Matching μ-Logic

Xiaohong Chen
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801–2302
Email: xc3@illinois.edu

Grigore Roşu
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801–2302
Email: grosu@illinois.edu

***Abstract*—Matching logic is a logic for specifying and reasoning about structure by means of patterns and pattern matching. This paper makes two contributions. First, it proposes a sound and complete proof system for matching logic in its full generality. Previously, sound and complete deduction for matching logic was known only for particular theories providing equality and membership. Second, it proposes matching μ-logic, an extension of matching logic with a least fixpoint μ-binder. It is shown that matching μ-logic captures as special instances many important logics in mathematics and computer science, including first-order logic with least fixpoints, modal μ-logic as well as dynamic logic and various temporal logics such as infinite/finite-trace linear temporal logic and computation tree logic, and notably reachability logic, the underlying logic of the K framework for programming language semantics and formal analysis. Matching μ-logic therefore serves as a unifying foundation for specifying and reasoning about fixpoints and induction, programming languages and program specification and verification.**

I. Motivation

Matching logic [1] (shortened as ML) is a first-order logic (FOL) variant for specifying and reasoning about structure by means of patterns and pattern matching. In the practice of *program verification*, ML is used to specify static properties of programs in reachability logic [2] (shortened as RL), which takes an operational semantics of a programming language as axioms and yields a program verifier that can prove any reachability properties of any programs written in that language. As a successful implementation of ML and RL, the $\mathbb{K}$ framework (http://kframework.org) has been used to define the formal semantics of various real languages such as C [3], Java [4], JavaScript [5], and to verify complex program properties [6].

A sound and complete Hilbert-style proof system $\mathcal{P}$ of ML is given in [1], whose proof of completeness is by reduction to pure predicate logic. However, the proof system $\mathcal{P}$ is only applicable to theories where a set of special *definedness symbols* are given together with appropriate axioms that can be used to define both equality and membership as derived constructs. This leaves the question of whether there is any proof system of ML that gives sound and complete deduction *for all theories*, open. Our first contribution is to answer this question by proposing a new proof system $\mathcal{H}$ of ML that is complete *without requiring definedness or any other symbols*.

Our second and main contribution was stimulated by limitations of RL itself as a logic to reason about dynamic behaviors of programs. Specifically, as its name suggests, RL can only define and reason about reachability claims. In particular, it is not capable of expressing liveness or many other interesting properties that temporal or dynamic logics can naturally express. Therefore, we propose *matching μ-logic* (shortened as MmL), which extends ML with a least fixpoint μ-binder. It turns out that MmL subsumes not only RL, but also a variety of common logics/calculi that are used to reason about fixpoints and induction, especially for program verification and model checking, including first-order logic with least fixpoints (LFP) [7], modal μ-logic [8] (as well as various temporal logics [9], [10] and dynamic logic (DL) [11]–[13]). For each of these we prove a *conservative extension result*, showing the faithfulness of our definitions.

We organize the rest of the paper as follows. We start with a quick but comprehensive overview of ML in Section II, and then present the new proof system $\mathcal{H}$ in Section III. We present MmL in Section IV, and show how to define recursive/co-recursive symbols as syntactic sugar in Section V. Then we discuss how MmL subsumes all the following: first-order logic with least fixpoints (Section VI); modal μ-logic and its fragment logics (Section VII); reachability logic (Section VIII). We compare with related work and conclude the paper with a proposal of future work in Sections IX and X.

II. Matching Logic Preliminaries

Matching logic (ML) is a variant of many-sorted FOL that makes no distinction between operation and predicate symbols, allowing them to be uniformly used to build *patterns*. Patterns define both structural and logical constraints, and are interpreted in models as sets of elements (those that *match* them). We offer a compact but comprehensive review of ML below. A detailed discussion of ML can be found in [1].

A. Matching logic syntax

Definition 1. A *matching logic signature* or simply a *signature* $\mathbb{\Sigma} = (S, \textsc{Var}, \Sigma)$ is a triple with a nonempty set S of *sorts*, an S-indexed set $\textsc{Var} = \{\textsc{Var}_s\}_{s\in S}$ of countably infinitely many *sorted variables* denoted $x{:}s, y{:}s$, etc., and an $(S^* \times S)$-indexed countable set $\Sigma = \{\Sigma_{s_1\ldots s_n,s}\}_{s_1,\ldots,s_n,s\in S}$ of *many-sorted symbols*. When $n = 0$, we write $\sigma \in \Sigma_{\lambda,s}$ and say σ is a *constant symbol*. Matching logic $\mathbb{\Sigma}$*-patterns*, or simply *($\mathbb{\Sigma}$-)patterns*, are defined inductively for all sorts $s, s', s_1, \ldots, s_n \in S$ as follows[1]:

[1]We use different primitives $\{\rightarrow, \neg, \forall\}$ than [1], which uses $\{\wedge, \neg, \exists\}$. These are more appropriate for our new proof system $\mathcal{H}$ (Fig. 1 in Section III).

$$\varphi_s ::= x{:}s \in \textsc{Var}_s \mid \varphi_s \to \varphi_s \mid \neg\varphi_s \mid \forall x{:}s'.\varphi_s \\ \mid \sigma(\varphi_{s_1}, \ldots, \varphi_{s_n}) \quad \text{if } \sigma \in \Sigma_{s_1 \ldots s_n, s}$$

We use $\textsc{Pattern}^{\mathsf{ML}}(\mathbb{\Sigma}) = \{\textsc{Pattern}^{\mathsf{ML}}_s(\mathbb{\Sigma})\}_{s\in S}$ to denote the S-indexed set of $\mathbb{\Sigma}$-patterns generated by the above grammar (modulo α-equivalence, see later). We feel free to drop the signature $\mathbb{\Sigma}$ and simply write $\textsc{Pattern}^{\mathsf{ML}} = \{\textsc{Pattern}^{\mathsf{ML}}_s\}_{s\in S}$.

The signature $\mathbb{\Sigma} = (S, \textsc{Var}, \Sigma)$ is abbreviated as (S, Σ) or just Σ when $\textsc{Var}$ and S are understood or not important. When we write a pattern, we assume it is well-formed without explicitly specifying the necessary conditions. When $\sigma \in \Sigma_{\lambda,s}$ is a constant symbol, we write σ to mean the pattern $\sigma()$. We adopt the following derived construct as syntactic sugar:

$$\varphi_1 \vee \varphi_2 \equiv \neg\varphi_1 \to \varphi_2 \qquad \exists x{:}s.\varphi \equiv \neg\forall x{:}s.\neg\varphi$$
$$\varphi_1 \wedge \varphi_2 \equiv \neg(\neg\varphi_1 \vee \neg\varphi_2) \qquad \top_s \equiv \exists x{:}s.x{:}s$$
$$\varphi_1 \leftrightarrow \varphi_2 \equiv (\varphi_1 \to \varphi_2) \wedge (\varphi_2 \to \varphi_1) \qquad \bot_s \equiv \neg\top_s$$

Note that "top" $\top_s$, the pattern that matches everything (see Proposition 5) is closed. We drop sort s whenever possible, so we write $x, y, \top, \bot$ instead of $x{:}s, y{:}s, \top_s, \bot_s$. Standard precedences are adopted to avoid parentheses. The scope of "$\forall$" and "$\exists$" goes as far as possible to the right.

As in FOL, "$\forall$" (and "$\exists$") are *binders*, and we adopt the standard notions of *free variables*, *α-renaming*, and *capture-avoiding substitution*. We use $FV(\varphi)$ to denote the set of all free variables in φ. When $FV(\varphi) = \emptyset$, we say φ is *closed*. We regard patterns that are α-equivalent *as the same*, i.e., $\varphi \equiv \varphi'$ if φ, φ' are α-equivalent. We write $\varphi[\psi/x]$ to mean the result of substituting ψ for every free occurrence of x in φ, where α-renaming happens implicitly to prevent variable capture. We abbreviate $\varphi[\psi_1/x_1]\ldots[\psi_n/x_n]$ as $\varphi[\psi_1/x_1, \ldots, \psi_n/x_n]$.

B. Matching logic semantics

ML symbols are interpreted as *relations*, and thus ML patterns evaluate to *sets of elements* (those "matching" them).

Definition 2. Let $\mathbb{\Sigma} = (S, \Sigma)$ be a signature. A *matching logic $\mathbb{\Sigma}$-model* $M = (\{M_s\}_{s\in S}, \_^M)$, or just a *($\mathbb{\Sigma}$-)model*, consists of:

- a nonempty carrier set M_s for each sort $s \in S$;
- an interpretation $\sigma_M : M_{s_1} \times \cdots \times M_{s_n} \to \mathcal{P}(M_s)$ for each $\sigma \in \Sigma_{s_1 \ldots s_n, s}$, where $\mathcal{P}(M_s)$ is the powerset of M_s.

For notational simplicity, we overload the letter M and use it to also mean the S-indexed set of carrier sets $\{M_s\}_{s\in S}$. The usual FOL models are special cases of ML models, where $|\sigma_M(a_1, \ldots, a_n)| = 1$ for all $a_1 \in M_{s_1}, \ldots, a_n \in M_{s_n}$. *Partial* FOL models are also special cases where $|\sigma_M(a_1, \ldots, a_n)| \leq 1$, since we can capture the undefinedness of the partial function σ_M on $a_1, \ldots, a_n$ with $\sigma_M(a_1, \ldots, a_n) = \emptyset$.

We tactically use the same letter σ_M to mean its *pointwise extension*, $\sigma_M : \mathcal{P}(M_{s_1}) \times \cdots \times \mathcal{P}(M_{s_n}) \to \mathcal{P}(M_s)$, defined as:

$$\sigma_M(A_1, \ldots, A_n) = \bigcup \{\sigma_M(a_1, \ldots, a_n) \mid a_1 \in A_1, \ldots, a_n \in A_n\}$$

for all $A_1 \subseteq M_{s_1}, \ldots, A_n \subseteq M_{s_n}$.

Proposition 3. *For all $A_i, A'_i \subseteq M_{s_i}$, $1 \leq i \leq n$, the pointwise extension σ_M has the following* property of propagation*:*

$$\sigma_M(A_1, \ldots, A_n) = \emptyset \text{ if } A_j = \emptyset \text{ for some } 1 \leq j \leq n,$$
$$\sigma_M(A_1 \cup A'_1, \ldots, A_n \cup A'_n) = \bigcup_{1\leq j\leq n, B_j \in \{A_j, A'_j\}} \sigma_M(B_1, \ldots, B_n),$$
$$\sigma(A_1, \ldots, A_n) \subseteq \sigma(A'_1, \ldots, A'_n) \text{ if } A_i \subseteq A'_i \text{ for all } 1 \leq i \leq n.$$

Definition 4. Let $\mathbb{\Sigma} = (S, \textsc{Var}, \Sigma)$ and let M be a $\mathbb{\Sigma}$-model. Given a function $\rho : \textsc{Var} \to M$, called an *$M$-valuation*, let its *extension* $\bar{\rho} : \textsc{Pattern}^{\mathsf{ML}} \to \mathcal{P}(M)$ be inductively defined as:

- $\bar{\rho}(x) = \{\rho(x)\}$, for all $x \in \textsc{Var}_s$;
- $\bar{\rho}(\varphi_1 \to \varphi_2) = M_s \setminus (\bar{\rho}(\varphi_1) \setminus \bar{\rho}(\varphi_2))$, for $\varphi_1, \varphi_2 \in \textsc{Pattern}_s$;
- $\bar{\rho}(\neg\varphi) = M_s \setminus \bar{\rho}(\varphi)$, for all $\varphi \in \textsc{Pattern}_s$;
- $\bar{\rho}(\forall x.\varphi) = \bigcap_{a\in M_{s'}} \overline{\rho[a/x]}(\varphi)$, for all $x \in \textsc{Var}_{s'}$;
- $\bar{\rho}(\sigma(\varphi_1, ..., \varphi_n)) = \sigma_M(\bar{\rho}(\varphi_1), ..., \bar{\rho}(\varphi_n))$, for $\sigma \in \Sigma_{s_1 \ldots s_n, s}$;

where "$\setminus$" is set difference and $\rho[a/x]$ denotes the M-valuation ρ' with $\rho'(x) = a$ and $\rho'(y) = \rho(y)$ for all $y \neq x$.

Intuitively, a pattern is evaluated to the set of all elements that "match" it. For example, the variable x (as a pattern) is matched by exactly one element, $\rho(x)$; the pattern $\neg\varphi$ is matched by exactly those that do not match φ; etc. The next proposition shows that all derived constructs have the expected semantics: "$\wedge$" means conjunction, "$\vee$" means disjunction, "$\top$" means the total set, "$\bot$" means the empty set, etc.

Proposition 5. *The following propositions hold:*

- $\bar{\rho}(\top_s) = M_s$ *and* $\bar{\rho}(\bot_s) = \emptyset$*;*
- $\bar{\rho}(\varphi_1 \wedge \varphi_2) = \bar{\rho}(\varphi_1) \cap \bar{\rho}(\varphi_2)$*;*
- $\bar{\rho}(\varphi_1 \vee \varphi_2) = \bar{\rho}(\varphi_1) \cup \bar{\rho}(\varphi_2)$*;*
- $\bar{\rho}(\varphi_1 \leftrightarrow \varphi_2) = M_s \setminus (\bar{\rho}(\varphi_1) \triangle \bar{\rho}(\varphi_2))$*, for* $\varphi_1, \varphi_2 \in \textsc{Pattern}_s$*;*
- $\bar{\rho}(\exists x.\varphi) = \bigcup_{a\in M_{s'}} \overline{\rho[a/x]}(\varphi)$*, for all* $x \in \textsc{Var}_{s'}$*;*

where "$\triangle$" is set symmetric difference.

Definition 6. We say a matching logic pattern φ holds in M, written $M \vDash_{\mathsf{ML}} \varphi$, if $\bar{\rho}(\varphi) = M$ for all $\rho : \textsc{Var} \to M$. Let Γ be a set of patterns, called *axioms*. We write $M \vDash_{\mathsf{ML}} \Gamma$ iff $M \vDash_{\mathsf{ML}} \varphi$ for all axioms $\varphi \in \Gamma$. We write $\Gamma \vDash_{\mathsf{ML}} \varphi$ iff $M \vDash_{\mathsf{ML}} \varphi$ for all models $M \vDash_{\mathsf{ML}} \Gamma$. When Γ is empty, we abbreviate $\Gamma \vDash_{\mathsf{ML}} \varphi$ as $\vDash_{\mathsf{ML}} \varphi$, and say that φ is *valid*. This is, a pattern is valid iff it is matched by all elements in all models. We call the pair $(\mathbb{\Sigma}, \Gamma)$ a *matching logic $\mathbb{\Sigma}$-theory*, or simply a *($\mathbb{\Sigma}$-)theory*. Model M is said to be *a model of the theory* $(\mathbb{\Sigma}, \Gamma)$ iff $M \vDash_{\mathsf{ML}} \Gamma$.

C. Important notations

Several mathematical instruments of practical importance, such as definedness, totality, equality, membership, set containment, functions and partial functions, and constructors, can all be defined using patterns. We give a compact summary of the definitions and notations that are needed in this paper.

Definition 7. For any (not necessarily distinct) sorts s, s', let us consider a unary symbol $\lceil\_\rceil^{s'}_s \in \Sigma_{s,s'}$, called the *definedness symbol*, and the pattern/axiom $\lceil x{:}s\rceil^{s'}_s$, called (Definedness).

We define *totality* "$\lfloor \_ \rfloor_s^{s'}$", *equality* "$=_s^{s'}$", *membership* "$\in_s^{s'}$", and *set containment* "$\subseteq_s^{s'}$" as derived constructs:

$$\lfloor \varphi \rfloor_s^{s'} \equiv \neg \lceil \neg \varphi \rceil_s^{s'} \qquad \varphi_1 =_s^{s'} \varphi_2 \equiv \lfloor \varphi_1 \leftrightarrow \varphi_2 \rfloor_s^{s'}$$
$$x \in_s^{s'} \varphi \equiv \lceil x \wedge \varphi \rceil_s^{s'} \qquad \varphi_1 \subseteq_s^{s'} \varphi_2 \equiv \lfloor \varphi_1 \rightarrow \varphi_2 \rfloor_s^{s'}$$

and feel free to drop the (not necessarily distinct) sorts s, s'.

The (Definedness) axiom ensures that $(\lceil \_ \rceil_s^{s'})_M(a) = M_{s'}$ in all models M for all $a \in M_s$. Therefore, for all valuations ρ, we have $\bar{\rho}(\lceil \varphi \rceil_s^{s'}) = M_{s'}$ if $\bar{\rho}(\varphi) \neq \emptyset$, and $\bar{\rho}(\lceil \varphi \rceil_s^{s'}) = \emptyset$ otherwise. That is, $\lceil \varphi \rceil_s^{s'}$ says, in the sort universe s', if φ is defined or not in its sort universe s. We can prove all constructs in Definition 7 have the expected semantics: $\bar{\rho}(\lfloor \varphi \rfloor_s^{s'}) = M_{s'}$ if $\bar{\rho}(\varphi) = M_s$, and $\bar{\rho}(\lfloor \varphi \rfloor_s^{s'}) = \emptyset$, otherwise; $\bar{\rho}(\varphi_1 =_s^{s'} \varphi_2) = M_{s'}$ if $\bar{\rho}(\varphi_1) = \bar{\rho}(\varphi_2)$, and $\bar{\rho}(\varphi_1 =_s^{s'} \varphi_2) = \emptyset$ otherwise; etc.

Functions and *partial functions* can be defined by axioms:

$$\text{(Function)} \qquad \exists y \,.\, \sigma(x_1, \ldots, x_n) = y$$
$$\text{(Partial Function)} \qquad \exists y \,.\, \sigma(x_1, \ldots, x_n) \subseteq y$$

(Function) requires $\sigma(x_1, \ldots, x_n)$ contains exactly one element and (Partial Function) requires it contains at least one element (recall y is evaluated to a singleton set). For brevity, we use the function notation $\sigma \colon s_1 \times \cdots \times s_n \rightarrow s$ to mean we automatically assume the (Function) axiom of σ. Similarly, partial functions are written as $\sigma \colon s_1 \times \cdots \times s_n \rightharpoonup s$.

Constructors are extensively used in building programs and data, as well as semantic structures to define and reason about languages and programs. They can be defined in the "no junk, no confusion" spirit [14]. Let $\mathbb{\Sigma} = (S, \Sigma)$ be a signature, let $C = \{c_i \in \Sigma_{s_i^1 \ldots s_i^{m_i}, s_i} \mid 1 \leq i \leq n\} \subseteq \Sigma$ be a set of *constructor symbols*, and consider the following axioms/patterns:

(No Junk) for all sorts $s \in S$:
$$\bigvee_{c_i \in C \text{ with } s_i = s} \exists x_i^1 : s_i^1 \ldots \exists x_i^{m_i} : s_i^{m_i} \,.\, c_i(x_i^1, \ldots, x_i^{m_i})$$
(No Confusion I) for all $i \neq j$ and $s_i = s_j$:
$$\neg(c_i(x_i^1, \ldots, x_i^{m_i}) \wedge c_j(x_j^1, \ldots, x_j^{m_j}))$$
(No Confusion II) for all $1 \leq i \leq n$:
$$(c_i(x_i^1, \ldots, x_i^{m_i}) \wedge c_i(y_i^1, \ldots, y_i^{m_i})) \rightarrow c_i(x_i^1 \wedge y_i^1, \ldots, x_i^{m_i} \wedge y_i^{m_i})$$

Intuitively, (No Junk) says everything is constructed; (No Confusion I) says different constructs build different things; and (No Confusion II) says constructors are injective. We refer to the the last two axioms as (No Confusion).

D. Defining first-order logic in matching logic

Given a FOL signature (S, Σ, Π) with *function symbols* Σ and *predicate symbols* Π, the *syntax* of FOL is given by:

$$t_s ::= x \in \text{Var}_s \mid f(t_{s_1}, \ldots, t_{s_n}) \text{ with } f \in \Sigma_{s_1 \ldots s_n, s}$$
$$\varphi ::= \pi(t_{s_1}, \ldots, t_{s_n}) \text{ with } \pi \in \Pi_{s_1 \ldots s_n} \mid \varphi \rightarrow \varphi \mid \neg \varphi \mid \forall x . \varphi$$

To subsume the syntax, we define a ML signature $\mathbb{\Sigma}^{\mathsf{FOL}} = (S^{\mathsf{FOL}}, \Sigma^{\mathsf{FOL}})$, where $S^{\mathsf{FOL}} = S \cup \{Pred\}$ contains a distinguished sort *Pred*, and $\Sigma^{\mathsf{FOL}} = \{f \colon s_1 \times \cdots \times s_n \rightarrow s \mid f \in \Sigma_{s_1 \ldots s_n, s}\} \cup \{\pi \colon s_1 \times \cdots \times s_n \rightarrow Pred \mid \pi \in \Pi_{s_1 \ldots s_n}\}$ contains FOL function symbols as ML functions and FOL predicate symbols as ML functions that return *Pred*. Let Γ^{FOL} be the resulting $\mathbb{\Sigma}^{\mathsf{FOL}}$-theory. Notice that we use the function notations so Γ^{FOL} contains the (Function) axioms for all symbols in Σ^{ML}.

