Bounded Modal Type Theory
Fine-Grained Scope-Safety in Modal Lambda-Calculus

***Abstract*—Modal lambda calculi provide theoretical foundations for scope-safe multi-stage programming, rejecting programs using resources from inaccessible stages. Meanwhile, there remain practical staged programs that are yet to be reasoned by existing proposals. Our *Bounded Modal Type Theory (BMTT)* fills this gap by incorporating *classifiers* into S4 modal types. Classifiers, as type-level objects representing variable scopes, enable fine-grained reasoning about resource visibility.**

We define sound and complete Kripke semantics for its logical counterpart, providing a guideline for the syntactic well-formedness of stages and scopes. We also prove confluence, strong normalization, and canonicity for its reduction semantics. These results demonstrate that BMTT would provide a theoretical foundation for various areas beyond multi-stage computation.

***Index Terms*—modal logic, modal lambda calculus, Kripke semantics, multi-stage programming**

I. Introduction

The well-known Curry–Howard isomorphism [1], [2] points out the correspondence between logic and programming languages. Programming language researchers have leveraged this correspondence to find sophisticated foundations for advanced type systems, validating them from logical perspectives. In this context, *modal lambda-calculi* were developed in explorations for computational interpretation of modal logic. Prior work found out that we can interpret several variants of modal lambda-calculi as type systems for effectful computation [3], guarded recursion [4]–[6], multi-stage programming [7]–[11], distributed computing [12] to name a few.

In this paper, we focus on applications to multi-stage programming, explaining concepts in modal calculi through concrete examples. Multi-Stage Programming (MSP) provides a syntactic mechanism to construct code fragments as first-class values that can be executed in later evaluation stages. In this context, a modal type $\Box A$ is interpreted as a type of a code representation for an expression typed A.

The issue we want to resolve in this paper is that **we do not have modal lambda-calculus that is granular enough to reason about *cross-stage persistence* in MSP**. We start with step-by-step explanations of key concepts in MSP and existing research before detailing this problem.

One of the core concerns in typing staged programs is *scope-safety*: as different stages do not always share variable environments, we want to ensure that there will not be scoping issues where invisible variables are accidentally used. We show example programs in the style of MetaML [13], [14] among other syntax proposed for MSP. We think its *quasi-quotation* syntax provides concise and clear semantics, and in fact influenced design of many staged programming languages, such as MetaOCaml [15], [16], Scala 3 [17] and Template Haskell [18].

The following function `spower` specializes a power function by taking a number `n` and returns a code fragment of a function that multiples its argument `n` times.

```
let rec spower_ n x =
  if n == 0 then `{ 1 }
  else `{ ~x * ~{ spower_ (n-1) x } } in
let spower n =
  `{ fun y => ~{ spower_ n `{ y } } }
```

Here a *quote* `` `{$M$} `` produces a code representation of M rather than evaluating it, and a *splice* `~{`M`}` embeds a code fragment from m into another quote. When we call `spower 3`, it finally returns `` `{fun x => x * x * x * 1} ``. To use the code fragment from `spower 3`, we can move the generated program to the next stage or evaluate it at the current stage using the **`run`** primitive, known as *run-time evaluation*. In the following program, we perform run-time evaluation on `spower 3` to define `power3`.

```
let power3 = run (spower 3)
(* fun y => y * y * y * 1 *)
```

The key insight in MSP is that variables are available only in the evaluation stage where they are defined. For example, the following program results in an error because it attempts to evaluates `x`, which is defined in a future stage.

```
`{ let x = 10 in ~{ spower x } }
```

In order to detect this sort of scoping errors, we need to care about visibility of variables at each point of a given program.

As mentioned earlier, certain modal lambda-calculi can be regarded as type systems that detect scoping errors in staged programs. Well-known calculi are **S4** modal lambda-calculus [7] and $\lambda_{\bigcirc}$, a calculus that corresponds to Linear-time Temporal logic (**LTL**) [8]. They provide different scoping disciplines, which leads to the well-known characterizations: **S4** modal types represent closed/persistent code fragments while **LTL** types represent open code fragments [19]. Here, persistent code fragments are those that can safely undergo run-time evaluation.

However, their type systems are known to be too coarse because we want to allow both openness and persistence [19]. This issue has motivated the exploration of novel modal calculi with fine-grained reasoning about scope-safety. For example, Taha et al. proposed a type system with *environment classifiers* [20], which is later formalized as a modal lambda-calculus $\lambda^{\triangleright}$ by Tsukada and Igarashi [9]. As another approach, Nanevski et al. presented *Contextual Modal Type Theory (CMTT)* that generalizes $\mathbf{S4}$ modal types to represent open yet persistent code fragments [10].

However, certain practical features for MSP remain unaddressed by existing modal lambda-calculi: now it is time to explain the concept of Cross-Stage Persistence (CSP) [14], [15], [21], [22]. Consider the following example.

```
let x = 10 in run `{ x + 10 }
```

Here, the variable x is defined at the current stage while used in the quote, which appears to be ill-scoped. However, run-time evaluation lifts the content of the quote to the current stage, and the variable is resolved to its definition properly. Hence, it is sometimes safe to use variables at (apparently) future stages, which is called CSP. The essence of CSP is that it requires a single code fragment to exibit both openness and persistence. $\lambda^{\triangleright}$ does not satisfy this requirement: it allows openness *or* persistence, not both. While CMTT seems to satisfy this requirement, but it cannot directly represent CSP due to its inherent nature; we will discuss this topic in Section VIII. In this sense, further granular scoping disciplines are required to type staged programs with CSP.

A. Our Contribution

To fill this gap, we present *Bounded Modal Type Theory (BMTT)*, a novel modal lambda-calculus that incorporates *classifiers* into $\mathbf{S4}$ modal types.

Classifiers are first-class objects in BMTT that represent visible resources. They can also be interpreted as variable environments or scopes. This idea is inspired from refined environment classifiers by Kiselyov et al. [23], which we compare in Section VIII.

With support of classifiers, BMTT maintains fine-grained scope-safety using two types. A *bounded modal type* $[\succeq^{\gamma} A]$ annotates a $\mathbf{S4}$ modal type with the classifier γ, which specifies the lower bound of resources for A to hold. A *polymorphic classifier type* $\forall \gamma_1 :\succeq \gamma_2 . A$ provides a first-order bounded polymorphism over classifiers, allowing flexible typing combined with bounded modal types.

We devote Section II for providing the intuition of how BMTT maintains scope-safety with classifiers. Section III provides its type system and reduction semantics. Section IV explains how BMTT encodes staged programs, including CSP, and the current gap between BMTT and applicactions to MSP. Section V defines a type-preserving embedding from $\lambda_{\bigcirc}$ into BMTT, which implies the potential applications of BMTT to practical languages with MetaML-style MSP.

At the same time, it should be emphasized that BMTT itself is a pure modal lambda-calculus. Section VI considers a logical counterpart of BMTT and defines sound and complete Kripke semantics for it. In fact, our model for BMTT worked as a guide for the syntactic design of BMTT, especially its judgment structure. Furthermore, we investigate metatheory of BMTT including strong normalization, confluence, canonicity and the subformula property in Section VII. These results validate BMTT from a logical perspective, and we expect that BMTT would provide theoretical foundations for various areas beyond MSP.

```
(a) ◎

(b) let x:int@γ1 = 10 in
    fun f:int->int@γ2 => ◎

(c) let x:int@γ1 = 10 in
    fun f:int->int@γ2 =>
      `{γ3:>! ◎ }

(d) let x:int@γ1 = 10 in
    fun f:int->int@γ2 =>
      `{γ3:>! ~{@γ2 ◎ } }

(e) let x:int@γ1 = 10 in
    fun f:int->int@γ2 =>
      `{γ3:>! ~{@γ2 λγ4:>γ3. ◎ } }
```

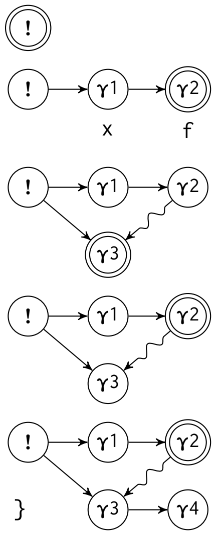


Fig. 1. Scoping and Staging Structure in BMTT, where $\longrightarrow$ and $\rightsquigarrow$ between nodes indicates scope and stage transitions, respectively

Note that we omit many details and proofs which can be find in Appendix.

II. Basic Intuition and Informal Descriptions

This section is devoted to providing basic concepts of BMTT: *classifier, position, scope transition and modal transition*. Classifiers are represented by γ, δ, and can be regarded to represent visible resources, variable environments or variable scopes either. BMTT heavily uses classifiers to maintain consistency of scopes and stages.

In BMTT, each part of a program is associated with a corresponding classifier, representing visible resources at the point. Given a specific part of a program, we refer to the corresponding classifier as the *position* of the part. In Figure 1, the left side provides partial programs with holes represented by ◎. The right side visualizes how BMTT maintains classifiers, where the node with double circle corresponds to the position of the ◎ in the left side.

(a) Here ◎ is put toplevel. Its position is the *initial classifier* **!**, a special classifier that represents empty resources. At this point, only the initial classifier exists on the right side, where double square indicates that it is the position of the hole.

(b) When we introduce new variables via **let** or **fun** in the hole, new classifiers are introduced that represent resources with the new variable in addition to the existing ones. In this case, declarations for x and f carry new classifiers γ_1 and γ_2, where x is visible at γ_1 and x, f are visible at γ_2. Here, you can see on the right side that arrows $\longrightarrow$ are drawn between classifiers, indicating the increase of resources. We call these arrows *scope transitions*. We write $\gamma \preceq \gamma'$ if there is an scope transition between γ and γ'. Note that scope transitions are reflexive and transitive while we omit derivable ones from the visualization. We can also see that declaring a new variable moves its position to the corresponding classifier; hence γ_2 is the new position in (b).

(c) Then we put a quote inside the hole, carrying a classifier declaration $\boldsymbol{\gamma_3} :\succeq \mathbf{!}$. It introduces a new classifier γ_3 with the lower bound $\mathbf{!}$, moving its position to γ_3. Hence, γ_3 represents abstract resources that are larger than $\mathbf{!}$. We see that there is a scope transition between $\mathbf{!}$ and γ_3. This quote also moves the stage of the hole, which is indicated by the *modal transition* from γ_2 to γ_3. This transition is visualized by the waving arrow $\rightsquigarrow$ on the right side, and we write $\gamma_2 \sqsubseteq \gamma_3$ to mean this modal transition. Modal transitions are reflexive and transitive, and also satisfy $\gamma \preceq \gamma' \Rightarrow \gamma \sqsubseteq \gamma'$. Therefore, any classifiers that are reachable by $\longrightarrow$ and $\rightsquigarrow$ in the visualizations have modal transitions.

(d) An unquote moves its position to somewhere in the past stages. It carries an existing classifier that is reachable by modal transition in *backward* direction. In this case, the unquote moves its position to γ_2 where we can confirm $\gamma_2 \sqsubseteq \gamma_3$.

(e) BMTT provides *classifier abstraction*, which introduces *polymorphic classifiers*. Here, a polymorhic classifier γ_4 is declared with the lower bound γ_3. The point of classifier abstraction is that it introduces a scope transition that is independent from its position. As you can see on the right side, a new scope transition is added from γ_3 to γ_4 without moving its position γ_2. We also have classifier application, but we omit it in this section because it does not affect classifiers and position.

As we have seen, BMTT keeps track of granular information on classifiers, positions and scope/modal transitions. The question for readers, then, is how we use it to ensure scope safety. First, we state the *well-scopedness principle*, which defines a scope-safe use of variables.

Principle 1 (Well-Scopedness)**.** *Supposing a variable x is declared with γ_1, a use of x is* well-scoped *if and only if its position γ_2 satisfies $\gamma_1 \preceq \gamma_2$.*

At any point, we know its position and the classifier of a variable declaration of x; hence, we can always check scope transition to see whether x is visible from the current position. We can lift this idea to general terms, which establishes an expected property for BMTT.

Principle 2 (Monotonicity)**.** *If a term M is well-scoped at the posision γ_1 and $\gamma_1 \preceq \gamma_2$ holds, then M is also well-scoped at the position γ_2.*

As computation proceeds, a part of a program may move to another place where it becomes ill-scoped. From this view, monotonicity provides the condition on places into which a program can move without scoping issues. The reduction semantics of BMTT takes care of this principle, and achieves static scope-safety guarantee. Therefore, we consider monotonicity the fundamental property in BMTT, capturing the essence of scope-safety.

If you are familiar with Kripke semantics for intuitionistic modal logic, concepts in this section might remind you of birelational Kripke semantics for intuitionistic variants of **S4**. That is correct, but only partly. Later in Section VI, we present our Kripke semantics for BMTT and demonstrate how scoping disciplines in BMTT fits into our models.

III. Formal Definitions

Now we look into the formal definitions of BMTT, and prove its basic properties based on the intuition provided in the last section.

A. Syntax and Typing Rules

Typing disciplines in BMTT incorporate classifiers, positions, scope transitions, and modal transitions. These notions are incorporated into *contexts*.

$$\Gamma, \Delta ::= \epsilon \mid \Gamma, \boldsymbol{x}{:}^{\gamma} A \mid \Gamma, \text{🔓}^{\gamma_1 :\succeq \gamma_2} \mid \Gamma, \text{🔒}^{\gamma} \mid \Gamma, \boldsymbol{\gamma_1} :\succeq \gamma_2$$

ϵ represents an empty context. Variable declaration $\boldsymbol{x}{:}^{\gamma} A$, open lock $\text{🔓}^{\gamma_1 :\succeq \gamma_2}$, closing lock 🔒^{γ}, and polymorphic classifier declaration $\boldsymbol{\gamma_1} :\succeq \gamma_2$ correspond to cases (b)–(e) in the last section, respectively.

Contexts in BMTT are responsible for maintaining 1. declared variables and classifiers 2. its position 3. transitions between classifiers. For the first one, we sometimes regard a context as a set of variable declarations, like $\boldsymbol{x}:{}^{\gamma} A \in \Gamma$. $\mathbf{Dom}_{\mathrm{C}}(\Gamma)$ and $\mathbf{Dom}_{\mathrm{V}}(\Gamma)$ stand for declared classifiers and variables in Γ, respectively. We write Γ_1, Γ_2 for a context that appends Γ_1 and Γ_2.

For the second one, we define a function $\mathrm{pos}(\Gamma)$, which returns the position of Γ.

$$\mathrm{pos}(\epsilon) = \mathbf{!} \quad \mathrm{pos}(\Gamma, \boldsymbol{x}{:}^{\gamma} A) = \gamma \quad \mathrm{pos}(\Gamma, \text{🔓}^{\gamma_1 :\succeq \gamma_2}) = \gamma_1$$
$$\mathrm{pos}(\Gamma, \text{🔒}^{\gamma}) = \gamma \qquad \mathrm{pos}(\Gamma, \boldsymbol{\gamma_1} :\succeq \gamma_2) = \mathrm{pos}(\Gamma)$$

For the sake of space, we introduce a shorthand notation for positions.

Notation 1. We write Γ^{γ} to represent Γ with its position γ. When we write a context with multiple meta-variables like $\Gamma_1^{\gamma_1}, \Gamma_2^{\gamma_2}$, then it means that $\mathrm{pos}(\Gamma_1) = \gamma_1$ and $\mathrm{pos}(\Gamma_1, \Gamma_2) = \gamma_2$ hold. Note that it does not mean $\mathrm{pos}(\Gamma_2) = \gamma_2$ because Γ_2 can be empty.

For the third one, we have judgments for a *scope transition* $\Gamma \vdash \gamma_1 \preceq \gamma_2$ and a *modal transition* $\Gamma \vdash \gamma_1 \sqsubseteq \gamma_2$. While we omit their definitions, they are basically designed so that the following lemma to hold. Note that judgments in BMTT require $\vdash \Gamma : \mathbf{ctx}$ by definition, stating well-formedness of Γ, which is defined later.

Lemma 1. *Judgments below are derivable.*

1) $\Gamma_1^{\gamma}, \boldsymbol{x}{:}^{\boldsymbol{\delta}} A, \Gamma_2 \vdash \gamma \preceq \delta$
2) $\Gamma_1^{\gamma_1}, \text{🔓}^{\boldsymbol{\delta} :\succeq \gamma_2}, \Gamma_2 \vdash \gamma_2 \preceq \delta$
3) $\Gamma_1^{\gamma_1}, \text{🔓}^{\boldsymbol{\delta} :\succeq \gamma_2}, \Gamma_2 \vdash \gamma_2 \sqsubseteq \delta$
4) $\Gamma_1^{\gamma_1}, \text{🔓}^{\boldsymbol{\delta} :\succeq \gamma_2}, \Gamma_2 \vdash \gamma_1 \sqsubseteq \delta$
5) $\Gamma_1, \boldsymbol{\delta} :\succeq \gamma, \Gamma_2 \vdash \gamma \preceq \delta$

Lemma 2. *Statements below hold. ($\trianglelefteq = \preceq$ or $\sqsubseteq$)*

1) $\gamma \in \mathbf{Dom}_{\mathrm{C}}(\Gamma) \implies \Gamma \vdash \gamma \trianglelefteq \gamma$.
2) $\Gamma \vdash \gamma_1 \trianglelefteq \gamma_2$ *and* $\Gamma \vdash \gamma_2 \trianglelefteq \gamma_3 \implies \Gamma \vdash \gamma_1 \trianglelefteq \gamma_3$.
3) $\Gamma \vdash \gamma_1 \preceq \gamma_2 \implies \Gamma \vdash \gamma_1 \sqsubseteq \gamma_2$.

$$\text{Var}\ \frac{\boldsymbol{x} :{}^{\boldsymbol{\gamma_1}} A \in \Gamma^{\gamma_2} \quad \Gamma^{\gamma_2} \vdash \gamma_1 \preceq \gamma_2}{\Gamma^{\gamma_2} \vdash x : A} \quad \to\text{-I}\ \frac{\Gamma, \boldsymbol{x}:^{\boldsymbol{\gamma}} A_1 \vdash M : A_2 \quad \gamma_1 \notin \mathbf{FC}(A_2)}{\Gamma \vdash \lambda \boldsymbol{x} : {}^{\gamma} A_1.\, M : A_1 \to A_2} \quad \to\text{-E}\ \frac{\Gamma \vdash M_1 : A_1 \to A_2 \quad \Gamma \vdash M_2 : A_1}{\Gamma \vdash M_1\ M_2 : A_2}$$

$$[]\text{-I}\ \frac{\Gamma, \text{🔓}^{\boldsymbol{\gamma_1} : \succeq \gamma_2} \vdash M : A \quad \gamma_1 \notin \mathbf{FC}(A)}{\Gamma \vdash \text{‘}\{^{\boldsymbol{\gamma_1} : \succeq \gamma_2} M\} : [^{\succeq \gamma_2} A]} \qquad []\text{-E}\ \frac{\Gamma^{\gamma_1}, \text{🔒}^{\gamma_2} \vdash M : [^{\succeq \gamma_3} A] \quad \Gamma^{\gamma_1} \vdash \gamma_3 \preceq \gamma_1}{\Gamma^{\gamma_1} \vdash \sim\{^{\gamma_2} M\} : A}$$

$$\forall\text{-I}\ \frac{\Gamma, \boldsymbol{\gamma_1} : \succeq \gamma_2 \vdash M : A}{\Gamma \vdash \lambda \boldsymbol{\gamma_1} : \succeq \gamma_2.\, M : \forall \boldsymbol{\gamma_1} : \succeq \gamma_2.\, A} \qquad \forall\text{-E}\ \frac{\Gamma \vdash M : \forall \boldsymbol{\gamma_1} : \succeq \gamma_2.\, A \quad \Gamma \vdash \gamma_2 \preceq \gamma_3}{\Gamma \vdash M \gamma_3 : A[\boldsymbol{\gamma_1} := \gamma_3]}$$

Fig. 2. Typing Rules

A judgment $\vdash \Gamma : \mathbf{ctx}$ states well-formedness of Γ. The rule WF-🔒 is worth mentioning, which ensures the context $\Gamma^{\gamma}, \text{🔒}^{\delta}$ to satisfy $\Gamma^{\gamma} \vdash \delta \sqsubseteq \gamma$. This condition corresponds to the behavior of unquote, as explained in case (d) of the last section.

$$\text{WF-🔒}\ \frac{\vdash \Gamma^{\gamma} : \mathbf{ctx} \qquad \Gamma^{\gamma} \vdash \delta \sqsubseteq \gamma}{\vdash \Gamma^{\gamma}, \text{🔒}^{\delta} : \mathbf{ctx}}$$

Other rules ensure that types and classifiers appear well-formed and defined in the context. We omit their straightforward definitions. Types and terms are defined in the syntax below.

$$\begin{array}{lll} \textbf{Types} & A, B & ::= p \mid A \to B \mid [^{\succeq \gamma} A] \mid \forall \boldsymbol{\gamma_1} : \succeq \gamma_2.\, A \\ \textbf{Terms} & M, N & ::= x \mid \lambda \boldsymbol{x} : {}^{\gamma} A.\, M \mid M\ N \mid \text{‘}\{^{\boldsymbol{\gamma_1} : \succeq \gamma_2} M\} \\ & & \quad \mid \ \sim\{^{\gamma} M\} \mid \lambda \boldsymbol{\gamma_1} : \succeq \gamma_2.\, M \mid M \gamma_1 \end{array}$$

$\forall \boldsymbol{\gamma_1} : \succeq \gamma_2.\, A$, ‘$\{^{\boldsymbol{\gamma_1} : \succeq \gamma_2} M\}$ and $\lambda \boldsymbol{\gamma_1} : \succeq \gamma_2.\, M$ binds γ_1 in A or M, respectively. $\lambda \boldsymbol{x} : {}^{\gamma} A.\, M$ binds x and γ in M. $\mathbf{FC}(A)$ and $\mathbf{FC}(M)$ represent a set of *free classifiers* in A and M, respectively. $\mathbf{FV}(M)$ represents a set of *free variables* in M. For each binding form, we assume that binding classifiers/variables can be renamed (Barendregt convention).

A judgment $\Gamma \vdash A : \mathbf{type}$ states well-formedness of A, ensuring free classifiers in A to be declared by Γ. Derivation rules are straightforward and omitted. A typing judgment $\Gamma \vdash M : A$ states that M has the type A under the context Γ at the position of Γ. Typing rules are listed in Figure 2.

We look into each typing rule. Var uses a variable x from Γ with the side condition $\Gamma^{\gamma_2} \vdash \gamma_1 \preceq \gamma_2$. This formalizes the well-scopedness principle, ensuring that x is visible from the current position.

$\to$-I and $\to$-E type a *function* $\lambda \boldsymbol{x} : {}^{\gamma} A.\, M$ and an *application* $M\ N$. They are almost the same as the common definition except that a lambda abstraction carries a classifier declaration and $\to$-I has a side condition $\gamma_1 \notin \mathbf{FC}(A_2)$ to avoid dependency. It should be feasible to extend function types to dependent classifier types like $\prod_{A_1 @ \gamma} A_2$, but we omitted it for simplicity.

[]-I introduces a *bounded modal type* $[^{\succeq \gamma_2} A]$ with a *quote* ‘$\{^{\boldsymbol{\gamma_1} : \succeq \gamma_2} M\}$ by discharging an open lock in a context. From the MSP perspective, this bounded modal type indicates that A holds for any classifier γ_1 in a future stage of pos(Γ) with more resources than γ_2. []-E provides $\gamma_1 = \text{pos}(\Gamma)$ as the classifier that satisfies this condition. From WF-🔒, we have $\Gamma^{\gamma_1} \vdash \gamma_2 \sqsubseteq \gamma_1$ for the first condition, and the side condition ensures the second condition $\Gamma^{\gamma_1} \vdash \gamma_3 \preceq \gamma_1$.

Rules for polymorphic classifier types are rather straightforward. $\forall$-I discharges polymorphic classifier declaration, introducing *polymorphic classifier type* $\forall \boldsymbol{\gamma_1} : \succeq \gamma_2.\, A$ with a *classifier abstraction* $\lambda \boldsymbol{\gamma_1} : \succeq \gamma_2.\, M$. On elimination, $\forall$-E requires a given classifier γ_3 to satisfy $\gamma_2 \preceq \gamma_3$. The type $A[\boldsymbol{\gamma_1} := \gamma_3]$ denotes the type obtained by replacing γ_1 occurring in A with γ_3.

B. Basic Properties

BMTT judgments have rich structural properties. First, weakening holds for some of the judgments.

Theorem 1 (Weakening). *Given* $\vdash \Gamma, \Delta : \mathbf{ctx}$*,*

1) $\Gamma \vdash A : \mathbf{type} \Rightarrow \Gamma, \Delta \vdash A : \mathbf{type}$.
2) $\Gamma \vdash \gamma_1 \trianglelefteq \gamma_2 \Rightarrow \Gamma, \Delta \vdash \gamma_1 \trianglelefteq \gamma_2$. *($\trianglelefteq$ = $\preceq$ or $\sqsubseteq$)*

However, weakening does not hold for typing judgments because typeability depends on a context's position. In this sense, a typing judgment pertains to *local* properties, whereas others pertain to *global* properties. Instead, we state a formalized version of the monotonicity principle. In the following definition, $\Delta_1 \# \Delta_2$ indicates that $\mathbf{Dom}_{\mathrm{C}}(\Delta_1)$, $\mathbf{Dom}_{\mathrm{C}}(\Delta_2)$, $\mathbf{Dom}_{\mathrm{V}}(\Delta_1)$, and $\mathbf{Dom}_{\mathrm{V}}(\Delta_2)$ are disjoint.

Theorem 2 (Monotonicty). *Given* $\Delta_1 \# \Delta_2$ *and* $\Gamma^{\gamma_1}, \Delta_1^{\gamma_2} \vdash \gamma_1 \preceq \gamma_2$*, then the following statements hold:*

1) $\vdash \Gamma, \Delta_2 : \mathbf{ctx} \implies \vdash \Gamma, \Delta_1, \Delta_2 : \mathbf{ctx}$.
2) $\Gamma, \Delta_2 \vdash A : \mathbf{type} \implies \Gamma, \Delta_1, \Delta_2 \vdash A : \mathbf{type}$.
3) $\Gamma, \Delta_2 \vdash \delta_1 \preceq \delta_2 \implies \Gamma, \Delta_1, \Delta_2 \vdash \delta_1 \preceq \delta_2$.
4) $\Gamma, \Delta_2 \vdash \delta_1 \sqsubseteq \delta_2 \implies \Gamma, \Delta_1, \Delta_2 \vdash \delta_1 \sqsubseteq \delta_2$.
5) $\Gamma, \Delta_2 \vdash M : A \implies \Gamma, \Delta_1, \Delta_2 \vdash M : A$.

Corollary 1. *If* $\Gamma_1 \vdash M : A$ *and* $\Gamma_1^{\gamma_1}, \Gamma_2^{\gamma_2} \vdash \gamma_1 \preceq \gamma_2$*, then* $\Gamma_1, \Gamma_2 \vdash M : A$.

As described in Section II, monotonicity states the essential property for scope-safety in BMTT. This theorem is used to prove base cases of Lemma 3, where a subterm moves its position without scoping errors.

We define meta-operations before stating other properties. Classifier substitution $\cdot[\boldsymbol{\gamma_1} := \gamma_2]$ operates on classifiers, contexts, types, and terms, substituting free occurrences of γ_1 with γ_2. Variable substitution $\cdot[\boldsymbol{\gamma_1} := \gamma_2, \boldsymbol{x} := M]$ operates on terms, substituting free occurrences of γ_1 and x with γ_2 and M, respectively.

Lemma 3 (Variable Substitution). *Let* $\Delta_1 = \Gamma_1^{\gamma_1}, \boldsymbol{x}:^{\boldsymbol{\gamma_2}} A, \Gamma_2$*, and* $\Delta_2 = \Gamma_1, \Gamma_2[\boldsymbol{\gamma_2} := \gamma_1]$. *Then, the following statements hold.*

1) $\vdash \Delta_1 : \mathbf{ctx} \implies \vdash \Delta_2 : \mathbf{ctx}$.
2) $\Delta_1 \vdash A : \mathbf{type} \implies \Delta_2 \vdash A[\boldsymbol{\gamma_2} := \gamma_1] : \mathbf{type}$.

3) $\Delta_1 \vdash \delta_1 \preceq \delta_2 \implies \Delta_2 \vdash \delta_1[\boldsymbol{\gamma_2}{:=}\gamma_1] \preceq \delta_2[\boldsymbol{\gamma_2}{:=}\gamma_1]$.
4) $\Delta_1 \vdash \delta_1 \sqsubseteq \delta_2 \implies \Delta_2 \vdash \delta_1[\boldsymbol{\gamma_2}{:=}\gamma_1] \sqsubseteq \delta_2[\boldsymbol{\gamma_2}{:=}\gamma_1]$.
5) $\Delta_1 \vdash M_1 \colon B$ *and* $\Gamma_1 \vdash M_2 \colon A$
$\implies \Delta_2 \vdash M_1[\boldsymbol{\gamma_2}{:=}\gamma_1, \boldsymbol{x}{:=}M_2] \colon B[\boldsymbol{\gamma_2}{:=}\gamma_1]$.

Lemma 4 (Rebasing)**.** *Let* $\Delta_1 = (\Gamma_1^{\gamma_1}, \text{🔒}^{\gamma_2}, \text{🔓}^{\boldsymbol{\gamma_3} :\succeq \gamma_4}, \Gamma_2)$, *and* $\Delta_2 = \Gamma_1, \Gamma_2[\boldsymbol{\gamma_3}{:=}\gamma_1]$. *Supposing* $\Gamma_1 \vdash \gamma_4 \preceq \gamma_1$, *the following statements hold,*

1) $\vdash \Delta_1 \colon \mathbf{ctx} \implies \vdash \Delta_2 \colon \mathbf{ctx}$.
2) $\Delta_1 \vdash A \colon \mathbf{type} \implies \Delta_2 \vdash A[\boldsymbol{\gamma_3}{:=}\gamma_1] \colon \mathbf{type}$.
3) $\Delta_1 \vdash \delta_1 \preceq \delta_2 \implies \Delta_2 \vdash \delta_1[\boldsymbol{\gamma_3}{:=}\gamma_1] \preceq \delta_2[\boldsymbol{\gamma_3}{:=}\gamma_1]$.
4) $\Delta_1 \vdash \delta_1 \sqsubseteq \delta_2 \implies \Delta_2 \vdash \delta_1[\boldsymbol{\gamma_3}{:=}\gamma_1] \sqsubseteq \delta_2[\boldsymbol{\gamma_3}{:=}\gamma_1]$.
5) $\Delta_1 \vdash M_1 \colon A \implies \Delta_2 \vdash M[\boldsymbol{\gamma_3}{:=}\gamma_1] \colon A[\boldsymbol{\gamma_3}{:=}\gamma_1]$.

