proof is not prohibited since the proof of the conversion is independent of this lemma.

Theorem 9 (Soundness of Base-level α-Conversion)

$$(\alpha)\frac{\Delta \Vdash M : (\Gamma, x : \alpha_2 \vdash \alpha_1) \qquad y \notin fv(M)}{[\![\Delta \Vdash \lambda x.M : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!] = [\![\Delta \Vdash \lambda y.M[y/x] : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!]}$$

Proof Suppose that $|\Delta| = d$, $|\Gamma| = n$ and let m be a morphism $[\![\Delta \Vdash M : (\Gamma, x : \alpha_2 \vdash \alpha_1)]\!]$.

$$[\![\Delta \Vdash \lambda x.M : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!] = \lambda_{[\![\Gamma]\!]} \circ m$$

$$\begin{aligned}
&[\![\Delta \Vdash \lambda y.M[y/x] : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!] \\
&= \lambda_{[\![\Gamma]\!]} \circ [\![\Delta \Vdash M[y/x] : (\Gamma, y : \alpha_2 \vdash \alpha_1)]\!] \\
&= \lambda_{[\![\Gamma]\!]} \circ \mathcal{S}ub_{[\![\Gamma]\!]\times[\![\alpha_2]\!]} \circ \langle \mathcal{C}(\langle \pi_1, \ldots, \pi_n, \pi_{n+2}\rangle, [\![\alpha_1]\!]) \circ m, tp_{[\![\alpha_2]\!]}(\pi_{n+1}) \circ !_{[\![\Delta]\!]} \rangle
\end{aligned}$$

Next, suppose that m maps a d-tuple $h_1, \ldots, h_d \in [\![\Delta]\!]$ to $f : [\![\Gamma]\!] \times [\![\alpha_2]\!] \longrightarrow [\![\alpha_1]\!]$. Then, the left side of the equation maps the d-tuple to $\lambda(f)$. On the other hand,

$$\begin{aligned}
&\lambda(f \circ \langle \pi_1, \ldots, \pi_n, \pi_{n+2}\rangle \circ \langle id_{[\![\Gamma]\!]\times[\![\alpha_2]\!]}, \pi_{n+1}\rangle) \\
=&\lambda(f \circ \langle \pi_1, \ldots, \pi_n, \pi_{n+1}\rangle) \\
=&\lambda(f)
\end{aligned}$$

Thus, $[\![\Delta \Vdash \lambda x.M : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!] = [\![\lambda y.M[y/x] : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!]$ holds.

Theorem 10 (Soundness of Base-level β-Conversion)

$$(\beta)\frac{\Delta \Vdash M_1 : (\Gamma, x : \alpha_2 \vdash \alpha_1) \qquad \Delta \Vdash M_2 : (\Gamma \vdash \alpha_2)}{[\![\Delta \Vdash (\lambda x.M_1)M_2 : (\Gamma \vdash \alpha_1)]\!] = [\![\Delta \Vdash M_1[M_2/x] : (\Gamma \vdash \alpha_1)]\!]}$$

Proof Suppose that $|\Delta| = d$ and let m_1, m_2 be the following morphisms:

$$\begin{aligned}
&m_1 : [\![\Delta \Vdash M_1 : (\Gamma, x : \alpha_2 \vdash \alpha_1)]\!] : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!] \times [\![\alpha_2]\!], [\![\alpha_1]\!]) \\
&m_2 : [\![\Delta \Vdash M_2 : (\Gamma \vdash \alpha_2)]\!] : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!], [\![\alpha_2]\!])
\end{aligned}$$

Suppose that m_1 and m_2 map a d-tuple $h_1, \ldots, h_d \in [\![\Delta]\!]$ to $f : [\![\Gamma]\!] \times [\![\alpha_2]\!] \longrightarrow [\![\alpha_1]\!]$ and $g : [\![\Gamma]\!] \longrightarrow [\![\alpha_2]\!]$, respectively. Then, the following equation holds.

$$\begin{aligned}
&[\![\Delta \Vdash (\lambda x.M_1)M_2 : (\Gamma \vdash \alpha_1)]\!] \\
&= \mathcal{C}([\![\Gamma]\!], ev) \circ \ltimes_{[\![\Gamma]\!]} \circ \langle [\![\Delta \Vdash \lambda x.M_1 : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!], m_2 \rangle \\
&= \mathcal{C}([\![\Gamma]\!], ev) \circ \ltimes_{[\![\Gamma]\!]} \circ \langle \lambda_{[\![\Gamma]\!]} \circ m_1, m_2 \rangle \text{ — } (\dagger)
\end{aligned}$$