Proposition 8. *For all FOL formulas φ, we have φ is a $\mathbb{\Sigma}^{FOL}$-pattern of sort Pred and $\vDash_{FOL} \varphi$ if and only if $\Gamma^{FOL} \vDash_{ML} \varphi$.*

E. Matching logic proof system $\mathcal{P}$ with definedness symbols

ML has a *conditional* sound and complete Hilbert-style proof system, which we refer to as $\mathcal{P}$ in this paper. We refer readers to [1] for details (see also Fig. 3). Here we denote its provability relation as $\Gamma \vdash_{\mathcal{P}} \varphi$. The proof system $\mathcal{P}$ can prove all patterns φ that are valid in Γ *under the condition that* Γ *contains definedness symbols and (Definedness) axioms*. In fact, many proof rules in $\mathcal{P}$ use the equality "=" and membership "$\in$" constructs, both of which are defined using the definedness symbols. This means $\mathcal{P}$ *is not applicable at all* to any theories that do not contain definedness symbols.

We wrap up this subsection by reviewing the soundness and completeness theorem of $\mathcal{P}$. In Section III, we propose a new ML proof system $\mathcal{H}$ that is sound and complete *without requiring the theories to contain definedness symbols*.

Theorem 9 (Soundness and Completeness of $\mathcal{P}$)**.** *For all theories Γ containing the definedness symbols and axioms (Definition 7) and all patterns φ, we have $\Gamma \vDash_{ML} \varphi$ iff $\Gamma \vdash_{\mathcal{P}} \varphi$.*

III. A New Proof System of Matching Logic

Our first main contribution in this paper is a new ML proof system $\mathcal{H}$ that is sound and complete *without requiring definedness symbols and axioms*, and thus extends the completeness result in [1], re-stated in Theorem 9.

We first need the following definition of *context*:

Definition 10. A *context* C is a pattern with a distinguished placeholder variable $\Box$. We write $C[\varphi]$ to mean the result of *replacing* $\Box$ with φ *without any α-renaming*, so free variables in φ may become bound in $C[\varphi]$, *different* from capture-avoiding substitution. A *single symbol context* has the form $C_\sigma \equiv \sigma(\varphi_1, \ldots, \varphi_{i-1}, \Box, \varphi_{i+1}, \ldots, \varphi_n)$ where $\sigma \in \Sigma_{s_1 \ldots s_n, s}$ and $\varphi_1, \ldots, \varphi_{i-1}, \varphi_{i+1}, \ldots, \varphi_n$ are patterns of appropriate sorts. A *nested symbol context* is inductively defined as:

- $\Box$ is a nested symbol context, called the *identity context*;
- if C_σ is a single symbol context, and C is a nested symbol context, then $C_\sigma[C[\Box]]$ is a nested symbol context.

Intuitively, a context C is a nested symbol context iff the path to $\Box$ in C contains only symbols and no logic connectives.

The proof system $\mathcal{H}$ (Fig. 1, above the double line) has 13 proof rules that are divided into four categories. The first category consists of the Łukasiewicz complete axiomatization of propositional logic [15] (four rules). The second category completes the (complete) axiomatization of first-order logic [16] (three rules). The third category contains four rules that capture the property of propagation (Proposition 3). The

$\mathcal{H}_\mu$ $\mathcal{H}$

(Proposition$_1$)	$\varphi_1 \to (\varphi_2 \to \varphi_1)$
(Proposition$_2$)	$(\varphi_1 \to (\varphi_2 \to \varphi_3)) \to (\varphi_1 \to \varphi_2) \to (\varphi_1 \to \varphi_3)$
(Proposition$_3$)	$(\neg\varphi_1 \to \neg\varphi_2) \to (\varphi_2 \to \varphi_1)$
(Modus Ponens)	$\dfrac{\varphi_1 \quad \varphi_1 \to \varphi_2}{\varphi_2}$
(Variable Substitution)	$\forall x.\varphi \to \varphi[y/x]$
($\forall$)	$\forall x.(\varphi_1 \to \varphi_2) \to (\varphi_1 \to \forall x.\varphi_2)$ if $x \notin FV(\varphi_1)$
(Universal Generalization)	$\dfrac{\varphi}{\forall x.\varphi}$
(Propagation$_\bot$)	$C_\sigma[\bot] \to \bot$
(Propagation$_\vee$)	$C_\sigma[\varphi_1 \vee \varphi_2] \to C_\sigma[\varphi_1] \vee C_\sigma[\varphi_2]$
(Propagation$_\exists$)	$C_\sigma[\exists x.\varphi] \to \exists x.C_\sigma[\varphi]$ if $x \notin FV(C_\sigma[\exists x.\varphi])$
(Framing)	$\dfrac{\varphi_1 \to \varphi_2}{C_\sigma[\varphi_1] \to C_\sigma[\varphi_2]}$
(Existence)	$\exists x.\, x$
(Singleton Variable)	$\neg(C_1[x \wedge \varphi] \wedge C_2[x \wedge \neg\varphi])$ where C_1 and C_2 are nested symbol contexts.
(Set Variable Substitution)	$\dfrac{\varphi}{\varphi[\psi/X]}$
(Pre-Fixpoint)	$\varphi[\mu X.\,\varphi/X] \to \mu X.\varphi$
(Knaster-Tarski)	$\dfrac{\varphi[\psi/X] \to \psi}{\mu X.\,\varphi \to \psi}$

Fig. 1. Sound and complete proof system $\mathcal{H}$ of matching logic (above the double line) and the proof system $\mathcal{H}_\mu$ of matching μ-logic

fourth category contains two technical proof rules that are needed for the completeness result of $\mathcal{H}$. Notice that unlike $\mathcal{P}$, all proof rules in $\mathcal{H}$ are general rules and do not depend on any special symbols such as the definedness symbols.

Definition 11. Let Γ be an axiom set and φ be a pattern. As usual, we write $\Gamma \vdash_{\mathcal{H}} \varphi$ iff φ can be proved by the proof system $\mathcal{H}$ with the patterns in Γ as additional axioms.

There are two interesting observations about $\mathcal{H}$. Firstly, the (Framing) rule allows us to lift the result of local reasoning through any symbol contexts, and thus supports *compositional reasoning* in ML. Secondly, the three propagation axioms plus the (Framing) rule inspire a close relationship between ML and modal logics, where the *ML symbols* and the *modal logic modalities* are dual to each other, as illustrated below:

Proposition 12. *Let $\sigma \in \Sigma_{s_1 \ldots s_n, s}$ and define its "dual" as $\bar{\sigma}(\varphi_1, \ldots, \varphi_n) \equiv \neg\sigma(\neg\varphi_1, \ldots, \neg\varphi_n)$. Then we have:*

- *(K): $\vdash_{\mathcal{H}} \bar{\sigma}(\varphi_1 \to \varphi'_1, \ldots, \varphi_n \to \varphi'_n) \to (\bar{\sigma}(\varphi_1, \ldots, \varphi_n) \to \bar{\sigma}(\varphi'_1, \ldots, \varphi'_n))$; and*
- *(N): $\vdash_{\mathcal{H}} \varphi_i$ implies $\vdash_{\mathcal{H}} \bar{\sigma}(\varphi_1, \ldots, \varphi_i, \ldots, \varphi_n)$.*

In particular, when $n = 1$, we obtain the normal modal logic (K) rule and (N) rule [17].

We present some important properties about the proof system $\mathcal{H}$. The first one is the soundness theorem.

Theorem 13 (Soundness of $\mathcal{H}$)**.** *$\Gamma \vdash_{\mathcal{H}} \varphi$ implies $\Gamma \vDash_{\mathsf{ML}} \varphi$.*

The second property is a version of *deduction theorem*, which requires definedness symbols and axioms.

Theorem 14 (Deduction Theorem of $\mathcal{H}$)**.** *Let Γ be an axiom set containing definedness symbols and axioms (see Definition 7), and let φ, ψ be two patterns. If $\Gamma \cup \{\psi\} \vdash_{\mathcal{H}} \varphi$ and the proof does not use (Universal Generalization) on free variables in ψ, then $\Gamma \vdash_{\mathcal{H}} \lfloor\psi\rfloor \to \varphi$. In particular, if ψ is closed, then $\Gamma \cup \{\psi\} \vdash_{\mathcal{H}} \varphi$ implies $\Gamma \vdash_{\mathcal{H}} \lfloor\psi\rfloor \to \varphi$. Notice that $\lfloor\psi\rfloor$ is an abbreviation of $\lfloor\psi\rfloor^s_{s'}$ if φ has sort s and ψ has sort s'. Also, the reverse theorem holds: $\Gamma \vdash_{\mathcal{H}} \lfloor\psi\rfloor \to \varphi$ implies $\Gamma \cup \{\psi\} \vdash_{\mathcal{H}} \varphi$, without any additional conditions.*

The verbose condition about (Universal Generalization) in Theorem 14 also appears in the deduction theorem in FOL (see, for example, [16]). Notice that we can not conclude $\Gamma \vdash_{\mathcal{H}} \psi \to \varphi$ in general. The theorem is proved by an induction on the length of the proof, but we here instead give an intuitive semantic explanation. Suppose $\Gamma \cup \{\psi\} \vDash_{\mathsf{ML}} \varphi$ for some closed pattern ψ (so we can ignore valuations). Then for all models $M \vDash_{\mathsf{ML}} \Gamma$, if ψ holds then φ also holds. This actually means $M \vDash_{\mathsf{ML}} \lfloor\psi\rfloor \to \varphi$, as $\lfloor\psi\rfloor$ is evaluated to the empty set if ψ does not hold in M. Note that $M \vDash_{\mathsf{ML}} \psi \to \varphi$ is too strong as a conclusion, for it requires the valuation of ψ is always contained in φ, even in models M where ψ does not hold.

The third property is that we can prove all proof rules in $\mathcal{P}$ using the new proof system $\mathcal{H}$, with definedness axioms as additional axioms. This immediately gives us the following completeness result of $\mathcal{H}$ as a corollary of Theorem 9.

Theorem 15. *For all axiom sets Γ containing (Definedness) axioms and all patterns φ, we have $\Gamma \vDash_{\mathsf{ML}} \varphi$ implies $\Gamma \vdash_{\mathcal{H}} \varphi$.*

Finally, we state our main completeness result for $\mathcal{H}$:

Theorem 16 (Completeness of $\mathcal{H}$). *$\vDash_{ML} \varphi$ implies $\vdash_{\mathcal{H}} \varphi$.*

The proof of Theorem 16 is rather complex (see Appendix D). We drew inspiration from Blackburn and Tzakova [18], who proved a completeness result for a version of *hybrid modal logic* with the $\forall$-binder, using a mixture of modal and first-order techniques: the idea of *canonical models* from modal logic and the idea of *witnessed sets* from first-order logic. Theorem 16 can be seen as a nontrivial generalization of the completeness result in [18]. Specifically, we extend the hybrid modal logic of [18] in two directions. First, we consider multiple sorts, each coming with its own universe of worlds and logical infrastructure; the approach in [18] has only one sort, that of "formulae". Second, we allow arbitrarily many modal operators of arbitrary arities (see Proposition 12); the approach in [18] only considers the usual, unary "necessity" modal operator $\Box\_$ (and its dual $\Diamond\_$). Polyadic, non-hybrid (i.e., without $\forall$-binder) variants of modal logic are known (see, e.g., [19]), but at our knowledge our work in this paper is the first to combine polyadic modal operators and FOL quantifiers.

IV. From Matching Logic to Matching μ-Logic

In this section, we extend ML with the least fixpoint μ-binder. We call the extended logic *matching μ-logic* (MmL), and study its syntax, semantics, and proof system. Many definitions, notations, and properties of ML that are introduced in Section II and III also work for MmL, so we only focus on parts where they differ to prevent redundancy.

A. Matching μ-logic syntax

Definition 17. A *matching μ-logic signature* $\mathbb{\Sigma} = (S, \textsc{Var}, \Sigma)$ or simply a *signature* is the same as a matching logic signature *except that* $\textsc{Var} = \textsc{EVar} \cup \textsc{SVar}$ is now a disjoint union of two S-indexed sets of variables: the *element variables* denoted as $x{:}s$, $y{:}s$, etc. in $\textsc{EVar}$, and the *set variables* denoted as $X{:}s, Y{:}s$, etc in $\textsc{SVar}$. *Matching μ-logic $\mathbb{\Sigma}$-patterns*, or simply *$\mathbb{\Sigma}$-patterns* or just *patterns*, are defined inductively by the following grammar for all sorts $s, s' \in S$:

$$\varphi_s ::= x{:}s \in \textsc{EVar}_s \mid X{:}s \in \textsc{SVar}_s \mid \cdots \\ \mid \mu X{:}s.\varphi_s \quad \text{if } \varphi_s \text{ is positive in } X{:}s,$$

where the "..." part is the same as in ML. We say φ_s is positive in $X{:}s$ if every free occurrence of $X{:}s$ is under an even number of negations, where for counting negations the formula $\varphi_1 \to \varphi_2$ is interpreted as $\neg\varphi_1 \vee \varphi_2$. We let $\textsc{Pattern}(\mathbb{\Sigma}) = \{\textsc{Pattern}_s\}_{s \in S}$ denote the set of all matching μ-logic $\mathbb{\Sigma}$-patterns and feel free to drop the signature $\mathbb{\Sigma}$.

From now on, we automatically assume we are talking about MmL unless we explicitly say otherwise.

Intuitively, element variables are like ML variables in that they are evaluated to *elements*, while set variables are evaluated to *subsets*. The least fixpoint pattern $\mu X{:}s.\varphi_s$ gives *the least solution* (under the subset relation) of the equation $X{:}s = \varphi_s$ of set variable $X{:}s$, and the condition of positive occurrence guarantees the existence of such a least solution. The notion of free variables, α-renaming, and capture-avoiding substitution are extended to set variables and the μ-binder. The dual version of the least fixpoint μ-binder is the *greatest fixpoint* ν-binder, defined as $\nu X{:}s.\varphi_s \equiv \neg\mu X{:}s.\neg\varphi_s[\neg X{:}s/X{:}s]$, given that φ_s is positive in $X{:}s$, (which implies that $\neg\varphi_s[\neg X{:}s/X{:}s]$ is also positive in $X{:}s$, justifying the definition).

B. Matching μ-logic semantics

We first review a variant of the Knaster-Tarski theorem [20]:

Theorem 18 (Knaster-Tarski)**.** *Let M be a nonempty set and $\mathcal{F}\colon \mathcal{P}(M) \to \mathcal{P}(M)$ be a monotone function, i.e., $\mathcal{F}(A) \subseteq \mathcal{F}(B)$ for all subsets $A \subseteq B$ of M. Then $\mathcal{F}$ has a unique least fixpoint $\mu\mathcal{F}$ and a unique greatest fixpoint $\nu\mathcal{F}$, where:*

$$\mu\mathcal{F} = \bigcap \{A \in \mathcal{P}(M) \mid \mathcal{F}(A) \subseteq A\},$$
$$\nu\mathcal{F} = \bigcup \{A \in \mathcal{P}(M) \mid A \subseteq \mathcal{F}(A)\}.$$

We call A a pre-fixpoint of $\mathcal{F}$ *whenever $\mathcal{F}(A) \subseteq A$, and a* post-fixpoint of $\mathcal{F}$ *whenever $A \subseteq \mathcal{F}(A)$.*

MmL models are exactly ML models where sorts are associated with their carrier sets and symbols are interpreted as relations. Valuations are extended such that element variables are mapped to elements and set variables are mapped to subsets. Patterns are evaluated the same way for the ML constructs, but extended with the valuation of least fixpoint patterns $\mu X{:}s.\,\varphi$ as the *true least fixpoints* in models. Formally:

Definition 19. Let $\mathbb{\Sigma} = (S, \textsc{Var}, \Sigma)$ be a signature with $\textsc{Var} = \textsc{EVar} \cup \textsc{SVar}$, and $M = (\{M_s\}_{s \in S}, \_^M)$ be a $\mathbb{\Sigma}$-model. A *valuation* $\rho\colon \textsc{Var} \to (M \cup \mathcal{P}(M))$ is a function such that $\rho(x) \in M_s$ for all $x \in \textsc{EVar}_s$ and $\rho(X) \in \mathcal{P}(M_s)$ for all $X \in \textsc{SVar}_s$. Its *extension* $\bar{\rho}\colon \textsc{Pattern} \to \mathcal{P}(M)$ is defined as in Definition 4, extended with:

- $\bar{\rho}(x) = \{\rho(x)\}$ for all $x \in \textsc{EVar}_s$;
- $\bar{\rho}(X) = \rho(X)$ for all $X \in \textsc{SVar}_s$;
- $\bar{\rho}(\mu X.\,\varphi) = \mu\mathcal{F}_{\varphi,X}$ for all $X{:}\textsc{SVar}_s$, where $\mathcal{F}_{\varphi,X}(A) = \overline{\rho[A/X]}(\varphi)$ for all $A \subseteq M_s$.

Here $\rho[A/X]$ denotes the valuation ρ' such that $\rho'(X) = A$ and $\rho'(Y) = \rho(Y)$ for all $Y \neq X$. Notice that we need to verify that $\mathcal{F}_{\varphi,X}$ is monotone. This is done by using the fact that φ is positive in X, and we omit the verification details. The notions $M \vDash \varphi$, $\Gamma \vDash \varphi$, and $M \vDash \Gamma$ for all MmL models M, patterns φ, and axiom sets Γ are defined in the expected way.

Proposition 20. *For all axiom sets Γ of matching logic patterns (without μ) and all matching logic patterns φ (without μ), we have $\Gamma \vDash_{ML} \varphi$ if and only if $\Gamma \vDash \varphi$.*

C. Example: capturing precisely term algebras

Many approaches to specifying formal semantics of programming languages are applications of *initial algebra semantics* [21]. In this subsection, we show how *term algebras*, a particular example of initial algebras, can be *precisely captured* using MmL patterns as axioms. For simplicity, we discuss only *single-sorted term algebras*, but the result can be extended to the many-sorted settings without any major technical difficulties using the techniques introduced in Section V.

Definition 21. Let $\mathbb{\Sigma} = (\{Term\}, \Sigma)$ be a signature with one sort *Term* and at least one constant symbol. A $\mathbb{\Sigma}$*-term* or simply a *term* is inductively defined as follows:

$$t ::= c \in \Sigma_{\lambda,Term} \mid c(t_1, \ldots, t_n) \text{ for } c \in \Sigma_{Term \ldots Term, Term}$$

The $\mathbb{\Sigma}$*-term algebra* $T^{\mathbb{\Sigma}} = (\{T^{\mathbb{\Sigma}}_{Term}\}, \_^{T^{\mathbb{\Sigma}}})$ consisting of:

- a carrier set $T^{\mathbb{\Sigma}}_{Term}$ containing all $\mathbb{\Sigma}$-terms;
- a function $c^{T^{\mathbb{\Sigma}}} \colon T^{\mathbb{\Sigma}}_{Term} \times \cdots \times T^{\mathbb{\Sigma}}_{Term} \to T^{\mathbb{\Sigma}}_{Term}$ for all symbols $c \in \Sigma_{Term \ldots Term, Term}$ defined as $c^{T^{\mathbb{\Sigma}}}(t_1, \ldots, t_n) = c(t_1, \ldots, t_n)$.

Proposition 22. *Let $\mathbb{\Sigma} = (\{Term\}, \Sigma)$ be a signature with one sort Term and at least one constant symbol. Define a $\mathbb{\Sigma}$-theory $\Gamma^{\text{term}}_{\mathbb{\Sigma}}$ with (Function) and (No Confusion) axioms[2] for all symbols in Σ, plus the following axiom:*

$$\text{(Inductive Domain)} \quad \mu D. \bigvee_{c \in \Sigma} c(D, \ldots, D)$$

Then for all $\mathbb{\Sigma}$-models $M \vDash \Gamma^{\text{term}}_{\mathbb{\Sigma}}$, M is isomorphic to $T^{\mathbb{\Sigma}}$. In addition, for all extended signatures $\mathbb{\Sigma}^+ \supseteq \mathbb{\Sigma}$ and $\mathbb{\Sigma}^+$-models $M \vDash \Gamma^{\text{term}}_{\mathbb{\Sigma}}$, we have $M\big|_{\mathbb{\Sigma}}$ is isomorphic to $T^{\mathbb{\Sigma}}$, where $M\big|_{\mathbb{\Sigma}}$ is the reduct model *of M over the sub-signature $\mathbb{\Sigma}$.*

Intuitively, the (Inductive Domain) axiom forces that for all models M, the carrier set M_{Term} must be the *the smallest set* that is closed under all symbols in Σ, while the (Function) and (No Confusion) axioms forces all symbols in Σ to be interpreted as injective functions, and different constructors construct different terms.