Lemma 5 (Classifier Substitution)**.** *Let* $\Delta_1 = \Gamma_1, \boldsymbol{\gamma_1} :\succeq \gamma_2, \Gamma_2$ *and* $\Delta_2 = \Gamma_1, \Gamma_2[\boldsymbol{\gamma_1}{:=}\gamma_3]$. *Given* $\Gamma_1 \vdash \gamma_2 \preceq \gamma_3$, *then the following statements hold.*

1) $\Delta_1 \vdash A \colon \mathbf{type} \implies \Delta_2 \vdash A[\boldsymbol{\gamma_1}{:=}\gamma_3] \colon \mathbf{type}$.
2) $\vdash \Delta_1 \colon \mathbf{ctx} \implies \vdash \Delta_2 \colon \mathbf{ctx}$.
3) $\Delta_1 \vdash \delta_1 \preceq \delta_2 \implies \Delta_2 \vdash \delta_1[\boldsymbol{\gamma_1}{:=}\gamma_3] \preceq \delta_2[\boldsymbol{\gamma_1}{:=}\gamma_3]$.
4) $\Delta_1 \vdash \delta_1 \sqsubseteq \delta_2 \implies \Delta_2 \vdash \delta_1[\boldsymbol{\gamma_1}{:=}\gamma_3] \sqsubseteq \delta_2[\boldsymbol{\gamma_1}{:=}\gamma_3]$.
5) $\Delta_1 \vdash M \colon A \implies \Delta_2 \vdash M[\boldsymbol{\gamma_1}{:=}\gamma_3] \colon A[\boldsymbol{\gamma_1}{:=}\gamma_3]$.

C. Logical Harmony and Reduction Semantics

The introduction and elimination rules in BMTT are well balanced with respect to local soundness and local completeness [24]. For the sake of space, we skim local soundness/completeness patterns in the following statements.

Lemma 6 (Local Soundness Patterns)**.**

1) $\Gamma^{\gamma_1} \vdash (\lambda \boldsymbol{x} : {}^{\gamma_2}A.\, M)\ N \colon B$
$\implies \Gamma \vdash M[\boldsymbol{\gamma_2}{:=}\gamma_1, \boldsymbol{x}{:=}N] \colon B$.
2) $\Gamma^{\gamma_1} \vdash \boldsymbol{\sim}\{^{\gamma_2}\,\text{‘}\{^{\boldsymbol{\gamma_3} :\succeq \gamma_4} M\}\} \colon A \implies \Gamma \vdash M[\boldsymbol{\gamma_3}{:=}\gamma_1] \colon A$.
3) $\Gamma \vdash (\lambda \boldsymbol{\gamma_1} :\succeq \gamma_2.\, M)\gamma_3 \colon A \implies \Gamma \vdash M[\boldsymbol{\gamma_1}{:=}\gamma_3] \colon A$.

Lemma 7 (Local Completeness Patterns)**.** *(δ is taken freshly)*

1) $\Gamma \vdash M \colon A \to B \implies \Gamma \vdash \lambda \boldsymbol{x} : {}^{\delta}A.\, (M\ x) \colon A \to B$.
2) $\Gamma^{\gamma_1} \vdash M \colon [^{\succeq \gamma_2} A] \Rightarrow \Gamma \vdash \text{‘}\{^{\boldsymbol{\delta} :\succeq \gamma_2} \boldsymbol{\sim}\{^{\gamma_1} M\}\} \colon [^{\succeq \gamma_2} A]$.
3) $\Gamma \vdash M \colon \forall \boldsymbol{\gamma_1} :\succeq \gamma_2.\, A$
$\implies \Gamma \vdash \lambda \boldsymbol{\delta} :\succeq \gamma_2.\, (M \delta) \colon \forall \boldsymbol{\gamma_1} :\succeq \gamma_2.\, A$.

Local soundness and completeness patterns can be regarded as β-reduction and η-expansion, respectively. We define β-reduction on raw terms, notated $M_1 @ \gamma \Rightarrow_\beta M_2$. Here the classifier γ stands for the position of M_1 and M_2. As you can see in the following definition, the position influences M_2.

Definition 1 (β-reduction)**.** $M_1 @ \gamma \Rightarrow_\beta M_2$ is defined to satisfy following rules (along with compatibility rules).

$$(\lambda \boldsymbol{x} : {}^{\gamma_2}A.\, M)\ N @ \gamma_1 \Rightarrow_\beta M[\boldsymbol{\gamma_2}{:=}\gamma_1, \boldsymbol{x}{:=}N]$$
$$\boldsymbol{\sim}\{^{\gamma_2}\,\text{‘}\{^{\boldsymbol{\gamma_3} :\succeq \gamma_4} M\}\} @ \gamma_1 \Rightarrow_\beta M[\boldsymbol{\gamma_3}{:=}\gamma_1]$$
$$(\lambda \boldsymbol{\gamma_2} :\succeq \gamma_3.\, M)\gamma_4 @ \gamma_1 \Rightarrow_\beta M[\boldsymbol{\gamma_2}{:=}\gamma_4]$$

Note that the compatibility rules need to maintain positions for terms that move positions, the full definitions of which can be find in Definition A.8. Also note that positions do not change the computational behavior (such as strong normalization) of β reduciton because classifier substitutions do not change the structure of terms. Finally, we prove subject reduction.

Theorem 3 (Subject Reduction)**.** *If* $\Gamma^{\gamma} \vdash M_1 \colon A$ *and* $M_1 @ \gamma \Rightarrow_\beta M_2$, *then* $\Gamma \vdash M_2 \colon A$.

D. Typing Examples

We show examples of typing judgments in Figure 3. K, T, 4^{-1} and 4 are generalization of common modal axioms. Mon and Mon^{-1} show that $[^{\succeq \gamma} A]$ and $\forall \boldsymbol{\gamma'} :\succeq \gamma.\, [^{\succeq \gamma'} A]$ are equivalent, demonstrating monotonicity in bounded modal types.

Polymorphic classifiers are useful in the case like K^{-1*} where they abstract classifiers shared by multiple bounded modal types. K^{-1*} is one of such interesting examples, which lifts a function on quotes to a quote of a function. Its type is similar to the axiom K^{-1} $(\bigcirc A \to \bigcirc B) \to \bigcirc (A \to B)$ for **LTL**, while BMTT is based on **S4** that lacks K^{-1}. In fact, we can embed $\lambda_{\bigcirc}$, a modal lambda calculus for linear-time temporal logic, to BMTT as we demonstrate in Section V.

We demonstrate more examples of typing in BMTT for staged programs in the next section.

IV. Staged Computational Account of BMTT

In this section, we briefly discuss how we can interpret staged programs using BMTT, and the gap between BMTT to practical applications.

As you have already seen, quotes in BMTT correspond to quotes. Meanwhile, we can consider unquotes to represent either splices or run-time evaluation, based on how many quotes it jumps out: a splice gets out of surrounding quotes while run-time evaluation stays at the same position. For types, a bounded modal type $[^{\succeq \gamma} A]$ represents code fragments of expressions typed A, under the variable environment bigger than γ.

We show how BMTT represents the `spower` example in Section I. Here we assume extensions with recursive definitions, integers and primitive operators, and we omit unnecessary classifiers.

```
let rec spower_[δ1:>!](n:int)(x:[int@δ1])
    :[int@δ1]@γ1 =
  if n == 0 then `{@_:>δ1 1 }
  else `{@δ2:>δ1
    ~{@γ1 x} * ~{@γ1 spower_[δ1] (n-1) x } } in
let spower(n:int):int->[int->int@!]@γ2 =
  `{@δ3:>! fun y:int@δ4 =>
      ~{@γ1 spower_[δ4] n `{@_:>δ4 y } } } in
let power4:int->int@γ3 = ~{@γ1 spower 4 }
```

In this program, the `spower_` function have the type `[δ1:>!](int->[int@δ1]->[int@δ1])` where a polymorphic classifier is introduced by `[δ1:>!]`. We want to call this function with `` `{@δ2 y } `` in the definition of `spower`, but δ2 is not defined at the point of the definition of `spower_`. That is why we use polymorphic classifiers to refer to classifiers that will be defined later. Most of unquotes in this program behave as splices, moving position to outside of surrounding quotes. However, the one in `power4` does not move its position and behaves as run-time evaluation.

$$\begin{array}{lll}
\vdash \lambda\boldsymbol{\gamma} :\succeq \mathbf{!}.\,\lambda\boldsymbol{x}^{\delta_1}.\,\lambda\boldsymbol{y}^{\delta_2}.\,\text{‘}\{^{\boldsymbol{\gamma'}:\succeq\gamma}\sim\{^{\delta_2}x\}\ \sim\{^{\delta_2}y\}\} & : \forall\boldsymbol{\gamma} :\succeq \mathbf{!}.\,[^{\succeq\gamma}A \to B] \to [^{\succeq\gamma}A] \to [^{\succeq\gamma}B] & \text{(K)} \\
\vdash \lambda\boldsymbol{x}^{\gamma}.\,\sim\{^{\gamma}x\} & : [^{\succeq\mathbf{!}}A] \to A & \text{(T)} \\
\vdash \lambda\boldsymbol{\gamma_1} :\succeq \mathbf{!}.\,\lambda\boldsymbol{x}^{\delta}.\,\text{‘}\{^{\boldsymbol{\gamma_2}:\succeq\gamma_1}\sim\{^{\gamma_2}\sim\{^{\delta}x\}\}\} & : \forall\boldsymbol{\gamma} :\succeq \mathbf{!}.\,[^{\succeq\gamma}[^{\succeq\gamma}A]] \to [^{\succeq\gamma}A] & (4^{-1}) \\
\vdash \lambda\boldsymbol{\gamma_1} :\succeq \mathbf{!}.\,\lambda\boldsymbol{x}^{\delta_1}.\,\text{‘}\{^{\boldsymbol{\gamma_2}:\succeq\gamma_1}\text{‘}\{^{\boldsymbol{\gamma_3}:\succeq\gamma_1}\sim\{^{\delta_1}x\}\}\} & : \forall\boldsymbol{\gamma} :\succeq \mathbf{!}.\,[^{\succeq\gamma}A] \to [^{\succeq\gamma}[^{\succeq\gamma}A]] & (4) \\
\vdash \lambda\boldsymbol{\gamma_1} :\succeq \mathbf{!}.\,\lambda\boldsymbol{x}^{\delta}.\,\lambda\boldsymbol{\gamma_2} :\succeq \gamma_1.\,\text{‘}\{^{\boldsymbol{\gamma_3}:\succeq\gamma_2}\sim\{^{\delta}x\}\} & : \forall\boldsymbol{\gamma_1} :\succeq \mathbf{!}.\,([^{\succeq\gamma_1}A] \to \forall\boldsymbol{\gamma_2} :\succeq \gamma_1.\,[^{\succeq\gamma_2}A]) & \text{(Mon)} \\
\vdash \lambda\boldsymbol{\gamma_1} :\succeq \mathbf{!}.\,\lambda\boldsymbol{x}^{\delta}.\,\text{‘}\{^{\boldsymbol{\gamma_3}:\succeq\gamma_1}\sim\{^{\delta}x\gamma_3\}\} & : \forall\boldsymbol{\gamma_1} :\succeq \mathbf{!}.\,((\forall\boldsymbol{\gamma_2} :\succeq \gamma_1.\,[^{\succeq\gamma_2}A]) \to [^{\succeq\gamma_1}A]) & (\text{Mon}^{-1}) \\
 & \multicolumn{2}{r}{\text{where } \gamma_2 \notin \mathbf{FC}(A)} \\
\vdash \lambda\boldsymbol{\gamma_1} :\succeq \mathbf{!}.\,\lambda\boldsymbol{x}^{\delta}.\,\text{‘}\{^{\boldsymbol{\gamma_3}:\succeq\gamma_1}\lambda\boldsymbol{y}^{\gamma_4}.\,\sim\{^{\delta}x\gamma_4\ \text{‘}\{^{\boldsymbol{\gamma_5}:\succeq\gamma_4}y\}\}\} & : \forall\boldsymbol{\gamma_1} :\succeq \mathbf{!}.\,(\forall\boldsymbol{\gamma_2} :\succeq \gamma_1.\,[^{\succeq\gamma_2}A] \to [^{\succeq\gamma_2}B]) \to [^{\succeq\gamma_1}A \to B] & (\text{K}^{-1*}) \\
 & \multicolumn{2}{r}{\text{where } \gamma_2 \notin \mathbf{FC}(A) \cup \mathbf{FC}(B)}
\end{array}$$

Fig. 3. Typing Examples

BMTT cannot type the example with a scoping issue in Section I as expected. In the program below, we want a classifier for ? such that $\boldsymbol{?} \sqsubseteq \gamma_5, \gamma_5 \preceq \boldsymbol{?}$. Only **γ5** satisfies this constraint, but the unquote does not move its position, behaving as run-time evaluation. Hence, we cannot type this unquote as a splice.

```
`{@γ4:>γ3 let x@γ5 = 10 in ~{@? spower x } }
```

The other example of run-time evaluation with CSP can be represented as follows.

```
let x:int@γ1 = 10 in ~{@γ1 `{γ2:>γ1 x + 10 } }
```

The unquote here does not change its position γ_1, interpreted as run-time evaluation. Inside the quote, x can be used because $\gamma_1 \preceq \gamma_2$ holds, and the whole quote is typed $[^{\succeq\gamma_1}\texttt{int}]$. This program remains well-scoped after reducing unquote and quote.

```
let x:int@γ1 = 10 in x + 10
```

Note that an unquote can sometimes be interpreted as *both* a splice and run-time evaluation, leading to ambiguity in the meaning of BMTT terms. In this sense, BMTT is not something that can be used for MSP as-is, while it can be amended by providing proof terms for modal transitions.

Finally, we formalize type-safe residualization [21], indicating adequacy as a type system for MSP (similar properties are called binding-time correctness [8] or eliminability [7]). It essentially states that if a well-typed code generator generates a code fragment, then the content of the fragment can be typed at the toplevel context, i.e., an empty context.

Theorem 4 (Type-Safe Residualization). *If* $\epsilon \vdash M : [^{\succeq\mathbf{!}}A]$ *and* M *is normal with regard to* $\Rightarrow_\beta$*, then* $M = \text{‘}\{^{\gamma:\succeq\mathbf{!}}M'\}$ *for some* M' *and* $\epsilon \vdash M'[\gamma{:=}\mathbf{!}] : A$ *is derivable.*

V. Embedding Other Modal Calculi

As we mentioned earlier, bounded modality in BMTT is based on **S4** Kripke/Fitch-style modal calculus [7], [25]–[27]. In fact, we can translate them to BMTT, where $\Box A$ is interpreted as $[^{\succeq\mathbf{!}}A]$.

Theorem 5. *There is a type-preserving translation from* **S4** *Kripke-style modal lambda-calculus by Davies and Pfenning [7] to BMTT.*

The detailed definitions and proofs can be found in Appendix C1.

More non-trivial is the relation to linear-time temporal lambda-calculus like $\lambda_{\bigcirc}$ by Davies [8], [28], and we focus on this topic. We sketch the definition of $\lambda_{\bigcirc}$ in Figure 4. $\lambda_{\bigcirc}$ influences design of many MetaML-style MSP languages. Therefore, we care about embedding $\lambda_{\bigcirc}$ to BMTT, which implies potential application of BMTT to practical type system for MSP.

$$\frac{x :^{k} A^{\circ} \in \Gamma^{\circ}}{\Gamma^{\circ} \vdash_k x : A^{\circ}} \qquad \frac{\Gamma^{\circ}, x :^{k} A^{\circ} \vdash_k M^{\circ} : B^{\circ}}{\Gamma^{\circ} \vdash_k \lambda x^{A^{\circ}}.\, M^{\circ} : A^{\circ} \to B^{\circ}}$$

$$\frac{\Gamma^{\circ} \vdash_k M^{\circ} : A^{\circ} \to B^{\circ} \qquad \Gamma^{\circ} \vdash_k N^{\circ} : A^{\circ}}{\Gamma^{\circ} \vdash_k M^{\circ}\ N^{\circ} : B^{\circ}}$$

$$\frac{\Gamma^{\circ} \vdash_{k+1} M^{\circ} : A^{\circ}}{\Gamma^{\circ} \vdash_k \mathbf{next}\,\{M^{\circ}\} : \bigcirc A^{\circ}} \qquad \frac{\Gamma^{\circ} \vdash_k M^{\circ} : \bigcirc A^{\circ}}{\Gamma^{\circ} \vdash_{k+1} \mathbf{prev}\,\{M^{\circ}\} : A^{\circ}}$$

$$(\lambda x^{A^{\circ}}.\, M^{\circ}{}_1)\ M^{\circ}{}_2 \Rightarrow_\beta M^{\circ}{}_1[x{:=}M^{\circ}{}_2]$$
$$\mathbf{prev}\,\{\mathbf{next}\,\{M^{\circ}\}\} \Rightarrow_\beta M^{\circ}$$

Fig. 4. Typing and Reduction in $\lambda_{\bigcirc}$ (omit compatibility for β)

This section demonstrates that such a translation can indeed be implemented. Our translation from $\lambda_{\bigcirc}$ basically adopts the idea of Murase et al., where they provide a type-preserving translation from two-level variant of $\lambda_{\bigcirc}$ to their calculus [11]. Furthermore, our translation supports translating the full $\lambda_{\bigcirc}$ with multi-levels, and we prove that it does not only preserves typeability, but also preserves to reduction semantics to some extent.

In essence, typing judgments of $\lambda_{\bigcirc}$ manage the position for each stage simultaneously. Our translation enables BMTT to emulate this behavior, which is defined in Figure 5. Note that we write $k+$ and $k-$ as shorthands for $k+1$ and $k-1$.

Our translation carries three data: current level k, upper-bound level l and mapping between levels and classifiers. This mapping between levels and classifier (or, simply *mapping*) $\gamma_0 \ldots \gamma_l$ is a sequence of $l+1$ classifiers, which maps a level n and corresponding classifier γ_n. This mapping encodes positions for each stage maintained by $\lambda_{\bigcirc}$. Given a mapping $\overrightarrow{\gamma}$, we write $\overrightarrow{\gamma}[k \mapsto \delta]$ to denote a new mapping that updates kth element of $\overrightarrow{\gamma}$ with δ.

Type $\llparenthesis A^\circ \rrparenthesis^{\gamma_{k+}\cdots\gamma_l}_{@k\le l},\ (\!(A^\circ)\!)^{\gamma_{k+}\cdots\gamma_l}_{@k\le l}$

$$\begin{aligned}
\llparenthesis p \rrparenthesis^{\overrightarrow{\gamma}}_{@k\le l} &= p \\
\llparenthesis A^\circ \to B^\circ \rrparenthesis^{\overrightarrow{\gamma}}_{@k\le l} &= (\!(A^\circ)\!)^{\overrightarrow{\gamma}}_{@k\le l} \to \llparenthesis B^\circ \rrparenthesis^{\overrightarrow{\gamma}}_{@k\le l} \\
\llparenthesis \bigcirc A^\circ \rrparenthesis^{\gamma_{k+}\cdots\gamma_l}_{@k\le l} &= [^{\succeq\gamma_{k+}} \llparenthesis A^\circ \rrparenthesis^{\gamma_{k+2}\cdots\gamma_l}_{@k+\le l}] \quad \text{if } k<l \\
(\!(A^\circ)\!)^{\gamma_{k+}\cdots\gamma_l}_{@k\le l} &= \forall_{\boldsymbol{\delta_{k+}}\ge\gamma_{k+}} \cdots \forall_{\boldsymbol{\delta_l}\ge\gamma_l} \llparenthesis A^\circ \rrparenthesis^{\delta_{k+}\ldots\delta_l}_{@k\le l}
\end{aligned}$$

Term $\llparenthesis M^\circ \rrparenthesis^{\gamma_0\cdots\gamma_l}_{@k\le l},\ (\!(M^\circ)\!)^{\gamma_0\cdots\gamma_l}_{@k\le l}$

$$\begin{aligned}
\llparenthesis x \rrparenthesis^{\gamma_0\cdots\gamma_l}_{@k\le l} &= x\gamma_{k+}\ldots\gamma_l \\
\llparenthesis \lambda x^{A^\circ}. M^\circ \rrparenthesis^{\gamma_0\cdots\gamma_l}_{@k\le l} &= \lambda \boldsymbol{x} :{}^{\delta} (\!(A^\circ)\!)^{\gamma_{k+}\cdots\gamma_l}_{@k\le l}. \llparenthesis M^\circ \rrparenthesis^{(\gamma_0\ldots\gamma_l)[k\mapsto\delta]}_{@k\le l} \\
\llparenthesis M^\circ\ N^\circ \rrparenthesis^{\gamma_0\cdots\gamma_l}_{@k\le l} &= \llparenthesis M^\circ \rrparenthesis^{\gamma_0\cdots\gamma_l}_{@k\le l}\ (\!(N^\circ)\!)^{\gamma_0\cdots\gamma_l}_{@k\le l} \\
\llparenthesis \textbf{next}\,\{M^\circ\} \rrparenthesis^{\gamma_0\cdots\gamma_l}_{@k\le l} &= \text{‘}\{^{\boldsymbol{\delta}:\succeq\gamma_{k+}} \llparenthesis M^\circ \rrparenthesis^{(\gamma_0\ldots\gamma_l)[k+\mapsto\delta]}_{@k+\le l}\} \quad \text{if } k<l \\
\llparenthesis \textbf{prev}\,\{M^\circ\} \rrparenthesis^{\gamma_0\cdots\gamma_l}_{@k\le l} &= \boldsymbol{\sim}\{^{\gamma_{k-}} \llparenthesis M^\circ \rrparenthesis^{\gamma_0\cdots\gamma_l}_{@k-\le l}\} \quad \text{if } k>0 \\
(\!(N^\circ)\!)^{\gamma_0\cdots\gamma_l}_{@k\le l} &= \lambda\boldsymbol{\delta_{k+}}\ge\gamma_{k+}\ldots\lambda\boldsymbol{\delta_l}\ge\gamma_l. \llparenthesis N^\circ \rrparenthesis^{\gamma_0\ldots\gamma_k,\delta_{k+}\ldots\delta_l}_{@k\le l}
\end{aligned}$$

Context $\Gamma^\circ @k \rightsquigarrow_{\le l} \Delta/\gamma_0\ldots\gamma_l$

$$\rightsquigarrow\text{-}\epsilon\ \frac{}{\epsilon@0 \rightsquigarrow_{\le l} \epsilon/\overrightarrow{!}} \qquad \rightsquigarrow\text{-🔒}\ \frac{\Gamma^\circ@k \rightsquigarrow_{\le l} \Delta/\overrightarrow{\gamma} \qquad k>0}{\Gamma^\circ@k- \rightsquigarrow_{\le l} \Delta, \text{🔒}^{\gamma_{k-}}/\overrightarrow{\gamma}} \qquad \rightsquigarrow\text{-V}\ \frac{\Gamma^\circ@k \rightsquigarrow_{\le l} \Delta/\gamma_0\ldots\gamma_l}{\Gamma^\circ, x:^k A^\circ@k \rightsquigarrow_{\le l} \Delta, \boldsymbol{x}:^{\boldsymbol{\delta}} \llparenthesis A^\circ \rrparenthesis^{\gamma_{k+}\cdots\gamma_l}_{@k\le l}/(\gamma_0\ldots\gamma_l)[k\mapsto\delta]}$$

$$\rightsquigarrow\text{-🔓}\ \frac{\Gamma^\circ@k \rightsquigarrow_{\le l} \Delta/\gamma_0\ldots\gamma_l \qquad k<l}{\Gamma^\circ@k+ \rightsquigarrow_{\le l} \Delta, \text{🔓}^{\boldsymbol{\delta}:\succeq\gamma_{k+}}/(\gamma_0\ldots\gamma_l)[k+\mapsto\delta]} \qquad \rightsquigarrow\text{-M}\ \frac{\Gamma^\circ@k \rightsquigarrow_{\le l} \Delta/\gamma_0\ldots\gamma_l}{\Gamma^\circ@k \rightsquigarrow_{\le l} \Delta, \boldsymbol{\delta_{k+}}:\succeq\gamma_{k+}\ldots\boldsymbol{\delta_l}:\succeq\gamma_l/\gamma_0\ldots\gamma_k,\delta_{k+}\ldots\delta_l}$$

Fig. 5. Translation from $\lambda_{\bigcirc}$ to BMTT (classifiers named δ or δ_k are taken freshly)

The basic idea of the translation is that annotating $\lambda_{\bigcirc}$ objects with classifiers based on a given mapping. The translation for bounded modal types, quotes and unquotes annotate appropriate classifiers using the mapping.

However, this approach alone is not sufficient to define a sound translation. For example, consider the following $\lambda_{\bigcirc}$ term.

$$\lambda x^{\bigcirc A^\circ}. \textbf{next}\,\{\lambda y^{B^\circ}. \textbf{prev}\,\{x\}\}$$

When translating the type $\bigcirc A^\circ$, the translation infers its classifier from the position of the next stage. However, such classifiers are different between when x is defined and when x is used, leading to inconsistency in the translation results. That is why we introduce polymorphic classifiers to address this gap, as shown in the definitions of $(\!(A^\circ)\!)^{\gamma_{k+}\cdots\gamma_l}_{@k\le l}$, $(\!(M^\circ)\!)^{\gamma_0\cdots\gamma_l}_{@k\le l}$ and $\rightsquigarrow$-M. These polymorphic classifiers are instantiated when a variable is used, applying classifiers at the point. In this sense, polymorphic classifiers play an essential role in this translation. We can confirm that it preserves typing.

Theorem 6. *If $\Gamma^\circ \vdash_k M^\circ : A^\circ$ holds in $\lambda_{\bigcirc}$, $\Gamma^\circ@k \rightsquigarrow_{\le l} \Delta/\gamma_0\ldots\gamma_l$ holds and l is sufficiently large, then $\Delta \vdash \llparenthesis M^\circ \rrparenthesis^{\gamma_0\cdots\gamma_l}_{@k\le l} : \llparenthesis A^\circ \rrparenthesis^{\gamma_{k+1}\cdots\gamma_l}_{@k\le l}$ holds in BMTT.*

In the rest of this section, we assume that l is sufficiently large to ensure that translation is defined. We further demonstrate that this translation is preserves reduction to some extent. We define a forgetful translation from BMTT to $\lambda_{\bigcirc}$.

$$\begin{aligned}
& & |x| &= x \\
|p| &= p & |\lambda\boldsymbol{x}:{}^{\gamma}A.\,M| &= \lambda x^{|A|}.\,|M| \\
|A\to B| &= |A|\to|B| & |M\ N| &= |M|\ |N| \\
|[^{\succeq\gamma}A]| &= \bigcirc|A| & |\text{‘}\{^{\gamma_1:\succeq\gamma_2}M\}| &= \textbf{next}\,\{|M|\} \\
& & |\boldsymbol{\sim}\{^{\gamma}M\}| &= \textbf{prev}\,\{|M|\}
\end{aligned}$$

We can confirm that the original $\lambda_{\bigcirc}$ term is restored by forgetting classifiers annotated by the translation.

Lemma 8. *If $\Gamma^\circ \vdash_k M^\circ : A^\circ$ and $\Gamma^\circ@k \rightsquigarrow_{\le l} \Delta/\gamma_0\ldots\gamma_l$, then $|\llparenthesis M^\circ \rrparenthesis^{\gamma_0\cdots\gamma_l}_{@k\le l}| = M^\circ$ and $|\llparenthesis A^\circ \rrparenthesis^{\gamma_{k+}\cdots\gamma_l}_{@k\le l}| = A^\circ$.*

We define *reduction ignoring classifiers*, denoted as $\Rrightarrow^*_\beta$, for BMTT terms. Then we prove that our translation is faithful to this reduction.

Definition 2. We write $M_1 \Rrightarrow^*_\beta M_2$ iff there exists M_3 such that $M_1@\gamma \Rightarrow^*_\beta M_3$ and $|M_3| = |M_2|$ for any γ.

Theorem 7. *Suppose $\Gamma^\circ \vdash_k {M^\circ}_1 : A^\circ$ and $\Gamma^\circ@k \rightsquigarrow_{\le l} \Delta/\overrightarrow{\gamma}$ hold. If ${M^\circ}_1 \Rightarrow_\beta {M^\circ}_2$, then $\llparenthesis {M^\circ}_1 \rrparenthesis^{\overrightarrow{\gamma}}_{@k\le l} \Rrightarrow^*_\beta \llparenthesis {M^\circ}_2 \rrparenthesis^{\gamma_0\cdots\gamma_l}_{@k\le l}$.*

Through these properties on the two translation $\llparenthesis\cdot\rrparenthesis$ and $|\cdot|$, we can regard BMTT as a granular calculus than $\lambda_{\bigcirc}$. In other words, BMTT can type strictly a larger number of staged programs than $\lambda_{\bigcirc}$ by annotating classifiers. Therefore, results in this section support the capability of BMTT for MetaML-style MSP along with the discussions in Section IV.

VI. The Corresponding Logic and Its Kripke Semantics

From the viewpoint of the Curry–Howard correspondence, BMTT can be regarded as a natural-deduction proof system for a kind of modal logic over intuitionistic logic. In this section, we consider this logic and provide a sound and complete Kripke semantics. In addition, we give a characterization in BMTT of modal logic **S4**, which is the logical background of the translation from **S4** modal λ-calculus mentioned in Section V.

Hereafter, we omit terms from judgments unless necessary, and types may also be referred to as *formula*.

Definition 3. The *logic* **BMTT** is defined to be $\{A \mid \varepsilon \vdash A\}$, the set of all formulae that can be proved without assumptions.

A. Logical Preliminaries

Since de Morgan's laws are not necessarily assumed in the intuitionistic setting, various counterparts for a single modal logic from the classical setting can be considered, depending on the axioms adopted. There are currently two main streams: *constructive modal logic* (e.g., [29], [30]) and *intuitionistic modal logic* (e.g., [31]–[33]). Several studies (e.g., [30], [34], [35]) have shown that constructive modal logics are more suitable for applications in computer science. For detailed background and examples, see, e.g., Mendler and Scheele [36].

Birelational Kripke models have been widely studied as a semantic foundation for both constructive and intuitionistic modal logics. Alechina et al. [37] introduced a birelational model for $\mathbf{CS4}$, a *constructive* variant of modal logic $\mathbf{S4}$:

Definition 4 ($\mathbf{CS4}$-model [37])**.** A $\mathbf{CS4}$-*model*[1] is a quadruple $\langle W, \preceq, R, V\rangle$, where

- $\langle W, \preceq\rangle$ is a nonempty preordered set,
- R is a preorder on W with condition:
  - *Left-persistency:* $(R\,;\preceq) \subseteq (\preceq\,; R)$, and
- V assigns each atom p to an upward-closed subset of W.

