Kripke-Style Contextual Modal Type Theory

Yuito Murase
The University of Tokyo
Tokyo, Japan
murase@lyon.is.s.u-tokyo.ac.jp

ABSTRACT

Under the Curry-Howard isomorphism, modal operators correspond to the type of closed code. Nanevski et al. generalized this result and proposed the contextual modal type theory. They introduced the notion of context that corresponds to free variables of code. Therefore the contextual modal type theory treats open code.

This paper provides another formulation of contextual modal type theory: Kripke-style contextual modal type theory. Our type system is based on the Kripke-style formulation of modal logic, whereas the original system is based on the dual-context formulation. The resulting system has Lisp-like quasiquotation, and hence we expect that KCMTT is adequate for the basis of syntactical metaprogramming.

KEYWORDS

Contextual Modal Type Theory, Lambda Calculus, Modal Logic, Metaprogramming

1 INTRODUCTION

The theory of modal calculi, which corresponds to intuitionistic modal logic through the Curry-Howard correspondence [11], have been studied since 1990s [1, 6, 8, 9]. It is known that some modalities correspond to types of closed code, that is, code without free variables. For example, the type $\Box A$ represents closed code that will be evaluated to the value of type A. From this perspective, modal calculi have been studied as a foundation for staged computation and run-time code generation [3].

The main restriction of modal calculi is that they can manipulate only closed codes. Nanevski et al [7] proposed a solution to this problem, as Contextual Modal Type Theory(CMTT). Contextual modal types are a generalized notion of modal types. They are allowed to have an environment in a modal operator. For example, the type $[x\colon A, y\colon B]C$ represents code that will be evaluated to the value of type C, *under the environment* $x\colon A, y\colon B$. As you can see, modal types are the special case of contextual modal types, where environments are always empty (and therefore code is closed). CMTT is based on Pfenning and Davies' dual-context modal type system [8] (we borrow the name 'dual-context' from Kavvos [4]). Their modal type system corresponds to S4 modal logic, and therefore CMTT corresponds to S4 modal logic.

In this paper, we propose another type system for CMTT. To distinguish from the original CMTT, we call our type system as Kripke-style CMTT (KCMTT). As the name shows, KCMTT is a generalization of the Kripke-style modal type system[3, 6, 9], where contexts form stack and terms have Lisp-like quasi-quotation. As a result, KCMTT provides four variations that correspond to K, T, K4, S4 respectively. KCMTT is different from the original CMTT at this point. The following table shows the position of KCMTT among related work.

	Modal Type	Contextual Modal Type
Dual-Context	[8]	[7]
Kripke-Style	[6, 9]	KCMTT

The paper is structured as follows. In Section 2, we provide Kripke-style contextual modal logic, which is the logic part of KCMTT. In section 3, we give the definition KCMTT in detail and show fundamental properties. Finally, we discuss future work and our motivation for KCMTT.

2 KRIPKE-STYLE CONTEXTUAL MODAL LOGIC

Before the definition of the type system, we introduce Kripke-style natural deduction for contextual modal logic(KCML). KCML is a natural extension of Pfenning and Davies' [8] Kripke-style modal logic. The fundamental idea of KCML and Kripke-style modal logic is the Kripke-style judgment, which has a stack of context.

First, we explain the notion of Kripke-style hypothetical judgment and then construct natural deduction system. We also show that our system is well-defined, that is, introduction and elimination rules for contextual modality satisfy local-soundness and local completeness [8].

2.1 Kripke-Style Hypothetical Judgment

First, we introduce Kripke-style hypothetical judgment, a generalization of hypothetical judgment. The idea of Kripke-style hypothetical judgment is not new: Martini and Masini [6] and Pfenning and Wong [9] initially proposed Kripke-style judgment around the same time, to construct modal calculi.

In a Kripke-style hypothetical judgment, hypotheses form a stack, where semicolons separate contexts. We write $A, B, \ldots$ for propositions, Γ for hypotheses, and Ψ for a stack of hypotheses.

$$\Gamma_m; \Gamma_{m-1}; \ldots; \Gamma_1 \vdash A$$

Informally, this judgment states the following fact from the viewpoint of Kripke semantics: for arbitrary world sequence $w_m \to w_{n-1} \to \ldots \to w_1$, the proposition A holds at the world w_1 if Γ_m holds in w_m, Γ_{m-1} holds in w_{m-1}, and so on. When the context stack has single context, it is equivalent to hypotetical judgment.