Proposition 22 immediately tells us that MmL cannot have a proof system that is both sound and complete, because one can capture precisely the model $(\mathbb{N}, +, \times)$ of natural numbers with addition and multiplication with MmL axioms, and the model $(\mathbb{N}, +, \times)$, by Gödel's first incompleteness theorem [22], is not axiomatizable.

Proposition 23. *Let $\mathbb{\Sigma} = (\{Nat\}, \{0 \in \Sigma_{\lambda,Nat}, succ \in \Sigma_{Nat,Nat}\})$ and the $\mathbb{\Sigma}$-theory $\Gamma^{\text{term}}_{\mathbb{\Sigma}}$ be defined as in Proposition 22, where the (Inductive Domain) takes the following form:*

$$\text{(Inductive Domain)} \quad \mu D \,.\, 0 \vee succ(D)$$

Let the signature $\mathbb{\Sigma}^{\mathbb{N}}$ extend $\mathbb{\Sigma}$ with two functions:

$$plus \colon Nat \times Nat \to Nat \qquad mult \colon Nat \times Nat \to Nat$$

and the $\mathbb{\Sigma}^{\mathbb{N}}$-theory $\Gamma^{\mathbb{N}}$ extend $\Gamma^{\text{term}}_{\mathbb{\Sigma}}$ with the standard axioms:

$$plus(0, y) = y \qquad plus(s(x), y) = s(plus(x, y))$$
$$mult(0, y) = 0 \qquad mult(s(x), y) = plus(y, mult(x, y))$$

Then, $\Gamma^{\mathbb{N}}$ captures precisely $(\mathbb{N}, +, \times)$, meaning that for all models $M \vDash \Gamma^{\mathbb{N}}$, M is isomorphic to $(\mathbb{N}, +, \times)$.

We finish this subsection by comparing Proposition 22 with the nontrivial result that the term algebra $T^{\mathbb{\Sigma}}$ has a *complete axiomatization* in FOL where the only predicate symbol is equality [23]. We refer to this complete FOL axiomatization as $\Gamma_{\mathsf{FOL}}(T^{\mathbb{\Sigma}})$. This means that for all FOL formulas φ, $\Gamma_{\mathsf{FOL}}(T^{\mathbb{\Sigma}}) \vDash_{\mathsf{FOL}} \varphi$ if and only if $T^{\mathbb{\Sigma}} \vDash_{\mathsf{FOL}} \varphi$. This result is *weaker* than Proposition 22, because by Löwenheim-Skolem theorem [24], the FOL theory $\Gamma_{\mathsf{FOL}}(T^{\mathbb{\Sigma}})$ has models of arbitrarily large cardinalities (if $\mathbb{\Sigma}$ contains at least one non-constant constructors), meaning that there are models $M \vDash_{\mathsf{FOL}} \Gamma_{\mathsf{FOL}}(T^{\mathbb{\Sigma}})$ with *strictly more elements* than $T^{\mathbb{\Sigma}}$, and thus cannot be isomorphic to $T^{\mathbb{\Sigma}}$. It is just the case that M (and all FOL models of $\Gamma_{\mathsf{FOL}}(T^{\mathbb{\Sigma}})$) satisfies exactly the same FOL formulas as $T^{\mathbb{\Sigma}}$, known in literature as *elementary equivalence*. Proposition 22, on the other hand, shows that the MmL theory $\Gamma^{\text{term}}_{\mathbb{\Sigma}}$ captures $T^{\mathbb{\Sigma}}$ *up to isomorphism.*

D. Matching μ-logic proof system

Proposition 23 implies that *MmL cannot have a sound and complete proof system*. The best we can do then is to aim for a *proof system that is good enough in practice*. We take the ML proof system $\mathcal{H}$ and extend it with three additional proof rules (see Fig. 1). Rules (Pre-Fixpoint) and (Knaster-Tarski) are standard proof rules about least fixpoints as in modal μ-logic [8]. Rule (Set Variable Substitution) allows us to prove from $\vdash \varphi$ any substitution $\vdash \varphi[\psi/X]$ for $X \in \text{SVar}$. Note the condition that X is a set variable is crucial. In general, we *cannot* prove from $\vdash \varphi$ that $\vdash \varphi[\psi/x]$ for $x \in \text{EVar}$, because it does not hold semantically. As shown in [1], it only holds when ψ is *functional*, that is, when ψ evaluates to a singleton set. Indeed, suppose that ψ is not functional, say it is the pattern $0 \vee succ(0)$ over the signature of natural numbers in Proposition 23, which evaluates to a set of two elements. Then we can pick φ to be the tautology $\exists y \,.\, x = y$, and then $\varphi[\psi/x]$ becomes $\exists y \,.\, \psi = y$, which states that ψ evaluates to a singleton set (the valuation of y), which is a contradiction.

We let $\mathcal{H}_\mu$ denote the extended 16-rule proof system in Fig. 1, and from here on we write $\Gamma \vdash \varphi$ instead of $\Gamma \vdash_{\mathcal{H}_\mu} \varphi$.

Theorem 24 (Soundness of $\mathcal{H}_\mu$)**.** *$\Gamma \vdash \varphi$ implies $\Gamma \vDash \varphi$.*

E. Instance: Peano arithmetic

The purpose of this subsection is to illustrate the power of the two proof rules (Pre-Fixpoint) and (Knaster-Tarski), by showing that they derive the (Induction) axiom schema in the FOL axiomatization of Peano arithmetic [25], [26]:

$$\text{(Induction)} \quad \varphi(0) \wedge \forall x.(\varphi(x) \to \varphi(succ(x))) \to \forall x.\varphi(x)$$

where $\varphi(x)$ is a FOL formula with a distinguished variable x.

We encode the FOL syntax of Peano arithmetic following the technique in Section II-D, that is, we define a signature $\mathbb{\Sigma}^{\mathsf{Peano}} = (\{Nat, Pred\}, \Sigma^{\mathbb{N}})$ where $\Sigma^{\mathbb{N}}$ is defined in Proposition 23 that contains the functions $0, succ, plus, mult$, and let Γ^{Peano} contain the same equations as axioms as $\Gamma^{\mathbb{N}}$. Notice that the only $\mathbb{\Sigma}^{\mathsf{Peano}}$-patterns of sort *Pred* are those built from equalities between two patterns of sort *Nat*.

Proposition 25. *Under the above notations, we have:*

$$\Gamma^{\mathsf{Peano}} \vdash \varphi(0) \wedge \forall x.(\varphi(x) \to \varphi(succ(x))) \to \forall x.\varphi(x).$$

[2] See Section II-C.

V. Defining Recursive Symbols as Syntactic Sugar

Intuitively, the least fixpoint pattern $\mu X.\varphi$ specifies *a recursive set* that satisfies the equation $X = \varphi$, where φ may contain recursive occurrences of X. For example, the pattern $\mu X.\, 0 \vee succ(succ(X))$ specifies the set of all even numbers, which conceptually defines *a recursive constant symbol*:

$$even \in \Sigma_{Nat,Nat} \qquad even =_{\text{lfp}} 0 \vee succ(succ(even)).$$

Here, "$=_{\text{lfp}}$" is merely a notation, meaning that we want *even* to be the least set that satisfies the equation (since the total set is always a trivial solution).

The challenge is how to generalize the above and define *recursive non-constant symbols*. For example, suppose we want to define a symbol $collatz \in \Sigma_{Nat,Nat}$ as follows:

$$collatz(n) =_{\text{lfp}} \\ n \vee (even(n) \wedge collatz(n/2)) \vee (odd(n) \wedge collatz(3n+1))$$

with the intuition that $collatz(n)$ gives the set of all numbers in the Collatz sequence[3] starting from n. However, the μ-binder in MmL can only be applied on *set variables*, not on *symbols*, so the following attempt is syntactically wrong:

$$collatz(n) = \mu\, \sigma(n)\,. \qquad \text{// } \mu \text{ can only bind a set variable} \\ n \vee (even(n) \wedge \sigma(n/2)) \vee (odd(n) \wedge \sigma(3n+1))$$

One possible solution could be to *extend* MmL with the above syntax and allow the μ-binder to quantify *symbol variables*, not only over *set variables*. The semantics and proof system could be extended accordingly. This is exactly how *first-order logic with least fixpoints* extends FOL [7]. But do we really have to? After all, our proof rules (Pre-Fixpoint) and (Knaster-Tarski) in Fig. 1 are nothing but a logical incarnation of the Knaster-Tarski theorem, which has been repeatedly demonstrated to serve as a solid if not the main foundation for recursion. Therefore, we conjecture that the $\mathcal{H}$ proof system in Fig. 1 is sufficient in practice, and thus would rather resist extending MmL. That is, we conjecture that it should be possible to *define* one's desired approach to recursion/induction/fixed-points using ordinary MmL *theories*; as an analogy, in Section II-C we showed how we can define definedness, totality, equality, membership, containment, functions, partial functions, etc. (see [1] for more) as theories, without a need to extend matching logic.

In particular, we can solve the recursive symbol challenge above by using the *principle of currying-uncurrying* to "mimic" the unary symbol $collatz \in \Sigma_{Nat,Nat}$ with a constant symbol $collatz \in \Sigma_{\lambda,Nat\otimes Nat}$, where $Nat\otimes Nat$ is the *product sort* (defined later; the intuition is that $Nat \otimes Nat$ has the product set $\mathbb{N} \times \mathbb{N}$ as its carrier set), and thus reducing the challenge of defining a least relation in $[\mathbb{N} \to \mathcal{P}(\mathbb{N})]$ to defining a least subset of $\mathcal{P}(\mathbb{N} \times \mathbb{N})$, without the need to extend the logic.

[3] A Collatz sequence starting from $n \geq 1$ is obtained by repeating the following procedure: if n is even then return $n/2$; otherwise, return $3n+1$.

A. Principle of currying-uncurrying and product sorts

The principle of currying-uncurrying [27], [28] is used in various settings (e.g., simply-typed lambda calculus [29]) as a means to reduce the study of multi-argument functions to the simpler single-argument functions. We here present the principle in its adapted form that fits best with our needs.

Proposition 26. *Let $M_{s_1}, \ldots, M_{s_n}, M_t$ be nonempty sets. The* principle of currying-uncurring *means the isomorphism*

$$\mathcal{P}(M_{s_1} \times \cdots \times M_{s_n} \times M_t) \underset{uncurry}{\overset{curry}{\rightleftharpoons}} [M_{s_1} \times \cdots \times M_{s_n} \to \mathcal{P}(M_t)]$$

defined for all $a_1 \in M_{s_1}, \ldots, a_n \in M_{s_n}, b \in M_t, \alpha \subseteq M_{s_1} \times \cdots \times M_{s_n} \times M_t$, and $f : M_{s_1} \times \cdots \times M_{s_n} \to \mathcal{P}(M_t)$ as:

$$curry(\alpha)(a_1, \ldots, a_n) = \{b \in M_t \mid (a_1, \ldots, a_n, b) \in \alpha\} \\ uncurry(f) = \{(a_1, \ldots, a_n, b) \mid b \in f(a_1, \ldots, a_n)\}.$$

In other words, we can mimic an n-ary symbol $\sigma \in \Sigma_{s_1 \ldots s_n, t}$ with a constant symbol of the *product sort* $s_1 \otimes \cdots \otimes s_n \otimes t$, whose (intended) carrier set is exactly the product set $M_{s_1} \times \ldots M_{s_n} \times M_t$. This leads to the following definition.

Definition 27. Let s, t be two sorts, not necessarily distinct. The *product sort* $s \otimes t$ is a sort that is different from s and t. The *pairing symbol* $\langle \_, \_ \rangle_{s,t} : s \times t \to s \otimes t$ is a function symbol and the *projection symbol* $\_(\_)_{s,t} : s \otimes t \times s \rightharpoonup t$ is a partial function symbol. These are governed by the axioms

(Injectivity)	$(\langle k, v \rangle = \langle k', v' \rangle) \to (k = k') \wedge (v = v')$
(Key-Value)	$\langle k, v \rangle(k') = (k = k') \wedge v$
(Product Domain)	$\exists k \exists v. \langle k, v \rangle$

forcing the carrier of $s \otimes t$ to be the product of the carriers of s and t, and pairing/projection are interpreted as expected.

Product of multiple sorts as well as the associated pairing/projection operations can be defined as derived constructs as follows. Let $s_1, \ldots, s_n, t$ be sorts, not necessarily distinct, and $\varphi_1, \ldots, \varphi_n, \varphi, \psi$ be patterns of appropriate sorts. We define:

$$s_1 \otimes \cdots \otimes s_n \otimes t \equiv s_1 \otimes (s_2 \otimes (\cdots \otimes (s_n \otimes t) \ldots)) \\ \langle \varphi_1, \ldots, \varphi_n, \varphi \rangle \equiv \langle \varphi_1, \langle \ldots, \langle \varphi_n, \varphi \rangle \ldots \rangle \rangle \\ \psi(\varphi_1, \ldots, \varphi_n) \equiv \psi(\varphi_1) \ldots (\varphi_n).$$

Notice that we tactically use the same syntax $\_(\_, \ldots, \_)$ for both symbol applications and projections to blur their distinction. In particular, if $\sigma \in \Sigma_{\lambda, s_1 \otimes \cdots \otimes s_n \otimes t}$ is a constant symbol of the product sort, then $\sigma(\varphi_1, \ldots, \varphi_n)$ is a well-formed pattern if $\varphi_1, \ldots, \varphi_n$ have appropriate sorts.

B. Defining recursive symbols in matching μ-logic

Definition 28. Let $\mathbb{\Sigma} = (S, \Sigma)$ be a signature and $\sigma \in \Sigma_{s_1 \ldots s_n, s}$. We use the notation $\sigma(x_1, \ldots, x_n) =_{\text{lfp}} \varphi$ to mean the axiom:

$$\sigma(x_1, \ldots, x_n) = \\ (\mu \sigma{:}s_1 \otimes \cdots \otimes s_n \otimes t. \exists x_1 \ldots \exists x_n. \langle x_1, \ldots, x_n, \varphi \rangle)(x_1, \ldots, x_n)$$

A symbol $\sigma \in \Sigma_{s_1 \ldots s_n, s}$ obeying this axiom is called *recursive*.

The following theorem says that if φ "behaves like a symbol", meaning that it has the property of propagation (Proposition 3), then we can obtain variants of (Pre-Fixpoint) and (Knaster-Tarski) for the recursive symbol σ.

Theorem 29. *Let $\mathbb{\Sigma}$ be a signature with a recursive symbol $\sigma \in \Sigma_{s_1 \ldots s_n, t}$ defined as $\sigma(x_1, \ldots, x_n) =_{\mathsf{lfp}} \varphi$. Let Γ be a $\mathbb{\Sigma}$-theory such that for all $\varphi_1, \ldots, \varphi_n$:*

$$\Gamma \vdash (\exists z_1 \ldots \exists z_n . z_1 \in \varphi_1 \wedge \cdots \wedge z_n \in \varphi_n \wedge \varphi[z_1/x_1, \ldots, z_n/x_n]) \rightarrow \varphi[\varphi_1/x_1, \ldots, \varphi_n/x_n].$$

Then the following hold:

- *Pre-Fixpoint: $\Gamma \vdash \varphi \rightarrow \sigma(x_1, \ldots, x_n)$;*
- *Knaster-Tarski: if $\Gamma \vdash \varphi[\psi/\sigma] \rightarrow \psi$ then $\Gamma \vdash \sigma(x_1, \ldots, x_n) \rightarrow \psi$, where $\varphi[\psi/\sigma]$ is the result of replacing all patterns of the form $\sigma(\varphi_1, \ldots, \varphi_n)$ in φ with $\psi[\varphi_1/x_1, \ldots, \varphi_n/x_n]$.*

VI. Instance: First-Order Logic with Least Fixpoints

First-order logic with least fixpoints (LFP) [7] extends the syntax of first-order logic formulas with:

$$\varphi ::= [\mathsf{lfp}_{R, x_1, \ldots, x_n} \varphi](t_1, \ldots, t_n)$$

where R is a *predicate variable* and φ is a formula that is positive in R. Intuitively, "$[\mathsf{lfp}_{R, x_1, \ldots, x_n} \varphi]$" behaves as the least fixpoint predicate of the operation that maps R to φ. Due to its complexity and our limited space, we skip the formal definition of the semantics and simply denote the validity relation in LFP as $\models_{\mathsf{LFP}} \varphi$. A comprehensive study on LFP can be found in [30].

Given the notations of recursive symbols defined in Section V, it is straightforward to subsume LFP by extending the theory Γ^{FOL} defined in Section II-D with:

$$[\mathsf{lfp}_{R, x_1, \ldots, x_n} \varphi](t_1, \ldots, t_n) \equiv (\mu R\colon s_1 \otimes \ldots \otimes s_n \otimes \mathit{Pred} . \exists x_1 \ldots \exists x_n . \langle x_1, \ldots, x_n, \varphi \rangle)(t_1, \ldots, t_n)$$

for all predicate variables R with argument sorts $s_1, \ldots, s_n$. What is different is that we add one additional axiom, $\forall x{:}\mathit{Pred} \forall y{:}\mathit{Pred} . x = y$, to constrain the (dummy) carrier set of *Pred* is a singleton set, so that all MmL models are also FOL/LFP models. This fact is used to prove the "only if" part in the next theorem.[4] We denote the resulting theory Γ^{LFP}.

Theorem 30. *If φ is any LFP formula, then $\models_{\mathsf{LFP}} \varphi$ iff $\Gamma^{\mathsf{LFP}} \models \varphi$.*

VII. Instances: Modal μ-Calculus and Temporal Logics

We have seen how MmL symbols and patterns can be used to specify both structure and constraints, such as terms (Section IV-C) and FOL (Section II-D), as well as various induction, recursion and least-fixed point schemas (Sections IV-E and V) over these. These suffice to express and prove program assertions, including complex state abstractions (see also how separation logic falls as a fragment of matching logic in [1]), in contexts where MmL is chosen as a static state assertion formalism in program verification frameworks based on Hoare logic [31], dynamic logic [11], or reachability logic [2]. However, as explained in Section I, our ultimate goal is to support not only static state assertions, but any program properties, including ones that are usually specified using Hoare, dynamic, or reachability logics. We start the discussion in this section, by showing how MmL symbols and patterns can also be used to specify *dynamic transition relations* such as modal μ-logic modalities and dynamic logic; in Section VIII we then discuss how MmL also subsumes reachability logic, which subsumes Hoare logic [6].

A. Modal μ-logic syntax, semantics, and proof system

The *syntax* of modal μ-logic [8] is parametric on a countably infinite set PVar of propositional variables. Modal μ-logic *formulas* are given by the grammar[5]:

$$\varphi ::= p \in \mathrm{PVar} \mid \varphi \wedge \varphi \mid \neg\varphi \mid \circ\varphi \mid \mu X . \varphi \text{ if } \varphi \text{ is positive in } X$$

Derived constructs are defined as usual, e.g., $\bullet\varphi \equiv \neg\circ\neg\varphi$. Modal μ-logic semantics is given using *transition systems* $\mathbb{S} = (S, R)$, with S a nonempty set of *states* and $R \subseteq S \times S$ a *transition relation*, and valuations $V\colon \mathrm{PVar} \rightarrow \mathcal{P}(S)$, as follows:

- $[\![p]\!]^{\mathbb{S}}_V = V(p)$;
- $[\![\varphi_1 \wedge \varphi_2]\!]^{\mathbb{S}}_V = [\![\varphi_1]\!]^{\mathbb{S}}_V \cap [\![\varphi_2]\!]^{\mathbb{S}}_V$;
- $[\![\neg\varphi]\!]^{\mathbb{S}}_V = S \setminus [\![\varphi]\!]^{\mathbb{S}}_V$;
- $[\![\circ\varphi]\!]^{\mathbb{S}}_V = \{s \in S \mid s\, R\, t \text{ implies } t \in [\![\varphi]\!]^{\mathbb{S}}_V \text{ for all } t \in S\}$;
- $[\![\mu X . \varphi]\!]^{\mathbb{S}}_V = \bigcap \{A \subseteq S \mid [\![\varphi]\!]^{\mathbb{S}}_{V[A/X]} \subseteq A\}$;

A modal μ-logic formula φ is *valid*, denoted $\models_\mu \varphi$, if for all transition systems $\mathbb{S}$ and all valuations V, we have $[\![\varphi]\!]^{\mathbb{S}}_V = S$. A proof system of modal μ-logic is firstly given in [8] and then proved to be complete in [32]. It extends the proof system of propositional logic with the following proof rules:

$$\text{(K)}\quad \circ(\varphi_1 \rightarrow \varphi_2) \rightarrow (\circ\varphi_1 \rightarrow \circ\varphi_2) \qquad \text{(N)}\quad \frac{\varphi}{\circ\varphi}$$

$$(\mu_1)\quad \varphi[(\mu X . \varphi)/X] \rightarrow \mu X . \varphi \qquad (\mu_2)\quad \frac{\varphi[\psi/X] \rightarrow \psi}{\mu X . \varphi \rightarrow \psi}$$

We denote the corresponding provability relation as $\vdash_\mu \varphi$. Notice that (K) and (N) are provable in MmL (Proposition 12), and (μ_1) and (μ_2) are exactly (Pre-Fixpoint) and (Knaster-Tarski). This means that we can easily mimic all modal μ-logic proofs in MmL (i.e. "(2) $\Rightarrow$ (3)" in Theorem 31).