Here, a model $\langle W, \preceq, V\rangle$ of intuitionistic logic and a model $\langle W, R, V\rangle$ of modal logic are combined together, with the left-persistency condition ensuring the axiom 4 be valid. It is known that several variants of $\mathbf{S4}$ can also be captured through additional constraints on the structure [33], [38].

At first glance, a $\mathbf{CS4}$-model might appear suitable as a model of BMTT, with scope and stage transitions (i.e., $\preceq$ and $\sqsubseteq$) interpreted as $\preceq$ and R, respectively. Unfortunately, this naive interpretation fails to be sound due to the insufficient interaction between the relations. BMTT admits the *stability*[2] condition $(\preceq) \subseteq R$, which reflects the intuition that stage transitions are more restrictive than scope transitions, but is not valid in the class of $\mathbf{CS4}$-models.

Imposing this condition would, in turn, establish the duality between $\Box$ and $\Diamond$ [29], which yet again conflicts with the computational interpretation intended for BMTT; such a rigid connection is difficult to reconcile with a constructive perspective. These issues suggest that birelational models does not provide an intuitive description of the relationship between scopes and stages and is therefore inappropriate for BMTT.

As for intuitionistic modal logics, there is another approach to relational semantics pioneered by Ewald [40]; a *predicate Kripke model* interprets modalities directly in a first-order structure via the *standard translation*. Simpson [33] investigated this semantics extensively, including its application to an intuitionistic variant of $\mathbf{S4}$:

Definition 5 ([33])**.** A *predicate* $\mathbf{IS4}$-*model* is a quintuple $\big\langle W, \preccurlyeq, \{D_w\}_{w\in W}, \{R_w\}_{w\in W}, \{V_w\}_{w\in W}\big\rangle$, where

- $\langle W, \preccurlyeq\rangle$ is a nonempty preordered set;
- Each $\langle D_w, R_w, V_w\rangle$ is an $\mathbf{S4}$-model; and
- If $w \preccurlyeq v$, then $D_w \subseteq D_v$, $R_w \subseteq R_v$, and $V_w(p) \subseteq V_v(p)$ for each atom p.

This approach is not applicable to constructive modal logics, which are incompatible with the standard translation, nor is it suitable for BMTT, as it does not cope with bounded modalities. Nevertheless, the predicate semantics provides an important insight: intuitionistically, domain structure is not known a priori, but is revealed progressively. This leads to our model structure of $\mathbf{BMTT}$.

B. Kripke Models

We now present our model. First, we introduce a $\mathbf{BMTT}$-*structure* based on a birelational model to represent the purely syntactic structure of a program with scopes and stages:

Definition 6 ($\mathbf{BMTT}$-structure)**.** A $\mathbf{BMTT}$-*structure* is a $\mathbf{CS4}$-model $\langle D, \preceq, \sqsubseteq, V\rangle$ with

- a *root* $\mathbf{!} \in D$, the least element with respect to $\preceq$, and
- the *stability* condition $(\preceq) \subseteq (\sqsubseteq)$, or equivalently:
  - *Left-stable:* $(\preceq\,;\sqsubseteq) \subseteq (\sqsubseteq)$, and
  - *Right-stable:* $(\sqsubseteq\,;\preceq) \subseteq (\sqsubseteq)$.

To capture the dynamics of syntactic structures, a $\mathbf{BMTT}$-*model* is defined as a family of $\mathbf{BMTT}$-structures, organized analogously to a predicate model:

Definition 7 ($\mathbf{BMTT}$-model)**.** A $\mathbf{BMTT}$-*model* $\mathfrak{M}$ is a quadruple $\big\langle W, \preccurlyeq, \{M_w\}_{w\in W}, \iota\big\rangle$, where

- $\langle W, \preccurlyeq\rangle$ is a nonempty preordered set;
- Each M_w is a $\mathbf{BMTT}$-structure $\langle D_w, \preceq_w, \sqsubseteq_w, V_w, \mathbf{!}_w\rangle$;
- ι is a family of structure-preserving functions $\iota_{w\preccurlyeq v} : D_w \to D_v$ for each $w \preccurlyeq v$ such that[3]:
  - For each $w \in W$, $\iota_{w\preccurlyeq w} = \mathrm{id}_{D_w}$, and
  - If $w \preccurlyeq v \preccurlyeq u$, then $\iota_{v\preccurlyeq u} \circ \iota_{w\preccurlyeq v} = \iota_{w\preccurlyeq u}$.

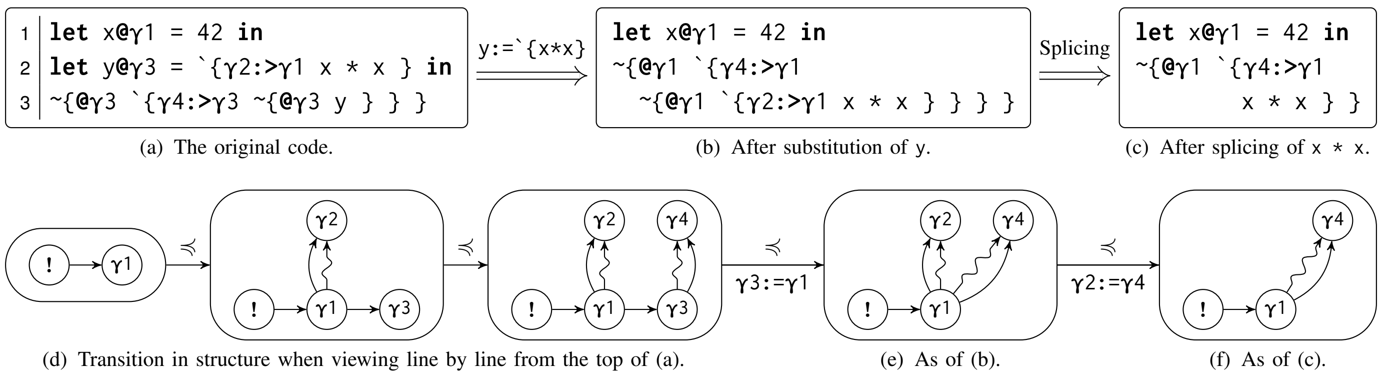
To illustrate the intuition behind the model, consider Figure 6. In the program shown in (a), as the compiler reads through the code from top to bottom, the number of classifiers (i.e., environments) it manages increases due to variable declarations and opening quotes. This process is captured as a sequence of growing $\mathbf{BMTT}$-structures, as depicted in (d). In addition to such a *static* evolution, (meta)program evaluation causes these structures to change *dynamically*; as the program is evaluated to (b) and then to (c), the structure is folded as in (e) and (f), reflecting the resolution of the declared classifiers.

In such a situation, relying on a single birelational model to validate scope-safety is insufficient. For instance, the quoted program `x * x` in (a) is eventually used at $\gamma 4$ in (c), yet $\gamma 4$ is not known to the compiler at the time `x * x` is constructed; that is, $\gamma 4$ is not present in the second $\mathbf{BMTT}$-structure of (d). This demonstrates that ensuring scope-safety requires validating not only for the environments currently known, but also for all environments that may appear during compilation. The $\mathbf{BMTT}$-model addresses this issue by integrating ideas from both birelational and predicate models, providing a semantic foundation for reasoning about scope-safety.

[1]For simplicity, we omit *fallible worlds* from the definition because $\bot$ is not considered in this paper, but our model can be extended to having them as well.

[2]The term "stable" is borrowed from Stell, Schmidt, and Rydeheard [39].

[3]In other words, a $\mathbf{BMTT}$-model is given by a functor from a small nonempty thin category $\langle W, \preccurlyeq\rangle$ to the category consisting of all $\mathbf{BMTT}$-structures with structure-preserving functions. Note that a morphism here is different from *p-morphism* or *bounded morphism*, commonly used notion of Kripke-model homomorphism, and does not need to preserve satisfaction.



```
let x@γ1 = 42 in
let y@γ3 = `{γ2:>γ1 x * x } in
~{@γ3 `{γ4:>γ3 ~{@γ3 y } } }
```

(a) The original code.

$\xRightarrow{\texttt{y:=`\{x*x\}}}$

```
let x@γ1 = 42 in
~{@γ1 `{γ4:>γ1
  ~{@γ1 `{γ2:>γ1 x * x } } } }
```

(b) After substitution of y.

$\xRightarrow{\text{Splicing}}$

```
let x@γ1 = 42 in
~{@γ1 `{γ4:>γ1
        x * x } }
```

(c) After splicing of `x * x`.

(d) Transition in structure when viewing line by line from the top of (a).

(e) As of (b).

(f) As of (c).

Fig. 6. Example of structural transitions represented by a **BMTT**-model, where $\longrightarrow$ and $\rightsquigarrow$ in node indicates scope and stage transitions, respectively.

From a logical perspective, such *variable domains* align with the intuitionistic setting; otherwise, it would result in a *constant-domain* interpretation of quantification. In other words, **BMTT** can be understood as a special kind of *intuitionistic first-order modal logic* in which each domain is also intuitionistic. This allows us to extend Simpson's predicate semantics to cover constructive modal logics, which shall be demonstrated in Section VI-E.

C. Formal Interpretation

A *w-assignment* ρ is a partial map from the set of all classifiers to D_w with $\mathbf{!} \mapsto \mathbf{!}_w$. Given a w-assignment ρ, define v-assignment $\iota_{w \preccurlyeq v}(\rho)$ for $v \succcurlyeq w$ as $\gamma \mapsto \iota_{w \preccurlyeq v}(\rho(\gamma))$ if $\rho(\gamma)$ is defined, and undefined otherwise. For simplicity, we assume that an assignment has sufficient domain of definition for interpretation.

The *satisfaction* of a formula A at $d \in D_w$ with ρ on $w \in W$ in $\mathfrak{M}$, written $w, d \Vdash^\rho A$, is defined as

$$w, d \Vdash^\rho A \qquad \text{iff } \forall v \succcurlyeq w.\ \big(v, \iota(d) \vDash^{\iota(\rho)} A\big),$$

where $w, d \vDash^\rho A$ is defined as follows:

$$\begin{aligned}
&w, d \vDash^\rho p && \text{iff } d \in V_w(p) \\
&w, d \vDash^\rho A \to B && \text{iff } \forall e \succeq_w d. \begin{cases} w, e \Vdash^\rho A \\ \qquad \Longrightarrow\ w, e \Vdash^\rho B \end{cases} \\
&w, d \vDash^\rho [\succeq^\gamma A] && \text{iff } \forall e \sqsupseteq_w d. \begin{cases} e \succeq_w \rho(\gamma) \\ \qquad \Longrightarrow\ w, e \Vdash^\rho A \end{cases} \\
&w, d \vDash^\rho \forall \boldsymbol{\gamma_2} :\succeq \gamma_1.\, A && \text{iff } \forall e \succeq_w \rho(\gamma_1).\ \big(w, d \Vdash^{\rho \cdot [\gamma_2 \mapsto e]} A\big)
\end{aligned}$$

The relation $w, d \vDash^\rho A$ only considers the local structure on w, which is insufficient for the monotonicity requirement regarding $\preccurlyeq$, so that $w, d \Vdash^\rho A$ checks for all $\preccurlyeq$-successors of w to satisfy A.

Unlike satisfaction of a formula, the accesibiity of each transition relation is independent of d and is interpreted as

$$w \Vdash^\rho \gamma_1 \trianglelefteq \gamma_2 \ \text{ iff } \ \rho(\gamma_1) \trianglelefteq \rho(\gamma_2). \qquad (\trianglelefteq \in \{\preceq, \sqsubseteq\})$$

Finally, the interpretation of a context Γ is determined based on its position $\mathrm{pos}(\Gamma)$ as follows:

$$\begin{aligned}
&w \Vdash^\rho \varepsilon && \text{iff always} \\
&w \Vdash^\rho \Gamma, \boldsymbol{x} :^\gamma A && \text{iff } \begin{cases} \bullet\ w \Vdash^\rho \Gamma, \\ \bullet\ w \Vdash^\rho \mathrm{pos}(\Gamma) \preceq \gamma, \text{ and} \\ \bullet\ w, \rho(\gamma) \Vdash^\rho A \end{cases} \\
&w \Vdash^\rho \Gamma, \text{🔓}^{\boldsymbol{\gamma_2} :\succeq \gamma_1} && \text{iff } \begin{cases} \bullet\ w \Vdash^\rho \Gamma, \\ \bullet\ w \Vdash^\rho \mathrm{pos}(\Gamma) \sqsubseteq \gamma_2, \text{ and} \\ \bullet\ w \Vdash^\rho \gamma_1 \preceq \gamma_2 \end{cases} \\
&w \Vdash^\rho \Gamma, \text{🔒}^\gamma && \text{iff } w \Vdash^\rho \Gamma \\
&w \Vdash^\rho \Gamma, \boldsymbol{\gamma_2} :\succeq \gamma_1 && \text{iff } \begin{cases} \bullet\ w \Vdash^\rho \Gamma \text{ and} \\ \bullet\ w \Vdash^\rho \gamma_1 \preceq \gamma_2 \end{cases}
\end{aligned}$$

Notice that 🔒^γ is not placing new assumptions here, but only serves to shift position back to γ. This is actually sufficient because the stage transition $\gamma \sqsubseteq \mathrm{pos}(\Gamma)$ is required in the formation rule WF-🔒 to be a consequence of Γ.

The following lemma is a semantical basis of scope-safety:

Lemma 9 (Monotonicity)**.** *Suppose $w \preccurlyeq v$ and $\iota(d) \preceq_v e$.*

1) *If $w, d \Vdash^\rho A$, then $v, e \Vdash^{\iota(\rho)} A$.*
2) *If $w \Vdash^\rho \gamma_1 \preceq \gamma_2$, then $v \Vdash^{\iota(\rho)} \gamma_1 \preceq \gamma_2$.*
3) *If $w \Vdash^\rho \gamma_1 \sqsubseteq \gamma_2$, then $v \Vdash^{\iota(\rho)} \gamma_1 \sqsubseteq \gamma_2$.*

Proof. (2) and (3) follows from the monotonicity of ι. (1) is shown by induction on A, where the case $A \equiv [\succeq^\gamma B]$ follows from the left-stability of $\sqsubseteq$. ☐

The *semantic consequence* relations are defined accordingly:

$$\begin{aligned}
&\Gamma \Vdash A && \text{iff } \forall \mathfrak{M}, w, \rho. \begin{cases} \mathfrak{M}, w \Vdash^\rho \Gamma \ \Longrightarrow \\ \quad \mathfrak{M}, w, \rho(\mathrm{pos}(\Gamma)) \Vdash^\rho A \end{cases} \\
&\Gamma \Vdash \gamma_1 \trianglelefteq \gamma_2 && \text{iff } \forall \mathfrak{M}, w, \rho. \begin{cases} \mathfrak{M}, w \Vdash^\rho \Gamma \ \Longrightarrow \\ \quad \mathfrak{M}, w \Vdash^\rho \gamma_1 \trianglelefteq \gamma_2 \end{cases}
\end{aligned}$$

where $\trianglelefteq \in \{\preceq, \sqsubseteq\}$. The soundness is now shown:

Theorem 8 (Kripke soundness)**.**

1) *If $\Gamma \vdash A$, then $\Gamma \Vdash A$.*
2) *If $\Gamma \vdash \gamma_1 \preceq \gamma_2$, then $\Gamma \Vdash \gamma_1 \preceq \gamma_2$.*
3) *If $\Gamma \vdash \gamma_1 \sqsubseteq \gamma_2$, then $\Gamma \Vdash \gamma_1 \sqsubseteq \gamma_2$.*

Proof. By induction on derivation, along with Lemma 9. ☐

D. Kripke Completeness

To prove completeness we use a *canonical-model* construction.

In semantics, the truth of a formula is always defined at each point of a model, whereas in syntax, only its validity at $\mathrm{pos}(\Gamma)$ is assertible under Γ; in this respect, BMTT differs from ordinary labeled proof systems (cf. e.g., [33], [41]). The restriction, however, does not pose a problem, as $\Gamma, \text{🔒}^{\mathbf{!}} \vdash [\succeq^{\gamma} A]$ can be used instead to represent the validity of A at γ under Γ. Here, the bounded modality expresses monotonicity, analogous to the **S4** modality in the Gödel–McKinsey–Tarski translation.

Definition 8 (Canonical model)**.** $\mathfrak{M}^{\mathrm{c}}$ is defined as follows:

- W^{c} is the set of all well-formed contexts.
- $\Gamma \preccurlyeq^{\mathrm{c}} \Delta$ iff $\exists \Gamma'. \big(\Delta \equiv \Gamma, \Gamma'\big)$.
- $M^{\mathrm{c}}_{\Gamma} = \langle D^{\mathrm{c}}_{\Gamma}, \preceq^{\mathrm{c}}_{\Gamma}, \sqsubseteq^{\mathrm{c}}_{\Gamma}, V^{\mathrm{c}}_{\Gamma}, \mathbf{!}^{\mathrm{c}}_{\Gamma}\rangle$ where
  - $D^{\mathrm{c}}_{\Gamma} = \mathbf{Dom}_{\mathrm{C}}(\Gamma)$ with $\mathbf{!}^{\mathrm{c}}_{\Gamma} = \mathbf{!}$;
  - $\gamma_1 \preceq^{\mathrm{c}}_{\Gamma} \gamma_2$ iff $\Gamma \vdash \gamma_1 \preceq \gamma_2$;
  - $\gamma_1 \sqsubseteq^{\mathrm{c}}_{\Gamma} \gamma_2$ iff $\Gamma \vdash \gamma_1 \sqsubseteq \gamma_2$;
  - $\gamma \in V^{\mathrm{c}}_{\Gamma}(p)$ iff $\Gamma, \text{🔒}^{\mathbf{!}} \vdash [\succeq^{\gamma} p]$.
- $\iota_{\Gamma \preccurlyeq \Delta} : D^{\mathrm{c}}_{\Gamma} \hookrightarrow D^{\mathrm{c}}_{\Delta}$ is the canonical injection.

Lemma 10. *$\mathfrak{M}^{\mathrm{c}}$ is a* $\mathbf{BMTT}$*-model.*

Proof. Most of the conditions are straightforward. Transitivity and stability are from Lemma 2. $\square$

The canonical model clarifies what inner and outer structures represent: each M_w models the structure formed by *classifiers*, whereas $\mathfrak{M}$ models the structure formed by *contexts*.

The *canonical* Γ*-assignment* $\rho^{\mathrm{c}}_{\Gamma}$ is an assignment that maps each $\gamma \in \mathbf{Dom}_{\mathrm{C}}(\Gamma)$ to itself, and we write $\Gamma, \gamma \Vdash^{\mathrm{c}} A$ if $\mathfrak{M}^{\mathrm{c}}, \Gamma, \gamma \Vdash^{\rho^{\mathrm{c}}_{\Gamma}} A$. The *truth lemma* is stated in the following form:

Lemma 11 (Truth lemma)**.** $\Gamma, \gamma \Vdash^{\mathrm{c}} A$ *iff* $\Gamma, \text{🔒}^{\mathbf{!}} \vdash [\succeq^{\gamma} A]$.

Proof. By induction on the size of A. $\square$

Theorem 9 (Kripke completeness)**.**

1) *If* $\Gamma \Vdash A$*, then* $\Gamma \vdash A$*.*
2) *If* $\Gamma \Vdash \gamma_1 \preceq \gamma_2$*, then* $\Gamma \vdash \gamma_1 \preceq \gamma_2$*.*
3) *If* $\Gamma \Vdash \gamma_1 \sqsubseteq \gamma_2$*, then* $\Gamma \vdash \gamma_1 \sqsubseteq \gamma_2$*.*

Proof. Shown by contrapositive, where we can take the canonical model as countermodel by Lemma 11. $\square$

We remark that the canonical model has just *increasing domains*, and thus $\preccurlyeq^{\mathrm{c}}$ represents only static evolution of the structure, like those in Figure 6(d). The dynamic transition in structure captured by a model shall be discussed in Section VII.

E. Comparison With CS4

Now we present a formal relationship between $\mathbf{BMTT}$ and $\mathbf{CS4}$ to confirm that $\mathbf{BMTT}$ is a generalization of $\mathbf{CS4}$. To align with the syntax of $\mathbf{CS4}$, we restrict $\mathbf{BMTT}$-formulae in this subsection as the following grammar:

$$A, B ::= p \mid A \to B \mid [\succeq^{\mathbf{!}} A].$$

We define $|A|^{\square}$ as the formula in which every $[\succeq^{\mathbf{!}} -]$ in A is replaced with a $\square$, turning it into a $\mathbf{CS4}$-formula.

As already described, the naive interpretation of BMTT in a $\mathbf{CS4}$-model is not sound due to the lack of the stability condition $(\preceq) \subseteq R$. However, the following lemma suggests a correct way of interpretation:

Lemma 12. *Given a* $\mathbf{CS4}$*-model* $M = \langle W, \preceq, R, V\rangle$*. Define* $\sqsubseteq$ *as* $(\preceq ; R)$*. Then* $M_* = \langle W, \preceq, \sqsubseteq, V\rangle$ *is a stable* $\mathbf{CS4}$*-model.*

Proof. The transitivity of $\sqsubseteq$ follows from the left-persistency of R, and the stability follows from the reflexivity of R. $\square$

The *submodel* M^w *of* M *generated by* $w \in W$ is given by restricting W to $\{v \in W \mid v \succeq w\}$, so that w serves as the root of M^w, making $\langle \{*\}, =, \{M^w_*\}, \{\mathrm{id}_W\}\rangle$ a *one-point* $\mathbf{BMTT}$*-model*, which also is referred to as M^w_*.

Lemma 13. $M^w, v \vDash_{\mathbf{CS4}} |A|^{\square}$ *iff* $M^w_*, *, v \Vdash^{[\mathbf{!} \mapsto w]} A$*.*

Proof. By induction on A. $\square$

Conversely, we construct a $\mathbf{CS4}$-model from a $\mathbf{BMTT}$-model by *flattening*:

Definition 9 (Flattening)**.** Let $\mathfrak{M} = \langle W, \preccurlyeq, \{M_w\}_{w \in W}, \iota\rangle$ be a $\mathbf{BMTT}$-model, with each M_w as $\langle D_w, \preceq_w, \sqsubseteq_w, V_w, \mathbf{!}_w\rangle$. Then $\mathfrak{M}_+ = \big\langle W_+, \preceq_+, R_+, V_+\big\rangle$ is defined as follows:

- $W_+ = \sum_{w \in W} D_w$;
- $\langle w, d\rangle \preceq_+ \langle w', d'\rangle$ iff $w \preccurlyeq w'$ & $\iota(d) \preceq_{w'} d'$;
- $\langle w, d\rangle \; R_+ \; \langle w', d'\rangle$ iff $w = w'$ & $d \sqsubseteq_w d'$; and
- $\langle w, d\rangle \in V_+(p)$ iff $d \in V_w(p)$.

Lemma 14. *$\mathfrak{M}_+$ is a* $\mathbf{CS4}$*-model.*

Proof. Left-persistency follows from the right-stability of $\sqsubseteq_w$. Note that stability does not preserved under flattening due to the outer intuitionistic transition $\preccurlyeq$. $\square$

Lemma 15. $\mathfrak{M}, w, d \Vdash^{\rho} A$ *iff* $\mathfrak{M}_+, \langle w, d\rangle \vDash_{\mathbf{CS4}} |A|^{\square}$*.*

Proof. By induction on A. $\square$

These lemmas indicate that the two constructions, namely M^w_* and $\mathfrak{M}_+$, are pseudo-inverse operations that preserve satisfaction, leading to the following characterization:

Theorem 10. *The* $\{\to, \square\}$*-fragment of* $\mathbf{CS4}$ *is isomorphic to the* $\{\to, [\succeq^{\mathbf{!}} -]\}$*-fragment of* $\mathbf{BMTT}$*.*

Proof. Follows from Lemmas 13 and 15, where $|-|^{\square}$ is the isomorphism. $\square$

Flattening offers another perspective on a $\mathbf{BMTT}$-model: $\mathfrak{M}$ "stratifies" the $\mathbf{CS4}$-model $\mathfrak{M}_+$ into the stable components $\{M_w\}_{w \in W}$ by decoupling its dynamic aspects from the object structure. While such stratification is necessary to determine the range of quantification, it does not always exist for every $\mathbf{CS4}$-model, which provides another reason why a $\mathbf{CS4}$-model is not suitable for BMTT.

VII. Metatheory

In this section, we show basic properties of BMTT: strong normalization, confluence, canonicity, and the subformula property. These properties ensure the computational adequacy of BMTT, as already stated in the form of Theorem 4.

A. Strong Normalization

Here, we define reducibility based on the Kripke semantics to demonstrate the dynamic aspects captured by our model, such as those shown in Figures 6(e) and 6(f).

To define reduciblity, we adopt the idea of using continuations from Lindley [42]. A *continuation* K is a term context defined by the following grammar:

$$K ::= [-] \mid K[[-]N] \mid K[\sim\{^{\gamma}\,[-]\}] \mid K[[-]\gamma].$$

Figure 7 defines the predicates $\mathcal{E}[\![A]\!]$ for terms and $\mathcal{K}[\![A]\!]$ for continuations, simultaneously inductively on A. To provide a proper computational interpretation, each predicate is indexed by a context Γ, which is particularly crucial in BMTT because: 1) the reduction is a position-aware relation; and 2) 🔒's cannot be removed from a context by abstraction.

The preorder $\preccurlyeq$ generalizes the canonical relation $\preccurlyeq^{\mathrm{c}}$ from Definition 8 while taking the position $\mathrm{pos}(\Gamma)$ into account to ensure typeability. To accommodate monotonicity, $\Gamma \vdash M \in \mathcal{E}[\![A]\!]$ verifies the SN-ability of M with all reducible continuations K under all $\preccurlyeq$-successor contexts Δ of Γ, whereas $\Gamma \vdash K \in \mathcal{K}[\![A]\!]$ is just validated locally within Γ.

Lemma 16. *If* $\Gamma \vdash M \in \mathcal{E}[\![A]\!]$*, then* $M \in \mathrm{SN}$.

Proof. If $\mathcal{E}$-id is applied, then obvious; otherwise follows from $\Gamma \vdash [-] \in \mathcal{K}[\![A]\!]$ by $\mathcal{K}$-id. □

Next, we consider the interpretation of a context. A *simultaneous substitution* σ is defined by the following grammar:

$$\sigma ::= \emptyset \mid \sigma \cdot [\gamma := \gamma', \boldsymbol{x} := M] \mid \sigma \cdot [\gamma := \gamma'],$$

where the symbol $\emptyset$ denotes an *empty substitution* that does not replace any variable or classifier. Figure 7 gives the definition of a *reducible substitution* $\Delta \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]$, whereby variables and classifiers declared in Γ are instantiated under Δ.

Lemma 17 (Fundamental property)**.** *Suppose* $\Gamma \vdash M : A$*. If* $\Delta \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]$*, then* $\Delta \vdash M\,\sigma \in \mathcal{E}[\![A\,\sigma]\!]$.

Proof. By induction on the derivation of $\Gamma \vdash M : A$. □

Theorem 11 (Strong normalization)**.** *A well-typed term is strongly normalizing with respect to* β*-reduction.*

Proof. Given $\Gamma \vdash M : A$. Taking $\Gamma \vdash \emptyset \in \mathcal{C}[\![\Gamma]\!]$ yields $\Gamma \vdash M \in \mathcal{E}[\![A]\!]$ by Lemma 17; thus $M \in \mathrm{SN}$ by Lemma 16. □

From a logical viewpoint, contexts here form a Kripke model similar to the canonical model $\mathfrak{M}^{\mathrm{c}}$, but with $\preccurlyeq$ instead of $\preccurlyeq^{\mathrm{c}}$ as the outer transition. In this model, we may regard

- $\Gamma \vdash M \in \mathcal{E}[\![A]\!]$ as a witness for $\Gamma \Vdash^{\mathrm{c}} A$, and
- $\Gamma \vdash K \in \mathcal{K}[\![A]\!]$ as a witness for $\Gamma \nVdash^{\mathrm{c}} A$ under certain assumption $\Gamma, \gamma \nVdash^{\mathrm{c}} B$, admitted by $[-]$.

If $\Delta \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]$ is viewed as a function $\iota \colon D^{\mathrm{c}}_{\Gamma} \to D^{\mathrm{c}}_{\Delta}$, another preorder $\preccurlyeq_{\mathcal{C}}$ is induced on W^{c}, which is actually consistent with $\preccurlyeq$ in the sense of Lemma 17. Nevertheless, only $\preccurlyeq$ can be used in the rule $\mathcal{E}$-blur to avoid circular definitions while guaranteeing monotonicity.

B. Other Properties

For the other properties, we only present definitions and statements here. See Appendix E for details.

Theorem 12 (Confluence)**.** *The* β*-reduction is confluent for well-typed terms.*

Corollary 2 (Uniqueness of β-normal form)**.** *A well-typed has a unique* β*-normal form.*

Definition 10.

1) A term is said to be *canonical* if its outermost term-former is for an introduction rule and *neutral* otherwise.
2) A *subformula* of a formula is a literal subexpression with some classifier maybe renamed.

Theorem 13 (Canonicity)**.** *If a term is well-typed, closed regarding term variable, and* β*-normal, then it is canonical.*

Theorem 14 (Subformula property)**.** *Suppose* $\Gamma \vdash M : A$*. If* M *is* β*-normal, then any subterm of* M *satisfies at least one of the following:*

a) *Its type is a subformula of* A*;*
b) *Its type is a subformula of* B *for some* $\boldsymbol{x} :^{\gamma} B \in \Gamma$.

VIII. Related Work

The idea of classifiers is inspired by `<NJ>` by Kiselyov et al. [23]. `<NJ>` use *refined environment classifiers* to achieve scope-safe code generation with run-time evaluation and effects. At the same time, `<NJ>` was formulated as a two-level calculus with code combinators, lacking capability of cross-stage persistence; hence, it does not serve directly as a theoretical foundation for MetaML-style MSP, nor is that the goal of `<NJ>`. They also mentioned the idea of polymorphic classifiers, but did not formalize it.

As the name of `<NJ>` suggests, they were conscious of its logical correspondence. Yamasaki and Kiselyov highlighted the connection between `<NJ>` and *hybrid logic*, especially that between a code type and a *satisfaction statement* $@_a A$ [43]. This statement means that A holds at the world represented by the *nominal* a [44]. They interpret nominals as available resources. We can find the similar idea in *hybrid logic framework* by Reed [45]. While they are similar to bounded modal types, satisfaction statements refer to absolute property and cannot represent nested code types. Hence, we think hybrid logic is not adequate for a foundation for MetaML-style MSP.