Recall that $\lambda_{[\![\Gamma]\!]}$, $\ltimes_{[\![\Gamma]\!]}$ and $\mathcal{C}([\![\Gamma]\!], ev)$ map morphisms in the following way:

$$\begin{aligned}
\lambda_{[\![\Gamma]\!]} &: \quad f \longmapsto \lambda(f) \\
\ltimes_{[\![\Gamma]\!]} &: (\lambda(f), g) \longmapsto \langle \lambda(f), g \rangle \\
\mathcal{C}([\![\Gamma]\!], ev) &: \langle \lambda(f), g \rangle \longmapsto ev \circ \langle \lambda(f), g \rangle
\end{aligned}$$

Since the following equation holds, the morphism (†) maps the d-tuple to a morphism $f \circ \langle id, g \rangle : [\![\Gamma]\!] \longrightarrow [\![\alpha_1]\!]$.

$$\begin{aligned}
ev \circ \langle \lambda(f), g \rangle &= ev \circ (\lambda(f) \times id_{[\![\alpha_2]\!]}) \circ \langle id_{[\![\Gamma]\!]}, g \rangle \\
&= f \circ \langle id_{[\![\Gamma]\!]}, g \rangle
\end{aligned}$$

On the other hand, the following equation holds.

$$\begin{aligned}
[\![\Delta \Vdash M_1[M_2/x] : (\Gamma \vdash \alpha_1)]\!] &= \|\Delta \Vdash M_1 : (\Gamma, x : \alpha_2 \vdash \alpha_1)\|_{m_2}^{\Gamma, x:\alpha_2} \\
&= \mathcal{S}ub_{[\![\Gamma]\!]} \circ \langle m_1, m_2 \rangle \text{ — } (\ddagger)
\end{aligned}$$

Let us also recall that $\mathcal{S}ub_{[\![\Gamma]\!]} : (f, g) \longmapsto f \circ \langle id_{[\![\Gamma]\!]}, g \rangle$. Thus, the morphism (‡) maps the d-tuple to a morphism $f \circ \langle id, g \rangle : [\![\Gamma]\!] \longrightarrow [\![\alpha_1]\!]$, which is equivalent to (†). Therefore, the following equation holds.

$$[\![\Delta \Vdash (\lambda x.M_1)M_2 : (\Gamma \vdash \alpha_1)]\!] = [\![\Delta \Vdash M_1[M_2/x] : (\Gamma \vdash \alpha_1)]\!]$$

Theorem 11 (Soundness of Base-level η-Conversion)

$$(\eta)\frac{\Delta \Vdash M : (\Gamma \vdash \alpha_2 \to \alpha_1) \qquad x \notin fv(M)}{[\![\Delta \Vdash \lambda x.Mx : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!] = [\![\Delta \Vdash M : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!]}$$

Proof Suppose that $|\Delta| = d$, $|\Gamma| = n$, and let m be a morphism $[\![\Delta \Vdash M : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!] : [\![\Delta]\!] \longrightarrow \mathcal{C}([\![\Gamma]\!].[\![\alpha_1]\!]^{[\![\alpha_2]\!]})$, which maps a d-tuple $h_1, \ldots, h_d \in [\![\Delta]\!]$ to $f : [\![\Gamma]\!] \longrightarrow [\![\alpha_1]\!]^{[\![\alpha_2]\!]}$

$$
\begin{aligned}
&[\![\Delta \Vdash \lambda x.Mx : (\Gamma \vdash \alpha_2 \to \alpha_1)]\!] \\
&= \lambda_{[\![\Gamma]\!]} \circ [\![\Delta \Vdash Mx : (\Gamma, x : \alpha_2 \vdash \alpha_1)]\!] \\
&= \lambda_{[\![\Gamma]\!]} \circ \mathcal{C}([\![\Gamma]\!] \times [\![\alpha_2]\!], ev) \circ \ltimes_{[\![\Gamma]\!]\times[\![\alpha_2]\!]} \circ \\
&\quad \langle [\![\Delta \Vdash M : (\Gamma, x : \alpha_2 \vdash \alpha_2 \to \alpha_1)]\!], [\![\Delta \Vdash x : (\Gamma, x : \alpha_2 \vdash \alpha_2)]\!] \rangle \\
&= \lambda_{[\![\Gamma]\!]} \circ \mathcal{C}([\![\Gamma]\!] \times [\![\alpha_2]\!], ev) \circ \ltimes_{[\![\Gamma]\!]\times[\![\alpha_2]\!]} \circ \\
&\quad \langle \mathcal{C}(\langle \pi_1, \ldots, \pi_n \rangle, [\![\alpha_1]\!]^{[\![\alpha_2]\!]}) \circ m, tp_{[\![\alpha_2]\!]}(\pi_{n+1}) \circ !_{[\![\Delta]\!]} \rangle \text{ — } (\dagger)
\end{aligned}
$$