In the rest of this section, we construct a natural deduction system on the Kripke-style hypothetical judgment. First, we define the following hyp rule, as we can use the assumption in the current world as the conclusion.

$$\text{hyp}\ \frac{A \in \Gamma}{\Psi; \Gamma \vdash A}$$

We construct natural deduction system of KCML in the following way. First, we define structural properties that Kripke-style judgment should satisfy. Afterward, we add rules for logical connectives so that those principles are formally proved as metatheory.

First, we define the following substitution principle. The substitution principle states that assumptions in a level can be replaced with other assumptions when we can conclude all of the former assumptions from the latter ones. This principle generalizes the usual substitution principle which substitutes a single variable. This style is useful when we reason quotations in Section 3.

Substitution Principle If $\Psi; A_1, \ldots, A_m; \Psi' \vdash B$ and $\Psi; \Gamma \vdash A_i$ holds for all $1 \leq i \leq n$, then $\Psi; \Gamma; \Psi' \vdash B$.

In addition to substitution principle, we can add two structural principles imposing some properties of world relations: reflexivity and transitivity.

As we said before, a stack of context is corresponds to a sequence of worlds. When the world relation satisfies reflexsivity, any two adjacent worlds in the stack can be same. Therefore it is natural to assume that we can merge them.

Reflexive Principle If $\Psi; \Gamma; \Gamma'; \Psi' \vdash A$, then $\Psi; \Gamma, \Gamma'; \Psi' \vdash A$.

Same discussion applies when the world relation satisfies transitivity. In this case, we can insert contexts between adjacent contexts.

Transitive Principle If $\Psi; \Gamma; \Psi' \vdash A$, then $\Psi; \ldots; \Gamma; \Psi' \vdash A$.

As a result, we have four variations of logic depending on whether we assume reflexivity and transitivity of the world relation. In classical modal logic [5], it is known that K, T, K4, and S4 modal logic correspond to those properties of the world relation. Therefore we identify symbols K, T, K4, S4 with those variations. In the rest of this paper, we write $\Psi \vdash_K A$ when we assume no properties of the world relation. We write $\Psi \vdash_T A$ when we assume reflexivity, $\Psi \vdash_{K4} A$ when we assume transitivity, and $\Psi \vdash_{S4} A$ when we assume both. We just write $\Psi \vdash A$ when we do not assume those conditions.

2.2 Kripke-Style Natural Deduction

Now we are ready to construct a natural deduction system for KCML. For simplicity, we consider the fragment with implication and contextual modality. Let us denote propositional variables with $P, Q, \ldots$. Propositions in KCML are inductively defined as follows.

Context $\Gamma ::= \cdot \mid A, \Gamma$

Propositions $A, B ::= P \mid A \to B \mid [\Gamma]A$

For a contextual modality $[\Gamma]A$, we call the formar part *context part*, and the latter *body part.*

Let us define introduction and elimination rules for logical connectives. For implications, their introduction and elimination rules are almost same as usual hypothetical judgment.

$$\to \text{I}\ \frac{\Psi; \Gamma, A \vdash B}{\Psi; \Gamma \vdash A \to B} \quad \to \text{E}\ \frac{\Psi; \Gamma \vdash A \to B \qquad \Psi; \Gamma \vdash A}{\Psi; \Gamma \vdash B}$$

Rules for implications are concerned with only the current world. Other worlds in the context stack are used only when we use contextual modal operator.

The introduction rule for contextual modality is defined as follows.

$$[\,]\text{I}\ \frac{\Psi; \Gamma \vdash A}{\Psi \vdash [\Gamma]A}$$

Kripke's multiple world semantics justifies this rule. Let us think of a special case where $\Psi; \Gamma; \cdot \vdash A$. The current world corresponds to the arbitrary world next to Γ, and we can interpret "A holds for any world next to Γ". By the definition of modal operator in Kripke's multiple world semantics, we conclude that "$\Box A$ holds at Γ". Contextual modality generalizes modal operator to have context.

When we assume neither reflexivity nor transitivity, the corresponding elimination rule is defines as follows. This rule states that A holds in the next world assuming Γ when $[B_1 \ldots B_m]A$ holds at the current world and B_i holds in the next world for each i assuming Γ. As you can see, introduction / elimination rules for modal operator interact with context stack by pushing and popping.

$$[\,]\text{E}\ \frac{\Psi \vdash [B_1 \ldots B_m]A \qquad \Psi; \Gamma \vdash B_i \text{ for } 1 \leq i \leq m}{\Psi; \Gamma \vdash A}$$

We can generalize this elimination rules to support reflexivity and transitivity as follows. Assuming reflexivity, we can identify the current world as the next world, and this corresponds to the case $l = 0$. Assuming transitivity, the lth next world is also the next world for $l > 1$.