The judgment of BMTT has elements of Fitch-style modal calculi [7], [11], [25], [26], labelled deduction [12], [33], [41], [46], [47] and nested sequents [48]–[51]. From the viewpoint of labelled deductive systems, BMTT manages both intuitionistic and modal transitions via labels, similar to the proof system by Marin et al. [47]. Meanwhile, modal transitions are almost entirely (except WF-🔒) implicitly managed by 🔓 and 🔒. These symbols are somewhat similar to nested sequents, although 🔓 and 🔒 do not necessarily need to appear as a pair in BMTT.

Xie et al. classified CSP into three kinds [22] and CSP in our paper is actually *heap-based CSP*. BER MetaOCaml supports heap-based CSP, but its type system gives up fully

$$\mathcal{E}\text{-blur}\ \frac{\forall\Delta \succcurlyeq \Gamma.\,\big(\Delta \vdash K \in \mathcal{K}[\![A]\!] \implies K[M] \in \mathrm{SN}\big)}{\Gamma \vdash M \in \mathcal{E}[\![A]\!]} \text{ where } \Gamma \preccurlyeq \Delta \text{ iff } \begin{cases} \bullet\ \Gamma \vdash \gamma_1 \trianglelefteq \gamma_2 \implies \Delta \vdash \gamma_1 \trianglelefteq \gamma_2; \\ \bullet\ \boldsymbol{x} :^{\gamma} A \in \Gamma \implies \boldsymbol{x} :^{\gamma} A \in \Delta; \text{ and} \\ \bullet\ \Delta \vdash \mathrm{pos}(\Gamma) \preceq \mathrm{pos}(\Delta) \end{cases}$$

$$\mathcal{E}\text{-id}\ \frac{\Gamma \vdash x : A}{\Gamma \vdash x \in \mathcal{E}[\![A]\!]} \qquad \mathcal{K}\text{-id}\ \frac{}{\Gamma \vdash [-] \in \mathcal{K}[\![A]\!]} \qquad \mathcal{K}\text{-}{\to}\ \frac{\Gamma \vdash N \in \mathcal{E}[\![A]\!] \qquad \Gamma \vdash K \in \mathcal{K}[\![B]\!]}{\Gamma \vdash K[[-]N] \in \mathcal{K}[\![A \to B]\!]}$$

$$\mathcal{K}\text{-}[]\ \frac{\Gamma \vdash \gamma \sqsubseteq \mathrm{pos}(\Gamma) \qquad \Gamma \vdash \gamma_1 \preceq \mathrm{pos}(\Gamma) \qquad \Gamma \vdash K \in \mathcal{K}[\![A]\!]}{\Gamma, \text{🔒}^{\gamma} \vdash K[\sim\{^{\gamma}[-]\}] \in \mathcal{K}[\![[^{\succeq\gamma_1} A]]\!]} \qquad \mathcal{K}\text{-}\forall\ \frac{\Gamma \vdash \gamma_1 \preceq \gamma \qquad \Gamma \vdash K \in \mathcal{K}[\![A[\boldsymbol{\gamma_2} \coloneqq \gamma]]\!]}{\Gamma \vdash K[[-]\gamma] \in \mathcal{K}[\![\forall \boldsymbol{\gamma_2} :\succeq \gamma_1.\, A]\!]}$$

$$\mathcal{C}\text{-id}\ \frac{}{\Gamma \vdash \emptyset \in \mathcal{C}[\![\Gamma]\!]} \qquad \mathcal{C}\text{-weak}\ \frac{\Delta \vdash \sigma \in \mathcal{C}[\![\Gamma]\!] \qquad \Delta \preccurlyeq \Delta'}{\Delta' \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]} \qquad \mathcal{C}\text{-}{\to}\ \frac{\Delta \vdash \sigma \in \mathcal{C}[\![\Gamma]\!] \qquad \Delta \vdash M \in \mathcal{E}[\![A\,\sigma]\!]}{\Delta \vdash \sigma \cdot [\boldsymbol{\gamma} \coloneqq \mathrm{pos}(\Delta), \boldsymbol{x} \coloneqq M] \in \mathcal{C}[\![\Gamma, \boldsymbol{x} :^{\gamma} A]\!]}$$

$$\mathcal{C}\text{-}[]\ \frac{\Delta, \text{🔒}^{\gamma} \vdash \sigma \in \mathcal{C}[\![\Gamma]\!] \qquad \Delta \vdash \gamma_1\,\sigma \preceq \mathrm{pos}(\Delta)}{\Delta \vdash \sigma \cdot [\boldsymbol{\gamma_2} \coloneqq \mathrm{pos}(\Delta)] \in \mathcal{C}[\![\Gamma, \text{🔓}^{\boldsymbol{\gamma_2} :\succeq \gamma_1}]\!]} \qquad \mathcal{C}\text{-}\forall\ \frac{\Delta \vdash \sigma \in \mathcal{C}[\![\Gamma]\!] \qquad \Delta \vdash \gamma_1\,\sigma \preceq \gamma}{\Delta \vdash \sigma \cdot [\boldsymbol{\gamma_2} \coloneqq \gamma] \in \mathcal{C}[\![\Gamma, \boldsymbol{\gamma_2} :\succeq \gamma_1]\!]} \qquad \mathcal{C}\text{-🔒}\ \frac{\Delta \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]}{\Delta, \text{🔒}^{\gamma\,\sigma} \vdash \sigma \in \mathcal{C}[\![\Gamma, \text{🔒}^{\gamma}]\!]}$$

Fig. 7. Reducibility. The predicates $\mathcal{E}[\![A]\!]$, $\mathcal{K}[\![A]\!]$, and $\mathcal{C}[\![\Gamma]\!]$ are for terms, for continuations, and for substitutions (contexts), respectively.

static detection of scoping errors [15]. To our knowledge, only $\lambda^{\triangleright\%}$ by Hanada and Igarashi provides sound type system with support of heap-based CSP [21]. However, it is an ad-hoc extension of $\lambda^{\triangleright}$ and its logical counterpart is not clear.

Last but not least, we compare CMTT. CMTT was first proposed by Nanevski et al. [10], annotateing a modal type with a *context* Γ instead of classifiers. Furthermore, Murase et al. proposed $\lambda^{\forall[]}$ by extending CMTT with *polymorphic contexts*, which plays a similar role as polymorphic classifiers in BMTT [11]. Contextual modal types and bounded modal types are alternative approaches to representing visible resources, each with its own pros and cons.

In the context of MSP, we think BMTT has two major advantages over CMTT. Firstly, quotes in CMTT need to bind all free variables in a code fragment: for example, a code fragment `` `{x+y} `` is represented as `` `{x,y. x+y} ``, binding `x` and `y`. Hence, $\lambda^{\forall[]}$ cannot directly encode a staged program like **`fun`** `` x => ~{f `{ x }} ``, in which the function binds a free variable within the quote. Instead, $\lambda^{\forall[]}$ annotates *explicit substitutions* in unquotes. In the $\lambda^{\forall[]}$-style program **`fun`** `` x => { ~{f `{x'. x'}}[x] } ``, the explicit substitution `[x]` annotates the unquote, stating that the free variable in a spliced code fragment will be replaced with `x`. However, the binding relation between the definition of `x` and `x'` in the quote is no longer clear until this unquote is reduced. This encoding of bindings also occurs in the example in I, and that is why we stated that CMTT cannot represent CSP directly. BMTT does not experience this issue as it represent open code fragments as open terms.

Secondly, polymorphic contexts in $\lambda^{\forall[]}$ is impredicative while polymorphic classifiers in BMTT are first-order. Therefore, BMTT would be easier to reason about from both theoretical and practical perspectives.

It should be also noted that we can combine bounded modal types and contextual modal types, like $[A \vdash B]^{\gamma}$, taking benefits from both types. BMTT and CMTT are complementary rather than conflicting paradigms.

IX. Conclusion and Future Work

Motivated by multi-stage programming, we introduced and studied BMTT, a modal type theory incorporating classifiers into **S4** modal types. We have demonstrated its potential for application to multi-stage programming, including embedding $\lambda_{\bigcirc}$ to BMTT. We have also defined Kripke semantics for its logical counterpart and investigated metatheory about its syntax and semantics. Although this paper emphasizes its MSP aspects, our development of Kripke semantics offers more granular models than existing proposals, and we expect it to provide valuable insights into constructive logic in general.

We are interested in whether we can incorporate classifiers and bounded modal types into other typing disciplines, including polymorphic types [52], dependent types [53]–[55], and linear types [56]. In particular, intensional analysis [17], [52], [57]–[59] would be both an interesting and challenging topic because quotes in BMTT are open by its inherent nature while Kavvos pointed out that closedness is required for intensional analysis [60]. We are also curious whether our methods can be applied to modalities other than **S4** by possibly incorporating classifiers into dual-context calculi [7], [61] or multimodal type theory [55].

References

[1] J.-Y. Girard, P. Taylor, and Y. Lafont, *Proofs and Types*, ser. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1989.

[2] M. H. Sørensen and P. Urzyczyn, *Lectures on the Curry–Howard Isomorphism*, ser. Studies in Logic and the Foundations of Mathematics. Elsevier, 2006, vol. 149.

[3] E. Moggi, "Notions of computation and monads," *Inf. Comput.*, vol. 93, no. 1, pp. 55–92, 1991. [Online]. Available: https://doi.org/10.1016/0890-5401(91)90052-4

[4] H. Nakano, "A modality for recursion," in *15th Annual IEEE Symposium on Logic in Computer Science, Santa Barbara, California, USA, June 26-29, 2000*. IEEE Computer Society, 2000, pp. 255–266. [Online]. Available: https://doi.org/10.1109/LICS.2000.855774

[5] A. Guatto, "A generalized modality for recursion," in *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, ser. LICS '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 482–491. [Online]. Available: https://doi.org/10.1145/3209108.3209148

[6] D. Gratzer, "A modal deconstruction of Löb induction," *Proc. ACM Program. Lang.*, vol. 9, no. POPL, Jan. 2025. [Online]. Available: https://doi.org/10.1145/3704866

[7] R. Davies and F. Pfenning, "A modal analysis of staged computation," *J. ACM*, vol. 48, no. 3, p. 555–604, May 2001.

[8] R. Davies, "A temporal logic approach to binding-time analysis," *Journal of the ACM*, vol. 64, no. 1, pp. 1:1–1:45, Mar. 2017. [Online]. Available: https://dl.acm.org/doi/10.1145/3011069

[9] T. Tsukada and A. Igarashi, "A logical foundation for environment classifiers," *Log. Methods Comput. Sci.*, vol. 6, no. 4, 2010. [Online]. Available: https://doi.org/10.2168/LMCS-6(4:8)2010

[10] A. Nanevski, F. Pfenning, and B. Pientka, "Contextual modal type theory," *ACM Trans. Comput. Log.*, vol. 9, no. 3, pp. 23:1–23:49, 2008. [Online]. Available: https://doi.org/10.1145/1352582.1352591

[11] Y. Murase, Y. Nishiwaki, and A. Igarashi, "Contextual modal type theory with polymorphic contexts," in *Programming Languages and Systems (ESOP 2023)*, T. Wies, Ed. Cham: Springer Nature Switzerland, 2023, pp. 281–308.

[12] T. M. VII, K. Crary, R. Harper, and F. Pfenning, "A symmetric modal lambda calculus for distributed computing," in *19th IEEE Symposium on Logic in Computer Science (LICS 2004), 14-17 July 2004, Turku, Finland, Proceedings*. IEEE Computer Society, 2004, pp. 286–295. [Online]. Available: https://doi.org/10.1109/LICS.2004.1319623

[13] W. Taha, "A sound reduction semantics for untyped CBN multi-stage computation. or, the theory of metaml is non-trivial (extended abstract)," in *Proceedings of the 2000 ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation (PEPM '00), Boston, Massachusetts, USA, January 22-23, 2000*, J. L. Lawall, Ed. ACM, 2000, pp. 34–43. [Online]. Available: https://doi.org/10.1145/328690.328697

[14] W. Taha and T. Sheard, "MetaML and multi-stage programming with explicit annotations," *Theor. Comput. Sci.*, vol. 248, no. 1-2, pp. 211–242, 2000. [Online]. Available: https://doi.org/10.1016/S0304-3975(00)00053-0

[15] O. Kiselyov, "The design and implementation of BER MetaOCaml - system description," in *Functional and Logic Programming - 12th International Symposium, FLOPS 2014, Kanazawa, Japan, June 4-6, 2014. Proceedings*, ser. Lecture Notes in Computer Science, M. Codish and E. Sumii, Eds., vol. 8475. Springer, 2014, pp. 86–102. [Online]. Available: https://doi.org/10.1007/978-3-319-07151-0_6

[16] ——, "Reconciling abstraction with high performance: A MetaOCaml approach," *Foundations and Trends in Programming Languages*, vol. 5, no. 1, pp. 1–101, 2018. [Online]. Available: http://dx.doi.org/10.1561/2500000038

[17] N. Stucki, J. I. Brachthäuser, and M. Odersky, "Multi-stage programming with generative and analytical macros," in *GPCE '21: Concepts and Experiences, Chicago, IL, USA, October 17 - 18, 2021*, E. Tilevich and C. D. Roover, Eds. ACM, 2021, pp. 110–122. [Online]. Available: https://doi.org/10.1145/3486609.3487203

[18] N. Xie, M. Pickering, A. Löh, N. Wu, J. Yallop, and M. Wang, "Staging with class: a specification for typed template Haskell," *Proc. ACM Program. Lang.*, vol. 6, no. POPL, pp. 1–30, 2022. [Online]. Available: https://doi.org/10.1145/3498723

[19] Y. Yuse and A. Igarashi, "A modal type system for multi-level generating extensions with persistent code," in *Proceedings of the 8th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming, July 10-12, 2006, Venice, Italy*, A. Bossi and M. J. Maher, Eds. ACM, 2006, pp. 201–212. [Online]. Available: https://doi.org/10.1145/1140335.1140360

[20] W. Taha and M. F. Nielsen, "Environment classifiers," in *Conference Record of POPL 2003: The 30th SIGPLAN-SIGACT Symposium on Principles of Programming Languages, New Orleans, Louisisana, USA, January 15-17, 2003*, A. Aiken and G. Morrisett, Eds. ACM, 2003, pp. 26–37. [Online]. Available: https://doi.org/10.1145/604131.604134

[21] Y. Hanada and A. Igarashi, "On Cross-Stage Persistence in Multi-Stage Programming," in *Functional and Logic Programming*, ser. Lecture Notes in Computer Science, M. Codish and E. Sumii, Eds. Cham: Springer International Publishing, 2014, pp. 103–118.

[22] N. Xie, L. White, O. Nicole, and J. Yallop, "MacoCaml: Staging composable and compilable macros," *Proc. ACM Program. Lang.*, vol. 7, no. ICFP, pp. 604–648, 2023. [Online]. Available: https://doi.org/10.1145/3607851

[23] O. Kiselyov, Y. Kameyama, and Y. Sudo, "Refined environment classifiers - type- and scope-safe code generation with mutable cells," in *Programming Languages and Systems - 14th Asian Symposium, APLAS 2016, Hanoi, Vietnam, November 21-23, 2016, Proceedings*, ser. Lecture Notes in Computer Science, A. Igarashi, Ed., vol. 10017, 2016, pp. 271–291. [Online]. Available: https://doi.org/10.1007/978-3-319-47958-3_15

[24] F. Pfenning and R. Davies, "A judgmental reconstruction of modal logic," *Math. Struct. Comput. Sci.*, vol. 11, no. 4, pp. 511–540, 2001. [Online]. Available: https://doi.org/10.1017/S0960129501003322

[25] S. Martini and A. Masini, *A Computational Interpretation of Modal Proofs*. Dordrecht: Springer Netherlands, 1996, pp. 213–241. [Online]. Available: https://doi.org/10.1007/978-94-017-2798-3_12

[26] R. Clouston, "Fitch-style modal lambda calculi," in *Proc. of Foundations of Software Science and Computation Structures*, C. Baier and U. Dal Lago, Eds. Cham: Springer International Publishing, 2018, pp. 258–275.

[27] N. Valliappan, F. Ruch, and C. Tomé Cortiñas, "Normalization for Fitch-style modal calculi," *Proc. ACM Program. Lang.*, vol. 6, no. ICFP, Aug. 2022. [Online]. Available: https://doi.org/10.1145/3547649

[28] R. Davies, "A temporal-logic approach to binding-time analysis," in *Proceedings 11th Annual IEEE Symposium on Logic in Computer Science*, Jul. 1996, pp. 184–195, iSSN: 1043-6871. [Online]. Available: https://ieeexplore.ieee.org/document/561317

[29] K. Kojima, "Semantical study of intuitionistic modal logics," Ph.D. dissertation, Kyoto University, 2012. [Online]. Available: http://hdl.handle.net/2433/157473

[30] D. Wijesekera, "Constructive modal logics I," *Annals of Pure and Applied Logic*, vol. 50, no. 3, pp. 271–301, 1990. [Online]. Available: https://10.1016/0168-0072(90)90059-B

[31] G. F. Servi, "Axiomatizations for some intuitionistic modal logics," *Rendiconti del Seminario Matematico Università e Politecnico di Torino*, vol. 42, no. 3, pp. 179–194, 1984.

[32] G. Plotkin and C. Stirling, "A framework for intuitionistic modal logics," in *Proceedings of the 1st Conference on Theoretical Aspects of Reasoning about Knowledge (TARK)*, 1986, pp. 399–406. [Online]. Available: https://dl.acm.org/doi/10.5555/1029786.1029823

[33] A. K. Simpson, "The proof theory and semantics of intuitionistic modal logic," Ph.D. dissertation, University of Edinburgh. College of Science and Engineering. School of Informatics, 1994. [Online]. Available: http://hdl.handle.net/1842/407

[34] S. Kobayashi, "Monad as modality," *Theoretical Computer Science*, vol. 175, no. 1, pp. 29–74, 1997. [Online]. Available: https://doi.org/10.1016/S0304-3975(96)00169-7

[35] A. Nanevski, "From dynamic binding to state via modal possibility," in *Proceedings of the 5th ACM SIGPLAN international conference on Principles and practice of declaritive programming*, 2003, pp. 207–218. [Online]. Available: https://doi.org/10.1145/888251.888271

[36] M. Mendler and S. Scheele, "Cut-free Gentzen calculus for multimodal $\mathbf{CK}$," *Information and Computation*, vol. 209, no. 12, pp. 1465–1490, 2011. [Online]. Available: https://doi.org/10.1016/j.ic.2011.10.003

[37] N. Alechina, M. Mendler, V. de Paiva, and E. Ritter, "Categorical and Kripke semantics for constructive $\mathbf{S4}$ modal logic," in *International Workshop on Computer Science Logic*. Springer, 2001, pp. 292–307. [Online]. Available: https://doi.org/10.1007/3-540-44802-0_21

[38] P. Balbiani, M. Diéguez, D. Fernández-Duque, and B. McLean, "Constructive $\mathbf{S4}$ modal logics with the finite birelational frame property," 2024. [Online]. Available: https://arxiv.org/abs/2403.00201

[39] J. G. Stell, R. A. Schmidt, and D. Rydeheard, "A bi-intuitionistic modal logic: Foundations and automation," *Journal of Logical and Algebraic Methods in Programming*, vol. 85, no. 4, pp. 500–519, 2016.

[40] W. B. Ewald, "Intuitionistic tense and modal logic," *The Journal of Symbolic Logic*, vol. 51, no. 1, pp. 166–179, 1986. [Online]. Available: https://doi.org/10.2307/2273953

[41] S. Negri, "Proof analysis in modal logic," *Journal of Philosophical Logic*, vol. 34, no. 5, pp. 507–544, 2005. [Online]. Available: https://doi.org/10.1007/s10992-005-2267-3

[42] S. Lindley, "Extensional rewriting with sums," in *International Conference on Typed Lambda Calculi and Applications*. Springer, 2007, pp. 255–271. [Online]. Available: https://doi.org/10.1007/978-3-540-73228-0_19

[43] S. Yamasaki and O. Kiselyov, "Code generating calculus and hybrid logic (in Japanese)," in *Informal Proceedings of the 20th JSSST Workshop on Programming and Programming languages*, 2018.

[44] T. Braüner, *Proof-Theory of Propositional Hybrid Logic*. Dordrecht: Springer Netherlands, 2011, pp. 21–57. [Online]. Available: https://doi.org/10.1007/978-94-007-0002-4_2

[45] J. Reed, "Hybridizing a logical framework," in *Proceedings of the International Workshop on Hybrid Logic, HyLo@FLoC 2006, Seattle, WA, USA, August 11, 2006*, ser. Electronic Notes in Theoretical Computer Science, P. Blackburn, T. Bolander, T. Braüner, V. de Paiva, and

J. Villadsen, Eds., vol. 174, no. 6. Elsevier, 2006, pp. 135–148. [Online]. Available: https://doi.org/10.1016/j.entcs.2006.11.030

[46] J. Reed and F. Pfenning, "Intuitionistic letcc via labelled deduction," *Electronic Notes in Theoretical Computer Science*, vol. 231, pp. 91–111, 2009, proceedings of the 5th Workshop on Methods for Modalities (M4M5 2007). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S157106610900036X

[47] S. Marin, M. Morales, and L. Straßburger, "A fully labelled proof system for intuitionistic modal logics," *Journal of Logic and Computation*, vol. 31, no. 3, pp. 998–1022, 05 2021. [Online]. Available: https://doi.org/10.1093/logcom/exab020

[48] R. A. Bull, "Cut elimination for propositional dynamic logic without *," *Math. Log. Q.*, vol. 38, no. 1, pp. 85–100, 1992. [Online]. Available: https://doi.org/10.1002/malq.19920380107

[49] R. Kashima, "Cut-free sequent calculi for some tense logics," *Stud Logica*, vol. 53, no. 1, pp. 119–136, 1994. [Online]. Available: https://doi.org/10.1007/BF01053026

[50] L. Straßburger, "Cut elimination in nested sequents for intuitionistic modal logics," in *Foundations of Software Science and Computation Structures*, F. Pfenning, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 209–224.

[51] R. Arisaka, A. Das, and L. Straßburger, "On nested sequents for constructive modal logics," *Log. Methods Comput. Sci.*, vol. 11, no. 3, 2015. [Online]. Available: https://doi.org/10.2168/LMCS-11(3:7)2015

[52] J. Jang, S. Gélineau, S. Monnier, and B. Pientka, "Mœbius: Metaprogramming using contextual types: The stage where System F can pattern match on itself," *Proc. ACM Program. Lang.*, vol. 6, no. POPL, Jan. 2022.

[53] D. Gratzer, J. Sterling, and L. Birkedal, "Implementing a modal dependent type theory," *Proc. ACM Program. Lang.*, vol. 3, no. ICFP, pp. 107:1–107:29, 2019. [Online]. Available: https://doi.org/10.1145/3341711

[54] A. Kawata and A. Igarashi, "A dependently typed multi-stage calculus," in *Programming Languages and Systems - 17th Asian Symposium, APLAS 2019, Nusa Dua, Bali, Indonesia, December 1-4, 2019, Proceedings*, ser. Lecture Notes in Computer Science, A. W. Lin, Ed., vol. 11893. Springer, 2019, pp. 53–72. [Online]. Available: https://doi.org/10.1007/978-3-030-34175-6_4

[55] D. Gratzer, G. A. Kavvos, A. Nuyts, and L. Birkedal, "Multimodal dependent type theory," *Log. Methods Comput. Sci.*, vol. 17, no. 3, 2021. [Online]. Available: https://doi.org/10.46298/lmcs-17(3:11)2021

[56] Y. Fukuda and A. Yoshimizu, "A linear-logical reconstruction of intuitionistic modal logic S4," in *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24-30, 2019, Dortmund, Germany*, ser. LIPIcs, H. Geuvers, Ed., vol. 131. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, pp. 20:1–20:24. [Online]. Available: https://doi.org/10.4230/LIPIcs.FSCD.2019.20

[57] L. Parreaux, A. Voizard, A. Shaikhha, and C. E. Koch, "Unifying analytic and statically-typed quasiquotes," *Proc. ACM Program. Lang.*, vol. 2, no. POPL, pp. 13:1–13:33, 2018. [Online]. Available: https://doi.org/10.1145/3158101

[58] L. Chen and H. Ko, "Realising intensional S4 and GL modalities," in *30th EACSL Annual Conference on Computer Science Logic, CSL 2022, February 14-19, 2022, Göttingen, Germany (Virtual Conference)*, ser. LIPIcs, F. Manea and A. Simpson, Eds., vol. 216. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, pp. 14:1–14:17. [Online]. Available: https://doi.org/10.4230/LIPIcs.CSL.2022.14

[59] J. Z. S. Hu and B. Pientka, "Layered modal type theory - where meta-programming meets intensional analysis," in *Programming Languages and Systems - 33rd European Symposium on Programming, ESOP 2024, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2024, Luxembourg City, Luxembourg, April 6-11, 2024, Proceedings, Part I*, ser. Lecture Notes in Computer Science, S. Weirich, Ed., vol. 14576. Springer, 2024, pp. 52–82. [Online]. Available: https://doi.org/10.1007/978-3-031-57262-3_3

[60] G. A. Kavvos, "Intensionality, intensional recursion and the Gödel-Löb axiom," *FLAP*, vol. 8, no. 8, pp. 2287–2312, 2021. [Online]. Available: https://collegepublications.co.uk/ifcolog/?00050

[61] ——, "Dual-context calculi for modal logic," *Log. Methods Comput. Sci.*, vol. 16, no. 3, 2020. [Online]. Available: https://lmcs.episciences.org/6722

Appendix

A. Detailed Definitions and Proofs of Section III

Definition A.1. $\mathbf{Dom}_{\mathrm{C}}(\Gamma)$ and $\mathbf{Dom}_{\mathrm{V}}(\Gamma)$ represent set of classifiers and variables declared in Γ, respectively.

$$
\begin{aligned}
\mathbf{Dom}_{\mathrm{C}}(\epsilon) &= \{\mathbf{!}\}\\
\mathbf{Dom}_{\mathrm{C}}(\Gamma, \boldsymbol{x}{:}^{\gamma} A) &= \mathbf{Dom}_{\mathrm{C}}(\Gamma) \cup \{\gamma\}\\
\mathbf{Dom}_{\mathrm{C}}(\Gamma, \text{🔓}^{\boldsymbol{\gamma_1}:\succeq\gamma_2}) &= \mathbf{Dom}_{\mathrm{C}}(\Gamma) \cup \{\gamma_1\}\\
\mathbf{Dom}_{\mathrm{C}}(\Gamma, \text{🔒}^{\gamma}) &= \mathbf{Dom}_{\mathrm{C}}(\Gamma)\\
\mathbf{Dom}_{\mathrm{C}}(\Gamma, \boldsymbol{\gamma_1} :\succeq \gamma_2) &= \mathbf{Dom}_{\mathrm{C}}(\Gamma) \cup \{\gamma_1\}\\
\mathbf{Dom}_{\mathrm{V}}(\epsilon) &= \{\}\\
\mathbf{Dom}_{\mathrm{V}}(\Gamma, \boldsymbol{x}{:}^{\gamma} A) &= \mathbf{Dom}_{\mathrm{V}}(\Gamma) \cup \{x\}\\
\mathbf{Dom}_{\mathrm{V}}(\Gamma, \text{🔓}^{\boldsymbol{\gamma_1}:\succeq\gamma_2}) &= \mathbf{Dom}_{\mathrm{V}}(\Gamma)\\
\mathbf{Dom}_{\mathrm{V}}(\Gamma, \text{🔒}^{\gamma}) &= \mathbf{Dom}_{\mathrm{V}}(\Gamma)\\
\mathbf{Dom}_{\mathrm{V}}(\Gamma, \boldsymbol{\gamma_1} :\succeq \gamma_2) &= \mathbf{Dom}_{\mathrm{V}}(\Gamma)
\end{aligned}
$$

Definition A.2. Scope transition judgment $\Gamma \vdash \gamma_1 \preceq \gamma_2$ and modal transition judgment $\Gamma \vdash \gamma_1 \sqsubseteq \gamma_2$ are derived by following rules. It also requires that $\vdash \Gamma \colon \mathbf{ctx}$ ($\trianglelefteq$ can be replaced with either of $\preceq$ or $\sqsubseteq$).

$$
\begin{aligned}
&\epsilon \vdash \mathbf{!} \trianglelefteq \mathbf{!}\\
&\Gamma, \boldsymbol{x}{:}^{\gamma} A \vdash \gamma \trianglelefteq \gamma\\
&\Gamma, \text{🔓}^{\boldsymbol{\gamma_1}:\succeq\gamma_2} \vdash \gamma_1 \trianglelefteq \gamma_1\\
&\Gamma, \boldsymbol{\gamma_1} :\succeq \gamma_2 \vdash \gamma_1 \trianglelefteq \gamma_1
\end{aligned}
$$

$$
\begin{aligned}
&\Gamma, \boldsymbol{x}{:}^{\boldsymbol{\gamma_2}} A \vdash \gamma_1 \trianglelefteq \gamma_2 \text{ iff } \Gamma \vdash \gamma_1 \trianglelefteq \mathrm{pos}(\Gamma)\\
&\Gamma, \text{🔓}^{\boldsymbol{\gamma_2}:\succeq\gamma_3} \vdash \gamma_1 \preceq \gamma_2 \text{ iff } \Gamma \vdash \gamma_1 \preceq \gamma_3\\
&\Gamma, \text{🔓}^{\boldsymbol{\gamma_2}:\succeq\gamma_3} \vdash \gamma_1 \sqsubseteq \gamma_2 \text{ iff } \Gamma \vdash \gamma_1 \sqsubseteq \gamma_3\\
&\qquad\qquad\qquad\qquad\quad \text{or } \Gamma \vdash \gamma_1 \sqsubseteq \mathrm{pos}(\Gamma)\\
&\Gamma, \boldsymbol{\gamma_2} :\succeq \gamma_3 \vdash \gamma_1 \trianglelefteq \gamma_2 \text{ iff } \Gamma \vdash \gamma_1 \trianglelefteq \gamma_3
\end{aligned}
$$

$$
\begin{aligned}
&\Gamma, \boldsymbol{x}{:}^{\boldsymbol{\gamma_3}} A \vdash \gamma_1 \trianglelefteq \gamma_2 \text{ iff } \gamma_2 \neq \gamma_3 \text{ and } \Gamma \vdash \gamma_1 \trianglelefteq \gamma_2\\
&\Gamma, \text{🔓}^{\boldsymbol{\gamma_3}:\succeq\gamma_4} \vdash \gamma_1 \trianglelefteq \gamma_2 \text{ iff } \gamma_2 \neq \gamma_3 \text{ and } \Gamma \vdash \gamma_1 \trianglelefteq \gamma_2\\
&\Gamma, \boldsymbol{\gamma_3} :\succeq \gamma_4 \vdash \gamma_1 \trianglelefteq \gamma_2 \text{ iff } \gamma_2 \neq \gamma_3 \text{ and } \Gamma \vdash \gamma_1 \trianglelefteq \gamma_2\\
&\Gamma, \text{🔒}^{\gamma_3} \vdash \gamma_1 \trianglelefteq \gamma_2 \text{ iff } \Gamma \vdash \gamma_1 \trianglelefteq \gamma_2
\end{aligned}
$$