B. Defining modal μ-logic in matching μ-logic

To subsume the syntax, we define a signature (of *transition systems*) $\mathbb{\Sigma}^{\mathsf{TS}} = (\{\mathit{State}\}, \{\bullet \in \Sigma^{\mu}_{\mathit{State}, \mathit{State}}\})$ where we call the symbol "$\bullet$" *one-path next*. We regard propositional variables in PVar as set variables. We write $\bullet\varphi$ instead of $\bullet(\varphi)$, and define $\circ\varphi \equiv \neg\bullet\neg\varphi$. Then every modal μ-logic formula φ is an MmL $\mathbb{\Sigma}^{\mathsf{TS}}$-pattern of sort *State*. Finally, note that no axioms are needed; let Γ^{μ} be the empty $\mathbb{\Sigma}^{\mathsf{TS}}$-theory.

An important observation is that the $\mathbb{\Sigma}^{\mathsf{TS}}$-models are *exactly* the transition systems, where $\bullet \in \Sigma^{\mathsf{TS}}_{\mathit{State}, \mathit{State}}$ is interpreted as

[4] We do not need that axiom in defining FOL in ML, as seen in Section II-D, because there the "if" part is proved via a proof theoretical approach, using the completeness proof system of FOL and the fact that we can mimic FOL proofs in ML (see [1]). Since LFP does not have a complete proof system, we have to add additional axioms to constrain more on the MmL models.

[5] The modal μ-logic literature often uses $\Box\varphi$ and $\Diamond\varphi$ instead of $\circ\varphi$ and $\bullet\varphi$. We here use the latter to avoid confusion with the "always" $\Box\varphi$ and "eventually" $\Diamond\varphi$ in LTL and CTL.

the transition relation R. Specifically, for any transition system $\mathbb{S} = (S, R)$, we can regard $\mathbb{S}$ as a $\mathbb{\Sigma}^{\mathsf{TS}}$-model where S is the carrier set of *State* and $\bullet_{\mathbb{S}}(t) = \{s \in S \mid s\, R\, t\}$ contains all R-*predecessors* of t. This might seem counter-intuitive at the first glance: why "one-path next" is interpreted as the predecessor instead of the successor of R? See the following illustration:

$$\begin{array}{ccccccc} \cdots & s & \xrightarrow{R} & s' & \xrightarrow{R} & s'' & \cdots \quad \text{// states} \\ & \bullet\bullet\varphi & & \bullet\varphi & & \varphi & \text{// patterns} \end{array}$$

In other words, $\bullet\varphi$ is matched by states which *have at least one next state* that satisfies φ, conforming to the intuition. Another interesting observation is about $\bullet\varphi$ and its dual, $\circ\varphi \equiv \neg\bullet\neg\varphi$, called "all-path next". The difference is that $\circ\varphi$ is matched by s if *for all* states t such that $s\, R\, t$, we have t matches φ. In particular, if s has no successor, then s matches $\circ\varphi$ for all φ. This is formally summarized in Proposition 32.

We now feel free to take any transition system $\mathbb{S}$ as an MmL $\mathbb{\Sigma}^{\mathsf{TS}}$-model. The following theorem shows that our definition of modal μ-logic in MmL is *faithful*, both syntactically and semantically. What is interesting about the theorem is its *proof*, which can be applied to other all logics discussed in this paper, and obtain similar results for all of them.

Theorem 31. *The following properties are equivalent for all modal μ-logic formulas φ: (1) $\vDash_\mu \varphi$; (2) $\vdash_\mu \varphi$; (3) $\Gamma^\mu \vdash \varphi$; (4) $\Gamma^\mu \vDash \varphi$; (5) $M \vDash \varphi$ for all $\mathbb{\Sigma}^{\mathsf{TS}}$-models M such that $M \vDash \Gamma^\mu$; (6) $\mathbb{S} \vDash_\mu \varphi$ for all transition systems $\mathbb{S}$.*

Proof sketch: We only need to prove "(2) $\Rightarrow$ (3)" and "(5) $\Rightarrow$ (6)", as the rest are already proved/known. (1) $\Longrightarrow$ (2) follows by the completeness of modal μ-logic, which is nontrivial but known. (2) $\Longrightarrow$ (3) follows by proving all modal μ-logic proof rules as theorems in MmL. (3) $\Longrightarrow$ (4) follows by the soundness of MmL (Theorem 24). (4) $\Longrightarrow$ (5) follows by definition. (5) $\Longrightarrow$ (6) follows by proving the contrapositive statement, "$\nvDash_\mu \varphi$ implies $\Gamma^\mu \nvDash \varphi$", by taking a transition system $\mathbb{S} = (S, R)$ and a valuation V such that $[\![\varphi]\!]^{\mathbb{S}}_V \neq S$, and showing that if we regard $\mathbb{S}$ as a $\mathbb{\Sigma}^{\mathsf{TS}}$-model and V as an $\mathbb{S}$-valuation in MmL, then $\mathbb{S} \vDash \Gamma^\mu$ and $\overline{V}(\varphi) \neq S$, which means that $\Gamma^\mu \nvDash \varphi$. Finally, (6) $\Longrightarrow$ (1) follows by definition. ∎

Therefore, modal μ-logic can be regarded as an empty theory in a vanilla MmL without quantifiers, over a signature containing only one sort and only one symbol, which is unary. It is worth mentioning that variants of modal μ-logic with more modal operators have been proposed (see [33] for a survey). At our knowledge, however, all such variants consider only unary modal operators and they are only required to obey the usual (K) and (N) proof rules of modal logic. In contrast, MmL allows polyadic symbols while still obeying the desired (K) and (N) rules (see Proposition 12), allows arbitrary further constraining axioms in MmL theories, and also quantification over element variables and many-sorted universes.

C. Studying transition systems in MmL

The above suggests that MmL may offer a unifying playground to specify and reason about transition systems, by means of $\mathbb{\Sigma}^{\mathsf{TS}}$-theories/models. We can define various temporal/dynamic operations and modalities as *derived constructs* from the basic "one-path next" symbol "$\bullet$" and the μ-binder, without the need to extend the syntax and semantics of the logic. We can constrain the models/transition systems of interest using *additional axioms*, without the need to modify/extend the proof system of the logic. In what follows, we show that by defining proper constructs as syntactic sugar and adding proper axioms, we can capture *precisely* LTL (both finite- and infinite-trace), CTL, dynamic logic (DL), and reachability logic (RL).

Let us add more temporal modalities as derived constructs (we have seen "all-path next" $\circ\varphi$ in Section VII-B):

$$\begin{array}{rl} \text{"eventually"} & \Diamond\varphi \equiv \mu X.\, \varphi \vee \bullet X \\ \text{"always"} & \Box\varphi \equiv \nu X.\, \varphi \wedge \circ X \\ \text{"until"} & \varphi_1\, U\, \varphi_2 \equiv \mu X.\, \varphi_2 \vee (\varphi_1 \wedge \bullet X) \\ \text{"well-founded"} & \mathsf{WF} \equiv \mu X. \circ X \end{array}$$

Proposition 32. *Let $\mathbb{S} = (S, R)$ be a transition system regarded as a $\mathbb{\Sigma}^{\mathsf{TS}}$-model, and let ρ be any valuation and $s \in S$. Then:*

- *$s \in \bar{\rho}(\bullet\varphi)$ if there exists $t \in S$ such that $s\, R\, t$, $t \in \bar{\rho}(\varphi)$; in particular, $s \in \bar{\rho}(\bullet\top)$ if s has an R-successor;*
- *$s \in \bar{\rho}(\circ\varphi)$ if for all $t \in S$ such that $s\, R\, t$, $t \in \bar{\rho}(\varphi)$; in particular, $s \in \bar{\rho}(\circ\bot)$ if s has no R-successor;*
- *$s \in \bar{\rho}(\Diamond\varphi)$ if there exists $t \in S$ such that $s\, R^*\, t$, $t \in \bar{\rho}(\varphi)$;*
- *$s \in \bar{\rho}(\Box\varphi)$ if for all $t \in S$ such that $s\, R^*\, t$, $t \in \bar{\rho}(\varphi)$;*
- *$s \in \bar{\rho}(\varphi_1\, U\, \varphi_2)$ if there exists $n \geq 0$ and $t_1, \ldots, t_n \in S$ such that $s\, R\, t_1\, R \cdots R\, t_n$, $t_n \in \bar{\rho}(\varphi_2)$, and $s, t_1, \ldots, t_{n-1} \in \bar{\rho}(\varphi_1)$;*
- *$s \in \bar{\rho}(\mathsf{WF})$ if s is R-well-founded, meaning that there is no infinite sequence $t_1, t_2, \cdots \in S$ with $s\, R\, t_1\, R\, t_2\, R \ldots$;*

where $R^* = \bigcup_{i \geq 0} R^i$ is the reflexive transitive closure of R.

D. Instances: temporal logics

Since MmL can define modal μ-logic (as an empty theory over a unary symbol), it is not surprising that it can also define various temporal logics such as LTL and CTL as theories whose axioms constrain the underlying transition relations. What is interesting, in our view, is that the resulting theories are simple, intuitive, and faithfully capture both the syntax (provability) and the semantics of these temporal logics.

1) Instance: infinite-trace LTL: The LTL syntax, namely

$$\varphi ::= p \in \mathrm{PVar} \mid \varphi \wedge \varphi \mid \neg\varphi \mid \circ\varphi \mid \varphi\, U\, \varphi$$

is already subsumed in MmL with the derived constructs we give in Section VII-C. Other common LTL modalities such as "always $\Box\varphi$" are defined from the "until U" modality in the usual way. Infinite-trace LTL takes as models transition systems whose transition relations are *linear* and *infinite into the future*. We assume readers are familiar with the semantics and proof system of infinite-trace LTL (if not, see [10]) and skip their formal definitions. We use "$\vDash_{\mathsf{infLTL}}$" and "$\vdash_{\mathsf{infLTL}}$" to denote infinite-trace LTL validity and provability, respectively.

To capture the characteristics of both "*infinite future*" and "*linear future*", we add the following two patterns as axioms:

$$(\textsc{Inf})\quad \bullet\top \qquad\qquad (\textsc{Lin})\quad \bullet\varphi \rightarrow \circ\varphi$$

and denote the resulting $\mathbb{\Sigma}^{\mathsf{TS}}$-theory as Γ^{infLTL}. Intuitively, (Inf) forces all states s to have at least one successor, and thus all traces are infinite, and (Lin) forces all states s to have only a *linear future*. The following theorem shows that our

definition of infinite-trace LTL is faithful both syntactically and semantically, proved exactly as Theorem 31.

Theorem 33. *The following properties are equivalent for all infinite-trace LTL formulas φ: (1) $\vdash_{\mathsf{infLTL}} \varphi$; (2) $\models_{\mathsf{infLTL}} \varphi$; (3) $\Gamma^{\mathsf{infLTL}} \vdash \varphi$; (4) $\Gamma^{\mathsf{infLTL}} \models \varphi$.*

Therefore, infinite-trace LTL can be regarded as a theory containing two axioms, (Inf) and (Lin), over the same signature as the theory corresponding to modal μ-logic.

2) Instance: finite-trace LTL: Finite execution traces play an important role in program verification and monitoring. Finite-trace LTL considers models that are *linear* but have only *finite future*. The following *syntax* of finite-trace LTL:

$$\varphi ::= p \in \text{PVar} \mid \varphi \wedge \varphi \mid \neg\varphi \mid \circ\varphi \mid \varphi\, U_w\, \varphi$$

differs from infinite-trace LTL in that the "until" modality "U" is *weak*, meaning that $\varphi_1 U_w \varphi_2$ does not necessarily imply that φ_2 eventually holds. Again, we assume readers are familiar with the semantics and proof system of finite-trace LTL (if not, see [10]) and use "$\models_{\mathsf{finLTL}}$" and "$\vdash_{\mathsf{finLTL}}$" to denote its validity and provability, respectively.

To subsume the above syntax, we define in MmL:

$$\text{"weak until"}\ \varphi_1\, U_w\, \varphi_2 \equiv \mu X.\varphi_2 \vee (\varphi_1 \wedge \circ X).$$

To capture the characteristics of both *finite future* and *linear future*, we add the following two patterns as axioms:

$$\text{(Fin)}\ \ \mathsf{WF} \equiv \mu X.\circ X \qquad \text{(Lin)}\ \ \bullet\varphi \rightarrow \circ\varphi$$

and call the resulting $\mathbb{\Sigma}^{\mathsf{TS}}$-theory Γ^{finLTL}. Intuitively, (Fin) forces all states to be well-founded, meaning that there is no infinite execution trace in the underlying transition systems.

Theorem 34. *The following properties are equivalent for all finite-trace LTL formula φ: (1) $\vdash_{\mathsf{finLTL}} \varphi$; (2) $\models_{\mathsf{finLTL}} \varphi$; (3) $\Gamma^{\mathsf{finLTL}} \vdash \varphi$; (4) $\Gamma^{\mathsf{finLTL}} \models \varphi$.*

Therefore, finite-trace LTL can be regarded as a theory containing two axioms, (Fin) and (Lin), over the same signature as the theory corresponding to modal μ-logic.

3) Instance: CTL: CTL's models are transition systems which are *infinite into the future* and allow states to have a *branching future* (rather than linear). The syntax of CTL is

$$\varphi ::= p \in \text{PVar} \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathsf{AX}\varphi \mid \mathsf{EX}\varphi \mid \varphi\ \mathsf{AU}\ \varphi \mid \varphi\ \mathsf{EU}\ \varphi$$

extended with the following derived constructs:

$$\mathsf{EF}\varphi \equiv \mathit{true}\ \mathsf{EU}\ \varphi \qquad \mathsf{AG}\varphi \equiv \neg\mathsf{EF}\neg\varphi$$
$$\mathsf{AF}\varphi \equiv \mathit{true}\ \mathsf{AU}\ \varphi \qquad \mathsf{EG}\varphi \equiv \neg\mathsf{AG}\neg\varphi$$

The names of the CTL modalities suggest their meaning: the first letter means either "on all paths" (A) or "on one path" (E), and the second letter means "next" (X), "until" (U), "always" (G), or "eventually" (F). For example, "AX" is "all-path next", "EU" is "one-path until", etc. We refer readers to [34] for CTL definitions, semantics and proof system. Here we denote its validity and provability as "$\models_{\mathsf{CTL}}$" and "$\vdash_{\mathsf{CTL}}$", respectively.

To define CTL as an MmL theory, we add only the axiom (Inf) for infinite future and use the following syntactic sugar:

$$\mathsf{AX}\varphi \equiv \circ\varphi \qquad \varphi_1\ \mathsf{AU}\ \varphi_2 \equiv \mu f.\, \varphi_2 \vee (\varphi_1 \wedge \circ f)$$
$$\mathsf{EX}\varphi \equiv \bullet\varphi \qquad \varphi_1\ \mathsf{EU}\ \varphi_2 \equiv \mu f.\, \varphi_2 \vee (\varphi_1 \wedge \bullet f)$$

The resulting $\mathbb{\Sigma}^{\mathsf{TS}}$-theory is denoted as Γ^{CTL}.

Theorem 35. *For all CTL formula φ, the following are equivalent: (1) $\vdash_{CTL} \varphi$; (2) $\models_{CTL} \varphi$; (3) $\Gamma^{CTL} \vdash \varphi$; (4) $\Gamma^{CTL} \models \varphi$.*

Therefore, CTL can be regarded as a theory over the same signature as the theory corresponding to modal μ-logic, but containing one axiom, (Inf). It may be insightful to look at all three temporal logics discussed in this section through the lenses of MmL, as theories over a unary symbol signature: modal μ-logic is the empty and thus the least constrained theory; CTL comes immediately next with only one axiom, (Inf), to enforce infinite traces; infinite-trace LTL further constrains with the linearity axiom (Lin); finally, finite-trace LTL replaces (Inf) with (Fin). We believe that MmL can serve as a convenient and uniform framework to define and study temporal logics. For example, finite-trace CTL can be trivially obtained as the theory containing only the axiom (Fin), LTL with both finite and infinite traces is the theory containing only the axiom (Lin), and CTL with unrestricted (finite or infinite branch) models is the empty theory (i.e., modal μ-logic).

E. Instance: dynamic logic

Dynamic logic (DL) [11]–[13] is a common logic used for program reasoning. The DL syntax is parametric in a set PVar of *propositional variables* and a set APgm of *atomic programs*, each belonging to a different formula syntactic category:

$$\varphi ::= p \in \text{PVar} \mid \varphi \rightarrow \varphi \mid \mathit{false} \mid [\alpha]\varphi$$
$$\alpha ::= a \in \text{APgm} \mid \alpha\,;\alpha \mid \alpha \cup \alpha \mid \alpha^* \mid \varphi?$$

The first line defines *propositional formulas*. The second line defines *program formulas*, which represent programs built from atomic ones with the primitive regular expression constructs. Define $\langle a\rangle\varphi \equiv \neg[\alpha](\neg\varphi)$. Common program constructs such as if-then-else, while-do, etc., can be defined as derived constructs using the four primitive ones; see [11]–[13]. We let "$\models_{\mathsf{DL}}$" and "$\vdash_{\mathsf{DL}}$" denote the validity and provability of DL.

It is known that DL can be embedded in the variant of modal μ-logic with multiple modalities [33]. The idea is to define a modality $[a]$ for every atomic program $a \in \text{APgm}$, and then to define the four program constructs as least/greatest fixpoints. We can easily adopt the same approach and associate an empty MmL theory to DL, over a signature containing as many unary symbols as atomic programs. However, MmL allows us to propose a better embedding, unrestricted by the limitations of modal μ-logic. Indeed, the embedding in [33] suffers from at least two limitations that we can avoid with MmL. First, sometimes transitions are not just labeled with discrete programs, such as in *hybrid systems* [35] and *cyber-physical systems* [36] where the labels are *continuous values* such as elapsing time. We cannot introduce for every time $t \in \mathbb{R}_{\geq 0}$ a modality $[t]$, as only countably many modalities are allowed. Instead, we may want to axiomatize the domains of such possibly continuous values and treat them as any other data. Second, we may want to quantify over such values, be they discrete or continuous, and we would not be able to do so (even in MmL) if they are encoded as modalities/symbols.

Let us instead define a signature (of *labeled transition systems*) $\mathbb{\Sigma}^{\mathsf{LTS}} = (\{State, Pgm\}, \Sigma^{\mathsf{LTS}}_{\lambda,Pgm} \cup \{\bullet \in \Sigma^{\mathsf{LTS}}_{Pgm\,State,State}\})$

where the "one-path next" $\bullet$ is a *binary symbol* taking an additional *Pgm* argument, and for all atomic programs $a \in \mathrm{AP}_{\mathrm{GM}}$ we add a constant symbol $a \in \Sigma^{\mathsf{LTS}}_{\lambda,Pgm}$. Just as all $\mathbb{\Sigma}^{\mathsf{TS}}$-models are exactly transition systems (Section VII-B), we have that all $\mathbb{\Sigma}^{\mathsf{LTS}}$-models are exactly labeled transition systems. We define compound programs as derived constructs as follows:

$$\langle\alpha\rangle\varphi \equiv \bullet(\alpha,\varphi) \qquad [\alpha]\varphi \equiv \neg\langle\alpha\rangle\neg\varphi$$

(Seq) $[\alpha\,;\beta]\varphi \equiv [\alpha][\beta]\varphi$ (Choice) $[\alpha \cup \beta]\varphi \equiv [\alpha]\varphi \wedge [\beta]\varphi$

(Test) $[\psi?]\varphi \equiv (\psi \rightarrow \varphi)$ (Iter) $[\alpha^*]\varphi \equiv \nu f.(\varphi \wedge [\alpha]f)$

Like for the embedding of modal μ-logic (Section VII-B), no axioms are needed. Let Γ^{DL} denote the empty $\mathbb{\Sigma}^{\mathsf{LTS}}$-theory.