$$[\,]\text{E}_l\ \frac{\Psi \vdash [B_1, \ldots, B_m]A \qquad \Psi; \Gamma_l; \ldots; \Gamma_1 \vdash B_i \text{ for } 1 \leq i \leq m}{\Psi; \Gamma_l; \ldots; \Gamma_1 \vdash A}$$

$$\text{where} \begin{cases} l = 1 & \text{for K} \quad l = 0, 1 \quad \text{for T} \\ l \geq 1 & \text{for K4} \quad l \geq 0 \quad \text{for S4} \end{cases}$$

Pfenning and Davies[8] stated that the elimination rule should not be too strong or too weak concerning the introduction rule, and proposed two conditions that introduction/elimination rules should satisfy: local soundness and local completeness. We should confirm that introduction/elimination rules for contextual modal types satisfy these conditions.

Let us think of the case of S4. The same discussion holds for K, T, and K4. Local soundness is the property that an elimination rule is not too strong with respect to the introduction rule. This property is shown by the following local reduction pattern where $n \geq 0$. This pattern demonstrates that we can omit introduction followed by elimination. $\mathcal{D}'$ is generated from $\mathcal{D}$ and $\mathcal{E}$, with substitution, reflexive, and transitive principle.

$$[\,]\text{E}_l\ \frac{[\,]\text{I}\ \dfrac{\overset{\mathcal{D}}{\Psi; A_1, \ldots, A_m \vdash B}}{\Psi \vdash [A_1, \ldots, A_m]B} \qquad \overset{\mathcal{E}}{\Psi; \Gamma_l; \ldots; \Gamma_1 \vdash A_i \text{ for } 1 \leq i \leq m}}{\Psi; \Gamma_l; \ldots; \Gamma_1 \vdash B}$$

$$\Downarrow R$$

$$\overset{\mathcal{D}'}{\Psi; \Gamma_l; \ldots; \Gamma_1 \vdash B}$$

On the other hand, local completeness is the property that an elimination rule is not too weak with respect to the introduction rule. This property is shown by the following local expansion pattern. This pattern demonstrates that original judgment (in this pattern, $\Psi \vdash [A_1, \ldots, A_m]B$) can be restored after elimination.

$$\begin{array}{c}\mathcal{D}\\ \Psi \vdash [A_1,\ldots,A_m]B\end{array}$$

$$\Downarrow E$$

$$[\,]\mathrm{E}_1\ \dfrac{\begin{array}{c}\mathcal{D}\\ \Psi \vdash [A_1,\ldots,A_m]B\end{array} \qquad \mathrm{hyp}\ \dfrac{}{\Psi; A_1,\ldots,A_m \vdash A_i}\ \text{for } 1 \le i \le m}{[\,]\mathrm{I}\ \dfrac{\Psi; A_1,\ldots,A_m \vdash B}{\Psi \vdash [A_1,\ldots,A_m]B}}$$

We call this natural deduction system KCML, which consists of Kripke-style judgment, the hyp rule, and the introduction and elimination rules for implication and contextual modality.

2.3 Fundamental Properties

Theorem 2.1. *(1)* $\Psi \vdash_K A \Rightarrow \Psi \vdash_X A$ *for* $X \in \{T, K4, S4\}$
(2) $\Psi \vdash_T A \Rightarrow \Psi \vdash_{S4} A$
(3) $\Psi \vdash_{K4} A \Rightarrow \Psi \vdash_{S4} A$

Proof. For (1), it easy to show that derivation tree of $\Psi \vdash_K A$ is also derivation tree of $\Psi \vdash_X A$ for $X \in \{T, K4, S4\}$. Same discussion for (2) and (3). □

Finally, we formally prove that KCML satisfies the substitution, reflexive, and transitive principles.

Theorem 2.2. *(1) For* $X \in \{K, T, K4, S4\}$, *if* $\Psi; A_1 \ldots A_m; \ldots \vdash_X B$ *and* $\Psi; \Gamma \vdash_X A_i$ *holds for all* $1 \le i \le m$, *then* $\Psi; \Gamma; \ldots \vdash_X B$.
(2) For $X \in \{T, S4\}$, *if* $\Psi; \Gamma; \Gamma'; \ldots \vdash_X B$, *then* $\Psi; \Gamma, \Gamma'; \ldots \vdash_X B$.
(3) For $X \in \{K4, S4\}$, *if* $\Psi; \Gamma; \ldots \vdash_X B$, *then* $\Psi; \ldots; \Gamma; \ldots \vdash_X B$.

Proof. By induction on the derivation rules. □

2.4 Examples

We show some examples provable in KCML.