$$
\begin{aligned}
tp_{[\![\alpha_2]\!]}(\pi_{n+1}) \circ !_{[\![\Delta]\!]} &: h_1, \ldots, h_d \longmapsto \pi_{n+1} \\
\mathcal{C}(\langle \pi_1, \ldots, \pi_n \rangle, [\![\alpha_1]\!]^{[\![\alpha_2]\!]}) &: f \longmapsto f \circ \langle \pi_1, \ldots, \pi_n \rangle \\
\ltimes_{[\![\Gamma]\!]\times[\![\alpha_2]\!]} &: (f \circ \langle \pi_1, \ldots, \pi_n \rangle, \pi_{n+1}) \longmapsto \langle f \circ \langle \pi_1, \ldots, \pi_n \rangle, \pi_{n+1} \rangle \\
\mathcal{C}([\![\Gamma]\!] \times [\![\alpha_2]\!], ev) &: \langle f \circ \langle \pi_1, \ldots, \pi_n \rangle, \pi_{n+1} \rangle \longmapsto ev \circ \langle f \circ \langle \pi_1, \ldots, \pi_n \rangle, \pi_{n+1} \rangle
\end{aligned}
$$

Thus, (†) maps the d-tuple to

$$
\begin{aligned}
&\lambda(ev \circ \langle f \circ \langle \pi_1, \ldots, \pi_n \rangle, \pi_{n+1} \rangle) \\
&= \lambda(ev \circ (f \times id_{[\![\alpha_2]\!]}) \circ \langle \pi_1, \ldots, \pi_n, \pi_{n+1} \rangle) \\
&= \lambda(ev \circ (f \times id_{[\![\alpha_2]\!]})) \\
&= f
\end{aligned}
$$

Therefore, (†) and m are identical morphisms.

7 Soundness of Meta-level Substitution and Conversions

Lemma 2 (Meta-level Substitution Lemma)

$$
\frac{\begin{array}{c}[\![\Delta, X : \tau' \Vdash M : \tau]\!] = m : [\![\Delta]\!] \times [\![\tau']\!] \longrightarrow [\![\tau]\!] \\ [\![\Delta \Vdash L : \tau']\!] = \ l \ : [\![\Delta]\!] \longrightarrow [\![\tau']\!]\end{array}}{[\![\Delta \Vdash M[L/X] : \tau]\!] = m \circ \langle id_{[\![\Delta]\!]}, l \rangle : [\![\Delta]\!] \longrightarrow [\![\tau]\!]}
$$

Proof By induction on the structure of the term M.

The substitution rule for meta-level variables is thus immune with respect to the binding of (base-level) variables, as the following equation implies.

$$
(\zeta X.\lambda x.X) \natural x = (\lambda x.X)[x/X] = \lambda x.x \tag{1}
$$

(23)

This means that MLC has a term that corresponds to the following map at the level of the object language, which is not the case in STLC.

(24) $$\phi \longmapsto \lambda x.\phi \tag{2}$$

Just as $\alpha/\beta/\eta$-conversions are sound in STLC, meta-level $\alpha/\beta/\eta$-conversions are sound in MLC.

Theorem 12 (**Soundness of Meta-level α -Conversion**)

$$(M\alpha)\frac{\Delta, X : \tau_2 \Vdash M : \tau_1 \qquad Y \notin fmv(M)}{[\![\Delta \Vdash \zeta X.M : \tau_2 \Rightarrow \tau_1]\!] = [\![\Delta \Vdash \zeta Y.M[Y/X] : \tau_2 \Rightarrow \tau_1]\!]}$$

Proof Suppose that $|\Delta| = d$.