Definition A.3. $\mathbf{FC}(A)$, $\mathbf{FC}(M)$ represent a set of free classifiers in A and M, respectively.

$$
\begin{aligned}
\mathbf{FC}(p) &= \{\}\\
\mathbf{FC}(A \to B) &= \mathbf{FC}(A) \cup \mathbf{FC}(B)\\
\mathbf{FC}([^{\succeq\gamma} A]) &= \mathbf{FC}(A) \cup \{\gamma\}\\
\mathbf{FC}(\forall \boldsymbol{\gamma_1} :\succeq \gamma_2 .\, A) &= (\mathbf{FC}(A) - \{\gamma_1\}) \cup \{\gamma_2\}\\
\mathbf{FC}(x) &= \{\}\\
\mathbf{FC}(\lambda \boldsymbol{x} : {}^{\gamma} A.\, M) &= \mathbf{FC}(M) \cup \{\gamma\}\\
\mathbf{FC}(M_1\ M_2) &= \mathbf{FC}(M_1) \cup \mathbf{FC}(M_2)\\
\mathbf{FC}(\text{‘}\{^{\gamma_1:\succeq\gamma_2} M\}) &= (\mathbf{FC}(M) \cup \{\gamma_2\}) - \{\gamma_1\}\\
\mathbf{FC}(\sim\{^{\gamma} M\}) &= \mathbf{FC}(M) \cup \{\gamma\}\\
\mathbf{FC}(\lambda \boldsymbol{\gamma_1} :\succeq \gamma_2 .\, M) &= (\mathbf{FC}(M) - \{\gamma_1\}) \cup \{\gamma_2\}\\
\mathbf{FC}(M \gamma) &= \mathbf{FC}(M) \cup \{\gamma\}
\end{aligned}
$$

Definition A.4. $\mathbf{FV}(M)$ represents a set of free variables in M.

$$
\begin{aligned}
\mathbf{FV}(x) &= \{x\}\\
\mathbf{FV}(\lambda \boldsymbol{x} : {}^{\gamma} A.\, M) &= \mathbf{FV}(M) - \{x\}\\
\mathbf{FV}(M_1\ M_2) &= \mathbf{FV}(M_1) \cup \mathbf{FV}(M_2)\\
\mathbf{FV}(\text{‘}\{^{\gamma_1:\succeq\gamma_2} M\}) &= \mathbf{FV}(M)\\
\mathbf{FV}(\sim\{^{\gamma} M\}) &= \mathbf{FV}(M)\\
\mathbf{FV}(\lambda \boldsymbol{\gamma_1} :\succeq \gamma_2 .\, M) &= \mathbf{FV}(M)\\
\mathbf{FV}(M \gamma) &= \mathbf{FV}(M)
\end{aligned}
$$

Definition A.5. A well-formed context judgment $\vdash \Gamma \colon \mathbf{ctx}$ and well-formed type judgment $\Gamma \vdash A \colon \mathbf{type}$ are derived by rules listed in Figure 8

Definition A.6. A classifier substitution $\cdot[\boldsymbol{\gamma_1}{:=}\gamma_2]$ is a meta operation on classifiers, types, terms and contexts, which replaces free occurrences of γ_1 with γ_2.

$$
\gamma_1[\boldsymbol{\gamma_2}{:=}\gamma_3] = \begin{cases} \gamma_3 & \text{if } \gamma_1 = \gamma_2\\ \gamma_1 & \text{otherwise} \end{cases}
$$

$$
\begin{aligned}
p[\boldsymbol{\gamma_1}{:=}\gamma_2] &= p\\
(A \to B)[\boldsymbol{\gamma_1}{:=}\gamma_2] &= A[\boldsymbol{\gamma_1}{:=}\gamma_2] \to B[\boldsymbol{\gamma_1}{:=}\gamma_2]\\
[^{\succeq\gamma_1} A][\boldsymbol{\gamma_2}{:=}\gamma_3] &= [^{\succeq\gamma_1[\boldsymbol{\gamma_2}:=\gamma_3]} A[\boldsymbol{\gamma_2}{:=}\gamma_3]]\\
(\forall \boldsymbol{\gamma_1} :\succeq \gamma_2 .\, A)[\boldsymbol{\gamma_3}{:=}\gamma_4] &= \forall \boldsymbol{\gamma_1} :\succeq \gamma_2[\boldsymbol{\gamma_3}{:=}\gamma_4].\, A[\boldsymbol{\gamma_3}{:=}\gamma_4]\\
&\qquad \text{where } \gamma_1 \notin \{\gamma_3, \gamma_4\}
\end{aligned}
$$

$$
\begin{aligned}
x[\boldsymbol{\gamma_1}{:=}\gamma_2] &= x\\
(\lambda \boldsymbol{x} : {}^{\gamma_1} A.\, M)[\boldsymbol{\gamma_2}{:=}\gamma_3] &= \lambda \boldsymbol{x} : {}^{\gamma_1} A[\boldsymbol{\gamma_2}{:=}\gamma_3].\, (M[\boldsymbol{\gamma_2}{:=}\gamma_3])\\
&\qquad \text{where } \gamma_1 \notin \{\gamma_2, \gamma_3\}\\
(M\ N)[\boldsymbol{\gamma_1}{:=}\gamma_2] &= (M[\boldsymbol{\gamma_1}{:=}\gamma_2])\ (N[\boldsymbol{\gamma_1}{:=}\gamma_2])\\
\text{‘}\{^{\gamma_1:\succeq\gamma_2} M\}[\boldsymbol{\gamma_3}{:=}\gamma_4] &= \text{‘}\{^{\gamma_1:\succeq\gamma_2[\boldsymbol{\gamma_3}:=\gamma_4]} M[\boldsymbol{\gamma_3}{:=}\gamma_4]\}\\
&\qquad \text{where } \gamma_1 \notin \{\gamma_3, \gamma_4\}\\
\sim\{^{\gamma_1} M\}[\boldsymbol{\gamma_2}{:=}\gamma_3] &= \sim\{^{\gamma_1[\boldsymbol{\gamma_2}:=\gamma_3]} M[\boldsymbol{\gamma_1}{:=}\gamma_2]\}\\
(\lambda \boldsymbol{\gamma_1} :\succeq \gamma_2 .\, M)[\boldsymbol{\gamma_3}{:=}\gamma_4] &= \lambda \boldsymbol{\gamma_1} :\succeq \gamma_2[\boldsymbol{\gamma_3}{:=}\gamma_4].\, (M[\boldsymbol{\gamma_3}{:=}\gamma_4])\\
&\qquad \text{where } \gamma_1 \notin \{\gamma_3, \gamma_4\}\\
(M \gamma_1)[\boldsymbol{\gamma_2}{:=}\gamma_3] &= (M[\boldsymbol{\gamma_2}{:=}\gamma_3])\gamma_1[\boldsymbol{\gamma_2}{:=}\gamma_3]
\end{aligned}
$$

$$
\begin{aligned}
\epsilon[\boldsymbol{\gamma_1}{:=}\gamma_2] &= \epsilon\\
(\boldsymbol{x}{:}^{\boldsymbol{\gamma_1}} A, \Gamma)[\boldsymbol{\gamma_2}{:=}\gamma_3] &= \boldsymbol{x}{:}^{\boldsymbol{\gamma_1}} A[\boldsymbol{\gamma_2}{:=}\gamma_3], \Gamma[\boldsymbol{\gamma_2}{:=}\gamma_3]\\
&\qquad \text{where } \gamma_1 \notin \{\gamma_2, \gamma_3\}\\
(\text{🔓}^{\boldsymbol{\gamma_1}:\succeq\gamma_2}, \Gamma)[\boldsymbol{\gamma_3}{:=}\gamma_4] &= \text{🔓}^{\boldsymbol{\gamma_1}:\succeq\gamma_2[\boldsymbol{\gamma_3}:=\gamma_4]}, \Gamma[\boldsymbol{\gamma_3}{:=}\gamma_4]\\
&\qquad \text{where } \gamma_1 \notin \{\gamma_3, \gamma_4\}\\
(\text{🔒}^{\gamma_1}, \Gamma)[\boldsymbol{\gamma_2}{:=}\gamma_3] &= \text{🔒}^{\gamma_1[\boldsymbol{\gamma_2}:=\gamma_3]}, \Gamma[\boldsymbol{\gamma_2}{:=}\gamma_3]\\
(\boldsymbol{\gamma_1} :\succeq \gamma_2, \Gamma)[\boldsymbol{\gamma_3}{:=}\gamma_4] &= \boldsymbol{\gamma_1} :\succeq \gamma_2[\boldsymbol{\gamma_3}{:=}\gamma_4], \Gamma[\boldsymbol{\gamma_3}{:=}\gamma_4]
\end{aligned}
$$

$$\text{WF-}\epsilon\ \frac{}{\vdash \epsilon : \mathbf{ctx}} \qquad \text{WF-Var}\ \frac{\vdash \Gamma : \mathbf{ctx} \qquad x \notin \mathbf{Dom}_{\mathrm{V}}(\Gamma) \qquad \Gamma \vdash A : \mathbf{type} \qquad \gamma \notin \mathbf{Dom}_{\mathrm{C}}(\Gamma)}{\vdash \Gamma, \boldsymbol{x}{:}^{\gamma} A : \mathbf{ctx}}$$

$$\text{WF-🔓}\ \frac{\vdash \Gamma : \mathbf{ctx} \qquad \gamma_1 \notin \mathbf{Dom}_{\mathrm{C}}(\Gamma) \qquad \gamma_2 \in \mathbf{Dom}_{\mathrm{C}}(\Gamma)}{\vdash \Gamma, 🔓^{\boldsymbol{\gamma_1}:\succeq\gamma_2} : \mathbf{ctx}} \qquad \text{WF-🔒}\ \frac{\vdash \Gamma^{\gamma} : \mathbf{ctx} \qquad \Gamma^{\gamma} \vdash \delta \sqsubseteq \gamma}{\vdash \Gamma^{\gamma}, 🔒^{\delta} : \mathbf{ctx}}$$

$$\text{WF-}{:}\succeq\ \frac{\vdash \Gamma : \mathbf{ctx} \qquad \gamma_1 \notin \mathbf{Dom}_{\mathrm{C}}(\Gamma) \qquad \gamma_2 \in \mathbf{Dom}_{\mathrm{C}}(\Gamma)}{\vdash \Gamma, \boldsymbol{\gamma_1} :\succeq \gamma_2 : \mathbf{ctx}}$$

$$\text{WF-}p\ \frac{\vdash \Gamma : \mathbf{ctx}}{\Gamma \vdash p : \mathbf{type}} \qquad \text{WF-}\rightarrow\ \frac{\Gamma \vdash A : \mathbf{type} \qquad \Gamma \vdash B : \mathbf{type}}{\Gamma \vdash A \rightarrow B : \mathbf{type}}$$

$$\text{WF-[]}\ \frac{\Gamma \vdash A : \mathbf{type} \qquad \gamma \in \mathbf{Dom}_{\mathrm{C}}(\Gamma)}{\Gamma \vdash [^{\succeq\gamma} A] : \mathbf{type}} \qquad \text{WF-}\forall\ \frac{\Gamma, \boldsymbol{\gamma_1} :\succeq \gamma_2 \vdash A : \mathbf{type}}{\Gamma \vdash \forall \boldsymbol{\gamma_1} :\succeq \gamma_2 .\, A : \mathbf{type}}$$

Fig. 8. Derivation Rules for Well-Formednes Judgments

where $\gamma_1 \notin \{\gamma_3, \gamma_4\}$

Definition A.7. A variable substitution $\cdot[\boldsymbol{\gamma_1}{:=}\gamma_2, \boldsymbol{x}{:=}M]$ is a meta operation on terms that replaces free occurrences of γ_1 and x with γ_2 and M, respectively. Its definition is given in Figure 9.

Definition A.8 (Full rules for Definition 1)**.** The full definition of derivation rules for $M_1@\gamma \Rightarrow_\beta M_2$ are follows.

Axioms

$$(\lambda \boldsymbol{x} : {}^{\gamma_2}A.\, M)\ N@\gamma_1 \Rightarrow_\beta M[\boldsymbol{\gamma_2}{:=}\gamma_1, \boldsymbol{x}{:=}N]$$
$${\sim}\{^{\gamma_2}\text{‘}\{^{\boldsymbol{\gamma_3}:\succeq\gamma_4} M\}\}@\gamma_1 \Rightarrow_\beta M[\boldsymbol{\gamma_3}{:=}\gamma_1]$$
$$(\lambda \boldsymbol{\gamma_2} :\succeq \gamma_3 .\, M)\gamma_4@\gamma_1 \Rightarrow_\beta M[\boldsymbol{\gamma_2}{:=}\gamma_4]$$

Compatibility Rules

$$\frac{M_1@\gamma_1 \Rightarrow_\beta M_2}{\lambda \boldsymbol{x} : {}^{\gamma_1}A.\, M_1@\gamma_2 \Rightarrow_\beta \lambda \boldsymbol{x} : {}^{\gamma_1}A.\, M_2}$$

$$\frac{M_1@\gamma_1 \Rightarrow_\beta M_2}{M_1\ N@\gamma_1 \Rightarrow_\beta M_2\ N}$$

$$\frac{M_1@\gamma_1 \Rightarrow_\beta M_2}{N\ M_1@\gamma_1 \Rightarrow_\beta N\ M_2}$$

$$\frac{M_1@\gamma_1 \Rightarrow_\beta M_2}{\text{‘}\{^{\boldsymbol{\gamma_1}:\succeq\gamma_2} M_1\}@\gamma_3 \Rightarrow_\beta \text{‘}\{^{\boldsymbol{\gamma_1}:\succeq\gamma_2} M_2\}}$$

$$\frac{M_1@\gamma_1 \Rightarrow_\beta M_2}{{\sim}\{^{\gamma_1} M_1\}@\gamma_2 \Rightarrow_\beta {\sim}\{^{\gamma_1} M_2\}}$$

$$\frac{M_1@\gamma_1 \Rightarrow_\beta M_2}{\lambda \boldsymbol{\gamma_2} :\succeq \gamma_3 .\, M_1@\gamma_1 \Rightarrow_\beta \lambda \boldsymbol{\gamma_2} :\succeq \gamma_3 .\, M_2}$$

$$\frac{M_1@\gamma_1 \Rightarrow_\beta M_2}{M_1\gamma_2@\gamma_1 \Rightarrow_\beta M_2\gamma_2}$$

Lemma 1 (On page 3)**.** *Judgments below are derivable.*

1) $\Gamma_1^{\gamma}, \boldsymbol{x}{:}^{\boldsymbol{\delta}} A, \Gamma_2 \vdash \gamma \preceq \delta$
2) $\Gamma_1^{\gamma_1}, 🔓^{\boldsymbol{\delta}:\succeq\gamma_2}, \Gamma_2 \vdash \gamma_2 \preceq \delta$
3) $\Gamma_1^{\gamma_1}, 🔓^{\boldsymbol{\delta}:\succeq\gamma_2}, \Gamma_2 \vdash \gamma_2 \sqsubseteq \delta$
4) $\Gamma_1^{\gamma_1}, 🔓^{\boldsymbol{\delta}:\succeq\gamma_2}, \Gamma_2 \vdash \gamma_1 \sqsubseteq \delta$
5) $\Gamma_1, \boldsymbol{\delta} :\succeq \gamma, \Gamma_2 \vdash \gamma \preceq \delta$

Proof. By induction on the size of Γ_2. Note that we need $\Gamma_1 \vdash \gamma \preceq \gamma$ in 4, which is ensured by $\vdash \Gamma : \mathbf{ctx}$. ☐

Lemma 2 (On page 3)**.** *Statements below hold. ($\trianglelefteq = \preceq$ or $\sqsubseteq$)*

1) $\gamma \in \mathbf{Dom}_{\mathrm{C}}(\Gamma) \implies \Gamma \vdash \gamma \trianglelefteq \gamma$.
2) $\Gamma \vdash \gamma_1 \trianglelefteq \gamma_2$ *and* $\Gamma \vdash \gamma_2 \trianglelefteq \gamma_3 \implies \Gamma \vdash \gamma_1 \trianglelefteq \gamma_3$.
3) $\Gamma \vdash \gamma_1 \preceq \gamma_2 \implies \Gamma \vdash \gamma_1 \sqsubseteq \gamma_2$.

Proof.
1) By induction on the size of Γ.
2) By induction on the derivation $\Gamma \vdash \gamma_2 \trianglelefteq \gamma_3$.
3) By induction on the derivation $\Gamma \vdash \gamma_1 \preceq \gamma_2$. Basically we construct the derivation of $\Gamma \vdash \gamma_1 \sqsubseteq \gamma_3$ from that of $\Gamma \vdash \gamma_1 \preceq \gamma_3$, where each derivation step is obtained by replacing $\preceq$ with $\sqsubseteq$.

☐

Lemma A.1.
1) $\Gamma \vdash A : \mathbf{type} \implies \vdash \Gamma : \mathbf{ctx}$
2) $\Gamma \vdash M : A \implies \vdash \Gamma : \mathbf{ctx}$

Proof.
1) By induction on the definvation of $\Gamma \vdash A : \mathbf{type}$.
2) By induction on the definvation of $\Gamma \vdash M : A$.

☐

Theorem 1 (Weakening (On page 4))**.** *Given* $\vdash \Gamma, \Delta : \mathbf{ctx}$,

1) $\Gamma \vdash A : \mathbf{type} \Rightarrow \Gamma, \Delta \vdash A : \mathbf{type}$.
2) $\Gamma \vdash \gamma_1 \trianglelefteq \gamma_2 \Rightarrow \Gamma, \Delta \vdash \gamma_1 \trianglelefteq \gamma_2$. *($\trianglelefteq = \preceq$ or $\sqsubseteq$)*

Proof. We prove 1 as a special case of the following statement.

- If $\Gamma, \Delta_1 \vdash A : \mathbf{type}$ and $\vdash \Gamma, \Delta_2, \Delta_1 : \mathbf{ctx}$ where $\Delta_1 = \boldsymbol{\gamma_1} :\succeq \gamma_1' \ldots \boldsymbol{\gamma_i} :\succeq \gamma_i'$ and $\Delta_1 \# \Delta_2$, then $\Gamma, \Delta_2, \Delta_1 \vdash A : \mathbf{type}$.

We can prove it by induction on the derivation of $\Gamma, \Delta_1 \vdash A : \mathbf{type}$. 2 can be proved by induction on the derivation of $\Gamma \vdash \gamma_1 \trianglelefteq \gamma_2$. ☐

$$
\begin{aligned}
x[\boldsymbol{\gamma_1}:=\gamma_2, \boldsymbol{y}:=M] &= \begin{cases} M & \text{where } x = y \\ x & \text{otherwise} \end{cases} \\
(\lambda \boldsymbol{x} : {}^{\gamma_1} A.\, M)[\boldsymbol{\gamma_2}:=\gamma_3, \boldsymbol{y}:=M] &= \lambda \boldsymbol{x} : {}^{\gamma_1} A[\boldsymbol{\gamma_2}:=\gamma_3].\, (M[\boldsymbol{\gamma_2}:=\gamma_3, \boldsymbol{y}:=M]) \\
& \text{ where } \gamma_1 \notin \{\gamma_2, \gamma_3\} \text{ and } x \neq y \\
(M_1\ M_2)[\boldsymbol{\gamma_1}:=\gamma_2, \boldsymbol{x}:=N] &= M_1[\boldsymbol{\gamma_1}:=\gamma_2, \boldsymbol{x}:=N]\ M_2[\boldsymbol{\gamma_1}:=\gamma_2, \boldsymbol{y}:=N] \\
\text{‘}\{^{\boldsymbol{\gamma_1} :\succeq \gamma_2} M\}[\boldsymbol{\gamma_3}:=\gamma_4, \boldsymbol{x}:=N] &= \text{‘}\{^{\boldsymbol{\gamma_1} :\succeq \gamma_2[\boldsymbol{\gamma_3}:=\gamma_4]} M[\boldsymbol{\gamma_3}:=\gamma_4, \boldsymbol{x}:=N]\} \\
& \text{ where } \gamma_1 \notin \{\gamma_3, \gamma_4\} \\
\mathord{\sim}\{^{\gamma_1} M\}[\boldsymbol{\gamma_2}:=\gamma_3, \boldsymbol{x}:=N] &= \mathord{\sim}\{^{\gamma_1[\boldsymbol{\gamma_2}:=\gamma_3]} M[\boldsymbol{\gamma_2}:=\gamma_3, \boldsymbol{x}:=N]\} \\
(\lambda \boldsymbol{\gamma_1} :\succeq \gamma_2.\, M)[\boldsymbol{\gamma_3}:=\gamma_4, \boldsymbol{x}:=N] &= \lambda \boldsymbol{\gamma_1} :\succeq \gamma_2[\boldsymbol{\gamma_3}:=\gamma_4].\, (M[\boldsymbol{\gamma_3}:=\gamma_4, \boldsymbol{x}:=N]) \\
& \text{ where } \gamma_1 \notin \{\gamma_3, \gamma_4\} \\
(M \gamma_1)[\boldsymbol{\gamma_2}:=\gamma_3, \boldsymbol{x}:=N] &= M[\boldsymbol{\gamma_2}:=\gamma_3, \boldsymbol{x}:=N]\gamma_1[\boldsymbol{\gamma_2}:=\gamma_3]
\end{aligned}
$$

Fig. 9. Definition of Variable Substitution

Theorem 2 (Monotonicty (On page 4))**.** *Given $\Delta_1 \# \Delta_2$ and $\Gamma^{\gamma_1}, \Delta_1^{\gamma_2} \vdash \gamma_1 \preceq \gamma_2$, then the following statements hold:*

1) $\vdash \Gamma, \Delta_2 : \mathbf{ctx} \implies \vdash \Gamma, \Delta_1, \Delta_2 : \mathbf{ctx}$.
2) $\Gamma, \Delta_2 \vdash A : \mathbf{type} \implies \Gamma, \Delta_1, \Delta_2 \vdash A : \mathbf{type}$.
3) $\Gamma, \Delta_2 \vdash \delta_1 \preceq \delta_2 \implies \Gamma, \Delta_1, \Delta_2 \vdash \delta_1 \preceq \delta_2$.
4) $\Gamma, \Delta_2 \vdash \delta_1 \sqsubseteq \delta_2 \implies \Gamma, \Delta_1, \Delta_2 \vdash \delta_1 \sqsubseteq \delta_2$.
5) $\Gamma, \Delta_2 \vdash M : A \implies \Gamma, \Delta_1, \Delta_2 \vdash M : A$.

Proof. By mutual induction the derivation of each judgment. We focus on the most important case in (3).

Case Derived from $\Gamma_1^{\gamma_1}, \Delta_3^{\delta_3} \vdash \delta_1 \preceq \delta_3$ where $\Delta_2 = \Delta_3, \boldsymbol{x}{:}^{\boldsymbol{\delta_2}} A$: We apply the induction hypothesis to get $\Gamma_1^{\gamma_1}, \Delta_1^{\gamma_2}, \Delta_3^{\delta_4} \vdash \delta_1 \preceq \delta_3$. Note that $\delta_3 = \delta_4$ does not necesarilly hold, and we need to check both cases. If $\delta_3 = \delta_4$, we can simply derive $\Gamma, \Delta_1, \Delta_2 \vdash \delta_1 \preceq \delta_2$. Otherwise, we have $\delta_3 = \gamma_1$ and $\delta_4 = \gamma_2$. As we have $\Gamma_1^{\gamma_1}, \Delta_1^{\gamma_2}, \Delta_3^{\delta_4} \vdash \gamma_1 \preceq \gamma_2$ from the assumption and weakening, we apply Lemma 2(2) to obtain $\Gamma, \Delta_1, \Delta_3^{\delta_4} \vdash \delta_1 \preceq \delta_4$. Therefore we can derive $\Gamma, \Delta_1, \Delta_2 \vdash \delta_1 \preceq \delta_2$.

Other cases in (3) and (4) works similarly, where we use the assumption and $\Gamma_1^{\gamma_1}, \Delta_1^{\gamma_2} \vdash \gamma_1 \preceq \gamma_2$ and Lemma 2(2) on demand. (1)(2)(5) are straightforward induction with some cases depending on (3)(4). □

Lemma 3 (Variable Substitution (On page 4))**.** *Let $\Delta_1 = \Gamma_1^{\gamma_1}, \boldsymbol{x}{:}^{\boldsymbol{\gamma_2}} A, \Gamma_2$, and $\Delta_2 = \Gamma_1, \Gamma_2[\boldsymbol{\gamma_2}:=\gamma_1]$. Then, the following statements hold.*

1) $\vdash \Delta_1 : \mathbf{ctx} \implies \vdash \Delta_2 : \mathbf{ctx}$.
2) $\Delta_1 \vdash A : \mathbf{type} \implies \Delta_2 \vdash A[\boldsymbol{\gamma_2}:=\gamma_1] : \mathbf{type}$.
3) $\Delta_1 \vdash \delta_1 \preceq \delta_2 \implies \Delta_2 \vdash \delta_1[\boldsymbol{\gamma_2}:=\gamma_1] \preceq \delta_2[\boldsymbol{\gamma_2}:=\gamma_1]$.
4) $\Delta_1 \vdash \delta_1 \sqsubseteq \delta_2 \implies \Delta_2 \vdash \delta_1[\boldsymbol{\gamma_2}:=\gamma_1] \sqsubseteq \delta_2[\boldsymbol{\gamma_2}:=\gamma_1]$.
5) $\Delta_1 \vdash M_1 : B$ *and* $\Gamma_1 \vdash M_2 : A$
   $\implies \Delta_2 \vdash M_1[\boldsymbol{\gamma_2}:=\gamma_1, \boldsymbol{x}:=M_2] : B[\boldsymbol{\gamma_2}:=\gamma_1]$.

Proof. By mutual induction on the defivation of the first judgment for each statements. To prove the case of typing judgment, we use Theorem 2 for the base case where M_1 is variable. □

Lemma 4 (Rebasing (On page 5))**.** *Let $\Delta_1 = (\Gamma_1^{\gamma_1}, \text{🔒}^{\gamma_2}, \text{🔓}^{\boldsymbol{\gamma_3} :\succeq \gamma_4}, \Gamma_2)$, and $\Delta_2 = \Gamma_1, \Gamma_2[\boldsymbol{\gamma_3}:=\gamma_1]$. Supposing $\Gamma_1 \vdash \gamma_4 \preceq \gamma_1$, the following statements hold,*

1) $\vdash \Delta_1 : \mathbf{ctx} \implies \vdash \Delta_2 : \mathbf{ctx}$.
2) $\Delta_1 \vdash A : \mathbf{type} \implies \Delta_2 \vdash A[\boldsymbol{\gamma_3}:=\gamma_1] : \mathbf{type}$.
3) $\Delta_1 \vdash \delta_1 \preceq \delta_2 \implies \Delta_2 \vdash \delta_1[\boldsymbol{\gamma_3}:=\gamma_1] \preceq \delta_2[\boldsymbol{\gamma_3}:=\gamma_1]$.
4) $\Delta_1 \vdash \delta_1 \sqsubseteq \delta_2 \implies \Delta_2 \vdash \delta_1[\boldsymbol{\gamma_3}:=\gamma_1] \sqsubseteq \delta_2[\boldsymbol{\gamma_3}:=\gamma_1]$.
5) $\Delta_1 \vdash M_1 : A \implies \Delta_2 \vdash M[\boldsymbol{\gamma_3}:=\gamma_1] : A[\boldsymbol{\gamma_3}:=\gamma_1]$.

Proof. By mutual induction on the first derivarion of each statements. □

Lemma 5 (Classifier Substitution (On page 5))**.** *Let $\Delta_1 = \Gamma_1, \boldsymbol{\gamma_1} :\succeq \gamma_2, \Gamma_2$ and $\Delta_2 = \Gamma_1, \Gamma_2[\boldsymbol{\gamma_1}:=\gamma_3]$. Given $\Gamma_1 \vdash \gamma_2 \preceq \gamma_3$, then the following statements hold.*

1) $\Delta_1 \vdash A : \mathbf{type} \implies \Delta_2 \vdash A[\boldsymbol{\gamma_1}:=\gamma_3] : \mathbf{type}$.
2) $\vdash \Delta_1 : \mathbf{ctx} \implies \vdash \Delta_2 : \mathbf{ctx}$.
3) $\Delta_1 \vdash \delta_1 \preceq \delta_2 \implies \Delta_2 \vdash \delta_1[\boldsymbol{\gamma_1}:=\gamma_3] \preceq \delta_2[\boldsymbol{\gamma_1}:=\gamma_3]$.
4) $\Delta_1 \vdash \delta_1 \sqsubseteq \delta_2 \implies \Delta_2 \vdash \delta_1[\boldsymbol{\gamma_1}:=\gamma_3] \sqsubseteq \delta_2[\boldsymbol{\gamma_1}:=\gamma_3]$.
5) $\Delta_1 \vdash M : A \implies \Delta_2 \vdash M[\boldsymbol{\gamma_1}:=\gamma_3] : A[\boldsymbol{\gamma_1}:=\gamma_3]$.

Proof. By mutual induction on the first derivarion of each statements. □

Lemma 6 (Local Soundness Patterns (On page 5))**.**

1) $\Gamma^{\gamma_1} \vdash (\lambda \boldsymbol{x} : {}^{\gamma_2} A.\, M)\ N : B$
   $\implies \Gamma \vdash M[\boldsymbol{\gamma_2}:=\gamma_1, \boldsymbol{x}:=N] : B$.
2) $\Gamma^{\gamma_1} \vdash \mathord{\sim}\{^{\gamma_2} \text{‘}\{^{\boldsymbol{\gamma_3} :\succeq \gamma_4} M\}\} : A \implies \Gamma \vdash M[\boldsymbol{\gamma_3}:=\gamma_1] : A$.
3) $\Gamma \vdash (\lambda \boldsymbol{\gamma_1} :\succeq \gamma_2.\, M)\gamma_3 : A \implies \Gamma \vdash M[\boldsymbol{\gamma_1}:=\gamma_3] : A$.

Proof. Easy to prove with Lemma 3, Lemma 4 and Lemma 5. □

Lemma 7 (Local Completeness Patterns (On page 5))**.** *(δ is taken freshly)*

1) $\Gamma \vdash M : A \to B \implies \Gamma \vdash \lambda \boldsymbol{x} : {}^{\delta} A.\, (M\ x) : A \to B$.
2) $\Gamma^{\gamma_1} \vdash M : [^{\succeq \gamma_2} A] \Rightarrow \Gamma \vdash \text{‘}\{^{\boldsymbol{\delta} :\succeq \gamma_2} \mathord{\sim}\{^{\gamma_1} M\}\} : [^{\succeq \gamma_2} A]$.
3) $\Gamma \vdash M : \forall \boldsymbol{\gamma_1} :\succeq \gamma_2.\, A$
   $\implies \Gamma \vdash \lambda \boldsymbol{\delta} :\succeq \gamma_2.\, (M \delta) : \forall \boldsymbol{\gamma_1} :\succeq \gamma_2.\, A$.