Theorem 36. *For all DL formulas φ, the following are equivalent: (1) $\vdash_{\mathsf{DL}} \varphi$; (2) $\vDash_{\mathsf{DL}} \varphi$; (3) $\Gamma^{\mathsf{DL}} \vdash \varphi$; (4) $\Gamma^{\mathsf{DL}} \vDash \varphi$.*

We point out that the iterative operator $[\alpha^*]\varphi$ is axiomatized with *two* axioms in the proof system of DL (see, e.g., [13]):

$$\text{(DL-Iter}_1\text{)} \qquad \varphi \wedge [\alpha][\alpha^*]\varphi \leftrightarrow [\alpha^*]\varphi$$

$$\text{(DL-Iter}_2\text{)} \qquad \varphi \wedge [\alpha^*](\varphi \rightarrow [\alpha]\varphi) \rightarrow [\alpha^*]\varphi$$

while we just regard it as syntactic sugar, via (Iter). One may argue that (Iter) desugars to the ν-binder, though, which obeys the proof rules (Pre-Fixpoint) and (Knaster-Tarski) that essentially have the same appearance as (DL-Iter$_1$) and (DL-Iter$_2$). We agree. And that is exactly why we think one should work in *one uniform and fixed logic*, such as MmL, where general fixpoint axioms are given to specify and reason about *any fixpoint properties* of *any domains* and to develop general-purpose automatic tools and provers, rather than designing special-purpose logics and tools that work on only certain domains and then extending existing logics or designing new logics when new domains are considered.

VIII. Instance: reachability logic

Reachability logic (RL) [2] is an approach to program verification using operational semantics. Different from other approaches such as Hoare-style verification, RL has a *language-independent* proof system that offers sound and relatively complete deduction for all languages. RL is the logic underlying the $\mathbb{K}$ framework [37], which has been used to define the formal semantics of various real languages such as C [3], Java [4], and JavaScript [5], yielding program verifiers for all these languages at no additional cost [6].

In spite of its generality w.r.t. languages, reachability logic is unfortunately limited to specifying and deriving only reachability properties. This limitation was one of the factors that motivated the development of MmL. Fig. 8 shows a few of RL's proof rules; notice that unlike Hoare logic's proof rules, RL's proof rules are not specific to any particular programming language. The programming language is given through its operational semantics as a set of axiom rules, to be used via the (Axiom) proof rule. The characteristic feature of RL is its (Circularity) rule, which supports reasoning about circular behavior and recursive program constructs. In this subsection, we show how RL is faithfully defined in MmL and all its proof rules, including (Circularity), can be proved in MmL.

$$\text{(Axiom)}\ \frac{\varphi_1 \Rightarrow \varphi_2 \in A}{A \vdash_C \varphi_1 \Rightarrow \varphi_2}$$

$$\text{(Transitivity)}\ \frac{A \vdash_C \varphi_1 \Rightarrow \varphi_2 \quad A \cup C \vdash \varphi_2 \Rightarrow \varphi_3}{A \vdash_C \varphi_1 \Rightarrow \varphi_3}$$

$$\text{(Consequence)}\ \frac{M^{\mathsf{cfg}} \vDash \varphi_1 \rightarrow \varphi_1' \quad A \vdash_C \varphi_1' \Rightarrow \varphi_2' \quad M^{\mathsf{cfg}} \vDash \varphi_2' \rightarrow \varphi_2}{A \vdash_C \varphi_1 \Rightarrow \varphi_2}$$

$$\text{(Circularity)}\ \frac{A \vdash_{C \cup \{\varphi_1 \Rightarrow \varphi_2\}} \varphi_1 \Rightarrow \varphi_2}{A \vdash_C \varphi_1 \Rightarrow \varphi_2}$$

Fig. 2. Some selected proof rules in the proof system of reachability logic

A. RL syntax, semantics, and proof system

RL is parametric in a model of ML (without μ) called the *configuration model*. Specifically, fix a signature (of *static program configurations*) $\mathbb{\Sigma}^{\mathsf{cfg}}$ which may have various sorts and symbols, among which there is a distinguished sort *Cfg*. Fix a $\mathbb{\Sigma}^{\mathsf{cfg}}$-model M^{cfg} called the *configuration model*, where M^{cfg}_{Cfg} is the set of all configurations. RL's formulas are called *reachability rules*, or simply *rules*, and have the form $\varphi_1 \Rightarrow \varphi_2$ where φ_1, φ_2 are ML (without μ) $\mathbb{\Sigma}^{\mathsf{cfg}}$-patterns. A *reachability system S* is a finite set of rules, which yields a transition system $\mathbb{S} = (M^{\mathsf{cfg}}_{Cfg}, R)$ where $s\,R\,t$ if there exist a rule $\varphi_1 \Rightarrow \varphi_2 \in S$ and an M^{cfg}-valuation ρ such that $s \in \bar{\rho}(\varphi_1)$ and $t \in \bar{\rho}(\varphi_2)$. A rule $\psi_1 \Rightarrow \psi_2$ is *S-valid*, denoted $S \vDash_{\mathsf{RL}} \psi_1 \Rightarrow \psi_2$, if for all M^{cfg}_{Cfg}-valuations ρ and configurations $s \in \bar{\rho}(\psi_1)$, either there is an infinite trace $s\,R\,t_1\,R\,t_2\,R\ldots$ in $\mathbb{S}$ or there is a configuration t such that $s\,R^*\,r$ and $t \in \bar{\rho}(\psi_2)$. Therefore, the validity in reachability logic is defined in the spirit of *partial correctness*.

The sound and relatively complete proof system of RL derives *reachability logic sequents* of the form $A \vdash_C \varphi_1 \Rightarrow \varphi_2$ where A (called *axioms*) and C (called *circularities*) are finite sets of rules. Initially we start with $A = S$ and $C = \emptyset$. As the proof proceeds, more rules can be added to C via (Circularity) and then moved to A via (Transitivity), which can then be used via (Axiom). We write $S \vdash_{\mathsf{RL}} \psi_1 \Rightarrow \psi_2$ to mean that $S \vdash_\emptyset \psi_1 \Rightarrow \psi_2$. Notice (Consequence) consults the configuration model M^{cfg} for validity, so the completeness result is *relative to* M^{cfg}. We recall the following result [2]:

Theorem 37. *For all reachability systems S satisfying some reasonable technical assumptions (see [2]) and all rules $\psi_1 \Rightarrow \psi_2$, we have $S \vDash_{\mathsf{RL}} \psi_1 \Rightarrow \psi_2$ if and only if $S \vdash_{\mathsf{RL}} \psi_1 \Rightarrow \psi_2$.*

B. Defining reachability logic in matching μ-logic

We define the extended signature $\mathbb{\Sigma}^{\mathsf{RL}} = \mathbb{\Sigma}^{\mathsf{cfg}} \cup \{\bullet \in \Sigma_{Cfg,Cfg}\}$ where "$\bullet$" is a unary symbol (one-path next). To capture the semantics of reachability rules $\varphi_1 \Rightarrow \varphi_2$, we define:

"weak eventually" $\Diamond_w\varphi \equiv \nu X.\,\varphi \vee \bullet X$ // equal to $\neg\mathsf{WF} \vee \Diamond\varphi$

"reaching star" $\varphi_1 \Rightarrow^* \varphi_2 \equiv \varphi_1 \rightarrow \Diamond_w\varphi_2$

"reaching plus" $\varphi_1 \Rightarrow^+ \varphi_2 \equiv \varphi_1 \rightarrow \bullet\Diamond_w\varphi_2$

Notice that the "weak eventually" $\Diamond_w\varphi$ is defined similarly to "eventually" $\Diamond\varphi \equiv \mu X.\,\varphi \vee \bullet X$, but instead of using least fixpoint μ-binder, we define it as a greatest fixpoint. One can prove that $\Diamond_w\varphi = \neg\mathsf{WF} \vee \Diamond\varphi$, that is, a configuration γ

satisfies $\Diamond_w \varphi$ if either it satisfies $\Diamond\varphi$, or it is not well-founded, meaning that there exists an infinite execution path from γ. Also notice that "reaching plus" $\varphi_1 \Rightarrow^+ \varphi_2$ is a stronger version of "reaching star", requiring that $\Diamond_w \varphi_2$ should hold *after at least one step*. This *progressive condition* is crucial to the soundness of RL reasoning: as shown in (Transitivity), circularities are flushed into the axiom set only *after one reachability step is established*. This leads us to the following translation from RL sequents to MmL patterns.

Definition 38. Given a rule $\varphi_1 \Rightarrow \varphi_2$, define the MmL pattern $\boxdot(\varphi_1 \Rightarrow \varphi_2) \equiv \Box(\varphi_1 \Rightarrow^+ \varphi_2)$ and extend it to a rule set A as follows: $\boxdot A \equiv \bigwedge_{\varphi_1 \Rightarrow \varphi_2 \in A} \boxdot(\varphi_1 \Rightarrow \varphi_2)$. Define the translation RL2MmL from RL sequents to MmL patterns as follows:

$$\text{RL2MmL}(A \vdash_C \varphi_1 \Rightarrow \varphi_2) = (\forall \boxdot A) \wedge (\forall \circ \boxdot C) \rightarrow (\varphi_1 \Rightarrow^\star \varphi_2)$$

where $\star = *$ if C is empty and $\star = +$ if C is nonempty. We use $\forall\varphi$ as a shorthand for $\forall \vec{x}.\varphi$ where $\vec{x} = FV(\varphi)$.

Hence, the translation of $A \vdash_C \varphi_1 \Rightarrow \varphi_2$ depends on whether C is empty or not. When C is nonempty, the RL sequent is *stronger* in that it requires *at least one step* in $\varphi_1 \Rightarrow \varphi_2$. Axioms (those in A) are also *stronger* than circularities (those in C) in that axioms *always* hold, while circularities only hold *after at least one step* because of the leading all-path next "$\circ$"; and since the "next" is a "weak" one, it does not matter which step is actually made as circularities hold on *all* next states.

Theorem 39. *Let* $\Gamma^{\mathsf{RL}} = \{\varphi \in \text{Pattern}^{ML}_{Cfg} \mid M^{\mathsf{cfg}} \vDash \varphi\}$ *be the set of all ML patterns (without* μ*) of sort Cfg that hold in* M^{cfg}*. For all RL systems S and rules* $\varphi_1 \Rightarrow \varphi_2$ *satisfying the same technical assumptions in [2], the following are equivalent: (1)* $S \vdash_{\mathsf{RL}} \varphi_1 \Rightarrow \varphi_2$*; (2)* $S \vDash_{\mathsf{RL}} \varphi_1 \Rightarrow \varphi_2$*; (3)* $\Gamma^{\mathsf{RL}} \vdash \text{RL2MmL}(S \vdash_\emptyset \varphi_1 \Rightarrow \varphi_2)$*; (4)* $\Gamma^{\mathsf{RL}} \vDash \text{RL2MmL}(S \vdash_\emptyset \varphi_1 \Rightarrow \varphi_2)$*.*

Therefore, provided that an oracle for validity of ML patterns (without μ) in M^{cfg} is available, the MmL proof system is capable of deriving any reachability property that can be derived with the RL proof system. This result makes MmL an even more fundamental logic foundation for the $\mathbb{K}$ framework and thus for programming language specification and verification than RL, because it can express significantly more properties than partial correctness reachability.

IX. Future and Related Work

We discuss future work, open problems, and related work.

A. Relation to modal logics

Due to the duality between ML symbols and modal logic modalities (Section III, Proposition 12), ML can be regarded as a non-trivial extension of modal logics. There are various directions to extend the basic propositional modal logic in the literature [17]. One is the *hybrid extension*, where first-order quantifiers "$\forall$" and "$\exists$" are added to the logic, as well as *state variables/names* that allow to specify one particular state. Another is the *polyadic extension*, where modalities can take not just one argument, but any number of arguments, and there can be multiple modalities. ML can be seen as a combination of both extensions, further extended with multiple sort universes. The completeness of $\mathcal{H}$ (Theorem 16) also extends the completeness results of its fragment logics, including hybrid modal logic [18] and many-sorted polyadic modal logic [38].

B. Stronger completeness results of $\mathcal{H}$

There are various notions of completeness in modal logics. We give three of them under the context of ML and its proof system $\mathcal{H}$, from the strongest to the weakest:

- Global completeness: $\Gamma \vDash_{\mathsf{ML}} \varphi$ implies $\Gamma \vdash_{\mathcal{H}} \varphi$;
- Strong local completeness: $\Gamma \vDash^{\text{loc}}_{\mathsf{ML}} \varphi$ implies $\Gamma \vdash^{\text{loc}}_{\mathcal{H}} \varphi$;
- Weak local completeness: $\vDash_{\mathsf{ML}} \varphi$ implies $\vdash_{\mathcal{H}} \varphi$;

where $\Gamma \vDash^{\text{loc}}_{\mathsf{ML}} \varphi$, called *local semantic entailment*, is defined as for all models M, all valuations ρ, and all $a \in M$, if $a \in \bar{\rho}(\psi)$ for all $\psi \in \Gamma$ then $a \in \bar{\rho}(\varphi)$; $\Gamma \vdash^{\text{loc}}_{\mathcal{H}} \varphi$, called *local provability*, is defined as there exists a finite subset $\Gamma_0 \subseteq_{fin} \Gamma$ such that $\vdash_{\mathcal{H}} \wedge\Gamma_0 \rightarrow \varphi$, where $\wedge\Gamma_0$ is the conjunction of all patterns in Γ_0. The completeness result for $\mathcal{H}$ that we present in Theorem 16 is a weak local completeness result, but the way we actually prove it is by proving the strong local completeness theorem and then let $\Gamma = \emptyset$. We did not present in this paper the strong local completeness theorem due to its complex form.

What is not known and left as future work is global completeness. Theorem 15 shows that global completeness holds when Γ contains definedness symbols and axioms.

C. Alternative semantics of matching μ-logic

MmL cannot have a sound and complete proof system because we can precisely define $(\mathbb{N}, +, \times)$ (see Proposition 23). On the other hand, the proof system $\mathcal{H}_\mu$ turned out to be strong enough to prove all the proof rules of all the proof systems of all the logics discussed in this paper. Therefore, a natural question is whether we can find alternative models for MmL that make $\mathcal{H}_\mu$ complete. A promising direction towards such an alternative semantics is to consider the so-called *Henkin semantics* or *general semantics*, where the least fixpoint pattern $\mu X.\varphi$ is not evaluated to the true least fixpoint in the models, but to *the least fixpoint that is definable in the logic*.

X. Conclusion

We made two main contributions in this paper. Firstly, we proposed a new sound and complete proof system $\mathcal{H}$ for matching logic (ML). Secondly, we extended ML with the least fixpoint μ-binder and proposed matching μ-logic (MmL). We showed the expressiveness of MmL by defining a variety of common logics about induction/fixpoints/verification in MmL. We hope that MmL may serve as a promising unifying foundation for specifying and reasoning about induction, fixpoints, as well as model checking and program verification.

References

[1] G. Roşu, "Matching logic," *Logical Methods in Computer Science*, vol. 13, no. 4, 2017.

[2] G. Roşu, A. Ştefănescu, c. Ciobâcă, and B. M. Moore, "One-path reachability logic," in *Proceedings of the 28th Symposium on Logic in Computer Science (LICS'13)*. IEEE, Jun. 2013, pp. 358–367.

[3] C. Hathhorn, C. Ellison, and G. Roşu, "Defining the undefinedness of C," in *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'15)*. ACM, Jun. 2015, pp. 336–345.

[4] D. Bogdănaş and G. Roşu, "K-Java: A complete semantics of Java," in *Proceedings of the 42nd Symposium on Principles of Programming Languages (POPL'15)*. ACM, Jan. 2015, pp. 445–456.

[5] D. Park, A. Ştefănescu, and G. Roşu, "KJS: A complete formal semantics of JavaScript," in *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'15)*. ACM, Jun. 2015, pp. 346–356.

[6] A. Ştefănescu, D. Park, S. Yuwen, Y. Li, and G. Roşu, "Semantics-based program verifiers for all languages," in *Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'16)*. ACM, Nov. 2016, pp. 74–91.

[7] Y. Gurevich and S. Shelah, "Fixed-point extensions of first-order logic," *Annals of pure and applied logic*, vol. 32, pp. 265–280, 1986.

[8] D. Kozen, "Results on the propositional μ-calculus," *Theoretical Computer Science*, vol. 27, no. 3, pp. 333–354, 1983, special Issue Ninth International Colloquium on Automata, Languages and Programming (ICALP) Aarhus, Summer 1982. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0304397582901256

[9] A. Pnueli, "The temporal logic of programs," in *Foundations of Computer Science, 1977., 18th Annual Symposium on*. IEEE, 1977, pp. 46–57.

[10] G. Roşu, "Finite-trace linear temporal logic: Coinductive completeness," *Formal methods in system design*, vol. 53, no. 1, pp. 138–163, 2018.

[11] M. J. Fischer and R. E. Ladner, "Propositional dynamic logic of regular programs," *Journal of computer and system sciences*, vol. 18, no. 2, pp. 194–211, 1979.

[12] D. Harel, "Dynamic logic," in *Handbook of philosophical logic*. Springer, 1984, pp. 497–604.

[13] D. Harel, D. Kozen, and J. Tiuryn, "Dynamic logic," in *Handbook of philosophical logic*. Springer, 2001, pp. 99–217.

[14] K. Futatsugi and J.-P. Jouannaud, *Algebra, meaning, and computation: Essays dedicated to Joseph A. Goguen on the occasion of his 65th birthday*. Springer Science & Business Media, 2006, vol. 4060.

[15] Y. Imai and K. Iséki, "On axiom systems of propositional calculi," *Proceedings of the Japan Academy*, vol. 41, no. 6, pp. 436–439, 1965.

[16] A. G. Hamilton, *Logic for mathematicians*. Cambridge University Press, 1988.

[17] P. Blackburn, J. van Benthem, and F. Wolter, *Handbook of modal logic*. Elsevier, 2006, vol. 3.

[18] P. Blackburn and M. Tzakova, "Hybrid completeness," *Logic Journal of IGPL*, vol. 6, no. 4, pp. 625–650, 1998.

[19] P. Blackburn, M. d. Rijke, and Y. Venema, *Modal logic*. New York, NY, USA: Cambridge University Press, 2001.

[20] A. Tarski, "A lattice-theoretical fixpoint theorem and its applications," *Pacific journal of Mathematics*, vol. 5, no. 2, pp. 285–309, 1955.

[21] J. A. Goguen, J. W. Thatcher, E. G. Wagner, and J. B. Wright, "Initial algebra semantics and continuous algebras," *Journal of the ACM*, vol. 24, no. 1, pp. 68–95, 1977.

[22] K. Gödel, *On formally undecidable propositions of principia Mathematica and related systems*. Courier corporation, 1992.

[23] A. I. Malc'ev, "Axiomatizable classes of locally free algebras of various type," *The Metamathematics of Algebraic Systems: Collected Papers*, vol. 1967, pp. 262–281, 1936.

[24] L. Löwenheim, "Über möglichkeiten im relativkalkül," *Mathematische Annalen*, vol. 76, no. 4, pp. 447–470, 1915.

[25] G. Peano, *Arithmetices principia: Nova methodo exposita*. Fratres Bocca, 1889.

[26] E. Mendelson, *Introduction to mathematical logic*. Springer, Boston, MA, 1979.

[27] M. Schönfinkel, "Über die Bausteine der mathematischen Logik," *Mathematische annalen*, vol. 92, no. 3-4, pp. 305–316, 1924.

[28] H. B. Curry, *Combinatory logic*. Amsterdam: North-Holland Pub. Co., 1958.

[29] A. Church, "A formulation of the simple theory of types," *The journal of symbolic logic*, vol. 5, no. 2, pp. 56–68, 1940.

[30] S. Kreutzer, "Pure and applied fixed-point logics," Ph.D. dissertation, Bibliothek der RWTH Aachen, 2002.