(1) $\vdash_K [C](A \to B) \to [C]A \to [C]B$
(2) $B \vdash_T [B]A \to A$
(3) $\vdash_{K4} [C]A \to [D][C]A$ (6) $\vdash_K [A]A$
(4) $\vdash_K [C]A \to [C, D]A$ (7) $\vdash_K [A]B \to [\,](A \to B)$
(5) $\vdash_K [C, C]A \to [C]A$ (8) $\vdash_K [\,](A \to B) \to [A]B$

In these examples, you can see that contextual modality generalizes modality. Each of (1), (2), (3) corresponds to the contextual version of axioms K, T, and 4. Note that (2) is derivable assuming reflexivity, and (3) assuming transitivity. Figure 1 gives the derivation tree of (1). We omit derivation trees for other examples.

$$\to\mathrm{I}\ \dfrac{\to\mathrm{I}\ \dfrac{[\,]\mathrm{I}\ \dfrac{\to\mathrm{E}\ \dfrac{[\,]\mathrm{E}_1\ \dfrac{\mathrm{hyp}\ \dfrac{}{\Gamma \vdash_K [C](A \to B)} \quad \mathrm{hyp}\ \dfrac{}{\Gamma; C \vdash_K C}}{\Gamma; C \vdash_K A \to B} \qquad \begin{array}{c}\vdots\\ \Gamma; C \vdash_K A\end{array}}{\Gamma; C \vdash_K B}}{\Gamma \vdash_K [C]B}}{[C](A \to B) \vdash_K [C]A \to [C]B}}{\vdash_K [C](A \to B) \to [C]A \to [C]B}$$

where $\Gamma := [C](A \to B), [C]A$

Figure 1: Example of a Derivation Tree

With the introduction rule for contextual modality, we can identify the proposition $[\Gamma]A$ with the hypothetical judgment $\Gamma \vdash A$ (not Kripke-style hypothetical judgment). The propositions (4), (5) and (6) represent weakening, contraction, and the hyp rule, demonstrating this idea.

The propositions (7) and (8) show that contextual modality is equivalent to modality with implication. In this sense, contextual modal logic is not stronger than modal logic. However, we think that contextual modality enables us a finer analysis of some notions. For example, our motivation for contextual modality is to reason binding manipulation on open code. We give detail in Section 4.

3 KRIPKE-STYLE CONTEXTUAL MODAL TYPE THEORY

In this section, we construct Kripke-style contextual modal type theory (KCMTT), a typed lambda calculus which corresponds to KCML through the Curry-Howard correspondence [11].

3.1 Type System

We write $x, y, \ldots$ for variables, τ for base types, $l, m, n \ldots$ for non-negative integers. The syntax of KCMTT is defined as follows.

Types	S, T	$::= \tau \mid S \to T \mid [S_1, \ldots, S_m]T$
Terms	M, N, L	$::= x \mid \lambda x{:}\, T.M \mid MN$
		$\mid\ `\langle x_1{:}T_1, \ldots, x_m{:}T_m\rangle M \mid\ ,_l\langle N_1, \ldots, N_m\rangle M$
Context	Γ	$::= \cdot \mid \Gamma, x{:}\, T$
Context Stack	Ψ	$::= \cdot \mid \Psi; \Gamma$
Judgment	J	$::= \Psi \vdash_X M{:}\, T\ (X \in \{K, K4, T, S4\})$

For a context $\Gamma = x_1 : T_1 \ldots x_m : T_m$, we define *the domain of* Γ as $dom(\Gamma) = x_1, \ldots, x_m$ and *the range of* Γ as $rg(\Gamma) = T_1, \ldots, T_m$. For a context stack $\Psi = \Gamma_1; \ldots; \Gamma_m$, we also define the domain of Ψ as $dom(\Psi) = dom(\Gamma_1), \ldots, dom(\Gamma_m)$. Let us think of a Kripke-style judgment $\Psi \vdash M : T$. For the case of K and K4, it is enough to assume that variables in the range of Γ are distinct for each context Γ in Ψ. However, in the case of T and S4 adjacent contexts can be merged by the reflexive principle, and therefore we assume that all variables in the domain of Ψ are distinct.

In KCMTT, two constructs are added to simply typed lambda calculus [11]: *quotation* and *unquotation*. From the viewpoint of staged computation, a quotation $`\langle\Gamma\rangle M$ can be interpreted as "a code of $M$ under the environment $\Gamma$". On the other hand, unquotation can be seen as "evaluation of the code $M$ through $l$ stages, giving the environmt $N_1, \ldots, N_m$". As a special case, 0-level unquotation can be interpreted as `eval` function in Lisp.