Proof. Easy to prove with Theorem 2. □

Theorem 3 (Subject Reduction (On page 5)). *If $\Gamma^{\gamma} \vdash M_1 \colon A$ and $M_1 @ \gamma \Rightarrow_{\beta} M_2$, then $\Gamma \vdash M_2 \colon A$.*

Proof. By induction on the derivation of $M_1 @ \gamma \Rightarrow_{\beta} M_2$. For base cases, we apply Lemma 6. □

B. Detailed Proofs of Section IV

Theorem 4 (Type-Safe Residualization (On page 6)). *If $\epsilon \vdash M \colon [{}^{\succeq \mathbf{!}} A]$ and M is normal with regard to $\Rightarrow_{\beta}$, then $M = \text{‘}\{^{\gamma : \succeq \mathbf{!}} M'\}$ for some M' and $\epsilon \vdash M'[\boldsymbol{\gamma} {:=} \mathbf{!}] \colon A$ is derivable.*

Proof. M is normal form and typed $[{}^{\succeq \mathbf{!}} A]$ under an empty context, []-I is the only applicable rule. Hence, there must be M' and γ such that $M = \text{‘}\{^{\gamma : \succeq \mathbf{!}} M'\}$ holdsfrom Theorem 13. It is derived from $\text{🔓}^{\gamma : \succeq \mathbf{!}} \vdash M' \colon A$ and $\gamma \notin \mathbf{FC}(A)$. We can see that $\text{🔒}^{\mathbf{!}}, \text{🔓}^{\gamma : \succeq \mathbf{!}} \vdash M' \colon A$ is also derivable from monotonicity, and then we apply rebasing to obtain $\epsilon \vdash M'[\boldsymbol{\gamma} {:=} \mathbf{!}] \colon A$. □

C. Detailed Definitions and Proofs of Section V

1) Embedding from Kripke-style **S4** *modal lambda-calculus:*

There are several variants of S4 Kripke/Fitch-style modal lambda-calculi with different term assignments [7], [25]–[27]. Here we use an **S4** Kripke-style modal lambda-calculus by Davies and Pfenning [7], which we call $\lambda_{\Box_{S4}}$ in this paper. In $\lambda_{\Box_{S4}}$, **unbox** carries an integer that represents the number of modal transitions, which makes the definition of our translation simpler.

The syntax and typing rules of $\lambda_{\Box_{S4}}$ are shown in Fig.10. Note that we tweak definitions around contexts to more like Fitch-style, which does not change the essence of $\lambda_{\Box_{S4}}$.

$\lambda_{\Box_{S4}}$ can be considered as restricted version of BMTT, where quoted code is always closed. This means that a box type $\Box A^{\blacksquare}$ corresponds to a bounded modal type with an initial classifier $[{}^{\succeq \gamma} A]$. The whole definition of the translation is provided in Figure 11.

The term translation $\llparenthesis M^{\blacksquare} \rrparenthesis^{\overrightarrow{\gamma}}$ carries a sequence of classifiers $\overrightarrow{\gamma}$, which represents positions for each past stage. The context translation judgment $\Gamma^{\blacksquare} \rightsquigarrow \Gamma / \overrightarrow{\gamma}$ states that $\Gamma^{\blacksquare}$ can be translated to Γ where positions of past states are $\overrightarrow{\gamma}$. We can prove this translation preserves typeability.

Lemma C.1. *$\Gamma \vdash \mathbf{!} \preceq \gamma$ and $\Gamma \vdash \mathbf{!} \sqsubseteq \gamma$ hold as long as $\gamma \in \mathbf{Dom}_{\mathrm{C}}(\Gamma)$.*

Proof. By induction on the derivation of $\Gamma \vdash \mathbf{!} \preceq \gamma$ and $\Gamma \vdash \mathbf{!} \sqsubseteq \gamma$. □

Theorem C.1. *If $\Gamma^{\blacksquare} \vdash M^{\blacksquare} : A^{\blacksquare}$ and $\Gamma^{\blacksquare} \rightsquigarrow \Gamma / \overrightarrow{\gamma}$, then $\Gamma \vdash \llparenthesis M^{\blacksquare} \rrparenthesis^{\overrightarrow{\gamma}} \colon \llparenthesis A^{\blacksquare} \rrparenthesis$ holds.*

Proof. By induction on the derivation of $\Gamma^{\blacksquare} \vdash M^{\blacksquare} : A^{\blacksquare}$. We demonstrate the case of $\Box$-E.

Case $\Box$-E: We have a derivation

$$\frac{{\Gamma^{\blacksquare}}_1 \vdash M^{\blacksquare\prime} : \Box A^{\blacksquare} \qquad k \ \text{🔓} \text{ in } {\Gamma^{\blacksquare}}_2}{{\Gamma^{\blacksquare}}_1, {\Gamma^{\blacksquare}}_2 \vdash \mathbf{unbox}_k M^{\blacksquare\prime} : A^{\blacksquare}}$$

where ${\Gamma^{\blacksquare}}_1, {\Gamma^{\blacksquare}}_2 = \Gamma^{\blacksquare}$. Here, we can assume that ${\Gamma^{\blacksquare}}_2$ starts with 🔓 or ${\Gamma^{\blacksquare}}_2$ is empty otherwise, by the weakening property of typing judgments in $\lambda_{\Box_{S4}}$. Decomposing $\overrightarrow{\gamma}$ to $\overrightarrow{\gamma}', \delta_0 \ldots \delta_k$, we derive ${\Gamma^{\blacksquare}}_1 \rightsquigarrow \Delta, \text{🔒}^{\delta_0} / \overrightarrow{\gamma}', \delta_0$ from ${\Gamma^{\blacksquare}}_1, {\Gamma^{\blacksquare}}_2 \rightsquigarrow \Delta / \overrightarrow{\gamma}', \delta_0 \ldots \delta_k$ where $\mathbf{pop}(({\Gamma^{\blacksquare}}_1, {\Gamma^{\blacksquare}}_2), k) = {\Gamma^{\blacksquare}}_1$. Then we can apply the induction hypothesis, and get $\Delta, \text{🔒}^{\delta_0} \vdash \llparenthesis M^{\blacksquare\prime} \rrparenthesis^{\overrightarrow{\gamma}', \delta_0} \colon [{}^{\succeq \mathbf{!}} \llparenthesis A^{\blacksquare} \rrparenthesis]$. We apply []-E to derive $\Delta \vdash \sim\{^{\delta_0} \llparenthesis M^{\blacksquare\prime} \rrparenthesis^{\overrightarrow{\gamma}', \delta_0}\} \colon \llparenthesis A^{\blacksquare} \rrparenthesis$, which is what we want. □

Variables	x, y		
Types	$A^{\blacksquare}, B^{\blacksquare}$	$::=$	$p \mid A^{\blacksquare} \to B^{\blacksquare} \mid \Box A^{\blacksquare}$
Terms	$M^{\blacksquare}, N^{\blacksquare}$	$::=$	$x \mid \lambda x^{A^{\blacksquare}}.\, M^{\blacksquare} \mid M^{\blacksquare}\ N^{\blacksquare}$
		$\mid$	$\mathbf{box}\,\{M^{\blacksquare}\} \mid \mathbf{unbox}_k M^{\blacksquare}$
Context	$\Gamma^{\blacksquare}, \Delta^{\blacksquare}$	$::=$	$\epsilon \mid \Gamma^{\blacksquare}, x \colon A^{\blacksquare} \mid \Gamma^{\blacksquare}, \text{🔓}$

$$\text{Var}\ \frac{x \colon A^{\blacksquare} \in \mathbf{tail}(\Gamma^{\blacksquare})}{\Gamma^{\blacksquare} \vdash x : A^{\blacksquare}}$$

$$\to\text{-I}\ \frac{\Gamma^{\blacksquare}, x \colon A^{\blacksquare} \vdash M^{\blacksquare} : B^{\blacksquare}}{\Gamma^{\blacksquare} \vdash \lambda x^{A^{\blacksquare}}.\, M^{\blacksquare} : A^{\blacksquare} \to B^{\blacksquare}}$$

$$\to\text{-E}\ \frac{\Gamma^{\blacksquare} \vdash M^{\blacksquare} : A^{\blacksquare} \to B^{\blacksquare} \qquad \Gamma^{\blacksquare} \vdash N^{\blacksquare} : A^{\blacksquare}}{\Gamma^{\blacksquare} \vdash M^{\blacksquare}\ N^{\blacksquare} : B^{\blacksquare}}$$

$$\Box\text{-I}\ \frac{\Gamma^{\blacksquare}, \text{🔓} \vdash M^{\blacksquare} : A^{\blacksquare}}{\Gamma^{\blacksquare} \vdash \mathbf{box}\,\{M^{\blacksquare}\} : \Box A^{\blacksquare}}$$

$$\Box\text{-E}\ \frac{\Gamma^{\blacksquare} \vdash M^{\blacksquare} : \Box A^{\blacksquare} \qquad k \ \text{🔓} \text{ appear in } \Delta^{\blacksquare}}{\Gamma^{\blacksquare}, \Delta^{\blacksquare} \vdash \mathbf{unbox}_k M^{\blacksquare} : A^{\blacksquare}}$$

$$\begin{aligned} \mathbf{tail}(\epsilon) &= \epsilon \\ \mathbf{tail}(\Gamma^{\blacksquare}, x \colon A^{\blacksquare}) &= \mathbf{tail}(\Gamma^{\blacksquare}), x \colon A^{\blacksquare} \\ \mathbf{tail}(\Gamma^{\blacksquare}, \text{🔓}) &= \epsilon \end{aligned}$$

Fig. 10. Syntax and Typing Rules of $\lambda_{\Box_{S4}}$

2) Emedding of $\lambda_{\bigcirc}$:

We give the detailed definitions of $\lambda_{\bigcirc}$ in Figure 12.

Lemma C.2. *If $\Gamma^{\circ} @ k \rightsquigarrow_{\leq l} \Delta / \gamma_0 \ldots \gamma_l$, then $pos(\Delta) = \gamma_k$.*

Proof. By induction on the derivation of $\Gamma^{\circ} @ k \rightsquigarrow_{\leq l} \Delta / \gamma_0 \ldots \gamma_l$. □

Lemma C.3. *If the derivation $\Gamma_1^{\circ}, \Gamma_2^{\circ} @ k_1 \rightsquigarrow_{\leq l} \Delta_1, \Delta_2 / \gamma_0 \ldots \gamma_l$ includes another derivation of $\Gamma_1^{\circ} @ k_2 \rightsquigarrow_{\leq l} \Delta_1 / \delta_0 \ldots \delta_l$, then $\Delta_1, \Delta_2 \vdash \gamma_n \preceq \delta_n$ for all n such that $0 \leq n \leq l$.*

Proof. By induction on the derivation of $\Gamma_1^{\circ}, \Gamma_2^{\circ} @ k_1 \rightsquigarrow_{\leq l} \Delta_1, \Delta_2 / \gamma_0 \ldots \gamma_l$. □

Lemma C.4. *If $\Gamma^{\circ} @ k \rightsquigarrow_{\leq l} \Delta / \gamma_0 \ldots \gamma_l$, then $\Delta \vdash \gamma_n \sqsubseteq \gamma_{n+}$ for any n such that $0 \leq n < k$.*

Proof. By induction on the derivation of $\Gamma^{\circ} @ k \rightsquigarrow_{\leq l} \Delta / \gamma_0 \ldots \gamma_l$. □

Lemma C.5. *Supposing $\llparenthesis A^{\circ} \rrparenthesis^{\gamma_k \ldots \gamma_l}_{@k \leq l}$ is defined, $\llparenthesis A^{\circ} \rrparenthesis^{\gamma_k \ldots \gamma_l}_{@k \leq l} [\gamma_k {:=} \delta_k] \ldots [\gamma_l {:=} \delta_l] = \llparenthesis A^{\circ} \rrparenthesis^{\delta_k \ldots \delta_l}_{@k \leq l}$*

Type $\llparenthesis A^{\blacksquare} \rrparenthesis$

$$\begin{aligned}
\llparenthesis p \rrparenthesis &= p \\
\llparenthesis A^{\blacksquare} \to B^{\blacksquare} \rrparenthesis &= \llparenthesis A^{\blacksquare} \rrparenthesis \to \llparenthesis B^{\blacksquare} \rrparenthesis \\
\llparenthesis \Box A^{\blacksquare} \rrparenthesis &= [^{\succeq \mathbf{!}} \llparenthesis A^{\blacksquare} \rrparenthesis]
\end{aligned}$$

Terms $\llparenthesis M^{\blacksquare} \rrparenthesis^{\overrightarrow{\gamma}}$

$$\begin{aligned}
\llparenthesis x \rrparenthesis^{\overrightarrow{\gamma}} &= x \\
\llparenthesis \lambda x^{A^{\blacksquare}}.\, M^{\blacksquare} \rrparenthesis^{\overrightarrow{\gamma}, \delta_1} &= \lambda \boldsymbol{x} :{}^{\delta_2} \llparenthesis A^{\blacksquare} \rrparenthesis.\ \llparenthesis M^{\blacksquare} \rrparenthesis^{\overrightarrow{\gamma}, \delta_2} \\
&\quad \text{where } \delta_2 \text{ is fresh} \\
\llparenthesis M^{\blacksquare}\ N^{\blacksquare} \rrparenthesis^{\overrightarrow{\gamma}} &= \llparenthesis M^{\blacksquare} \rrparenthesis^{\overrightarrow{\gamma}}\ \llparenthesis N^{\blacksquare} \rrparenthesis^{\overrightarrow{\gamma}} \\
\llparenthesis \mathbf{box}\,\{M^{\blacksquare}\} \rrparenthesis^{\overrightarrow{\gamma}} &= \text{‘}\{^{\boldsymbol{\delta}: \succeq \mathbf{!}} \llparenthesis M^{\blacksquare} \rrparenthesis^{\overrightarrow{\gamma}, \delta}\} \\
&\quad \text{where } \delta \text{ is a fresh classifier} \\
\llparenthesis \mathbf{unbox}_k M^{\blacksquare} \rrparenthesis^{\overrightarrow{\gamma}, \delta_0 \ldots \delta_k} &= \boldsymbol{\sim}\{^{\delta_0} \llparenthesis M^{\blacksquare} \rrparenthesis^{\overrightarrow{\gamma}, \delta_0}\}
\end{aligned}$$

Contexts $\Gamma^{\blacksquare} \rightsquigarrow \Gamma / \overrightarrow{\gamma}$

$$\frac{}{\epsilon \rightsquigarrow \epsilon / \mathbf{!}}$$

$$\frac{\Gamma^{\blacksquare} \rightsquigarrow \Gamma / \overrightarrow{\gamma}, \delta_1 \qquad \delta_2 \text{ is fresh}}{\Gamma^{\blacksquare}, x\colon A^{\blacksquare} \rightsquigarrow \Gamma, \boldsymbol{x}{:}^{\boldsymbol{\delta_2}} \llparenthesis A^{\blacksquare} \rrparenthesis / \overrightarrow{\gamma}, \delta_2}$$

$$\frac{\Gamma^{\blacksquare} \rightsquigarrow \Gamma / \overrightarrow{\gamma} \qquad \delta \text{ is fresh}}{\Gamma^{\blacksquare}, \text{🔓} \rightsquigarrow \Gamma, \text{🔓}^{\boldsymbol{\delta}: \succeq \mathbf{!}} / \overrightarrow{\gamma}, \delta}$$

$$\frac{\Gamma^{\blacksquare} \rightsquigarrow \Gamma / \overrightarrow{\gamma}, \delta_0 \ldots \delta_k}{\mathbf{pop}(\Gamma^{\blacksquare}, k) \rightsquigarrow \Gamma, \text{🔒}^{\delta_0} / \overrightarrow{\gamma}, \delta_0}$$

$$\begin{aligned}
\mathbf{pop}(\Gamma^{\blacksquare}, 0) &= \Gamma^{\blacksquare} \\
\mathbf{pop}((\Gamma^{\blacksquare}, x\colon A^{\blacksquare}), k+) &= \mathbf{pop}(\Gamma^{\blacksquare}, k+) \\
\mathbf{pop}((\Gamma^{\blacksquare}, \text{🔓}), k+) &= \mathbf{pop}(\Gamma^{\blacksquare}, k)
\end{aligned}$$

Fig. 11. Translation from $\lambda_{\Box_{S4}}$ to BMTT

Proof. By induction on the structure of A°. □

Theorem 6 (On page 7)**.** *If* $\Gamma^\circ \vdash_k M^\circ : A^\circ$ *holds in* $\lambda_{\bigcirc}$, $\Gamma^\circ @k \rightsquigarrow_{\leq l} \Delta / \gamma_0 \ldots \gamma_l$ *holds and* l *is sufficiently large, then* $\Delta \vdash \llparenthesis M^\circ \rrparenthesis^{\gamma_0 \ldots \gamma_l}_{@k \leq l} : \llparenthesis A^\circ \rrparenthesis^{\gamma_{k+1} \ldots \gamma_l}_{@k \leq l}$ *holds in BMTT.*

Proof. By induction on the derivation of $\Gamma^\circ \vdash_k M^\circ : A^\circ$. We provide non-trivial cases for Var, $\to$-E and $\bigcirc$-E.

Case Var: By inversion, we have

$$\frac{}{\Gamma^\circ_1, x :^k A^\circ, \Gamma^\circ_2 \vdash_k x : A^\circ}$$

where $\Gamma^\circ = \Gamma^\circ_1, x :^k A^\circ, \Gamma^\circ_2$ and $M^\circ = x$. In the derivation of $\Gamma^\circ @k \rightsquigarrow_{\leq l} \Delta / \gamma_0 \ldots \gamma_l$, we have a following derivation by $\rightsquigarrow$-V, where $\Delta = \Delta_1, \boldsymbol{x}{:}^{\boldsymbol{\delta'}} (\!(A^\circ)\!)^{\delta_{k+} \ldots \delta_l}_{@k \leq l}, \Delta_2$ for some Δ_2.

$$\frac{\Gamma^\circ_1 @k \rightsquigarrow_{\leq l} \Delta_1 / \delta_0 \ldots \delta_l}{\Gamma^\circ_1, x :^k A^\circ @k \rightsquigarrow_{\leq l} \Delta_1, \boldsymbol{x}{:}^{\boldsymbol{\delta'}} (\!(A^\circ)\!)^{\delta_{k+} \ldots \delta_l}_{@k \leq l} / (\delta_0 \ldots \delta_l)[k \mapsto \delta']}$$

Variables	x, y		
Staging Level	k	$\in$	$\mathbb{N}$
Types	A°, B°	$::=$	$p \mid A^\circ \to B^\circ \mid \bigcirc A^\circ$
Terms	M°, N°	$::=$	$x \mid \lambda x^{A^\circ}.\, M^\circ \mid M^\circ\ N^\circ$
		$\mid$	$\mathbf{next}\,\{M^\circ\} \mid \mathbf{prev}\,\{M^\circ\}$
Context	$\Gamma^\circ, \Delta^\circ$	$::=$	$\epsilon \mid \Gamma^\circ, x :^k A^\circ$

$$\boxed{\Gamma^\circ \vdash_k M^\circ : A^\circ}$$

$$\text{Var}\ \frac{x :^k A^\circ \in \Gamma^\circ}{\Gamma^\circ \vdash_k x : A^\circ} \qquad \to\text{-I}\ \frac{\Gamma^\circ, x :^k A^\circ \vdash_k M^\circ : B^\circ}{\Gamma^\circ \vdash_k \lambda x^{A^\circ}.\, M^\circ : A^\circ \to B^\circ}$$

$$\to\text{-E}\ \frac{\Gamma^\circ \vdash_k M^\circ : A^\circ \to B^\circ \qquad \Gamma^\circ \vdash_k N^\circ : A^\circ}{\Gamma^\circ \vdash_k M^\circ\ N^\circ : B^\circ}$$

$$\bigcirc\text{-I}\ \frac{\Gamma^\circ \vdash_{k+1} M^\circ : A^\circ}{\Gamma^\circ \vdash_k \mathbf{next}\,\{M^\circ\} : \bigcirc A^\circ}$$

$$\bigcirc\text{-E}\ \frac{\Gamma^\circ \vdash_k M^\circ : \bigcirc A^\circ}{\Gamma^\circ \vdash_{k+1} \mathbf{prev}\,\{M^\circ\} : A^\circ}$$

$$\boxed{M^\circ{}_1 \Rightarrow_\beta M^\circ{}_2}$$

Axioms

$$(\lambda x^{A^\circ}.\, M^\circ{}_1)\ M^\circ{}_2 \Rightarrow_\beta M^\circ{}_1[x{:=}M^\circ{}_2] \qquad \beta\text{-}\to$$

$$\mathbf{prev}\,\{\mathbf{next}\,\{M^\circ\}\} \Rightarrow_\beta M^\circ \qquad \beta\text{-}\bigcirc$$

Compatibility

$$\frac{M^\circ{}_1 \Rightarrow_\beta M^\circ{}_2}{\lambda x^{A^\circ}.\, M^\circ{}_1 \Rightarrow_\beta \lambda x^{A^\circ}.\, M^\circ{}_2}$$

$$\frac{M^\circ{}_1 \Rightarrow_\beta M^\circ{}_2}{M^\circ{}_1\ N^\circ \Rightarrow_\beta M^\circ{}_2\ N^\circ} \qquad \frac{M^\circ{}_1 \Rightarrow_\beta M^\circ{}_2}{N^\circ\ M^\circ{}_1 \Rightarrow_\beta N^\circ\ M^\circ{}_2}$$

$$\frac{M^\circ{}_1 \Rightarrow_\beta M^\circ{}_2}{\mathbf{next}\,\{M^\circ{}_1\} \Rightarrow_\beta \mathbf{next}\,\{M^\circ{}_2\}}$$

$$\frac{M^\circ{}_1 \Rightarrow_\beta M^\circ{}_2}{\mathbf{prev}\,\{M^\circ{}_1\} \Rightarrow_\beta \mathbf{prev}\,\{M^\circ{}_2\}}$$

Fig. 12. Detailed Definition of $\lambda_{\bigcirc}$

For such δ', we have $\Delta \vdash \delta' \preceq \gamma_k$ from Lemma C.3. From C.2, we have $\text{pos}(\Delta) = \gamma_k$. Therefore, $\Delta \vdash \delta' \preceq \text{pos}(\Delta)$ holds, and we can use Var to derive $\Delta \vdash x\colon (\!(A^\circ)\!)^{\delta_{k+} \ldots \delta_l}_{@k \leq l}$. From the definition, $(\!(A^\circ)\!)^{\delta_{k+} \ldots \delta_l}_{@k \leq l} = \forall_{\boldsymbol{\delta''_{k+}} \geq \delta_{k+}} \ldots \forall_{\boldsymbol{\delta''_l} \geq \delta_l} \llparenthesis A^\circ \rrparenthesis^{\delta''_{k+} \ldots \delta''_l}_{@k \leq l}$. Also, $\Delta \vdash \delta_n \preceq \gamma_n$ for any n such that $0 \leq n \leq l$ from Lemma C.3. Hence, we can apply $\forall$-E as follows.

$$\frac{\begin{array}{c}\Delta \vdash x\colon \forall_{\boldsymbol{\delta''_{k+}} \geq \delta_{k+}} \ldots \forall_{\boldsymbol{\delta''_l} \geq \delta_l} \llparenthesis A^\circ \rrparenthesis^{\delta''_{k+} \ldots \delta''_l}_{@k \leq l} \\ \Delta \vdash \delta_n \preceq \gamma_n \text{ for any } n \text{ s.t. } k+ \leq n \leq l\end{array}}{\Delta \vdash x\gamma_{k+} \ldots \gamma_l\colon \llparenthesis A^\circ \rrparenthesis^{\delta''_{k+} \ldots \delta''_l}_{@k \leq l} [\delta''_{k+}{:=}\gamma_{k+}] \ldots [\delta''_l{:=}\gamma_l]}$$

We have $\llparenthesis A^\circ \rrparenthesis^{\delta''_{k+} \ldots \delta''_l}_{@k \leq l} [\delta''_{k+}{:=}\gamma_{k+}] \ldots [\delta''_l{:=}\gamma_l] = \llparenthesis A^\circ \rrparenthesis^{\gamma_{k+} \ldots \gamma_l}_{@k \leq l}$ from C.5. Therefore this is what we want.

Case $\to$-E: By inversion, we have

$$\frac{\Gamma^\circ \vdash_k {M^\circ}_1 : B^\circ \to A^\circ \qquad \Gamma^\circ \vdash_k {M^\circ}_2 : B^\circ}{\Gamma^\circ \vdash_k {M^\circ}_1\ {M^\circ}_2 : A^\circ}$$

where $M^\circ = {M^\circ}_1\ {M^\circ}_2$. For the first assumption, we apply the induction hypothesis and get $\Delta \vdash \llparenthesis {M^\circ}_1 \rrparenthesis^{\gamma_0 \cdots \gamma_l}_{@k \le l} \colon (\!(B^\circ)\!)^{\gamma_{k+} \cdots \gamma_l}_{@k \le l} \to \llparenthesis A^\circ \rrparenthesis^{\gamma_{k+} \cdots \gamma_l}_{@k \le l}$. For the second assumption, we first derive $\Gamma^\circ @k \rightsquigarrow_{\le l} \Delta, \boldsymbol{\gamma'_{k+}} {:\succeq}\, \gamma_{k+} \cdots \boldsymbol{\gamma'_l} {:\succeq}\, \gamma_l / \gamma_0 \dots \gamma_k, \gamma'_{k+} \cdots \gamma'_l$ by $\rightsquigarrow$-M. Then we use the inducion hypothesis with this context to get $\Delta, \boldsymbol{\gamma'_{k+}} {:\succeq}\, \gamma_{k+} \cdots \boldsymbol{\gamma'_l} {:\succeq}\, \gamma_l \vdash \llparenthesis {M^\circ}_2 \rrparenthesis^{\gamma_0 \cdots \gamma_k, \gamma'_{k+} \cdots \gamma'_l}_{@k \le l} : \llparenthesis B^\circ \rrparenthesis^{\gamma'_{k+} \cdots \gamma'_l}_{@k \le l}$. We apply $\forall$-I multiple times to get $\Delta \vdash (\!({M^\circ}_2)\!)^{\gamma_0 \cdots \gamma_l}_{@k \le l} : (\!(B^\circ)\!)^{\gamma_{k+} \cdots \gamma_l}_{@k \le l}$. Finally we apply $\to$-E to obtain $\Delta \vdash \llparenthesis {M^\circ}_1 \rrparenthesis^{\gamma_0 \cdots \gamma_l}_{@k \le l}\ (\!({M^\circ}_2)\!)^{\gamma_0 \cdots \gamma_l}_{@k \le l} : \llparenthesis A^\circ \rrparenthesis^{\gamma_{k+} \cdots \gamma_l}_{@k \le l}$, which is what we want.

Case $\bigcirc$-E: By inversion, we have

$$\frac{\Gamma^\circ \vdash_k M^{\circ\prime} : \bigcirc A^\circ}{\Gamma^\circ \vdash_k \textbf{prev}\,\{M^{\circ\prime}\} : A^\circ}$$

where $\textbf{prev}\,\{M^{\circ\prime}\} = M^\circ$. We use $\rightsquigarrow$-🔒 to derive $\Gamma^\circ @k- \rightsquigarrow_{\le l} \Delta, \text{🔒}^{\gamma_{k-}} / \gamma_0 \dots \gamma_l$. Here, we can confirm that $\vdash \Delta, \text{🔒}^{\gamma_{k-}} : \mathbf{ctx}$ by Lemma C.4. Then we apply the induction hypothesis to get $\Delta, \text{🔒}^{\gamma_{k-}} \vdash \llparenthesis M^{\circ\prime} \rrparenthesis^{\gamma_0 \cdots \gamma_l}_{@k- \le l} : [{\succeq}\gamma_k\, \llparenthesis A^\circ \rrparenthesis^{\gamma_{k+} \cdots \gamma_l}_{@k \le l}]$. As $\Delta, \text{🔒}^{\gamma_{k-}} \vdash \gamma_k \preceq \mathrm{pos}(\Delta)$ holds from C.2, and we can apply []-E to derive $\Delta \vdash \boldsymbol{\sim}\{^{\gamma_{k-}} \llparenthesis M^{\circ\prime} \rrparenthesis^{\gamma_0 \cdots \gamma_l}_{@k- \le l}\} : \llparenthesis A^\circ \rrparenthesis^{\gamma_{k+} \cdots \gamma_l}_{@k \le l}$, which is what we want. □

Lemma 8 (On page 7)**.** *If* $\Gamma^\circ \vdash_k M^\circ : A^\circ$ *and* $\Gamma^\circ @k \rightsquigarrow_{\le l} \Delta / \gamma_0 \dots \gamma_l$*, then* $|\llparenthesis M^\circ \rrparenthesis^{\gamma_0 \cdots \gamma_l}_{@k \le l}| = M^\circ$ *and* $|\llparenthesis A^\circ \rrparenthesis^{\gamma_{k+} \cdots \gamma_l}_{@k \le l}| = A^\circ$.

Proof. By induction on the structure of M° and A°. □

Lemma C.6. *Suppose* $\Gamma \vdash M_1 \colon A$ *and* $\Gamma \vdash M_2 \colon A$*. If* $|M_1| \Rrightarrow_\beta |M_2|$*, then* $M_1 \Rrightarrow^*_\beta M_2$*.*

Proof. By induction on the derivation of $|M_1| \Rrightarrow_\beta |M_2|$. We focus on the case for β-$\to$. β-$\bigcirc$ works in the similar manner, and other cases are straightforward.