[31] C. A. R. Hoare, "An axiomatic basis for computer programming," *Communications of ACM*, vol. 12, no. 10, pp. 576–580, 1969.

[32] I. Walukiewicz, "Completeness of Kozen's axiomatisation of the propositional μ-calculus," *Information and Computation*, vol. 157, no. 1-2, pp. 142–182, 2000.

[33] G. Lenzi, "The modal μ-calculus: A survey," *Task quarterly*, vol. 9, no. 3, pp. 293–316, 2005.

[34] E. A. Emerson, "Temporal and modal logic," in *Formal Models and Semantics*. Elsevier, 1990, pp. 995–1072.

[35] R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho, "Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems," in *Hybrid systems*. Springer, 1993, pp. 209–229.

[36] E. A. Lee, "Cyber physical systems: Design challenges," in *11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*. IEEE, 2008, pp. 363–369.

[37] G. Rosu, "K—A semantic framework for programming languages and formal analysis tools," in *Dependable Software Systems Engineering*, ser. NATO Science for Peace and Security, D. Peled and A. Pretschner, Eds. IOS Press, 2017.

[38] I. Leustean and N. Moanga, "A many-sorted polyadic modal logic," *CoRR*, vol. abs/1803.09709, 2018. [Online]. Available: http://arxiv.org/abs/1803.09709

Appendix A
Matching Logic Proof System $\mathcal{P}$

We show the matching logic proof system $\mathcal{P}$ proposed in [1] in Fig. 3.

Appendix B
Proof of Theorem 13

We prove the soundness theorem of $\mathcal{H}$ (Theorem 13). We only discuss ML (without μ) in this section, so we drop all unnecessary annotations. Specifically, we abbreviate "$\vDash_{\mathsf{ML}}$" as "$\vDash$" and "$\vdash_{\mathcal{H}}$" as "$\vdash$".

We write $\rho_1 \overset{x}{\sim} \rho_2$ to mean that ρ_1, ρ_2 differ only at x. As in FOL, we can prove that $\bar{\rho}(\forall x.\varphi) = \bigcap_{a\in M} \overline{\rho[a/x]}(\varphi) = \bigcap\{\bar{\rho'}(\varphi) \mid \rho' \overset{x}{\sim} \rho\}$ and $\bar{\rho}(\exists x.\varphi) = \bigcup_{a\in M} \rho[a/x](\varphi) = \bigcup\{\bar{\rho'}(\varphi) \mid \rho' \overset{x}{\sim} \rho\}$.

Lemma 40. *The following propositions hold:*

1) $\vDash \varphi_1 \to (\varphi_2 \to \varphi_1)$
2) $\vDash \varphi_1 \to (\varphi_2 \to \varphi_3) \to (\varphi_1 \to \varphi_2) \to (\varphi_1 \to \varphi_3)$
3) $\vDash (\neg\varphi_1 \to \neg\varphi_2) \to (\varphi_2 \to \varphi_1)$
4) $M, \rho \vDash \varphi_1$ *and* $M, \rho \vDash \varphi_1 \to \varphi_2$ *imply* $M, \rho \vDash \varphi_2$
5) $\vDash \forall x.\varphi \to \varphi[y/x]$
6) $\vDash \forall x.(\varphi_1 \to \varphi_2) \to \varphi_1 \to \forall x.\varphi_2$ *if* $x \notin FV(\varphi_1)$
7) $M \vDash \varphi$ *implies* $M \vDash \forall x.\varphi$
8) $\vDash C_\sigma[\bot] \to \bot$
9) $\vDash C_\sigma[\varphi_1 \vee \varphi_2] \to C_\sigma[\varphi_1] \vee C_\sigma[\varphi_2]$
10) $\vDash C_\sigma[\exists x.\varphi] \to \exists x.C_\sigma[\varphi]$ *if* $x \notin FV(C_\sigma[\exists x.\varphi])$
11) $M, \rho \vDash \varphi_1 \to \varphi_2$ *implies* $M, \rho \vDash C_\sigma[\varphi_1] \to C_\sigma[\varphi_2]$
12) $\vDash \exists x.x$
13) $\vDash \neg(C_1[x \wedge \varphi] \wedge C_2[x \wedge \neg\varphi])$

where $\varphi, \varphi_1, \varphi_2, \varphi_3$ are patterns, x, y are variables, σ is a symbol, C_σ is a single symbol context, C_1, C_2 are nested symbol contexts, M is a model, and ρ is a valuation.

Proof: Some of the propositions are proved in [1]. To make this proof self-contained, we present the proofs of all propositions. Let M be a model and ρ be a valuation.

(1) $\bar{\rho}(\varphi_1 \to (\varphi_2 \to \varphi_1)) = \bar{\rho}(\neg\varphi_1) \cup \bar{\rho}(\varphi_2 \to \varphi_1) = (M \setminus \bar{\rho}(\varphi_1)) \cup \bar{\rho}(\neg\varphi_2) \cup \bar{\rho}(\varphi_1) = M$.

(2) $\bar{\rho}(\varphi_1 \to (\varphi_2 \to \varphi_3) \to (\varphi_1 \to \varphi_2) \to (\varphi_1 \to \varphi_3)) = \bar{\rho}(\neg(\varphi_1 \to (\varphi_2 \to \varphi_3))) \cup \bar{\rho}((\varphi_1 \to \varphi_2) \to (\varphi_1 \to \varphi_3)) = (\bar{\rho}(\varphi_1) \cap (\bar{\rho}(\neg(\varphi_2 \to \varphi_3))) \cup \bar{\rho}(\neg(\varphi_1 \to \varphi_2)) \cup \bar{\rho}(\varphi_1 \to \varphi_3) = (\bar{\rho}(\varphi_1) \cap \bar{\rho}(\varphi_2) \cap (M \setminus \bar{\rho}(\varphi_3))) \cup (\bar{\rho}(\varphi_1) \cap (M \setminus \bar{\rho}(\varphi_2))) \cup (M \setminus \bar{\rho}(\varphi_1)) \cup \bar{\rho}(\varphi_3) = M$.

(3) $\bar{\rho}(\neg\varphi_1 \to \neg\varphi_2) \to (\varphi_2 \to \varphi_1) = \bar{\rho}(\neg(\neg\varphi_1 \to \varphi_2)) \cup \bar{\rho}(\varphi_2 \to \varphi_1) = (M \setminus \bar{\rho}(\neg\varphi_1 \to \neg\varphi_2)) \cup (M \setminus \bar{\rho}(\varphi_2)) \cup \bar{\rho}(\varphi_1) = (M \setminus (\bar{\rho}(\neg\neg\varphi_1) \cup \bar{\rho}(\neg\varphi_2))) \cup (M \setminus \bar{\rho}(\varphi_2)) \cup \bar{\rho}(\varphi_1) = (M \setminus ((M \setminus M \setminus \bar{\rho}(\varphi_1)) \cup (M \setminus \bar{\rho}(\varphi_2)))) \cup (M \setminus \bar{\rho}(\varphi_2)) \cup \bar{\rho}(\varphi_1) = (M \setminus (\bar{\rho}(\varphi_1) \cup (M \setminus \bar{\rho}(\varphi_2)))) \cup (M \setminus \bar{\rho}(\varphi_2)) \cup \bar{\rho}(\varphi_1) = M$.

(4) $M, \rho \vDash \varphi_1 \to \varphi_2$, so $\bar{\rho}(\varphi_1 \to \varphi_2) = (M \setminus \bar{\rho}(\varphi_1)) \cup \bar{\rho}(\varphi_2) = M$, and thus $\bar{\rho}(\varphi_1) \subseteq \bar{\rho}(\varphi_2)$. Because $M, \rho \vDash \varphi_1$, $\bar{\rho}(\varphi_1) = M$, and thus $\bar{\rho}(\varphi_2) = M$.

(5) $\bar{\rho}(\forall x.\varphi \to \varphi[y/x]) = (M \setminus \bar{\rho}(\forall x.\varphi)) \cup \bar{\rho}(\varphi[y/x]) = (M \setminus \bigcap_{\rho'}(\overline{\rho'}(\varphi))) \cup \overline{\rho'_y}(\varphi)$ where $\rho'_y = \rho[\rho(y)/x]$ and $\rho' \overset{x}{\sim} \rho$. Notice that $\rho_y \overset{x}{\sim} \rho$. Thus $\bigcap_{\rho'}(\overline{\rho'}(\varphi)) \subseteq \overline{\rho'_y}(\varphi)$, and $(M \setminus \bigcap_{\rho'}(\overline{\rho'}(\varphi))) \cup \overline{\rho'_y}(\varphi) = M$.

(6) If suffices to show $\bar{\rho}(\forall x.(\varphi_1 \to \varphi_2) \subseteq \bar{\rho}(\varphi_1 \to \forall x.\varphi_2)$. Notice that $\bar{\rho}(\forall x.(\varphi_1 \to \varphi_2)) = \bigcap_{\rho'} \overline{\rho'}((\varphi_1 \to \varphi)) = \bigcap_{\rho'}((M \setminus \overline{\rho'}(\varphi_1)) \cup \overline{\rho'}(\varphi_2))$ where $\rho' \overset{x}{\sim} \rho$. Since $x \notin FV(\varphi_1)$, $\overline{\rho'}(\varphi_1) = \bar{\rho}(\varphi_1)$, and thus $\bigcap_{\rho'}((M \setminus \overline{\rho'}(\varphi_1)) \cup \overline{\rho'}(\varphi_2)) = \bigcap_{\rho'}((M \setminus \bar{\rho}(\varphi_1)) \cup \overline{\rho'}(\varphi_2)) = (M \setminus \bar{\rho}(\varphi_1)) \cup \bigcap_{\rho'}(\overline{\rho'}(\varphi_2)) = \bar{\rho}(\varphi_1 \to \forall x.\varphi_2)$.

(7) $\bar{\rho}(\forall x.\varphi) = \bigcap_{\rho'} \overline{\rho'}(\varphi)$ where $\rho' \overset{x}{\sim} \rho$, so it suffices to show $\overline{\rho'}(\varphi) = M$ for any ρ'. Since $\vDash \varphi$, we have $M, \rho' \vDash \varphi$, and thus $\overline{\rho'}(\varphi) = M$.

(8) $\bar{\rho}(C_\sigma[\bot] \to \bot) = M \setminus \bar{\rho}(C_\sigma[\bot])$, so it suffices to show $\bar{\rho}(C_\sigma[\bot]) = \emptyset$. In fact, $\bar{\rho}(\sigma(\ldots \bot \ldots)) = \sigma_M(\ldots \bar{\rho}(\bot) \ldots) = \sigma_M(\ldots \emptyset \ldots) = \emptyset$.

(9) It suffices to show $\bar{\rho}(C_\sigma(\varphi_1 \vee \varphi_2)) \subseteq \bar{\rho}(C_\sigma[\varphi_1] \vee C_\sigma[\varphi_2])$. In fact, $\bar{\rho}(C_\sigma(\varphi_1 \vee \varphi_2)) = \bar{\rho}(\sigma(\ldots(\bar{\rho}(\varphi_1) \cup \bar{\rho}(\varphi_2)) \ldots)) = \sigma_M(\ldots \bar{\rho}(\varphi_1) \ldots) \cup \sigma_M(\ldots \bar{\rho}(\varphi_2) \ldots) = \bar{\rho}(C_\sigma[\varphi_1]) \cup \bar{\rho}(C_\sigma[\varphi_2]) = \bar{\rho}(C_\sigma[\varphi_1] \vee C_\sigma[\varphi_2])$.

(10) It suffices to show $\bar{\rho}(C_\sigma[\exists x.\varphi]) \subseteq \bar{\rho}(\exists x.C_\sigma[\varphi])$. In fact, $\bar{\rho}(C_\sigma[\exists x.\varphi]) = \bar{\rho}(\sigma(\ldots \exists x.\varphi \ldots)) = \sigma_M(\ldots \bar{\rho}(\exists x.\varphi) \ldots) = \sigma_M(\ldots \bigcup_{\rho'} \overline{\rho'}(\varphi) \ldots) = \bigcup_{\rho'} \sigma_M(\ldots \overline{\rho'}(\varphi) \ldots) = \bar{\rho}(\exists x.C_\sigma[\varphi])$ where $\rho' \overset{x}{\sim} \rho$. Notice that we can move the big union $\bigcup_{\rho'}$ from the argument to the top without affecting other arguments because $x \notin FV(C_\sigma[\exists x.\varphi])$.

(11) It suffices to show $\bar{\rho}(C_\sigma[\varphi_1]) \subseteq \bar{\rho}(C_\sigma[\varphi_2])$. Notice that $\vDash \varphi_1 \to \varphi_2$, so $\bar{\rho}(\varphi_1) \subseteq \bar{\rho}(\varphi_2)$, and thus, $\bar{\rho}(C_\sigma[\varphi_1]) = \sigma_M(\ldots \bar{\rho}(\varphi_1) \ldots) \subseteq \sigma_M(\ldots \bar{\rho}(\varphi_2) \ldots) = \bar{\rho}(C_\sigma[\varphi_2])$.

(12) $\bar{\rho}(\exists x.x) = \bigcup_{\rho'}(\overline{\rho'}(x)) = \bigcup_{\rho'}\{\rho'(x)\}$ where $\rho' \overset{x}{\sim} \rho$. Notice $\bigcup_{\rho'}\{\rho'(x)\} = \bigcup_{a\in M}\{a\} = M$. ∎

(13) It suffices to show that either $\bar{\rho}(C_1[x \wedge \varphi])$ or $\bar{\rho}(C_2[x \wedge \neg\varphi])$ is the empty set. For every nested symbol context C, use the same technique in (8) and structural induction, we can prove that if $\bar{\rho}(\psi) = \emptyset$ then $\bar{\rho}(C[\psi]) = \emptyset$. Therefore, we just need to prove that either $\bar{\rho}(x \wedge \varphi)$ or $\bar{\rho}(x \wedge \neg\varphi)$ is the empty set. If $\rho(x) \notin \bar{\rho}(\varphi)$, then the former is empty. Otherwise, the latter is empty.

Now we are ready to prove Theorem 13.

Proof of Theorem 13: Carry out induction on the length of the Hilbert-style proof $\Gamma \vdash \varphi$.

(Base Case). Suppose the length is 1. Then φ is either an axiom in $\mathcal{H}$ or $\varphi \in \Gamma$. If φ is an axiom, then $\vDash \varphi$ by Lemma 40. If $\varphi \in \Gamma$, then $\Gamma \vDash \varphi$ by definition.

(Induction Step). Suppose the proof $\Gamma \vdash \varphi$ has $n+1$ steps:

$$\varphi_1, \ldots, \varphi_n, \varphi_{n+1} \qquad \text{with } \varphi_{n+1} \equiv \varphi$$

By induction hypothesis, $\Gamma \vDash \varphi_1$,..., $\Gamma \vDash \varphi_n$. If φ is an axiom or $\varphi \in \Gamma$, then $\vDash \varphi$ by for the same reason as in (Base Case). If the last step is one of (Modus Ponens), (Universal Generalization), or (Framing), then $\Gamma \vDash \varphi$ by Lemma 40, cases (4), (7), and (11), respectively. ∎

Appendix C
Properties of Proof System $\mathcal{H}$

We discuss properties of $\mathcal{H}$. In particular, we prove Proposition 12 and Theorem 14.

Our final goal is to prove all proof rules in $\mathcal{P}$ using the proof system $\mathcal{H}$ plus (Definedness) axioms, i.e., Theorem 15.

We only discuss ML (without μ) in this section, so we drop all unnecessary annotations. Specifically, we abbreviate "$\vDash_{\mathsf{ML}}$"

(Propositional Tautology)	φ, if φ is a proposition tautology over patterns of the same sort
(Modus Ponens)	$\dfrac{\varphi_1 \quad \varphi_1 \to \varphi_2}{\varphi_2}$
(Functional Substitution)	$(\forall x.\varphi) \wedge (\exists y.\varphi' = y) \to \varphi[\varphi'/x]$ if $y \notin FV(\varphi')$
($\forall$)	$\forall x.(\varphi_1 \to \varphi_2) \to (\varphi_1 \to \forall x.\varphi_2)$ if $x \notin FV(\varphi_1)$
(Universal Generalization)	$\dfrac{\varphi}{\forall x.\varphi}$
(Equality Introduction)	$\varphi = \varphi$
(Equality Elimination)	$(\varphi_1 = \varphi_2) \wedge \psi[\varphi_1/x] \to \psi[\varphi_2/x]$
(Membership Introduction)	$\dfrac{\varphi}{\forall x.(x \in \varphi)}$ if $x \notin FV(\varphi)$
(Membership Elimination)	$\dfrac{\forall x.(x \in \varphi)}{\varphi}$ if $x \notin FV(\varphi)$
(Membership Variable)	$(x \in y) = (x = y)$
(Membership$_\neg$)	$(x \in \neg\varphi) = \neg(x \in \varphi)$
(Membership$_\wedge$)	$(x \in \varphi_1 \wedge \varphi_2) = (x \in \varphi_1) \wedge (x \in \varphi_2)$
(Membership$_\exists$)	$(x \in \exists y.\varphi) = \exists y.(x \in \varphi)$, where x and y distinct.
(Membership Symbol)	$x \in C_\sigma[\varphi] = \exists y.(y \in \varphi) \wedge (x \in C_\sigma[y])$ if $y \notin FV(C_\sigma[\varphi])$

Fig. 3. Sound and complete matching logic proof system $\mathcal{P}$ with definedness symbols

as "$\models$" and "$\vdash_{\mathcal{H}}$" as "$\vdash$". We sometimes call a *nested symbol context* just a *symbol context* (see Definition 10).

Proposition 41 (Sound FOL Reasoning)**.** *Let $\mathbb{\Sigma} = (S, \Sigma)$ be a matching logic signature. Let (S, Π, F) be any first-order logic signature with $\Pi = \{\Pi_s\}_{s \in S}$ a set of constant predicate symbols and $F = \emptyset$ a set of function symbols. For any predicate logic formula $\Psi(\pi_1, \ldots, \pi_n)$ where $\pi_1, \ldots, \pi_n \in \Pi$, if $\Psi(\pi_1, \ldots, \pi_n)$ is derivable in FOL, then $\Psi(\varphi_1, \ldots, \varphi_n)$ is derivable in matching logic, where φ_i has the same sort as π_i for $1 \leq i \leq n$.*

Proof: Notice that $F = \emptyset$, so the only FOL terms are variables. Under that condition, the first seven rules in Fig. 1 form a complete FOL proof system as in [16]. ∎

Proposition 42 (Sound Frame Reasoning)**.** *For any $\sigma \in \Sigma_{s_1 \ldots s_n, s}$ and $\varphi_i, \varphi_i' \in \text{Pattern}_{s_i}$ such that $\Gamma \vdash \varphi_i \to \varphi_i'$ for any $1 \leq i \leq n$, then $\Gamma \vdash \sigma(\varphi_1, \ldots, \varphi_n) \to \sigma(\varphi_1', \ldots, \varphi_n')$. For any symbol context C and φ_i, φ_i' such that $\Gamma \vdash \varphi_i \to \varphi_i'$, then $\Gamma \vdash C[\varphi] \to C[\varphi_i']$.*

Proof: For the first case, it suffices to show that

$$\begin{aligned}
&\Gamma \vdash \sigma(\varphi_1, \varphi_2, \ldots, \varphi_n) \to \sigma(\varphi_1', \varphi_2, \ldots, \varphi_n)\\
&\Gamma \vdash \sigma(\varphi_1', \varphi_2, \ldots, \varphi_n) \to \sigma(\varphi_1', \varphi_2', \ldots, \varphi_n)\\
&\ldots\\
&\Gamma \vdash \sigma(\varphi_1', \varphi_2', \ldots, \varphi_n) \to \sigma(\varphi_1', \varphi_2', \ldots, \varphi_n')
\end{aligned}$$

which directly follow by (Framing).

For the second case, the proof is by structure induction on C. If C is the identity context, the conclusion is obvious. If C has the form $C_\sigma[C']$, the conclusion follows from induction hypothesis and (Framing). ∎

Proposition 43 (Propagation through Symbol Contexts)**.** *For any symbol context C and patterns $\varphi_1, \varphi_2, \varphi$, the following propositions hold.*

- $\Gamma \vdash C[\bot] \leftrightarrow \bot$
- $\Gamma \vdash C[\varphi_1 \vee \varphi_2] \leftrightarrow C[\varphi_1] \vee C[\varphi_2]$
- $\Gamma \vdash C[\exists x.\varphi] \leftrightarrow \exists x.C[\varphi]$ *if* $x \notin FV(C[\exists x.\varphi])$

The following results are often useful in practice, whose proofs can be obtained by standard propositional reasoning with the above propositions:

- $\Gamma \vdash C[\varphi_1 \vee \varphi_2]$ *iff* $\Gamma \vdash C[\varphi_1] \vee C[\varphi_2]$
- $\Gamma \vdash C[\exists x.\varphi]$ *iff* $\Gamma \vdash \exists x.C[\varphi]$ *if* $x \notin FV(C[\exists x.\varphi])$

Proof: The proof is by structure induction on the symbol context C. If C is the identity context then the conclusion is obvious. Now assume $C = C_\sigma[C']$ where C' is a symbol context for which the conclusion holds.