The typing rules of KCMTT are defined as follows. Those rules correspond to deduction rules of KCML in Section 2. In the rest of this paper, we assume that all terms are typed: for any term M, there exists a type judgment $\Psi \vdash M : T$. We also identify terms under α-equivalence in Definition 3.5, and hence we can rename bound variables.

$$(\mathrm{Var})\ \frac{x : T \in \Gamma}{\Psi; \Gamma \vdash x : T} \quad (\mathrm{Abs})\ \frac{\Psi; \Gamma, x{:}\, T \vdash M : S}{\Psi; \Gamma \vdash \lambda x{:}\, T.M : T \to S}$$

$$(\mathrm{App})\ \frac{\Psi; \Gamma \vdash M : T \to S \qquad \Psi; \Gamma \vdash N : T}{\Psi; \Gamma \vdash MN : S}$$

$$(\text{Quo})\ \frac{\Psi;\Gamma \vdash M : T}{\Psi \vdash `\langle\Gamma\rangle M : [rg(\Gamma)]T}$$

$$(\text{Unq})_l\ \frac{\Psi \vdash M : [T_1,\dots,T_m]S \qquad \begin{array}{c}\Psi;\Gamma_l;\dots;\Gamma_1 \vdash N_i : T_i\\ \text{for } 1 \le i \le m\end{array}}{\Psi;\Gamma_l;\dots;\Gamma_1 \vdash\ ,_{l\langle N_1,\dots,N_m\rangle}M : S}$$

$$\text{where}\begin{cases} l = 1 & \text{for K} \quad l = 0,1 \quad \text{for T}\\ l \ge 1 & \text{for K4} \quad l \ge 0 \quad \text{for S4}\end{cases}$$

In KCMTT, free variables have levels which correspond to the depth of the context stack. For $l \ge 1$, the set of level-l free variables in a term M, $FV_l(M)$, is defined as follows.

$$\begin{aligned}
FV_l(x) &= \begin{cases}\{x\} & \text{when } l = 1\\ \varnothing & \text{otherwise}\end{cases}\\
FV_l(\lambda x{:}\,T.M) &= \begin{cases}FV_l(M) - \{x\} & \text{when } l = 1\\ FV_l(M) & \text{otherwise}\end{cases}\\
FV_l(MN) &= FV_l(M) \cup FV_l(N)\\
FV_l(`\langle\Gamma\rangle M) &= FV_{l+1}(M)\\
FV_l(,_{k\langle N_1,\dots,N_m\rangle}M) &\\
&= \begin{cases}\bigcup_{1\le i\le m} FV_l(N_i) & \text{when } l \le k\\ FV_{l-k}(M) \cup \bigcup_{1\le i\le m} FV_l(N_i) & \text{otherwise}\end{cases}
\end{aligned}$$

For a judgment $\Gamma_m;\dots;\Gamma_1 \vdash M : T$, $FV_l(M)$ corresponds to the l-th context Γ_l. We can formally state this idea by the following lemma.

LEMMA 3.1. *If* $\Psi;\Gamma_l;\dots;\Gamma_1 \vdash M{:}\,T$, *then* $FV_l(M) \subseteq dom(\Gamma_l)$.

PROOF. By induction on the derivation. □

A quotation $`\langle\Gamma\rangle M$ corresponds to the $[\,]$-introduction rule in KCML. Γ is required to include all level-1 free variables in M by the (Quo) rule, and therefore there are no ill-formed codes with "undeclared variables". An unquotation $,_{l\langle N_1,\dots,N_m\rangle}M$ corresponds to the $[\,]$-elimination rule in KCML. In contrast to quotation, it substitutes all level-1 free variables in M with $N_1,\dots,N_m$.

3.2 Substitution

For $l \ge 1$, a substitution $[N_1/x_1,\dots,N_m/x_m]_l$ is a meta operation that maps a term to a term. It substitutes level-l free variables $x_1,\dots,x_m$ with terms $N_1,\dots,N_m$, respectively.