Case β-$\to$: Let $|M_1| = (\lambda x^{B^\circ}. {N^\circ}_1)\ {N^\circ}_2$. Then $|M_2| = {N^\circ}_1[x{:=}{N^\circ}_2]$. M_1 has the form of $C[\lambda \boldsymbol{x} : {}^\gamma B. M_3]\ M_4$, where ${N^\circ}_1 = |M_3|$, ${N^\circ}_2 = |M_4|$, and C is a evaluation context that consists of only classifier abstraction and classifier applications. The condition $\Gamma \vdash M_1 \colon A$ ensures that $\Gamma \vdash C[\lambda \boldsymbol{x} : {}^\gamma B. M_3] \colon B \to A$. Hence, C consists of redexes that can be reduced. Therefore, we have $C[\lambda \boldsymbol{x} : {}^\gamma B. M_3] @\mathrm{pos}(\Gamma) \Rrightarrow^*_\beta \lambda \boldsymbol{x} : {}^\gamma B. M_3$, and $M_1 @\mathrm{pos}(\Gamma) \Rrightarrow^*_\beta M_3[\boldsymbol{\gamma}{:=}\mathrm{pos}(\Gamma), \boldsymbol{x}{:=}M_4]$. Here it is easy to confirm ${N^\circ}_1[x{:=}{N^\circ}_2] = |M_3[\boldsymbol{\gamma}{:=}\mathrm{pos}(\Gamma), \boldsymbol{x}{:=}M_4]|$, and we have $|M_3[\boldsymbol{\gamma}{:=}\mathrm{pos}(\Gamma), \boldsymbol{x}{:=}M_4]| = |M_2|$. Hence we get $M_1 \Rrightarrow^*_\beta M_2$ from the definition of $\Rrightarrow^*_\beta$. □

Theorem 7 (On page 7)**.** *Suppose* $\Gamma^\circ \vdash_k {M^\circ}_1 : A^\circ$ *and* $\Gamma^\circ @k \rightsquigarrow_{\le l} \Delta / \overrightarrow{\gamma}$ *hold. If* ${M^\circ}_1 \Rrightarrow_\beta {M^\circ}_2$*, then* $\llparenthesis {M^\circ}_1 \rrparenthesis^{\overrightarrow{\gamma}}_{@k \le l} \Rrightarrow^*_\beta \llparenthesis {M^\circ}_2 \rrparenthesis^{\gamma_0 \cdots \gamma_l}_{@k \le l}$.

Proof. Direct result from Theorem 6, Lemma 8 and Lemma C.6. □

D. Detailed Proofs of Section VI

Lemma 9 (Monotonicity (On page 9))**.** *Suppose* $w \preccurlyeq v$ *and* $\iota(d) \preceq_v e$*.*

1) *If* $w, d \Vdash^\rho A$*, then* $v, e \Vdash^{\iota(\rho)} A$*.*
2) *If* $w \Vdash^\rho \gamma_1 \preceq \gamma_2$*, then* $v \Vdash^{\iota(\rho)} \gamma_1 \preceq \gamma_2$*.*
3) *If* $w \Vdash^\rho \gamma_1 \sqsubseteq \gamma_2$*, then* $v \Vdash^{\iota(\rho)} \gamma_1 \sqsubseteq \gamma_2$*.*

Proof. (2) and (3) follows from monotonicity of ι. For (1), it suffices to check that for all $d' \succeq_w d$, if $w, d \vDash^\rho A$, then $w, d' \vDash^\rho A$. We proceed by induction on A.

Case $A \equiv p$: Follows from that $V_w(p)$ is upward-closed.

Case $A \equiv B \to C$: By definition.

Case $A \equiv [{\succeq}\gamma\, B]$: Suppose $w, d \vDash^\rho [{\succeq}\gamma\, B]$ and $d \preceq_w d'$. Take $e \sqsupseteq_w d'$ such that $e \succeq_w \rho(\gamma)$. By left-stability we have $d \sqsubseteq_w e$, so that $w, e \Vdash^\rho B$ since $w, d \vDash^\rho [{\succeq}\gamma\, B]$, which implies $w, e \vDash^\rho [{\succeq}\gamma\, B]$.

Case $A \equiv \forall \boldsymbol{\gamma_2} :\succeq \gamma_1. B$: Immediate from the IH. □

Corollary D.1. *If* $w \preccurlyeq v$ *and* $w \Vdash^\rho \Gamma$*, then* $v \Vdash^{\iota(\rho)} \Gamma$*.*

Lemma D.1 (Substitution)**.** $w, d \Vdash^{\rho \cdot [\gamma_2 \mapsto \rho(\gamma_1)]} A$ *iff* $w, d \Vdash^\rho A[\boldsymbol{\gamma_2} \coloneqq \gamma_1]$*.*

Proof. By induction on A. □

Theorem 8 (Kripke soundness (On page 9))**.**

1) *If* $\Gamma \vdash A$*, then* $\Gamma \Vdash A$*.*
2) *If* $\Gamma \vdash \gamma_1 \preceq \gamma_2$*, then* $\Gamma \Vdash \gamma_1 \preceq \gamma_2$*.*
3) *If* $\Gamma \vdash \gamma_1 \sqsubseteq \gamma_2$*, then* $\Gamma \Vdash \gamma_1 \sqsubseteq \gamma_2$*.*

Proof. (2) and (3) are mostly straightforward, so we show (1) here. We proceed by induction on derivation.

Suppose $\Gamma \vdash A$. Assuming $w \Vdash^\rho \Gamma$, we show w, $\mathrm{pos}(\Gamma) \Vdash^\rho A$. We analyze the last rule of the derivation:

Case Var: Assume

$$\frac{\boldsymbol{x} :^\gamma A \in \Gamma \qquad \Gamma \vdash \gamma \preceq \mathrm{pos}(\Gamma)}{\Gamma \vdash A}$$

By assumption we have $w, \rho(\gamma) \Vdash^\rho A$, and from (2), $\rho(\gamma) \preceq_w \rho(\mathrm{pos}(\Gamma))$ holds. By Lemma 9 we see $w, \rho(\mathrm{pos}(\Gamma)) \Vdash^\rho A$.

Case $\to$-I: Assume

$$\frac{\Gamma, \boldsymbol{x} :^\gamma B \vdash C \qquad \gamma \notin \mathbf{FC}(C)}{\Gamma \vdash B \to C}$$

Take $v \succcurlyeq w$ and $d \succeq_v \iota(\rho(\mathrm{pos}(\Gamma)))$, and suppose $v, d \Vdash^{\iota(\rho)} B$. By Corollary D.1 we have $v \Vdash^{\iota(\rho)} \Gamma$, so letting $\rho' = \iota(\rho) \cdot [\gamma \mapsto d]$ we obtain $v \Vdash^{\rho'} \Gamma, \boldsymbol{x} :^\gamma B$. By the IH, $v, d \Vdash^{\rho'} C$ holds, and so does $v, d \Vdash^{\iota(\rho)} C$ as $\gamma \notin \mathbf{FC}(C)$, which implies $w, \rho(\mathrm{pos}(\Gamma)) \Vdash^\rho B \to C$.

Case $\to$-E: Assume

$$\frac{\Gamma \vdash B \to C \qquad \Gamma \vdash B}{\Gamma \vdash C}$$

By the IH, we have $w, \rho(\mathrm{pos}(\Gamma)) \Vdash^\rho B \to C$ and w, $\rho(\mathrm{pos}(\Gamma)) \Vdash^\rho B$. Since $w \preccurlyeq w$ and $\rho(\mathrm{pos}(\Gamma)) \preceq_w \rho(\mathrm{pos}(\Gamma))$, we obtain $w, \rho(\mathrm{pos}(\Gamma)) \Vdash^\rho C$.

Case []-I: Assume

$$\frac{\Gamma, \faUnlock^{\gamma_2 : \succeq \gamma_1} \vdash B \qquad \gamma_2 \notin \mathbf{FC}(B)}{\Gamma \vdash [\succeq^{\gamma_1} B]}$$

Take $v \succcurlyeq w$ and $d \sqsupseteq_v \iota(\rho(\mathrm{pos}(\Gamma)))$. By Corollary D.1 we have $v \Vdash^{\iota(\rho)} \Gamma$, so letting $\rho' = \iota(\rho) \cdot [\gamma_2 \mapsto d]$ we obtain $v \Vdash^{\rho'} \Gamma, \faUnlock^{\gamma_2 : \succeq \gamma_1}$. By the IH, $v, d \Vdash^{\rho'} B$ holds, and so does $v, d \Vdash^{\iota(\rho)} B$ as $\gamma_2 \notin \mathbf{FC}(B)$, which implies $w, \rho(\mathrm{pos}(\Gamma)) \Vdash^{\rho} [\succeq^{\gamma_1} B]$.

Case []-E: Assume

$$\frac{\Gamma, \faLock^{\gamma} \vdash [\succeq^{\gamma_1} B] \qquad \Gamma \vdash \gamma_1 \preceq \mathrm{pos}(\Gamma)}{\Gamma \vdash B}$$

From $w \Vdash^{\rho} \Gamma$, we have $w \Vdash^{\rho} \Gamma, \faLock^{\gamma}$, and by (2), also $\rho(\gamma_1) \preceq_w \rho(\mathrm{pos}(\Gamma))$. Since $\Gamma, \faLock^{\gamma}$ is well-formed, there should be a subderivation of $\Gamma \vdash \gamma \sqsubseteq \mathrm{pos}(\Gamma)$, which gives $\rho(\gamma) \sqsubseteq_w \rho(\mathrm{pos}(\Gamma))$ by (3). Applying the IH to $\Gamma, \faLock^{\gamma} \vdash [\succeq^{\gamma_1} B]$, we have $w, \rho(\gamma) \Vdash^{\rho} [\succeq^{\gamma_1} B]$, and from $w \preccurlyeq w$, we see $w, \rho(\mathrm{pos}(\Gamma)) \Vdash^{\rho} B$.

Case $\forall$-I: Assume

$$\frac{\Gamma, \boldsymbol{\gamma_2} :\succeq \gamma_1 \vdash B}{\Gamma \vdash \forall \boldsymbol{\gamma_2} :\succeq \gamma_1 .\, B}$$

Take $v \succcurlyeq w$ and $d \succeq_v \iota(\rho(\gamma_1))$. By Corollary D.1 we have $v \Vdash^{\iota(\rho)} \Gamma$, so letting $\rho' = \iota(\rho) \cdot [\gamma_2 \mapsto d]$ we obtain $v \Vdash^{\rho'} \Gamma, \boldsymbol{\gamma_2} :\succeq \gamma_1$. By the IH, $v, \iota(\rho(\mathrm{pos}(\Gamma))) \Vdash^{\rho'} B$ holds, which implies $w, \rho(\mathrm{pos}(\Gamma)) \Vdash^{\rho} \forall \boldsymbol{\gamma_2} :\succeq \gamma_1 .\, B$.

Case $\forall$-E: Assume

$$\frac{\Gamma \vdash \forall \boldsymbol{\gamma_2} :\succeq \gamma_1 .\, B \qquad \Gamma \vdash \gamma_1 \preceq \gamma}{\Gamma \vdash B[\boldsymbol{\gamma_2} := \gamma]}$$

By the IH, we have $w, \rho(\mathrm{pos}(\Gamma)) \Vdash^{\rho} \forall \boldsymbol{\gamma_2} :\succeq \gamma_1 .\, B$, and from (2), $\rho(\gamma_1) \preceq_w \rho(\gamma)$ holds. Since $w \preccurlyeq w$, we obtain $w, \rho(\mathrm{pos}(\Gamma)) \Vdash^{\rho \cdot [\gamma_2 \mapsto \rho(\gamma)]} B$, and Lemma D.1 yields $w, \rho(\mathrm{pos}(\Gamma)) \Vdash^{\rho} B[\boldsymbol{\gamma_2} := \gamma]$. ☐

Lemma D.2. *The following rules are admissible:*

1) *Inversion of* []-I*:*

$$\frac{\Gamma \vdash [\succeq^{\gamma} A]}{\Gamma, \faUnlock^{\boldsymbol{\gamma'} : \succeq \gamma} \vdash A}$$

2) *General weakening for top-level subderivations:*

$$\frac{\Gamma, \faLock^{!}, \Gamma' \vdash A}{\Gamma, \Delta, \faLock^{!}, \Gamma' \vdash A} \qquad \frac{\Gamma, \faLock^{!}, \Gamma' \vdash \gamma_1 \trianglelefteq \gamma_2}{\Gamma, \Delta, \faLock^{!}, \Gamma' \vdash \gamma_1 \trianglelefteq \gamma_2}$$

where $\trianglelefteq \in \{\preceq, \sqsubseteq\}$.

Proof.

1) Using Theorem 2, we have

$$[]\text{-E}\ \frac{\dfrac{\Gamma \vdash [\succeq^{\gamma} A]}{\Gamma, \faUnlock^{\boldsymbol{\gamma'} : \succeq \gamma}, \faLock^{\mathrm{pos}(\Gamma)} \vdash [\succeq^{\gamma} A]}}{\Gamma, \faUnlock^{\boldsymbol{\gamma'} : \succeq \gamma} \vdash A}$$

2) By induction on derivation. ☐

Lemma 11 (Truth lemma (On page 10))**.** $\Gamma, \gamma \Vdash^{\mathrm{c}} A$ *iff* $\Gamma, \faLock^{!} \vdash [\succeq^{\gamma} A]$.

Proof. By induction on the size of A, using Lemma D.2.

Case $A \equiv p$: By definition.

Case $A \equiv B \to C$:

($\Longleftarrow$): Suppose $\Gamma, \faLock^{!} \vdash [\succeq^{\gamma} B \to C]$. Take $\Delta \succcurlyeq^{\mathrm{c}} \Gamma$ and $\delta \succeq^{\mathrm{c}}_{\Delta} \gamma$ satisfying $\Delta, \delta \Vdash^{\rho^{\mathrm{c}}_{\Gamma}} B$. Since $\rho^{\mathrm{c}}_{\Delta} \restriction_{\mathbf{Dom}_{\mathrm{C}}(\Gamma)} = \rho^{\mathrm{c}}_{\Gamma}$, we have $\Delta, \delta \Vdash^{\mathrm{c}} B$, and the IH yields $\Delta, \faLock^{!} \vdash [\succeq^{\delta} B]$. Then we can derive

$$\to\text{-E}\ \frac{\dfrac{\dfrac{\Gamma, \faLock^{!} \vdash [\succeq^{\gamma} B \to C]}{\Delta, \faLock^{!} \vdash [\succeq^{\gamma} B \to C]}}{\Delta, \faLock^{!}, \faUnlock^{\boldsymbol{\delta'} : \succeq \delta} \vdash B \to C} \qquad \dfrac{\Delta, \faLock^{!} \vdash [\succeq^{\delta} B]}{\Delta, \faLock^{!}, \faUnlock^{\boldsymbol{\delta'} : \succeq \delta} \vdash B}}{[]\text{-I}\ \dfrac{\Delta, \faLock^{!}, \faUnlock^{\boldsymbol{\delta'} : \succeq \delta} \vdash C}{\Delta, \faLock^{!} \vdash [\succeq^{\delta} C]}}$$

By the IH, $\Delta, \delta \Vdash^{\mathrm{c}} C$ holds, and so does $\Delta, \delta \Vdash^{\rho^{\mathrm{c}}_{\Gamma}} C$, which implies $\Gamma, \gamma \Vdash^{\mathrm{c}} B \to C$.

($\Longrightarrow$): We proceed by contrapositive. Suppose $\Gamma, \faLock^{!} \nvdash [\succeq^{\gamma} B \to C]$. Let $\Delta \equiv \Gamma, \faLock^{!}, \faUnlock^{\boldsymbol{\gamma'} : \succeq \gamma}, \boldsymbol{x} :^{\boldsymbol{\delta}} B$. Then we must have $\Delta, \faLock^{!} \nvdash [\succeq^{\delta} C]$; otherwise we could derive

$$[]\text{-I}\ \frac{\to\text{-I}\ \dfrac{[]\text{-E}\ \dfrac{\Delta, \faLock^{!} \vdash [\succeq^{\delta} C]}{\Gamma, \faLock^{!}, \faUnlock^{\boldsymbol{\gamma'} : \succeq \gamma}, \boldsymbol{x} :^{\boldsymbol{\delta}} B \vdash C}}{\Gamma, \faLock^{!}, \faUnlock^{\boldsymbol{\gamma'} : \succeq \gamma} \vdash B \to C}}{\Gamma, \faLock^{!} \vdash [\succeq^{\gamma} B \to C]}$$

a contradiction. By the IH, we have $\Delta, \delta \nVdash^{\mathrm{c}} C$, and also $\Delta, \delta \nVdash^{\rho^{\mathrm{c}}_{\Gamma}} C$ as $\mathbf{FC}(C) \subseteq \mathbf{Dom}_{\mathrm{C}}(\Gamma)$. Applying a similar argument to

$$[]\text{-I}\ \frac{\text{Var}\ \dfrac{}{\Delta, \faLock^{!}, \faUnlock^{\boldsymbol{\delta'} : \succeq \delta} \vdash B}}{\Delta, \faLock^{!} \vdash [\succeq^{\delta} B]}$$

yields $\Delta, \delta \Vdash^{\rho^{\mathrm{c}}_{\Gamma}} B$. Since $\Delta \succcurlyeq^{\mathrm{c}} \Gamma$ and $\delta \succeq^{\mathrm{c}}_{\Delta} \gamma$, we see $\Gamma, \gamma \nVdash^{\mathrm{c}} B \to C$.

Case $A \equiv [\succeq^{\gamma_1} B]$:

($\Longleftarrow$): Suppose $\Gamma, \faLock^{!} \vdash [\succeq^{\gamma} [\succeq^{\gamma_1} B]]$. Take $\Delta \succcurlyeq^{\mathrm{c}} \Gamma$ and $\delta \sqsupseteq^{\mathrm{c}}_{\Delta} \gamma$ satisfying $\delta \succeq^{\mathrm{c}}_{\Delta} \gamma_1$. Then we can derive

$$[]\text{-I}\ \frac{[]\text{-E}\ \dfrac{[]\text{-E}\ \dfrac{\dfrac{\Gamma, \faLock^{!} \vdash [\succeq^{\gamma} [\succeq^{\gamma_1} B]]}{\Delta, \faLock^{!}, \faUnlock^{\boldsymbol{\delta'} : \succeq \delta}, \faLock^{\gamma}, \faLock^{!} \vdash [\succeq^{\gamma} [\succeq^{\gamma_1} B]]}}{\Delta, \faLock^{!}, \faUnlock^{\boldsymbol{\delta'} : \succeq \delta}, \faLock^{\gamma} \vdash [\succeq^{\gamma_1} B]}}{\Delta, \faLock^{!}, \faUnlock^{\boldsymbol{\delta'} : \succeq \delta} \vdash B}}{\Delta, \faLock^{!} \vdash [\succeq^{\delta} B]}$$

By the IH, $\Delta, \delta \Vdash^{\mathrm{c}} B$ holds, and so does $\Delta, \delta \Vdash^{\rho^{\mathrm{c}}_{\Gamma}} B$, which implies $\Gamma, \gamma \Vdash^{\mathrm{c}} [\succeq^{\gamma_1} B]$.

($\Longrightarrow$): We proceed by contrapositive. Suppose $\Gamma, \faLock^{!} \nvdash [\succeq^{\gamma} [\succeq^{\gamma_1} B]]$. Let $\Delta \equiv \Gamma, \faLock^{!}, \faUnlock^{\boldsymbol{\gamma'} : \succeq \gamma}, \faUnlock^{\boldsymbol{\delta} : \succeq \gamma_1}$. Then we must have $\Delta, \faLock^{!} \nvdash [\succeq^{\delta} B]$; otherwise we could derive

$$[]\text{-I}\ \frac{[]\text{-I}\ \dfrac{[]\text{-E}\ \dfrac{\Delta, \faLock^{!} \vdash [\succeq^{\delta} B]}{\Gamma, \faLock^{!}, \faUnlock^{\boldsymbol{\gamma'} : \succeq \gamma}, \faUnlock^{\boldsymbol{\delta} : \succeq \gamma_1} \vdash B}}{\Gamma, \faLock^{!}, \faUnlock^{\boldsymbol{\gamma'} : \succeq \gamma} \vdash [\succeq^{\gamma_1} B]}}{\Gamma, \faLock^{!} \vdash [\succeq^{\gamma} [\succeq^{\gamma_1} B]]}$$

a contradiction. By the IH, we have $\Delta, \delta \nVdash^{\mathrm{c}} B$ and hence $\Delta, \delta \nVdash^{\rho^{\mathrm{c}}_{\Gamma}} B$. As $\Delta \succcurlyeq^{\mathrm{c}} \Gamma$ and $\delta \sqsupseteq^{\mathrm{c}}_{\Delta} \gamma$ with $\delta \succeq^{\mathrm{c}}_{\Delta} \gamma_1$, we see $\Gamma, \gamma \nVdash^{\mathrm{c}} [\succeq^{\gamma_1} B]$.

Case $A \equiv \forall \boldsymbol{\gamma_2} :\succeq \gamma_1 .\, B$:

($\Longleftarrow$): Suppose $\Gamma, \text{🔒}^{!} \vdash [{\succeq}^{\gamma} \forall \boldsymbol{\gamma_2} :\succeq \gamma_1 .\, B]$. Take $\Delta \succcurlyeq^{\mathrm{c}} \Gamma$ and $\delta \succeq^{\mathrm{c}}_{\Delta} \gamma_1$. Then we can derive

$$\begin{array}{rc}
 & \Gamma, \text{🔒}^{!} \vdash [{\succeq}^{\gamma} \forall \boldsymbol{\gamma_2} :\succeq \gamma_1 .\, B] \\ \hdashline
 & \Delta, \text{🔒}^{!} \vdash [{\succeq}^{\gamma} \forall \boldsymbol{\gamma_2} :\succeq \gamma_1 .\, B] \\ \hdashline
 & \Delta, \text{🔒}^{!}, \text{🔓}^{\gamma' :\succeq \gamma} \vdash \forall \boldsymbol{\gamma_2} :\succeq \gamma_1 .\, B \\ \hline
\forall\text{-E} & \Delta, \text{🔒}^{!}, \text{🔓}^{\gamma' :\succeq \gamma} \vdash B[\boldsymbol{\gamma_2} := \delta] \\ \hline
[]\text{-I} & \Delta, \text{🔒}^{!} \vdash [{\succeq}^{\gamma} B[\boldsymbol{\gamma_2} := \delta]]
\end{array}$$

By the IH, $\Delta, \gamma \Vdash^{\mathrm{c}} B[\boldsymbol{\gamma_2} := \delta]$ holds, and so does $\Delta, \gamma \Vdash^{\rho^{\mathrm{c}}_{\Gamma} \cdot [\gamma_2 \mapsto \delta]} B$, which yields $\Gamma, \gamma \Vdash^{\mathrm{c}} \forall \boldsymbol{\gamma_2} :\succeq \gamma_1 .\, B$.

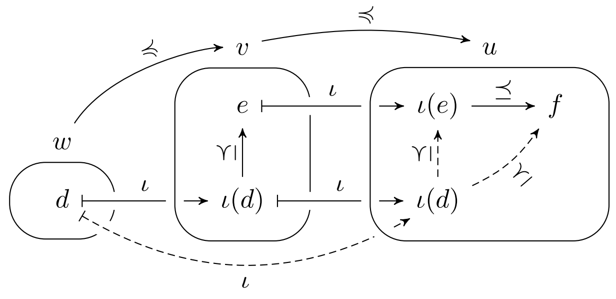
($\Longrightarrow$): We proceed by contrapositive. Suppose $\Gamma, \text{🔒}^{!} \nvdash [{\succeq}^{\gamma} \forall \boldsymbol{\gamma_2} :\succeq \gamma_1 .\, B]$. Let $\Delta \equiv \Gamma, \text{🔒}^{!}, \text{🔓}^{\gamma' :\succeq \gamma}, \boldsymbol{\gamma_2} :\succeq \gamma_1$. Then we must have $\Delta, \text{🔒}^{!} \nvdash \forall \boldsymbol{\gamma_2} :\succeq \gamma_1 .\, B$; otherwise we could derive

$$\begin{array}{rc}
 & \Delta, \text{🔒}^{!} \vdash [{\succeq}^{\gamma} B] \\ \hline
[]\text{-E} & \Gamma, \text{🔒}^{!}, \text{🔓}^{\gamma' :\succeq \gamma}, \boldsymbol{\gamma_2} :\succeq \gamma_1 \vdash B \\ \hline
\forall\text{-I} & \Gamma, \text{🔒}^{!}, \text{🔓}^{\gamma' :\succeq \gamma} \vdash \forall \boldsymbol{\gamma_2} :\succeq \gamma_1 .\, B \\ \hline
[]\text{-I} & \Gamma, \text{🔒}^{!} \vdash [{\succeq}^{\gamma} \forall \boldsymbol{\gamma_2} :\succeq \gamma_1 .\, B]
\end{array}$$

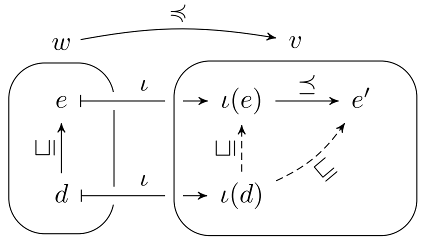
a contradiction. By the IH, we have $\Delta, \gamma \nVdash^{\mathrm{c}} B$, and hence $\Delta, \gamma \nVdash^{\rho^{\mathrm{c}}_{\Gamma} \cdot [\gamma_2 \mapsto \gamma_2]} B$. Since $\Delta \succcurlyeq^{\mathrm{c}} \Gamma$, we see $\Gamma, \gamma \Vdash^{\mathrm{c}} \forall \boldsymbol{\gamma_2} :\succeq \gamma_1 .\, B$. ☐

Lemma 14 (On page 10). *$\mathfrak{M}_{+}$ is a* **CS4***-model.*

Proof. Most of the conditions are straightforward. The transitivity of $\preceq_{+}$ follows from that of $\preceq_u$: if $\langle w, d\rangle \preceq_{+} \langle v, e\rangle$ and $\langle v, e\rangle \preceq_{+} \langle u, f\rangle$, then $\langle w, d\rangle \preceq_{+} \langle u, f\rangle$:



Left-persistency follows from right-stability: if $\langle w, d\rangle\ R_{+}\ \langle w, e\rangle \preceq_{+} \langle v, e'\rangle$, then $\langle w, d\rangle \preceq_{+} \langle v, \iota(d)\rangle\ R_{+}\ \langle v, e'\rangle$:



☐

Lemma 13 (On page 10). *$M^w, v \vDash_{\mathbf{CS4}} |A|^{\Box}$ iff $M^w_*, *, v \Vdash^{[! \mapsto w]} A$.*

Proof. By induction on A. There are three cases:

Case $A \equiv p$:

$$\begin{aligned}
M^w, v \vDash_{\mathbf{CS4}} |p|^{\Box} &\iff |p|^{\Box} \in V(v) \\
&\iff p \in V_*(v) \\
&\iff M^w_*, *, v \Vdash^{[! \mapsto w]} p
\end{aligned}$$

Case $A \equiv B \to C$:

$$\begin{aligned}
& M^w, v \vDash_{\mathbf{CS4}} |B \to C|^{\Box} \\
\iff & \forall u \succeq v. \begin{cases} M^w, u \vDash_{\mathbf{CS4}} |B|^{\Box} \\ \quad \implies M^w, u \vDash_{\mathbf{CS4}} |C|^{\Box} \end{cases} \\
\iff & \forall u \succeq_* v. \begin{cases} M^w_*, *, u \Vdash^{[! \mapsto w]} B \\ \quad \implies M^w_*, *, u \Vdash^{[! \mapsto w]} C \end{cases} && \text{(IH)} \\
\iff & M^w_*, *, v \Vdash^{[! \mapsto w]} B \to C
\end{aligned}$$

Case $A \equiv [{\succeq}^{!} B]$:

$$\begin{aligned}
& M^w, v \vDash_{\mathbf{CS4}} |[{\succeq}^{!} B]|^{\Box} \\
\iff & \forall v' \succeq v. \forall u \in R(v').\, \big(M^w, u \vDash_{\mathbf{CS4}} |B|^{\Box}\big) \\
\iff & \forall u \sqsupseteq_* v.\, \big(M^w, u \vDash_{\mathbf{CS4}} |B|^{\Box}\big) \\
\iff & \forall u \sqsupseteq_* v.\, \big(M^w *, u \Vdash^{[! \mapsto w]} B\big) && \text{(IH)} \\
\iff & M^w_*, v \Vdash^{[! \mapsto w]} [{\succeq}^{!} B]
\end{aligned}$$

☐

Lemma 15 (On page 10). *$\mathfrak{M}, w, d \Vdash^{\rho} A$ iff $\mathfrak{M}_{+}, \langle w, d\rangle \vDash_{\mathbf{CS4}} |A|^{\Box}$.*

Proof. By induction on A. There are three cases:

Case $A \equiv p$:

$$\begin{aligned}
\mathfrak{M}, w, d \Vdash^{\rho} p &\iff \forall v \succcurlyeq w.\, \iota(d) \in V_v(p) \\
&\iff d \in V_w(p) \\
&\iff \langle w, d\rangle \in V_{+}(p) \\
&\iff \mathfrak{M}_{+}, \langle w, d\rangle \vDash_{\mathbf{CS4}} |p|^{\Box}
\end{aligned}$$

Case $A \equiv B \to C$:

$$\begin{aligned}
& \mathfrak{M}, w, d \Vdash^{\rho} B \to C \\
\iff & \forall v \succcurlyeq w. \forall e \succeq_v \iota(d). \begin{cases} \mathfrak{M}, v, e \Vdash^{\iota(\rho)} B \\ \quad \implies \mathfrak{M}, v, e \Vdash^{\iota(\rho)} C \end{cases} \\
\iff & \forall v \succcurlyeq w. \forall e \succeq_v \iota(d). \begin{cases} \mathfrak{M}_{+}, \langle v, e\rangle \vDash_{\mathbf{CS4}} |B|^{\Box} \\ \quad \implies \mathfrak{M}_{+}, \langle v, e\rangle \vDash_{\mathbf{CS4}} |C|^{\Box} \end{cases} && \text{(IH)} \\
\iff & \forall \langle v, e\rangle \succeq_{+} \langle w, d\rangle. \begin{cases} \mathfrak{M}_{+}, \langle v, e\rangle \vDash_{\mathbf{CS4}} |B|^{\Box} \\ \quad \implies \mathfrak{M}_{+}, \langle v, e\rangle \vDash_{\mathbf{CS4}} |C|^{\Box} \end{cases} \\
\iff & \mathfrak{M}_{+}, \langle w, d\rangle \vDash_{\mathbf{CS4}} |B \to C|^{\Box}
\end{aligned}$$

Case $A \equiv [{\succeq}^{!} B]$:

$$\begin{aligned}
& \mathfrak{M}, w, d \Vdash^{\rho} [{\succeq}^{!} B] \\
\iff & \forall v \succcurlyeq w. \forall e \sqsupseteq_v \iota(d).\, \Big(\mathfrak{M}, v, e \Vdash^{\iota(\rho)} B\Big) \\
\iff & \forall v \succcurlyeq w. \forall e \sqsupseteq_v \iota(d).\, \Big(\mathfrak{M}_{+}, \langle v, e\rangle \vDash_{\mathbf{CS4}} |B|^{\Box}\Big) && \text{(IH)} \\
\iff & \forall \langle v, d'\rangle \succeq_{+} \langle w, d\rangle. \\
& \quad \forall \langle v, e\rangle \in R_{+}(\langle v, d'\rangle).\, \Big(\mathfrak{M}_{+}, \langle v, e\rangle \vDash_{\mathbf{CS4}} |B|^{\Box}\Big) && (\dagger) \\
\iff & \mathfrak{M}_{+}, \langle w, d\rangle \vDash_{\mathbf{CS4}} |[{\succeq}^{!} B]|^{\Box}
\end{aligned}$$

where the left-to-right direction of (†) follows from left-stability:

$$\begin{tikzcd}
 & u & d' \arrow[r, "\sqsubseteq"] & e \\
v \arrow[ur, bend left, "\preccurlyeq"] & d \arrow[r, mapsto, "\iota"] & \iota(d) \arrow[u, "\gamma|"] \arrow[ur, dashed, "\sqsubseteq"'] &
\end{tikzcd}$$

□

E. Detailed Proofs of Section VII

Lemma E.1 (SN-closure)**.**

1) *If* $N \in \mathrm{SN}$ *and* $K[M[\boldsymbol{\gamma} \coloneqq \gamma', \boldsymbol{x} \coloneqq N]] \in \mathrm{SN}$*, then* $K[(\lambda \boldsymbol{x} :^{\gamma} A.\, M)N] \in \mathrm{SN}$.
2) *If* $K[M[\boldsymbol{\gamma_2} \coloneqq \gamma']] \in \mathrm{SN}$*, then* $K[\sim\{^{\gamma_3}\, {}^{\backprime}\{^{\boldsymbol{\gamma_2} :\succeq \gamma_1} M\}\}] \in \mathrm{SN}$.
3) *If* $K[M[\boldsymbol{\gamma_2} \coloneqq \gamma_3]] \in \mathrm{SN}$*, then* $K[(\lambda \boldsymbol{\gamma_2} :\succeq \gamma_1.\, M)\gamma_3] \in \mathrm{SN}$.