Firstly, let us prove $\Gamma \vdash C_\sigma[C'[\bot]] \leftrightarrow \bot$. The implication from right to left is by simple propositional reasoning. For the other direction, notice by induction hypothesis $\Gamma \vdash C'[\bot] \to \bot$ and by (Framing) $\Gamma \vdash C_\sigma[C'[\bot]] \to C_\sigma[\bot]$. In addition by (Propagation$_\bot$), $\Gamma \vdash C_\sigma[\bot] \to \bot$, and the rest of the proof is by standard propositional reasoning.

Secondly, let us prove $\Gamma \vdash C_\sigma[C'[\varphi_1 \vee \varphi_2]] \leftrightarrow C_\sigma[C'[\varphi_1]] \vee C_\sigma[C'[\varphi_2]]$. For the implication from right to left, it suffices to prove $\Gamma \vdash C_\sigma[C'[\varphi_i]] \to C_\sigma[C'[\varphi_1 \vee \varphi_2]]$ for $i = 1, 2$. By (Framing), it suffices to prove $\Gamma \vdash C'[\varphi_i] \to C'[\varphi_1 \vee \varphi_2]$, which follows from the induction hypothesis. For the implication from left to right, the proof is the same as how we proved $\Gamma \vdash C_\sigma[C'[\bot]] \to \bot$, while instead of (Propagation$_\bot$) we use (Propagation$_\vee$).

Finally, let us prove $\Gamma \vdash C_\sigma[C'[\exists x.\varphi]] \leftrightarrow \exists x.C_\sigma[C'[\varphi]]$ for $x \notin FV(C_\sigma[C'[\exists x.\varphi]])$. In fact the proof is the same as above, while instead of (Propagation$_\vee$) we use (Propagation$_\exists$). ∎

Proposition 44 (Congruence of Provably Equivalence)**.** *For any context C (not necessarily just symbol context), $\Gamma \vdash \varphi_1 \leftrightarrow \varphi_2$ implies $\Gamma \vdash C[\varphi_1] \leftrightarrow C[\varphi_2]$.*

Proof: The proof is by induction on the structure of C. If C is the identity context the conclusion is obvious. If C is of the form $\neg C'$, $\psi \to C'$, or $C' \to \psi$ where C' is a context and ψ is a pattern (notice ψ does not have the placeholder variable $\Box$ in it), the conclusion is by standard propositional reasoning. If C has the form $\forall x.C'$, the conclusion follows from standard FOL reasoning. If C has the form $C_\sigma[C']$, the conclusion follows from Proposition 42. ∎

Proposition 44 allows us to in-place replace φ_1 for φ_2 *in any context* as long as $\vdash \varphi_1 \leftrightarrow \varphi_2$, which is obviously a powerful and convenient result. In some of the following proofs, when we carry out structural induction on a pattern φ, we take $I = \{\wedge, \neg, \exists\}$ as primitives instead of $J = \{\to, \neg, \forall\}$ for technical simplicity. Proposition 44 justifies this approach, as we can transform any pattern φ to another pattern, say φ^I, that uses only constructs in I and $\vdash \varphi \leftrightarrow \varphi^I$. Then, φ and φ^I are interchangeable in any context.

Definition 45. Define the dual of a symbol σ as follows:

$$\bar{\sigma}(\varphi_1, \ldots, \varphi_n) \equiv \neg\sigma(\neg\varphi_1, \ldots, \neg\varphi_n).$$

Lemma 46. *$\Gamma \vdash \varphi$ implies $\Gamma \vdash \neg C[\neg\varphi]$ for symbol context C.*

Proof:

1	φ	hypothesis
2	$\neg\varphi \to \bot$	by 1, FOL reasoning
3	$C[\neg\varphi] \to C[\bot]$	by 2, (Framing)
4	$C[\bot] \to \bot$	by (Propagation)
5	$C[\neg\varphi] \to \bot$	by 3 and 4, FOL reasoning
6	$\neg C[\neg\varphi]$	by 5, FOL reasoning

∎

Now we are ready to prove Proposition 12.

Proof of Proposition 12: Let the single symbol context $C_\sigma = \sigma(\varphi_1, \ldots, \varphi_{i-1}, \Box, \varphi_{i+1}, \ldots, \varphi_n)$ for some symbol $\sigma \in \Sigma$.

(K). Note that we just need to prove the case of one argument, i.e., to prove $\vdash \neg C_\sigma[\neg(\varphi \to \varphi')] \to \neg C_\sigma[\neg\varphi] \to \neg C_\sigma[\neg\varphi']$. The case of multiple arguments can be incrementally proved by simple propositional reasoning.

To prove the "one argument" case, we apply simple propositional reasoning and obtain $\vdash C_\sigma[\varphi \wedge \varphi'] \vee C_\sigma[\neg\varphi] \vee \neg C_\sigma[\neg\varphi']$. By Proposition 43, the goal becomes $\vdash C_\sigma[(\varphi \wedge \varphi') \vee \neg\varphi] \vee \neg C_\sigma[\neg\varphi']$, i.e., $\vdash C_\sigma[\varphi' \vee \neg\varphi] \vee \neg C_\sigma[\neg\varphi']$. By Proposition 43 again, we obtain $\vdash C_\sigma[\varphi'] \vee C_\sigma[\neg\varphi] \vee \neg C_\sigma[\neg\varphi']$. Done.

(N) is proved in Lemma 46, letting C to be C_σ. ∎

In what follows, we move towards proving Theorem 15, by showing that all proof rules of $\mathcal{P}$ in Fig. 3 can be proved in $\mathcal{H}$. We will need (a lot of) lemmas.

The next lemma is useful in establishing an equality.

Lemma 47. *$\Gamma \vdash \varphi_1 \leftrightarrow \varphi_2$ implies $\Gamma \vdash \varphi_1 = \varphi_2$.*

Proof:

1	$\varphi_1 \leftrightarrow \varphi_2$	hypothesis
2	$\neg\lceil\neg(\varphi_1 \leftrightarrow \varphi_2)\rceil$	by 1, Lemma 46
3	$\varphi_1 = \varphi_2$	by 2, definition of equality

∎

Lemma 48. *(Equality Introduction) can be proved in $\mathcal{H}$.*

Proof:

1	$\varphi \leftrightarrow \varphi$	propositional tautology
2	$\varphi = \varphi$	by 1, Lemma 47

∎

Lemma 49. *(Membership Introduction) can be proved in $\mathcal{H}$.*

Proof:

1	φ	hypothesis
2	$\varphi \to (x \to \varphi)$	(Proposition$_1$)
3	$x \to \varphi$	by 1 and 2, (Modus Ponens)
4	$x \to x$	propositional tautology
5	$x \to x \wedge \varphi$	by 3 and 4, FOL reasoning
6	$\lceil x\rceil \to \lceil x \wedge \varphi\rceil$	by 5, (Framing)
7	$\lceil x\rceil$	definedness axiom
8	$\lceil x \wedge \varphi\rceil$	by 6 and 7, (Modus Ponens)
9	$x \in \varphi$	by 8, definition of membership
10	$\forall x.(x \in \varphi)$	by 9, (Universal Generalization)

∎

Lemma 50. *(Membership Elimination) can be proved in $\mathcal{H}$.*

Proof:

1	$\forall x.(x \in \varphi)$	hypothesis
2	$(\forall x.(x \in \varphi)) \to x \in \varphi$	(Variable Substitution)
3	$x \in \varphi$	by 1 and 2, (Modus Ponens)
4	$\lceil x \wedge \varphi\rceil$	by 3, definition of membership
5	$\neg(\lceil x \wedge \varphi\rceil \wedge (x \wedge \neg\varphi))$	(Singleton Variable)
6	$\lceil x \wedge \varphi\rceil \to (x \to \varphi)$	by 5, FOL reasoning
7	$x \to \varphi$	by 4 and 6, (Modus Ponens)
8	$\forall x.(x \to \varphi)$	by 7, (Universal Generalization)
9	$(\exists x.x) \to \varphi$	by 8, FOL reasoning
10	$\exists x.x$	(Existence)
11	φ	by 10 and 9, (Modus Ponens)

∎

Lemma 51. *(Membership Variable) can be proved in $\mathcal{H}$.*

Proof: By Lemma 47, we just need to prove both $\vdash (x \in y) \to (x = y)$ and $\vdash (x = y) \to (x \in y)$. We first prove $\vdash (x = y) \to (x \in y)$.

1	$\lceil x \rceil$	definedness axiom
2	$\lceil x \rceil \vee \lceil y \rceil$	by 1, FOL reasoning
3	$\lceil x \vee y \rceil$	by 2, Proposition 43
4	$\lceil \neg(x \leftrightarrow y) \vee (x \wedge y) \rceil$	by 3, FOL reasoning
5	$\lceil \neg(x \leftrightarrow y) \rceil \vee \lceil x \wedge y \rceil$	by 4, Proposition 43
6	$\neg\lceil \neg(x \leftrightarrow y) \rceil \rightarrow \lceil x \wedge y \rceil$	by 5, FOL reasoning
7	$(x = y) \rightarrow (x \in y)$	by 6, definition

We then prove $\vdash (x \in y) \rightarrow (x = y)$.

1	$\neg(\lceil x \wedge y \rceil \wedge \lceil x \wedge \neg y \rceil)$	by (Singleton Variable)
2	$\neg(\lceil x \wedge y \rceil \wedge \lceil \neg x \wedge y \rceil)$	by (Singleton Variable)
3	$\lceil x \wedge y \rceil \rightarrow \neg\lceil x \wedge \neg y \rceil$	by 1, FOL reasoning
4	$\lceil x \wedge y \rceil \rightarrow \neg\lceil \neg x \wedge y \rceil$	by 2, FOL reasoning
5	$\lceil x \wedge y \rceil$ $\rightarrow \neg\lceil x \wedge \neg y \rceil \wedge \neg\lceil \neg x \wedge y \rceil$	by 3, 4, FOL reasoning
6	$\lceil x \wedge y \rceil$ $\rightarrow \neg(\lceil x \wedge \neg y \rceil \vee \lceil \neg x \wedge y \rceil)$	by 5, FOL reasoning
7	$\lceil x \wedge y \rceil$ $\rightarrow \neg\lceil (x \wedge \neg y) \vee (\neg x \wedge y) \rceil$	by 6, Proposition 43
8	$\lceil x \wedge y \rceil \rightarrow \neg\lceil \neg(x \leftrightarrow y) \rceil$	by 7, FOL reasoning
9	$(x \in y) \rightarrow (x = y)$	by 8, definition

■

Lemma 52. *(Membership$_\neg$) can be proved in $\mathcal{H}$.*

Proof: We first prove $\vdash (x \in \neg\varphi) \rightarrow \neg(x \in \varphi)$.

1	$\neg(\lceil x \wedge \varphi \rceil \wedge \lceil x \wedge \neg\varphi \rceil)$	by (Singleton Variable)
2	$\lceil x \wedge \neg\varphi \rceil \rightarrow \neg\lceil x \wedge \varphi \rceil$	by 1, FOL reasoning
3	$(x \in \neg\varphi) \rightarrow \neg(x \in \varphi)$	by 2, definition

We then prove $\vdash \neg(x \in \varphi) \rightarrow (x \in \neg\varphi)$.

1	$\lceil x \rceil$	definedness axiom
2	$\lceil (x \wedge \varphi) \vee (x \wedge \neg\varphi) \rceil$	by 1, FOL reasoning
3	$\lceil x \wedge \varphi \rceil \vee \lceil x \wedge \neg\varphi \rceil$	by 2, Proposition 43
4	$\neg\lceil x \wedge \varphi \rceil \rightarrow \lceil x \wedge \neg\varphi \rceil$	by 3, FOL reasoning
5	$\neg(x \in \varphi) \rightarrow (x \in \neg\varphi)$	by 4, definition

■

Lemma 53. $\vdash (x \in (\varphi_1 \vee \varphi_2)) \leftrightarrow (x \in \varphi_1) \vee (x \in \varphi_2)$.

Proof: Use (Propagation$_\vee$) and FOL reasoning. ■

Lemma 54. *(Membership$_\wedge$) can be proved in $\mathcal{H}$.*

Proof: Use Lemma 52 and 53, and the fact that $\vdash \varphi_1 \wedge \varphi_2 \leftrightarrow \neg(\neg\varphi_1 \vee \neg\varphi_2)$. ■

Lemma 55. *(Membership$_\exists$) can be proved in $\mathcal{H}$.*

Proof: Use (Propagation$_\exists$) and FOL reasoning. ■

The following is a useful lemma about definedness symbols.

Lemma 56. $\vdash C[\varphi] \rightarrow \lceil \varphi \rceil$ *for any symbol context C.*

Proof: Let x be a fresh variable in the following proof.

1	$\lceil x \rceil$	definedness axiom
2	$\lceil x \rceil \vee \lceil \varphi \rceil$	by 1, FOL reasoning
3	$\lceil x \vee \varphi \rceil$	by 2, Proposition 43
4	$\lceil x \wedge \neg\varphi \vee \varphi \rceil$	by 3, FOL reasoning
5	$\lceil x \wedge \neg\varphi \rceil \vee \lceil \varphi \rceil$	by 4, Proposition 43
6	$C[x \wedge \varphi] \rightarrow \neg\lceil x \wedge \neg\varphi \rceil$	by (Singleton Variable)
7	$\neg\lceil x \wedge \neg\varphi \rceil \rightarrow \lceil \varphi \rceil$	by 5, FOL reasoning
8	$C[x \wedge \varphi] \rightarrow \lceil \varphi \rceil$	by 6 and 7, FOL reasoning
9	$\forall x.(C[x \wedge \varphi] \rightarrow \lceil \varphi \rceil)$	by 8, FOL reasoning
10	$(\exists x.C[x \wedge \varphi]) \rightarrow \lceil \varphi \rceil$	by 9, FOL reasoning
11	$\varphi \rightarrow (\exists x.x) \wedge \varphi$	by (Existence)
12	$\varphi \rightarrow \exists x.(x \wedge \varphi)$	by 11, FOL reasoning
13	$C[\varphi] \rightarrow C[\exists x.(x \wedge \varphi)]$	by 12, (Framing)
14	$C[\exists x.(x \wedge \varphi)] \rightarrow \lceil \varphi \rceil$	by 10, Proposition 43
15	$C[\varphi] \rightarrow \lceil \varphi \rceil$	by 13, 14, FOL reasoning

■

Corollary 57. $\vdash C_\sigma[\varphi] \rightarrow \lceil \varphi \rceil$ *and* $\vdash \lfloor \varphi \rfloor \rightarrow \neg C_\sigma[\neg\varphi]$ *for all symbols σ. In particular,* $\vdash \varphi \rightarrow \lceil \varphi \rceil$ *and* $\vdash \lfloor \varphi \rfloor \rightarrow \varphi$.

We are now ready to prove the deduction theorem (Theorem 14).

Proof of Theorem 14: Carry out induction on the length of the proof $\Gamma \cup \{\psi\} \vdash \varphi$.

(Base Case). Suppose the length is one, then either φ is an axiom in $\mathcal{H}$ or $\varphi \in \Gamma \cup \{\psi\}$. In either case, it is obvious that $\Gamma \vdash \lfloor \psi \rfloor \rightarrow \varphi$ (noticing Corollary 57 for the case φ is ψ).

(Induction Step). Suppose the proof $\Gamma \cup \{\psi\} \vdash \varphi$ has $n+1$ steps:

$$\varphi_1, \ldots, \varphi_n, \varphi.$$

If φ is an axiom in $\mathcal{H}$ or $\varphi \in \Gamma \cup \{\psi\}$, then $\Gamma \vdash \lfloor \psi \rfloor \rightarrow \varphi$ for the same reason as (Base Case). If the last step is (Modus Ponens) on φ_i and φ_j for some $1 \leq i, j \leq n$ such that φ_j has the form $\varphi_i \rightarrow \varphi$, by induction hypothesis, $\Gamma \vdash \lfloor \psi \rfloor \rightarrow \varphi_i$ and $\Gamma \vdash \lfloor \psi \rfloor \rightarrow (\varphi_i \rightarrow \varphi)$. By FOL reasoning, $\Gamma \vdash \lfloor \psi \rfloor \rightarrow \varphi$. If the last step is (Universal Generalization) on φ_i for some $1 \leq i \leq n$, then φ must have the form $\forall x.\varphi_i$ where x does not occur free in ψ. By induction hypothesis, $\Gamma \vdash \lfloor \psi \rfloor \rightarrow \varphi_i$. By FOL reasoning, $\Gamma \vdash \lfloor \psi \rfloor \rightarrow \forall x.\varphi_i$.

If the last step is (Framing) on φ_i for some $1 \leq i \leq n$, then φ_i must have the form $\varphi_i' \rightarrow \varphi_i''$, and φ must have the form $C_\sigma[\varphi_i'] \rightarrow C_\sigma[\varphi_i'']$ for some symbol σ. By induction hypothesis, $\Gamma \vdash \lfloor \psi \rfloor \rightarrow (\varphi_i' \rightarrow \varphi_i'')$. We now prove $\Gamma \vdash \lfloor \psi \rfloor \rightarrow (C_\sigma[\varphi_i'] \rightarrow C_\sigma[\varphi_i''])$.

1	$\lfloor \psi \rfloor \rightarrow (\varphi_i' \rightarrow \varphi_i'')$	hypothesis
2	$\varphi_i' \rightarrow \varphi_i'' \vee \lceil \neg\psi \rceil$	by 1, FOL reasoning
3	$C_\sigma[\lceil \neg\psi \rceil] \rightarrow \lceil \neg\psi \rceil$	Corollary 57
4	$C_\sigma[\varphi_i']$ $\rightarrow C_\sigma[\varphi_i'' \vee \lceil \neg\psi \rceil]$	by 2, (Framing)
5	$C_\sigma[\varphi_i']$ $\rightarrow C_\sigma[\varphi_i''] \vee C_\sigma[\lceil \neg\psi \rceil]$	by 4, Proposition 43
6	$C_\sigma[\varphi_i''] \vee C_\sigma[\lceil \neg\psi \rceil]$ $\rightarrow C_\sigma[\varphi_i''] \vee \lceil \neg\psi \rceil$	by 3, FOL reasoning
7	$C_\sigma[\varphi_i'] \rightarrow C_\sigma[\varphi_i''] \vee \lceil \neg\psi \rceil$	by 5, 6, FOL reasoning
8	$\lfloor \psi \rfloor \rightarrow (C_\sigma[\varphi_i'] \rightarrow C_\sigma[\varphi_i''])$	by 7, FOL reasoning

■

Lemma 58. *(Equality Elimination) can be proved in $\mathcal{H}$.*

Proof: Recall the definition of equality $(\varphi_1 = \varphi_2) \equiv \lfloor \varphi_1 \leftrightarrow \varphi_2 \rfloor$. Theorem 14 together with Proposition 44 give us a nice way to deal with equality premises. To prove $\vdash (\varphi_1 = \varphi_2) \to (\psi[\varphi_1/x] \to \psi[\varphi_2/x])$, we apply Theorem 14 and prove $\{\varphi_1 \leftrightarrow \varphi_2\} \vdash \psi[\varphi_1/x] \to \psi[\varphi_2/x]$, which is proved by Proposition 44. Note that the (formal) proof given in Proposition 44 does not use (Universal Generalization) at all, so the conditions of Theorem 14 are satisfied. ∎

Lemma 59. *(Functional Substitution) can be proved in $\mathcal{H}$.*

Proof: Let z be a fresh variable that does not occur free in φ and φ', and is distinct from x. Notice the side condition that y does not occur free in φ'.