Substitution is inductively defined as follows. We denote σ for a content of substitution. For $\sigma = N_1/x_1,\dots,N_m/x_m$, we define $FV_l(\sigma) = \bigcup_{1\le i\le m} FV_l(N_i)$ and $dom(\sigma) = x_1,\dots,x_m$. We assume that all variables in $dom(\sigma)$ are distinct.

$$\begin{aligned}
x[\sigma]_l &= \begin{cases}N & \text{when } l = 1 \text{ and } N/x \in \sigma\\ x & \text{otherwise}\end{cases}\\
(MN)[\sigma]_l &= (M[\sigma]_l)(N[\sigma]_l)\\
(\lambda x{:}\,A.M)[\sigma]_l &= \begin{cases}\lambda x{:}\,A.(M[\sigma]_l) & \text{when } l = 1,\ x \notin dom(\sigma) \text{ and } x \notin FV_1(\sigma)\\ \lambda x{:}\,A.(M[\sigma]_l) & \text{when } l > 1\end{cases}\\
(`\langle\Gamma\rangle M)[\sigma]_l &= `\langle\Gamma\rangle(M[\sigma]_{l+1})\\
(,_{k\langle N_1\dots N_m\rangle}M)[\sigma]_l &= \begin{cases},_{k\langle N_1[\sigma]_l\dots N_m[\sigma]_l\rangle}M & \text{when } l \le k\\ ,_{k\langle N_1[\sigma]_l\dots N_m[\sigma]_l\rangle}M[\sigma]_{l-k} & \text{otherwise}\end{cases}
\end{aligned}$$

Substitution corresponds to rewriting proof trees with the substitution principle in KCML. The following substitution lemma formally states the substitution principle in KCMTT.

LEMMA 3.2 (SUBSTITUTION LEMMA). *If* $\Psi;x_1{:}\,S_1,\dots,x_m : S_m;\Gamma_{l-1};\dots;\Gamma_1 \vdash M{:}\,T$ *and* $\Psi;\Gamma \vdash N_i : S_i$ *for all* $1 \le i \le m$ *, then* $\Psi;\Gamma;\Gamma_{l-1};\dots;\Gamma_1 \vdash M[\sigma]_l{:}\,T$ *, where* $\sigma = N_1/x_1,\dots,N_m/x_m$.

PROOF. By induction on the derivation rules. □

Note that this substitution is capture-avoiding one, though there is apparently no collision check for quotation. It works because the substitution and the bindings of the quotation are at different levels. As a result of substitution lemma, we can state that weakening, exchange, and single substitution preserves types.

3.3 Level Substitution

For $l \ge 1$ and $m \ge 0$, a level substitution $\uparrow_l^m$ is a meta operation that maps a term to a term.

$$\begin{aligned}
x \uparrow_l^m &= x\\
(MN) \uparrow_l^m &= (M \uparrow_l^m)(N \uparrow_l^m)\\
(\lambda x{:}\,A.M) \uparrow_l^m &= \lambda x{:}\,A.(M \uparrow_l^m)\\
(`\langle\Gamma\rangle M) \uparrow_l^m &= `\langle\Gamma\rangle(M \uparrow_{l+1}^m)\\
(,_{k\langle N_1,\dots,N_n\rangle}M) \uparrow_l^m &= \begin{cases},_{k+m-1\langle N_1\uparrow_l^m,\dots,N_n\uparrow_l^m\rangle}M & \text{when } l \le k\\ ,_{k\langle N_1\uparrow_l^m,\dots,N_n\uparrow_l^m\rangle}M \uparrow_{l-k}^m & \text{otherwise}\end{cases}
\end{aligned}$$

The idea of the level substitution may not be intuitive. Proof theoretically, it corresponds to rewriting proof-trees with reflexive/transitive principles. The following lemmas formally state those principles.

LEMMA 3.3 (LEVEL SUBSTITUTION LEMMA). *(i) For* $X \in \{T, S4\}$*, if* $\Psi;\Gamma_{l+1};\Gamma_l;\dots;\Gamma_1 \vdash_X M{:}\,T$ *, then* $\Psi;\Gamma_{l+1},\Gamma_l;\dots;\Gamma_1 \vdash_X M \uparrow_l^0{:}\,T$ *.*
(ii) For $X \in \{K4, S4\}$ *and* $m > 1$*, if* $\Psi;\Gamma_{l+1};\Gamma_l;\dots;\Gamma_1 \vdash_X M{:}\,T$ *, then* $\Psi;\Gamma_{l+1};\Psi';\Gamma_l;\dots;\Gamma_1 \vdash_X M \uparrow_l^m{:}\,T$ *where* Ψ' *is size-*$(m-1)$ *stack of empty contexts.*

PROOF. By induction on the derivation rules. □

Note that a level substitution $\uparrow_1^1$ is identity on terms. Therefore the level substituion lemma for the K variant is trivial.

3.4 Equivalence on Terms

Now we are ready to define α-equivalence, β-reduction and η-expansion rules.