Proof. By contrapositive. □

Lemma E.2 (Reducibility-closure)**.** *The following rules are admissible:*

$$\mathcal{E}\text{-}{\to}\ \frac{\forall \Delta \succcurlyeq \Gamma.\ \begin{pmatrix} \Delta \vdash N \in \mathcal{E}[\![A]\!] \implies \\ \Delta \vdash M[\boldsymbol{\gamma} \coloneqq \mathrm{pos}(\Delta), \boldsymbol{x} \coloneqq N] \in \mathcal{E}[\![B]\!] \end{pmatrix}}{\Gamma \vdash \lambda \boldsymbol{x} :^{\gamma} A.\, M \in \mathcal{E}[\![A \to B]\!]}$$

$$\mathcal{E}\text{-}[]\ \frac{\forall \big(\Delta, \text{🔒}^{\delta}\big) \succcurlyeq \Gamma.\ \begin{pmatrix} \Delta \vdash \delta \sqsubseteq \mathrm{pos}(\Delta) \\ \&\ \ \Delta \vdash \gamma_1 \preceq \mathrm{pos}(\Delta) \implies \\ \Delta \vdash M[\boldsymbol{\gamma_2} \coloneqq \mathrm{pos}(\Delta)] \in \mathcal{E}[\![A]\!] \end{pmatrix}}{\Gamma \vdash {}^{\backprime}\{^{\boldsymbol{\gamma_2} :\succeq \gamma_1} M\} \in \mathcal{E}[\![[\succeq \gamma_1\, A]]\!]}$$

$$\mathcal{E}\text{-}\forall\ \frac{\forall \Delta \succcurlyeq \Gamma.\ \begin{pmatrix} \Delta \vdash \gamma_1 \preceq \gamma \implies \\ \Delta \vdash M[\boldsymbol{\gamma_2} \coloneqq \gamma] \in \mathcal{E}[\![A[\boldsymbol{\gamma_2} \coloneqq \gamma]]\!] \end{pmatrix}}{\Gamma \vdash \lambda \boldsymbol{\gamma_2} :\succeq \gamma_1.\, M \in \mathcal{E}[\![\forall \boldsymbol{\gamma_2} :\succeq \gamma_1.\, A]\!]}$$

Proof. By simply checking the condition of $\mathcal{E}$-blur.

Proof of $\mathcal{E}$*-*$\to$*:* Suppose $\Delta \succcurlyeq \Gamma$. Take $\Delta \vdash K \in \mathcal{K}[\![A \to B]\!]$ to show $K[\lambda \boldsymbol{x} :^{\gamma} A.\, M] \in \mathrm{SN}$. For the last rule of the derivation of K, there are two possibilities:

Case $\mathcal{K}$-id: We may assume without loss of generality that $x \notin \mathbf{Dom}_{\mathrm{V}}(\Delta)$ and $\gamma \notin \mathbf{Dom}_{\mathrm{C}}(\Delta)$. Let $\Delta' \equiv \Delta, \boldsymbol{x} :^{\gamma} A$. Then we have $\Gamma \preccurlyeq \Delta'$ and $\Delta' \vdash x \in \mathcal{E}[\![A]\!]$. By assumption, we obtain $\Delta' \vdash M[\boldsymbol{\gamma} \coloneqq \gamma, \boldsymbol{x} \coloneqq x] \equiv M \in \mathcal{E}[\![B]\!]$, and Lemma 16 yields $M \in \mathrm{SN}$. Thus, $\lambda \boldsymbol{x} :^{\gamma} A.\, M \in \mathrm{SN}$.

Case $\mathcal{K}$-$\to$: Then we have $K \equiv K'[[-]N]$ for some $\Delta \vdash N \in \mathcal{E}[\![A]\!]$ and $\Delta \vdash K' \in \mathcal{K}[\![B]\!]$. By assumption we have $\Delta \vdash M[\boldsymbol{\gamma} \coloneqq \mathrm{pos}(\Delta), \boldsymbol{x} \coloneqq N] \in \mathcal{E}[\![B]\!]$, and hence $K'[M[\boldsymbol{\gamma} \coloneqq \mathrm{pos}(\Delta), \boldsymbol{x} \coloneqq N]] \in \mathrm{SN}$. By Lemma E.1 we see $K'[(\lambda \boldsymbol{x} :^{\gamma} A.\, M)N] \in \mathrm{SN}$.

Proof of $\mathcal{E}$*-*$[]$*:* Suppose $\big(\Delta, \text{🔒}^{\delta}\big) \succcurlyeq \Gamma$. Take $\Delta, \text{🔒}^{\delta} \vdash K \in \mathcal{K}[\![[\succeq \gamma_1\, A]]\!]$ to show $K[{}^{\backprime}\{^{\boldsymbol{\gamma_2} :\succeq \gamma_1} M\}] \in \mathrm{SN}$. For the last rule of the derivation of K, there are two possibilities:

Case $\mathcal{K}$-id: We may assume without loss of generality that $\gamma_2 \notin \mathbf{Dom}_{\mathrm{C}}(\Delta)$. Let $\Delta' \equiv \Delta, \text{🔒}^{\delta}, \text{🔓}^{\boldsymbol{\gamma_2} :\succeq \gamma_1}$. Then we have $\Gamma \preccurlyeq \big(\Delta', \text{🔒}^{\delta}\big)$ with $\Delta' \vdash \delta \sqsubseteq \gamma_2$ and $\Delta' \vdash \gamma_1 \preceq \gamma_2$. By assumption we have $\Delta' \vdash M[\boldsymbol{\gamma_2} \coloneqq \gamma_2] \equiv M \in \mathcal{E}[\![A]\!]$, and Lemma 16 yields $M \in \mathrm{SN}$. Thus, ${}^{\backprime}\{^{\boldsymbol{\gamma_2} :\succeq \gamma_1} M\} \in \mathrm{SN}$.

Case $\mathcal{K}$-$[]$: Then we have $K \equiv K'[\sim\{^{\delta}\,[-]\}]$ for some $\Delta \vdash K \in \mathcal{K}[\![A]\!]$ with $\Delta \vdash \delta \sqsubseteq \mathrm{pos}(\Delta)$ and $\Delta \vdash \gamma_1 \preceq \mathrm{pos}(\Delta)$. By assumption we have $\Delta \vdash M[\boldsymbol{\gamma_2} \coloneqq \mathrm{pos}(\Delta)] \in \mathcal{E}[\![A]\!]$, and hence $K'[M[\boldsymbol{\gamma_2} \coloneqq \mathrm{pos}(\Delta)]] \in \mathrm{SN}$. By Lemma E.1 we see $K'[\sim\{^{\delta}\, {}^{\backprime}\{^{\boldsymbol{\gamma_2} :\succeq \gamma_1} M\}\}] \in \mathrm{SN}$.

Proof of $\mathcal{E}$*-*$\forall$*:* Suppose $\Delta \succcurlyeq \Gamma$. Take $\Delta \vdash K \in \mathcal{K}[\![\forall \boldsymbol{\gamma_2} :\succeq \gamma_1.\, A]\!]$ to show $K[\lambda \boldsymbol{\gamma_2} :\succeq \gamma_1.\, M] \in \mathrm{SN}$. For the last rule of the derivation of K, there are two possibilities:

Case $\mathcal{K}$-id: We may assume without loss of generality that $\gamma_2 \notin \mathbf{Dom}_{\mathrm{C}}(\Delta)$. Let $\Delta' \equiv \Delta, \boldsymbol{\gamma_2} :\succeq \gamma_1$. Then we have $\Gamma \preccurlyeq \Delta'$ with $\Delta' \vdash \gamma_1 \preceq \gamma_2$. By assumption we obtain $\Delta' \vdash M[\boldsymbol{\gamma_2} \coloneqq \gamma_2] \equiv M \in \mathcal{E}[\![A]\!]$, and Lemma 16 yields $M \in \mathrm{SN}$. Thus, $\lambda \boldsymbol{\gamma_2} :\succeq \gamma_1.\, M \in \mathrm{SN}$.

Case $\mathcal{K}$-$\forall$: Then we have $K \equiv K'[[-]\gamma]$ for some $\Delta \vdash K' \in \mathcal{K}[\![A[\boldsymbol{\gamma_2} \coloneqq \gamma]]\!]$ with $\Delta \vdash \gamma_1 \preceq \gamma$. By assumption we obtain $\Delta \vdash M[\boldsymbol{\gamma_2} \coloneqq \gamma] \in \mathcal{E}[\![A[\boldsymbol{\gamma_2} \coloneqq \gamma]]\!]$, and hence $K'[M[\boldsymbol{\gamma_2} \coloneqq \gamma]] \in \mathrm{SN}$. By Lemma E.1 we see $K'[(\forall \boldsymbol{\gamma_2} :\succeq \gamma_1.\, M)\gamma] \in \mathrm{SN}$. □

Lemma E.3 (Monotonicity w.r.t. $\preccurlyeq$)**.** *Suppose* $\Gamma \preccurlyeq \Delta$*. If* $\Gamma \vdash M \in \mathcal{E}[\![A]\!]$*, then* $\Delta \vdash M \in \mathcal{E}[\![A]\!]$.

Proof. If $\mathcal{E}$-id is applied, then it can also be applied to x in Δ as $\Delta \succcurlyeq \Gamma$; otherwise, take $\Delta' \succcurlyeq \Delta$ and $\Delta' \vdash K \in \mathcal{K}[\![A]\!]$. Then we have $\Gamma \preccurlyeq \Delta \preccurlyeq \Delta'$, and thus $K[M] \in \mathrm{SN}$ as $\Gamma \vdash M \in \mathcal{E}[\![A]\!]$, which implies $\Delta \vdash M \in \mathcal{E}[\![A]\!]$. □

Lemma E.4 (Monotonicity regarding $\mathcal{C}[\![\Gamma]\!]$)**.** *Suppose* $\Delta \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]$.

1) *It holds that* $\Delta \vdash \mathrm{pos}(\Gamma)\,\sigma \preceq \mathrm{pos}(\Delta)$.
2) *If* $\Gamma \vdash \gamma_1 \preceq \gamma_2$*, then* $\Delta \vdash \gamma_1\,\sigma \preceq \gamma_2\,\sigma$.
3) *If* $\Gamma \vdash \gamma_1 \sqsubseteq \gamma_2$*, then* $\Delta \vdash \gamma_1\,\sigma \sqsubseteq \gamma_2\,\sigma$.

Proof. By induction on the derivation of $\Delta \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]$. □

Lemma E.5. *Suppose*

$$\begin{array}{c} \vdots \\ \Delta \vdash \sigma \in \mathcal{C}[\![\Gamma]\!] \\ \vdots \\ \Delta' \vdash \sigma \cdot \sigma' \in \mathcal{C}[\![\Gamma, \Gamma']\!] \end{array}$$

If $\Gamma \preccurlyeq \Gamma, \Gamma'$*, then* $\Delta' \preccurlyeq \Delta$.

$$\begin{tikzcd}
\Gamma, \Gamma' \arrow[r, "\sigma \cdot \sigma'"] & \Delta' \\
\Gamma \arrow[u, "\preccurlyeq"] \arrow[r, "\sigma"] & \Delta \arrow[u, dashed, "\preccurlyeq"']
\end{tikzcd}$$

Proof. Check all conditions for $\Delta \preccurlyeq \Delta'$.

To show $\Delta' \vdash \mathrm{pos}(\Delta) \preceq \mathrm{pos}(\Delta')$, we proceed by induction on derivation. If either $\mathcal{C}$-weak or $\mathcal{C}$-$\forall$ is applied, then it follows from the IH; otherwise, we have $\mathrm{pos}(\Gamma, \Gamma')\,(\sigma \cdot \sigma') \equiv \mathrm{pos}(\Delta')$. By Lemma E.4, it holds that

- $\Delta' \vdash \mathrm{pos}(\Gamma)\,\sigma \preceq \mathrm{pos}(\Gamma, \Gamma')$ and
- $\Delta' \vdash \mathrm{pos}(\Gamma, \Gamma')\,\sigma \preceq \mathrm{pos}(\Delta')$,

which implies $\Delta' \vdash \mathrm{pos}(\Delta) {\preceq} \mathrm{pos}(\Delta')$.

The other conditions are shown by straightforward induction. □

Lemma 17 (Fundamental property (On page 11)). *Suppose* $\Gamma \vdash M : A$. *If* $\Delta \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]$, *then* $\Delta \vdash M\,\sigma \in \mathcal{E}[\![A\,\sigma]\!]$.

Proof. By induction on the derivation of $\Gamma \vdash M : A$. Taking $\Delta \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]$, We analyze the last rule of the derivation:

Case Var: If $x \notin \mathbf{Dom}_{\mathrm{V}}(\sigma)$, then it follows from Lemma E.3; otherwise, it follows from Lemma E.5.

Case $\to$-I: Assume

$$\frac{\Gamma, \boldsymbol{x} :^{\gamma} B \vdash N : C \qquad \gamma \notin \mathbf{FC}(C)}{\Gamma \vdash \lambda \boldsymbol{x} :^{\gamma} B.\, N : B \to C}$$

Take $\Delta' \succcurlyeq \Delta$ with $\Delta' \vdash P \in \mathcal{E}[\![B\,\sigma]\!]$. Then we have

$$\text{$\mathcal{C}$-$\to$}\ \frac{\text{$\mathcal{C}$-weak}\ \dfrac{\Delta \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]}{\Delta' \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]} \qquad \Delta' \vdash P \in \mathcal{E}[\![B\,\sigma]\!]}{\Delta' \vdash \sigma \cdot [\boldsymbol{\gamma} \coloneqq \mathrm{pos}(\Delta'), \boldsymbol{x} \coloneqq P] \in \mathcal{C}[\![\Gamma, \boldsymbol{x} :^{\gamma} B]\!]}$$

By the IH, we have

$$\begin{aligned}&\Delta' \vdash N\,(\sigma \cdot [\boldsymbol{\gamma} \coloneqq \mathrm{pos}(\Delta'), \boldsymbol{x} \coloneqq P]) \\ &\quad \equiv (N\,\sigma)[\boldsymbol{\gamma} \coloneqq \mathrm{pos}(\Delta'), \boldsymbol{x} \coloneqq P] \in \mathcal{E}[\![C\,\sigma]\!],\end{aligned}$$

and by $\mathcal{E}$-$\to$, we see

$$\begin{aligned}&\Delta \vdash \lambda \boldsymbol{x} :^{\gamma} B\,\sigma.\, N\,\sigma \\ &\quad \equiv (\lambda \boldsymbol{x} :^{\gamma} B.\, N)\,\sigma \in \mathcal{E}[\![(B \to C)\,\sigma]\!].\end{aligned}$$

Case $\to$-E: Assume

$$\frac{\Gamma \vdash N : B \to C \qquad \Gamma \vdash P : B}{\Gamma \vdash NP : C}$$

Take $\Delta' \succcurlyeq \Delta$ with $\Delta \vdash K \in \mathcal{K}[\![C\,\sigma]\!]$. Then we have

$$\text{IH}\ \frac{\text{$\mathcal{C}$-weak}\ \dfrac{\Delta \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]}{\Delta' \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]} \qquad \Gamma \vdash P : B}{\Delta' \vdash P\,\sigma \in \mathcal{E}[\![B\,\sigma]\!]}$$

and

$$\text{$\mathcal{K}$-$\to$}\ \frac{\Delta' \vdash K \in \mathcal{K}[\![C\,\sigma]\!] \qquad \Delta' \vdash P\,\sigma \in \mathcal{E}[\![B\,\sigma]\!]}{\Delta' \vdash K[[-](P\,\sigma)] \in \mathcal{E}[\![(B \to C)\,\sigma]\!]}$$

Applying the IH to N yields $\Delta \vdash N\,\sigma \in \mathcal{E}[\![(B \to C)\,\sigma]\!]$, so that $K[(NP)\,\sigma] \in \mathrm{SN}$, implying $\Delta \vdash (NP)\,\sigma \in \mathcal{E}[\![C\,\sigma]\!]$.

Case []-I: Assume

$$\frac{\Gamma, 🔓^{\boldsymbol{\gamma_2} :\succeq \gamma_1} \vdash N : B \qquad \gamma_2 \notin \mathbf{FC}(B)}{\Gamma \vdash \text{`}\{^{\boldsymbol{\gamma_2} :\succeq \gamma_1} N\} : [^{\succeq \gamma_1} B]}$$

Take $\Delta', 🔒^{\delta} \succcurlyeq \Delta$ with $\Delta' \vdash \delta \sqsubseteq \mathrm{pos}(\Delta')$ and $\Delta' \vdash \gamma_1\,\sigma \preceq \mathrm{pos}(\Delta')$. Then we have

$$\text{$\mathcal{C}$-[]}\ \frac{\text{$\mathcal{C}$-weak}\ \dfrac{\Delta, 🔒^{\delta} \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]}{\Delta', 🔒^{\delta} \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]} \qquad \Delta' \vdash \gamma_1\,\sigma \preceq \mathrm{pos}(\Delta')}{\Delta' \vdash \sigma \cdot [\boldsymbol{\gamma_2} \coloneqq \mathrm{pos}(\Delta')] \in \mathcal{C}[\![\Gamma, 🔓^{\boldsymbol{\gamma_2} :\succeq \gamma_1}]\!]}$$

By the IH, we have

$$\begin{aligned}&\Delta' \vdash N\,(\sigma \cdot [\boldsymbol{\gamma_2} \coloneqq \mathrm{pos}(\Delta')]) \\ &\quad \equiv (N\,\sigma)[\boldsymbol{\gamma_2} \coloneqq \mathrm{pos}(\Delta')] \in \mathcal{E}[\![B\,\sigma]\!],\end{aligned}$$

and by $\mathcal{E}$-[], we see

$$\begin{aligned}&\Delta \vdash \text{`}\{^{\boldsymbol{\gamma_2} :\succeq \gamma_1\,\sigma} N\,\sigma\} \\ &\quad \equiv \text{`}\{^{\boldsymbol{\gamma_2} :\succeq \gamma_1} N\}\,\sigma \in \mathcal{E}[\![[^{\succeq \gamma_1} B]\,\sigma]\!].\end{aligned}$$

Case []-E: Assume

$$\frac{\Gamma, 🔒^{\gamma} \vdash N : [^{\succeq \gamma_1} B] \qquad \Gamma \vdash \gamma_1 \preceq \mathrm{pos}(\Gamma)}{\Gamma \vdash \sim\!\{^{\gamma} N\} : B}$$

Take $\Delta' \succcurlyeq \Delta$ with $\Delta' \vdash K \in \mathcal{K}[\![B\,\sigma]\!]$. Using Lemma E.4 we have

$$\frac{\dfrac{\Gamma \vdash \gamma \sqsubseteq \mathrm{pos}(\Gamma)}{\Delta' \vdash \gamma\,\sigma \sqsubseteq \mathrm{pos}(\Gamma)\,\sigma} \qquad \dfrac{}{\Delta' \vdash \mathrm{pos}(\Gamma)\,\sigma \preceq \mathrm{pos}(\Delta')}}{\Delta' \vdash \gamma\,\sigma \sqsubseteq \mathrm{pos}(\Delta')}$$

$$\frac{\dfrac{\Gamma \vdash \gamma_1 \preceq \mathrm{pos}(\Gamma)}{\Delta' \vdash \gamma\,\sigma \sqsubseteq \mathrm{pos}(\Gamma)\,\sigma} \qquad \dfrac{}{\Delta' \vdash \mathrm{pos}(\Gamma)\,\sigma \preceq \mathrm{pos}(\Delta')}}{\Delta' \vdash \gamma_1\,\sigma \preceq \mathrm{pos}(\Delta')}$$

and $\Delta', 🔒^{\gamma\,\sigma} \vdash K[\sim\!\{^{\gamma\,\sigma}\,[-]\}] \in \mathcal{K}[\![[^{\succeq \gamma_1} B]\,\sigma]\!]$ by $\mathcal{K}$-[]. In addition, we have

$$\text{IH}\ \frac{\text{$\mathcal{C}$-🔒}\ \dfrac{\text{$\mathcal{C}$-weak}\ \dfrac{\Delta \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]}{\Delta' \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]}}{\Delta', 🔒^{\gamma\,\sigma} \vdash \sigma \in \mathcal{C}[\![\Gamma, 🔒^{\gamma}]\!]} \qquad \Gamma, 🔒^{\gamma} \vdash N : [^{\succeq \gamma_1} B]}{\Delta', 🔒^{\gamma\,\sigma} \vdash N\,\sigma \in \mathcal{E}[\![[^{\succeq \gamma_1} B]\,\sigma]\!]}$$

and therefore $K[\sim\!\{^{\gamma\,\sigma} N\,\sigma\}] \equiv K[(\sim\!\{^{\gamma} N\})\,\sigma] \in \mathrm{SN}$, implying $\Delta \vdash (\sim\!\{^{\gamma} N\})\,\sigma \in \mathcal{E}[\![B\,\sigma]\!]$.

Case $\forall$-I: Assume

$$\frac{\Gamma, \boldsymbol{\gamma_2} :\succeq \gamma_1 \vdash N : B}{\Gamma \vdash \lambda \boldsymbol{\gamma_2} :\succeq \gamma_1.\, N : \forall \boldsymbol{\gamma_2} :\succeq \gamma_1.\, B}$$

Take $\Delta' \succcurlyeq \Delta$ with $\Delta' \vdash \gamma_1\,\sigma \preceq \gamma$. Then we have

$$\text{$\mathcal{C}$-$\forall$}\ \frac{\text{$\mathcal{C}$-weak}\ \dfrac{\Delta \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]}{\Delta' \vdash \sigma \in \mathcal{C}[\![\Gamma]\!]} \qquad \Delta' \vdash \gamma_1\,\sigma \preceq \gamma}{\Delta' \vdash \sigma \cdot [\boldsymbol{\gamma_2} \coloneqq \gamma] \in \mathcal{C}[\![\Gamma, \boldsymbol{\gamma_2} :\succeq \gamma_1]\!]}$$

By the IH, we have $\Delta' \vdash (N\,\sigma)[\boldsymbol{\gamma_2} \coloneqq \gamma] \in \mathcal{E}[\![B\,\sigma]\!]$, and by $\mathcal{E}$-$\forall$ we see

$$\begin{aligned}&\Delta \vdash \lambda \boldsymbol{\gamma_2} :\succeq \gamma_1\,\sigma.\, N\,\sigma \\ &\quad \equiv (\lambda \boldsymbol{\gamma_2} :\succeq \gamma_1.\, N)\,\sigma \in \mathcal{E}[\![(\forall \boldsymbol{\gamma_2} :\succeq \gamma_1.\, B)\,\sigma]\!].\end{aligned}$$

Case $\forall$-E: Assume

$$\frac{\Gamma \vdash N : \forall \boldsymbol{\gamma_2} :\succeq \gamma_1.\, B \qquad \Gamma \vdash \gamma_1 \preceq \gamma}{\Gamma \vdash N\gamma : B[\boldsymbol{\gamma_2} \coloneqq \gamma]}$$

Take $\Delta' \succcurlyeq \Delta$ with $\Delta \vdash K \in \mathcal{K}[\![(B[\boldsymbol{\gamma_2} \coloneqq \gamma])\,\sigma]\!]$. Using Lemma E.4 we have

$$\text{$\mathcal{C}$-$\forall$}\ \frac{\dfrac{\Gamma \vdash \gamma_1 \preceq \gamma}{\Delta' \vdash \gamma_1\,\sigma \preceq \gamma\,\sigma} \qquad \Delta' \vdash K \in \mathcal{K}[\![(B[\boldsymbol{\gamma_2} \coloneqq \gamma])\,\sigma]\!]}{\Delta' \vdash K[[-](\gamma\,\sigma)] \in \mathcal{C}[\![(\forall \boldsymbol{\gamma_2} :\succeq \gamma_1.\, B)\,\sigma]\!]}$$

Applying the IH to N yields $\Delta' \vdash N\,\sigma \in \mathcal{E}[\![(\forall \boldsymbol{\gamma_2} :\succeq \gamma_1.\, B)\,\sigma]\!]$, so that $K[(N\gamma)\,\sigma] \in \mathrm{SN}$, implying $\Delta \vdash (N\gamma)\,\sigma \in \mathcal{E}[\![(B[\boldsymbol{\gamma_2} := \gamma])\,\sigma]\!]$. ☐

Theorem 12 (Confluence (On page 11))**.** *The β-reduction is confluent for well-typed terms.*

Proof. From Newman's lemma [2] and Theorem 11, we need only to prove weak confluence. It is done by induction on β, which is straightforward because there are no critical pairs. ☐

Lemma E.6. *Suppose $\Gamma \vdash M : A$. If M is β-normal and neutral, then there exists some $\boldsymbol{x} :^{\gamma} B \in \Gamma$ such that A is a subformula of B.*

Proof. By induction on derivation. For the last rule of the derivation, there are four possibilities:

Case Var: Obvious.

Case $\to$-E: Assume

$$\frac{\Gamma \vdash N : B \to C \qquad \Gamma \vdash P : B}{\Gamma \vdash NP : C}$$

Since NP is β-normal, N is β-normal and neutral. By the IH there exists some $\boldsymbol{x} :^{\boldsymbol{\delta}} D \in \Gamma$ such that $B \to C$ is a subformula of D. Then C is also a subformula of D, hence $\boldsymbol{x} :^{\boldsymbol{\delta}} D$ meets the condition.

Case []-E: Assume

$$\frac{\Gamma, \text{🔒}^{\gamma} \vdash N : [\succeq^{\gamma_1} B] \qquad \Gamma \vdash \gamma_1 \preceq \mathrm{pos}(\Gamma)}{\Gamma \vdash \sim\!\{^{\gamma} N\} : B}$$

Since $\sim\!\{^{\gamma} N\}$ is β-normal, N is β-normal and neutral. By the IH there exists some $\boldsymbol{x} :^{\boldsymbol{\delta}} D \in \Gamma, \text{🔒}^{\gamma}$ such that $[\succeq^{\gamma_1} B]$ is a subformula of D. Then B is also a subformula of D, hence $\boldsymbol{x} :^{\boldsymbol{\delta}} D$ meets the condition.

Case $\forall$-E: Assume

$$\frac{\Gamma \vdash N : \forall \boldsymbol{\gamma_2} :\succeq \gamma_1.\, B \qquad \Gamma \vdash \gamma_1 \preceq \gamma}{\Gamma \vdash N\gamma : B[\boldsymbol{\gamma_2} := \gamma]}$$

Since $N\gamma$ is β-normal, N is β-normal and neutral. By the IH there exists some $\boldsymbol{x} :^{\boldsymbol{\delta}} D \in \Gamma$ such that $\forall \boldsymbol{\gamma_2} :\succeq \gamma_1.\, B$ is a subformula of D. Then $B[\boldsymbol{\gamma_2} := \gamma]$ is also a subformula of D, hence $\boldsymbol{x} :^{\boldsymbol{\delta}} D$ meets the condition. Notice that classifier renaming $[\boldsymbol{\gamma_2} := \gamma]$ here is allowed in the definition of subformula. ☐

Theorem 13 (Canonicity (On page 11))**.** *If a term is well-typed, closed regarding term variable, and β-normal, then it is canonical.*

Proof. If not canonical, by Lemma E.6 it contains a free variable, which contradicts the assumption. ☐

Theorem 14 (Subformula property (On page 11))**.** *Suppose $\Gamma \vdash M : A$. If M is β-normal, then any subterm of M satisfies at least one of the following:*

a) *Its type is a subformula of A;*
b) *Its type is a subformula of B for some $\boldsymbol{x} :^{\gamma} B \in \Gamma$.*

Proof. By induction on derivation. Since the term M itself clearly satisfies (a), it suffices to check condition for proper subterms.

Case Var: No proper subterm exists.

Case $\to$-I: Assume

$$\frac{\Gamma, \boldsymbol{y} :^{\gamma} B \vdash N : C \qquad \gamma \notin \mathbf{FC}(C)}{\Gamma \vdash \lambda \boldsymbol{y} :^{\gamma} B.\, N : B \to C}$$

Since all proper subterms of $\lambda \boldsymbol{y} :^{\gamma} B.\, N$ are a subterm of N, by the IH there are three possibilities for their types:

Subcase 1: A subformula of C. Then it is also a subformula of $B \to C$, so (a) holds.

Subcase 2: A subformula of B. This is also the case (a).

Subcase 3: A subformula of D for some $\boldsymbol{x} :^{\boldsymbol{\delta}} D \in \Gamma$. Yields (b).

Case $\to$-E: Assume

$$\frac{\Gamma \vdash N : B \to C \qquad \Gamma \vdash P : B}{\Gamma \vdash NP : C}$$

Then N is β-normal and neutral, so applying Lemma E.6 to N, we see that there exists some $\boldsymbol{x} :^{\boldsymbol{\delta}} D \in \Gamma$ such that $B \to C$ is a subformula of D. Together with the IH, we see that (b) holds for any proper subterm of NP.

Case []-I: Follows from the IH.

Case []-E: Similar to the case $\to$-E.

Case $\forall$-I: Follows from the IH.

Case $\forall$-E: Similar to the case $\to$-E. ☐