$$\begin{aligned}
&[\![\Delta \Vdash \zeta X.M : \tau_2 \Rightarrow \tau_1]\!] \\
&= \lambda([\![\Delta, X : \tau_2 \Vdash M : \tau_1]\!]) \\
&= \lambda([\![\Delta, X : \tau_2 \Vdash M : \tau_1]\!] \circ \langle \pi_1, \ldots, \pi_{d+1} \rangle) \\
&= \lambda([\![\Delta, X : \tau_2 \Vdash M : \tau_1]\!] \circ \langle \pi_1, \ldots, \pi_d, \pi_{d+2} \rangle \circ \langle id_{[\![\Delta]\!] \times [\![\tau_2]\!]}, \pi_{d+1} \rangle) \\
&= \lambda([\![\Delta, Y : \tau_2, X : \tau_2 \Vdash M : \tau_1]\!] \circ \langle id_{[\![\Delta]\!] \times [\![\tau_2]\!]}, [\![\Delta, Y : \tau_2 \Vdash Y : \tau_2]\!] \rangle) \\
&= \lambda([\![\Delta, Y : \tau_2 \Vdash M[Y/X] : \tau_1]\!]) \\
&= [\![\Delta \Vdash \zeta Y.M[Y/X] : \tau_2 \Rightarrow \tau_1]\!]
\end{aligned}$$

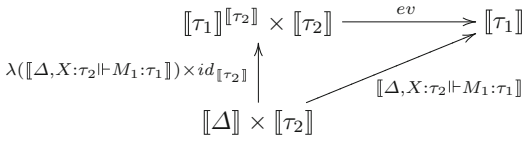
Theorem 13 (**Soundness of Meta-level β-Conversion**)

$$(M\beta)\frac{\Delta, X : \tau_2 \Vdash M_1 : \tau_1 \qquad \Delta \Vdash M_2 : \tau_2}{[\![\Delta \Vdash (\zeta X.M_1)\not\zeta M_2 : \tau_1]\!] = [\![\Delta \Vdash M_1[M_2/X] : \tau_1]\!]}$$

This is proved by a standard triangular identity in Cartesian closed categories.

Proof

$$
\begin{aligned}
&[\![\Delta \Vdash (\zeta X.M_1) \not\downarrow M_2 : \tau_1]\!] \\
&= ev \circ \langle [\![\Delta \Vdash \zeta X.M_1 : \tau_2 \Rightarrow \tau_1]\!], [\![\Delta \Vdash M_2 : \tau_2]\!] \rangle \\
&= ev \circ \langle \lambda([\![\Delta, X : \tau_2 \Vdash M_1 : \tau_1]\!]), [\![\Delta \Vdash M_2 : \tau_2]\!] \rangle \\
&= [\![\Delta, X : \tau_2 \Vdash M_1 : \tau_1]\!] \circ \langle id_{[\![\Delta]\!]}, [\![\Delta \Vdash M_2 : \tau_2]\!] \rangle \\
&= [\![\Delta \Vdash M_1[M_2/X] : \tau_1]\!]
\end{aligned}
$$



Theorem 14 (Soundness of Meta-level η-Conversion)

$$
(M\eta)\frac{\Delta \Vdash M : \tau_2 \Rightarrow \tau_1 \qquad X \notin fmv(M)}{[\![\Delta \Vdash \zeta X.M \not\downarrow X : \tau_2 \Rightarrow \tau_1]\!] = [\![\Delta \Vdash M : \tau_2 \Rightarrow \tau_1]\!]}
$$

Proof Suppose that $|\Delta| = d$.

$$
\begin{aligned}
&[\![\Delta \Vdash \zeta X.M \not\downarrow X : \tau_2 \Rightarrow \tau_1]\!] \\
&= \lambda([\![\Delta, X : \tau_2 \Vdash M \not\downarrow X : \tau_1]\!]) \\
&= \lambda(ev \circ \langle [\![\Delta, X : \tau_2 \Vdash M : \tau_2 \Rightarrow \tau_1]\!], [\![\Delta, X : \tau_2 \Vdash X : \tau_2]\!] \rangle) \\
&= \lambda(ev \circ \langle [\![\Delta \Vdash M : \tau_2 \Rightarrow \tau_1]\!] \circ \langle \pi_1, \ldots, \pi_d \rangle, \pi_{d+1} \rangle) \\
&= \lambda(ev \circ ([\![\Delta \Vdash M : \tau_2 \Rightarrow \tau_1]\!] \times id_{[\![\tau_2]\!]}) \circ \langle \pi_1, \ldots, \pi_d, \pi_{d+1} \rangle) \\
&= \lambda(ev \circ ([\![\Delta \Vdash M : \tau_2 \Rightarrow \tau_1]\!] \times id_{[\![\tau_2]\!]})) \\
&= [\![\Delta \Vdash M : \tau_2 \Rightarrow \tau_1]\!]
\end{aligned}
$$

8 Conclusion

We have presented a revised syntax, an equational theory, and a revised categorical semantics of MLC, by which we fixed several problems faced by previous formulations of MLC (Bekki 2009; Bekki and Asai 2010), and proved the soundness of the theory, including base-level/meta-level substitution and $\alpha/\beta/\eta$-conversions.