Definition 3.4. Let $\sim$ be a binary relation on terms. $\sim$ is *compatible* iff it satisfies the following conditions.

$$M_1 \sim M_2 \Rightarrow \lambda x{:}\,T.M_1 \sim \lambda x{:}\,T.M_2$$
$$M_1 \sim M_2 \Rightarrow (M_1N) \sim (M_2N)$$
$$M_1 \sim M_2 \Rightarrow (NM_1) \sim (NM_2)$$
$$M_1 \sim M_2 \Rightarrow `\langle\Gamma\rangle M_1 \sim `\langle\Gamma\rangle M_2$$

$$M_1 \sim M_2 \Rightarrow\ ,l\langle N_1 \ldots N_m \rangle M_1 \sim\ ,l\langle N_1 \ldots N_m \rangle M_2$$

$$M_1 \sim M_2 \Rightarrow\ ,l\langle L_1 \ldots L_m, M_1, L'_1 \ldots L'_n \rangle N \sim\ ,l\langle L_1 \ldots L_m, M_2, L'_1 \ldots L'_n \rangle N$$

Definition 3.5. α-equivalence $=_\alpha$ is the least transitive, reflexive, and compatible relation on terms satisfying the following:

$$\lambda x{:}\, T.M =_\alpha \lambda y{:}\, T.(M[x/y]_1)$$

$$`\langle x_1{:}T_1, \ldots, x_m{:}T_m \rangle M =_\alpha\ `\langle y_1{:}T_1, \ldots, y_m{:}T_m \rangle (M[y_1/x_1, \ldots, y_m/x_m]_1)$$

Definition 3.6. β-reduction $\overset{R}{\Rightarrow}$ and η-expansion $\overset{E}{\Rightarrow}$ are the least compatible relations on terms which satisfy the following:

$$(\lambda x{:}\, A.M)N \overset{R}{\Rightarrow} M[N/x]_1$$

$$,k\langle N_1, \ldots, N_m \rangle `\langle x_1{:}T_1, \ldots, x_m{:}T_m \rangle M \overset{R}{\Rightarrow} M \uparrow_1^k [N_1/x_1, \ldots, N_m/x_m]_1$$

$$M \overset{E}{\Rightarrow} \lambda x{:}\, T.Mx \quad \text{when } \Psi \vdash M : T \to S$$

$$M \overset{E}{\Rightarrow}\ `\langle x_1{:}T_1, \ldots, x_m{:}T_m \rangle (\,,1\langle \vec{x} \rangle M) \quad \text{when } \Psi \vdash M : [T_1, \ldots, T_m]S$$

Finally, we get subject reduction and expansion.

THEOREM 3.7 (SUBJECT REDUCTION/EXPANSION). *(i) If* $\Psi \vdash M{:}\, T$ *and* $M \overset{R}{\Rightarrow} N$*, then* $\Psi \vdash N{:}\, T$.
(ii) If $\Psi \vdash M{:}\, T$ *and* $M \overset{E}{\Rightarrow} N$*, then* $\Psi \vdash N{:}\, T$.

PROOF. By induction on the definition of $\overset{R}{\Rightarrow}$ and $\overset{E}{\Rightarrow}$. The base cases are proved by Lemma 3.2 and 3.3. □

3.5 Examples

The following examples show KCMTT type judgments which correspond to examples in Section 2. We use X, Y for higher level variables, and a, b for lower ones.

(1) $\vdash_K \lambda X{:}\, [C](A \to B).\lambda Y{:}\, [C]A.`\langle a{:}C \rangle (\,,1\langle a \rangle X)(\,,1\langle a \rangle Y)$
$: [C](A \to B) \to [C]A \to [C]B$
(2) $a : B \vdash_T \lambda X{:}\, [B]A.\,,0\langle a \rangle X : [B]A \to A$
(3) $\vdash_{K4} \lambda X{:}\, [C]A.`\langle a{:}D \rangle `\langle b{:}C \rangle\,,2\langle b \rangle X : [C]A \to [D][C]A$
(4) $\vdash_K \lambda X{:}\, [C]A.`\langle a{:}C, b{:}D \rangle\,,1\langle a \rangle X : [C]A \to [C, D]A$
(5) $\vdash_K \lambda X{:}\, [C, C]A.`\langle a{:}C \rangle\,,1\langle a, a \rangle X : [C, C]A \to [C]A$
(6) $\vdash_K `\langle a{:}A \rangle a : [A]A$
(7) $\vdash_K \lambda X{:}\, [A]B.`\langle\rangle(\lambda a{:}\, A.\,,1\langle a \rangle X) : [A]B \to [](A \to B)$
(8) $\vdash_K \lambda X{:}\, [](A \to B).`\langle a{:}A \rangle(\,,1\langle\rangle X)a : [](A \to B) \to [A]B$

4 FUTURE WORK

In this paper, we introduced the overview of KCMTT.

There are many problems to be solved. This paper only provides subject reduction/expansion and does not prove confluency and strong normalization. We expect that proofs in previous work of Kripke-style modal type theory may be helpful. Comparison between S4 KCMTT and the original CMTT is also necessary. We expect that they have equal expressiveness, but otherwise the difference can be interesting. Davies and Pfenning[3] provide mutual translation between S4 Kripke-style modal type theory and dual-context modal type theory. This translation may help us to solve the problem.

Our goal is to construct a type system for syntactical metaprogramming such as macro system in Common Lisp [12] or Template Haskell [10]. Although those implementations give a great extensibility to programming languages, they are known not to be type-safe. In other words, code with syntactic extension, even if it is well-typed, may extend to ill-typed code. Therefore we want a type system for syntactical metaprogramming that guarantees type-safety of syntactic extension.

The basic idea of such metaprogramming is quasiquotation syntax, which enables programmers to construct code. There are some formal systems which provide Lisp-like quasiquotation such as Kripke-style modal calculi [3] and linear temporal calculi [2]. However, they are not capable of binding manipulation: Kripke-style modal calculi only treat closed code, and linear temporal calculi do not allow access to free variables in open code. We think KCMTT can be the basis for type system of syntactical metaprogramming because it provides Lisp-like quasiquotation and allows access to free variables in open code.

With comparison to CMTT, we think that KCMTT is preferable as the type system for this purpose. First, it has quasiquotation constructs. Second, it is sufficiently *weak* as a logical system. We do not need runtime code evaluation for syntactical metaprogramming. It is known that the T axiom corresponds to runtime code evaluation, and therefore we may omit assumption on the reflexivity. KCMTT provides K and K4 variants, and we think those variants perceive the nature of syntactical metaprogramming.

ACKNOWLEDGMENTS

We thank Yuichi Nishiwaki and members of Hagiya Laboratory for valuable discussion. We also appreciate anonymous referees for helpful comments and suggestions.

REFERENCES

[1] Gianluigi Bellin, Valeria de Paiva, and Eike Ritter. 2001. Extended Curry-Howard correspondence for a basic constructive modal logic. In *Proceedings of Methods for Modalities*. http://ukpmc.ac.uk/abstract/CIT/485479
[2] Rowan Davies. 2017. A Temporal Logic Approach to Binding-Time Analysis. *J. ACM* 64, 1 (2017), 1–45.
[3] Rowan Davies and Frank Pfenning. 2001. A Modal analysis of Staged Computation. *J. ACM* 48, 3 (2001), 555–604. https://doi.org/10.1145/382780.382785
[4] G. A. Kavvos. 2017. Dual-Context Calculi for Modal Logic. In *Logic in Computer Science*. Reykjavik. arXiv:1602.04860 http://arxiv.org/abs/1602.04860
[5] Saul A. Kripke. 1963. Semantical Analysis of Modal Logic I Normal Modal Propositional Calculi. *Mathematical Logic Quarterly* 9, 5-6 (1963), 67–96. https://doi.org/10.1002/malq.19630090502
[6] Simone Martini and Andrea Masini. 1996. A Computational Interpretation of Modal Proofs. In *Proof Theory of Modal Logic*. 213–241. https://doi.org/10.1007/978-94-017-2798-3_12
[7] Aleksandar Nanevski, Frank Pfenning, and Brigitte Pientka. 2008. Contextual modal type theory. *ACM Transactions on Computational Logic* 9, 3 (2008), 1–49. https://doi.org/10.1145/1352582.1352591
[8] Frank Pfenning and Rowan Davies. 2001. A Judgmental Reconstruction of Modal Logic. *Mathematical Structures in Computer Science* 11, 4 (2001), 511–540. https://doi.org/10.1017/S0960129501003322
[9] Frank Pfenning and H. C. Wong. 1995. On a Modal λ-Calculus for S4. *Electronic Notes in Theoretical Computer Science* 1, C (1995), 515–534. https://doi.org/10.1016/S1571-0661(04)00028-3
[10] Tim Sheard and Simon Peyton Jones. 2002. Template Meta-programming for Haskell. *SIGPLAN Not.* 37, 12 (Dec. 2002), 60–75. https://doi.org/10.1145/636517.636528
[11] Morten Heine Sørensen and Pawel Urzyczyn. 2006. *Lectures on the Curry-Howard Isomorphism, Volume 149 (Studies in Logic and the Foundations of Mathematics)*. Elsevier Science Inc., New York, NY, USA.
[12] Guy L. Steele, Jr. 1990. *Common LISP: The Language (2Nd Ed.)*. Digital Press, Newton, MA, USA.