By virtue of this revision, we provide a more reliable foundation for the research based on monadic translations in terms of internal monads in Bekki (2009), Bekki and Asai (2010), Hayashishita and Bekki (2011), which enables us to maintain and further develop the result of those analyses.

MLC promises to find application in at least two areas: in linguistics, we are ready to investigate other linguistic phenomena to which monadic analyses may

yield appropriate structures; and in theory of programming languages, we may be able to apply MLC to such tasks as code generation and partial evaluation, among others.

References

Bekki, D. (2009). Monads and meta-lambda calculus. In H. Hattori, T. Kawamura, T. Ide', M. Yokoo & Y. Murakami (Eds.), *New Frontiers in Artificial Intelligence Conference and Workshops, (JSAI 2008), Asahikawa, Japan, June 2008, Revised Selected Papers from LENLS5* (Vol. LNAI 5447, pp. 193–208), Springer.

Bekki, D., & Asai, K. (2010). Representing covert movements by delimited continuations. In K. Nakakoji, Y. Murakami & E. McCready (Eds.), *New Frontiers in Artificial Intelligence JSAI-isAI 2009 Workshops, Tokyo, Japan, November 2009, Selected Papers from LENLS7* (Vol. LNAI 6284, pp. 161–180). Heidelberg: Springer.

Crole, R. L. (1993) *Categories for types*. Cambridge: Cambridge University Press.

Danvy, O., & Filinski, A. (1990) Abstracting control. *LFP90, the 1990 ACM Conference on Lisp and Functional Programming* (pp. 151–160).

Hayashishita, J. R., & Bekki, D. (2011). Conjoined nominal expressions in Japanese: Interpretation through monad, In *The Eighth International Workshop on Logic and Engineering of Natural Language Semantics (LENLS8)* (pp. 139–152). JSAI International Symposia on AI 2011, Takamatsu, Kagawa, Japan.

Kock, A. (1970). Strong functors and monoidal monads. Various Publications Series 11, Aarhus Universitet, August 1970.

Lambek, J. (1980). From λ-calculus to cartesian closed categories. In: J. P. Seldin & J. R. Hindley (Eds.), *To H. B. Curry: Essays on combinatory logic, lambda calculus and formalism*. New York: Academic Press.

Lambek, J., & Scott, P. J. (1986). *Introduction to higher order categorical logic*. Cambridge: Cambridge University Press.

MacLane, S. (1997). *Categories for the working mathematician. Graduate texts in mathematics* (2nd ed.). New Yok: Springer.

Masuko, M., & Bekki, D. (2011). Categorical semantics of meta-lambda calculus. *The 13th JSSST Workshop on Programming and Programming Languages (PPL2011) (in Japanese)* (pp. 60–74), Joozankei, Japan.

Moggi, E. (1989). Computational lambda-calculus and monads. *Fourth Annual IEEE Symposium on Logic in Computer, Science* (pp. 14–23).

Ogata, N. (2008). Towards computational non-associative lambek lambda-calculi for formal pragmatics. *The Fifth International Workshop on Logic and Engineering of Natural Language Semantics (LENLS2008) in Conjunction with the 22nd Annual Conference of the Japanese Society for Artificial Intelligence* (pp. 79–102), Asahikawa, Japan.

Shan, C. C. (2001). Monads for natural language semantics. In K. Striegnitz (Ed.), The ESSLLI-2001 student session (13th European summer school in logic, language and information, 2001). pp. 285–298.

Unger, C. (2011). Dynamic semantics as monadic computation. *The Eighth International Workshop on Logic and Engineering of Natural Language Semantics (LENLS8)* (pp. 153–164). JSAI International Symposia on AI 2011, Takamatsu, Kagawa, Japan.

Wadler, P. (1992). Comprehending monads. *Mathematical structure in computer science* (Vol. 2, pp. 461–493). Cambridge: Cambridge University Press. (an ealier version appears in Conference on Lisp and Functional Programming, Nice, France, ACM, June 1990).