1	$\varphi' = z \leftrightarrow z = \varphi'$	definition
2	$z = \varphi' \to (\varphi[z/x] \to \varphi[\varphi'/x])$	Lemma 58
3	$(\forall x.\varphi) \to \varphi[z/x]$	by axiom
4	$\varphi' = z \to ((\forall x.\varphi) \to \varphi[z/x])$	FOL reasoning
5	$\varphi' = z \to (\varphi[z/x] \to \varphi[\varphi'/x])$	FOL reasoning
6	$\varphi' = z \to ((\forall x.\varphi) \to \varphi[\varphi'/x])$	FOL reasoning
7	$\forall z.(\varphi' = z \to ((\forall x.\varphi) \to \varphi[\varphi'/x]))$	by 6
8	$(\exists z.\varphi' = z) \to ((\forall x.\varphi) \to \varphi[\varphi'/x])$	FOL reasoning
9	$(\forall x.\varphi) \wedge (\exists z.\varphi' = z) \to \varphi[\varphi'/x]$	FOL reasoning
10	$(\forall x.\varphi) \wedge (\exists y.\varphi' = y) \to \varphi[\varphi'/x]$	FOL reasoning

∎

Lemma 60. $\vdash C_\sigma[\varphi_1 \wedge (x \in \varphi_2)] = C_\sigma[\varphi_1] \wedge (x \in \varphi_2)$.

Proof: We first prove $\vdash C_\sigma[\varphi_1 \wedge (x \in \varphi_2)] \to C_\sigma[\varphi_1] \wedge (x \in \varphi_2)$. By FOL reasoning, it suffices to show both $\vdash C_\sigma[\varphi_1 \wedge (x \in \varphi_2)] \to C_\sigma[\varphi_1]$ and $\vdash C_\sigma[\varphi_1 \wedge (x \in \varphi_2)] \to (x \in \varphi_2)$. The first follows immediately by (Framing) and FOL reasoning. The second can be proved as:

1	$\lceil x \rceil$
2	$\lceil (x \wedge \neg\varphi_2) \vee (x \wedge \varphi_2) \rceil$
3	$\lceil x \wedge \neg\varphi_2 \rceil \vee \lceil x \wedge \varphi_2 \rceil$
4	$\neg\lceil x \wedge \neg\varphi_2 \rceil \to \lceil x \wedge \varphi_2 \rceil$
5	$C_\sigma[\lceil x \wedge \varphi_2 \rceil] \to \neg\lceil x \wedge \neg\varphi_2 \rceil$
6	$C_\sigma[\lceil x \wedge \varphi_2 \rceil] \to \lceil x \wedge \varphi_2 \rceil$
7	$C_\sigma[\varphi_1 \wedge \lceil x \wedge \varphi_2 \rceil] \to C_\sigma[\lceil x \wedge \varphi_2 \rceil]$
8	$C_\sigma[\varphi_1 \wedge \lceil x \wedge \varphi_2 \rceil] \to \lceil x \wedge \varphi_2 \rceil$
9	$C_\sigma[\varphi_1 \wedge (x \in \varphi_2)] \to (x \in \varphi_2)$

∎

Lemma 61. $\vdash \exists y.((x = y) \wedge \varphi) = \varphi[x/y]$ *where x, y distinct.*

Proof: The proof is by induction on the structural of φ and Lemma 60. ∎

Lemma 62. $\vdash \varphi = \exists y.(\lceil y \wedge \varphi \rceil \wedge y)$ *if* $y \notin FV(\varphi)$.

Proof: We first prove $\vdash \exists y.(\lceil y \wedge \varphi \rceil \wedge y) \to \varphi$.

1	$\neg(\lceil y \wedge \varphi \rceil \wedge (y \wedge \neg\varphi))$	(Singleton Variable)
2	$\lceil y \wedge \varphi \rceil \wedge y \to \varphi$	by 1, FOL reasoning
3	$\forall y.(\lceil y \wedge \varphi \rceil \wedge y \to \varphi)$	by 2, axiom
4	$\exists y.(\lceil y \wedge \varphi \rceil \wedge y) \to \varphi$	by 3, FOL reasoning

We then prove $\vdash \varphi \to \exists y.(\lceil y \wedge \varphi \rceil \wedge y)$. Let x be a fresh variable distinct from y.

1	$x \in \varphi \to x \in \varphi$
2	$x \in \varphi \to \lceil x \wedge \varphi \rceil$
3	$x \in \varphi \to \lceil x \wedge \lceil x \wedge \varphi \rceil \rceil$
4	$x \in \varphi \to x \in \lceil x \wedge \varphi \rceil$
5	$x \in \varphi \to \exists y.(x = y \wedge x \in \lceil y \wedge \varphi \rceil)$
6	$x \in \varphi \to \exists y.(x \in y \wedge x \in \lceil y \wedge \varphi \rceil)$
7	$x \in \varphi \to \exists y.(x \in (y \wedge \lceil y \wedge \varphi \rceil))$
8	$x \in \varphi \to x \in \exists y.(y \wedge \lceil y \wedge \varphi \rceil)$
9	$x \in (\varphi \to \exists y.(y \wedge \lceil y \wedge \varphi \rceil))$
10	$\forall x.(x \in (\varphi \to \exists y.(y \wedge \lceil y \wedge \varphi \rceil)))$
11	$\varphi \to \exists y.(y \wedge \lceil y \wedge \varphi \rceil)$

∎

Lemma 63. *(Membership Symbol) is provable in $\mathcal{H}$.*

Proof: We first prove $\vdash x \in C_\sigma[\varphi] \to \exists y.(y \in \varphi \wedge x \in C_\sigma[y])$. Let $\Psi \equiv \exists y.(y \in \varphi \wedge x \in C_\sigma[y])$.

1	$\exists y.(y \in \varphi \wedge x \in C_\sigma[y]) \to \Psi$
2	$\exists y.(\lceil y \wedge \varphi \rceil \wedge x \in C_\sigma[y]) \to \Psi$
3	$\exists y.(\lceil x \wedge \lceil y \wedge \varphi \rceil \rceil \wedge x \in C_\sigma[y]) \to \Psi$
4	$\exists y.(x \in \lceil y \wedge \varphi \rceil \wedge x \in C_\sigma[y]) \to \Psi$
5	$\exists y.(x \in (\lceil y \wedge \varphi \rceil \wedge C_\sigma[y])) \to \Psi$
6	$x \in \exists y.(\lceil y \wedge \varphi \rceil \wedge C_\sigma[y]) \to \Psi$
7	$x \in \exists y.C_\sigma[\lceil y \wedge \varphi \rceil \wedge y] \to \Psi$
8	$x \in C_\sigma[\exists y.\lceil y \wedge \varphi \rceil \wedge y] \to \Psi$
9	$x \in C_\sigma[\varphi] \to \Psi$

We then prove $\vdash \exists y.(y \in \varphi \wedge x \in C[y]) \to x \in C[\varphi]$. In fact, we just need to apply the same derivation as above on $\vdash \Psi \to \exists y.(y \in \varphi \wedge x \in C[y])$.

∎

We are now ready to prove Theorem 15.

Proof of Theorem 15: By the completeness of $\mathcal{P}$ (Theorem 9), we have $\Gamma \vdash_{\mathcal{P}} \varphi$. We have shown that all proof rules in $\mathcal{P}$ are provable in $\mathcal{H}$ with (Definedness) axioms, so $\Gamma \vdash_{\mathcal{H}} \varphi$. ∎

Appendix D
Proof of Theorem 16

We prove the completeness theorem of $\mathcal{H}$ (Theorem 16). We only discuss ML (without μ) in this section, so we drop all unnecessary annotations. Specifically, we abbreviate "$\vDash_{\mathsf{ML}}$" as "$\vDash$"; "$\vdash_{\mathcal{H}}$" as "$\vdash$"; "$\text{Pattern}^{\mathsf{ML}}$" as "Pattern", etc.

For simplicity of some technical proofs, we assume that $\{\wedge, \neg, \exists\}$ is our set of primitives, instead of $\{\to, \neg, \forall\}$. This is justified by Proposition 44.

Our proof technique was mainly inspired by [18].

Lemma 64 (Substitution Lemma)**.** $\bar{\rho}(\varphi[y/x]) = \overline{\rho[\rho(y)/x]}(\varphi)$.

Proof: Carry out induction on the structure of φ. The only nontrivial case is when $\varphi \equiv \exists z.\psi$. Without loss of generality, let us assume z is distinct from x and y. If not, apply α-renaming to make them different. Then

$$
\begin{aligned}
&\bar{\rho}((\exists z.\psi)[y/x]) \\
\equiv\ &\bar{\rho}(\exists z.(\psi[y/x])) \\
\equiv\ &\bigcup\{\overline{\rho_1}(\psi[y/x]) \mid \rho_1 \overset{z}{\sim} \rho\} \\
\equiv\ &\bigcup\{\overline{\rho_1'}(\psi) \mid \rho_1 \overset{z}{\sim} \rho \text{ and } \rho_1' = \rho_1[\rho_1(y)/x]\} \\
\equiv\ &\bigcup\{\overline{\rho_1'}(\psi) \mid \rho_1 \overset{z}{\sim} \rho \text{ and } \rho_1' = \rho_1[\rho(y)/x]\} \\
\equiv\ &\bigcup\{\overline{\rho_1'}(\psi) \mid \rho_1 \overset{z}{\sim} \rho[\rho(y)/x]\} \\
\equiv\ &\bigcup\{\overline{\rho_1'}(\psi) \mid \rho_1 \overset{z}{\sim} \rho'\} \\
\equiv\ &\overline{\rho'}(\exists z.\psi)
\end{aligned}
$$

■

Definition 65 (Local Provability)**.** Let s be a sort, $H_s \subseteq \textsc{Pattern}_s$ be a pattern set, and φ_s be a pattern of sort s. We write $H_s \Vdash_s \varphi_s$, if there exists a finite subset $\Delta_s \subseteq_{fin} H_s$ such that $\emptyset \vdash_s \bigwedge \Delta_s \to \varphi_s$, where $\bigwedge \Delta_s$ is the conjunction of all patterns in Δ_s. When Δ_s is the empty set, $\bigwedge \Delta_s$ is $\top_s$. Let $H = \{H_s\}_{s\in S}$ be a family set of patterns. We write $H \Vdash_s \varphi_s$ if $H_s \Vdash_s \varphi_s$. We drop sort subscripts when there is no confusion.

Definition 66 (Consistent Sets)**.** Let Γ_s be a pattern set of sort s. We say Γ_s is *consistent*, if $\Gamma_s \nVdash \bot_s$. Γ_s is a maximal consistent set (MCS) if any strict extension of it is inconsistent. By abuse of language, we say $\Gamma = \{\Gamma_s\}_{s\in S}$ is consistent if every Γ_s is consistent, and Γ is an MCS if every Γ_s is an MCS.

Like the local provability relation, consistency is also a local property. Pattern set Γ_s is consistent (or an MCS) only depends on itself. A useful intuition about consistent sets is that they provide consistent "views" of patterns. Recall that patterns in matching logic match elements in domain. Intuitively speaking, a pattern set Γ_s is inconsistent if it contains patterns that cannot match common elements in any models and valuations. In other words, if Γ_s is consistent, then there exist a model M and a valuation ρ, and an element a in the model, such that all patterns in Γ_s match a, i.e., $a \in \bar{\rho}(\varphi)$ for all pattern $\varphi \in \Gamma_s$. If Γ_s is in addition an MCS, adding any pattern $\psi \notin \Gamma_s$ will lead to inconsistency, and thus $a \notin \bar{\rho}(\psi_s)$. Therefore, we can think of the MCS Γ_s representing that particular element a, with all patterns in Γ_s matching it while patterns outside Γ_s not. This useful intuition motivates the definition of canonical models that consist MCSs as elements (see Definition 70), and the Truth Lemma that says "Matching = Membership in MCSs", connecting syntax and semantics, (see Lemma 79). They play an important role in proving the completeness result, including both local and global completeness theorems. The rest of the section is all about making this intuition work.

Proposition 67 (MCS Properties)**.** *Given an MCS Γ and patterns $\varphi, \varphi_1, \varphi_2$ of the same sort s. The following propositions hold.*

1) *$\varphi \in \Gamma$ if and only if $\Gamma \Vdash \varphi$; In particular, if $\vdash \varphi$ then $\varphi \in \Gamma$;*
2) *$\neg\varphi \in \Gamma$ if and only if $\varphi \notin \Gamma$;*
3) *$\varphi_1 \wedge \varphi_2 \in \Gamma$ if and only if $\varphi_1 \in \Gamma$ and $\varphi_2 \in \Gamma$; In general, for any finite pattern set Δ, $\bigwedge \Delta \in \Gamma$ if and only if $\Delta \subseteq \Gamma$;*
4) *$\varphi_1 \vee \varphi_2 \in \Gamma$ if and only if $\varphi_1 \in \Gamma$ or $\varphi_2 \in \Gamma$; In general, for any finite pattern set Δ, $\bigvee \Delta \in \Gamma$ if and only if $\Delta \cap \Gamma \neq \emptyset$; As a convention, when $\Delta = \emptyset$, $\bigvee \Delta$ is $\bot$;*
5) *$\varphi_1, \varphi_1 \to \varphi_2 \in \Gamma$ implies $\varphi_2 \in \Gamma$; In particular, if $\vdash \varphi_1 \to \varphi_2$, then $\varphi_1 \in \Gamma$ implies $\varphi_2 \in \Gamma$.*

Proof: Standard propositional reasoning. ■

Definition 68 (Witnessed MCSs)**.** Let Γ be an MCS of sort s. Γ is a witnessed MCS, if for any pattern $\exists x.\varphi \in \Gamma$, there is a variable y such that $(\exists x.\varphi) \to \varphi[y/x] \in \Gamma$. By abuse use of language, we say the family set $\Gamma = \{\Gamma_s\}_{s\in S}$ is a witnessed MCS if every Γ_s is a witnessed MCS.

In the following, we show any consistent set Γ can be extended to a witnessed MCS Γ^+. The extension, however, requires an extension of the set of variables. To see why such an extension is needed, consider the following example. Let $\mathbb{\Sigma} = (S, \textsc{Var}, \Sigma)$ be a signature, $s \in S$ be a sort, and $\Gamma = \{\neg x \mid x \in \textsc{Var}_s\}$ be a pattern set containing all variable negations. We leave it for the readers to show that Γ is consistent. Here, we claim the consistent set Γ cannot be extended to a witnessed MCS Γ^+ in the signature $\mathbb{\Sigma}$. The proof is by contradiction. Assume Γ^+ exists. By Proposition 67 and (Existence), $\exists x.x \in \Gamma^+$. Because Γ^+ is a witnessed MCS, there is a variable y such that $(\exists x.x) \to y \in \Gamma^+$, and by Proposition 67, $y \in \Gamma^+$. On the other hand, $\neg y \in \Gamma \subseteq \Gamma^+$. This contradicts the consistency of Γ^+.

Lemma 69 (Extension Lemma)**.** *Let $\mathbb{\Sigma} = (S, \textsc{Var}, \Sigma)$ be a signature and Γ be a consistent set of sort $s \in S$. Extend the variable set $\textsc{Var}$ to $\textsc{Var}^+$ with countably infinitely many new variables, and denoted the extended signature as $\mathbb{\Sigma}^+ = (\textsc{Var}^+, S, \Sigma)$. There exists a pattern set Γ^+ in the extended signature $\mathbb{\Sigma}$ such that $\Gamma \subseteq \Gamma^+$ and Γ^+ is a witnessed MCS.*

Proof: We use $\textsc{Pattern}_s$ and $\textsc{Pattern}_s^+$ denote the set of all patterns of sort s in the original and extended signatures, respectively. Enumerate all patterns $\varphi_1, \varphi_2, \cdots \in \textsc{Pattern}_s^+$. For every sort s, enumerate all variables $\aleph_1{:}s, \aleph_2{:}s, \ldots$ in $\textsc{Var}_s^+ \setminus \textsc{Var}_s$. We will construct a non-decreasing sequence of pattern sets $\Gamma_0 \subseteq \Gamma_1 \subseteq \Gamma_2 \cdots \subseteq \textsc{Pattern}_s^+$, with $\Gamma_0 = \Gamma$. Notice that Γ_0 contains variables only in $\textsc{Var}$. Eventually, we will let $\Gamma^+ = \bigcup_{i\geq 0} \Gamma_i$ and prove it has the intended properties.

For every $n \geq 1$, we define Γ_n as follows. If $\Gamma_{n-1} \cup \{\varphi_n\}$ is inconsistent, then $\Gamma_n = \Gamma_{n-1}$. Otherwise,

if φ_n is not of the form $\exists x{:}s'.\psi$:

$$\Gamma_n = \Gamma_{n-1} \cup \{\varphi_n\}$$

if $\varphi_n \equiv \exists x{:}s'.\psi$ and $\aleph_i{:}s'$ is the first variable in $\textsc{Var}_{s'}^+ \setminus \textsc{Var}_{s'}$ that does not occur free in Γ_{n-1} and ψ:

$$\Gamma_n = \Gamma_{n-1} \cup \{\varphi_n\} \cup \{\psi[\aleph_i{:}s'/x{:}s']\}$$

Notice that in the second case, we can always pick a variable $\aleph_i{:}s'$ that satisfies the conditions because by construction, $\Gamma_{n-1} \cup \{\varphi_n\}$ uses at most finitely many variables in $\textsc{Var}^+ \setminus \textsc{Var}$.

We show that Γ_n is consistent for every $n \geq 0$ by induction. The base case is to show Γ_0 is consistent in the extended signature. Assume it is not. Then there exists a finite subset

$\Delta_0 \subseteq_{fin} \Gamma_0$ such that $\vdash \bigwedge \Delta_0 \to \bot$. The proof of $\bigwedge \Delta_0 \to \bot$ is a finite sequence of patterns in PATTERN^+. We can replace every occurrence of the variable $\mathbb{y} \in \text{VAR}^+ \setminus \text{VAR}$ ($\mathbb{y}$ can have any sort) with a variable $y \in \text{VAR}$ that has the same sort as $\mathbb{y}$ and does not occur (no matter bound or free) in the proof. By induction on the length of the proof, the resulting sequence is also a proof of $\bigwedge \Delta_0 \to \bot$, and it consists of only patterns in PATTERN. This contradicts the consistency of Γ_0 as a subset of PATTERN_s, and this contradiction finishes our proof of the base case.

Now assume Γ_{n-1} is consistent for $n \geq 1$. We will show Γ_n is also consistent. If $\Gamma_{n-1} \cup \{\varphi_n\}$ is inconsistent or φ_n does not have the form $\exists x{:}s'.\psi$, Γ_n is consistent by construction. Assume $\Gamma_{n-1} \cup \{\varphi_n\}$ is consistent, $\varphi_n \equiv \exists x{:}s'.\psi$, but $\Gamma_n = \Gamma_{n-1} \cup \{\varphi_n\} \cup \{\psi[\mathbb{x}_i{:}s'/x{:}s']\}$ is not consistent. Then there exists a finite subset $\Delta \subseteq_{fin} \Gamma_{n-1} \cup \{\varphi_n\}$ such that $\vdash \bigwedge \Delta \to \neg\psi[\mathbb{x}_i{:}s'/x{:}s']$. By (UNIVERSAL GENERALIZATION), $\vdash \forall \mathbb{x}_i{:}s'.(\bigwedge \Delta \to \neg\psi[\mathbb{x}_i{:}s'/x{:}s'])$. Notice that $\mathbb{x}_i{:}s' \notin FV(\bigwedge \Delta)$ by construction, so by FOL reasoning $\vdash \bigwedge \Delta \to \neg\exists \mathbb{x}_i{:}s'.(\psi[\mathbb{x}_i{:}s'/x{:}s'])$. Since $\mathbb{x}_i{:}s' \notin FV(\psi)$, by α-renaming, $\exists \mathbb{x}_i{:}s'.(\psi[\mathbb{x}_i{:}s'/x{:}s']) \equiv \exists x{:}s'.\psi \equiv \varphi_n$, and thus $\vdash \bigwedge \Delta \to \neg\varphi_n$. This contradicts the assumption that $\Gamma_{n-1} \cup \{\varphi_n\}$ is consistent.

Since Γ_n is consistent for any $n \geq 0$, $\Gamma^+ = \bigcup_n \Gamma_n$ is also consistent. This is because the derivation that shows inconsistency would use only finitely many patterns in Γ^+. In addition, we know Γ^+ is maximal and witnessed by construction. ∎

We will prove that for every witnessed MCS $\Gamma = \{\Gamma_s\}_{s \in S}$, there exists a model M and a valuation ρ such that for every $\varphi \in \Gamma_s$, $\bar{\rho}(\varphi) \neq \emptyset$. The next definition defines the canonical model which contains all witnessed MCSs as its elements. We will construct our intended model M as a submodel of the canonical model.

Definition 70 (Canonical Model)**.** Given a signature $\mathbb{\Sigma} = (S, \Sigma)$. The canonical model $W = (\{W_s\}_{s \in S}, \_{}_W)$ consists of

- a carrier set $W_s = \{\Gamma \mid \Gamma \text{ is a witnessed MCS of sort } s\}$ for every sort $s \in S$;
- an interpretation $\sigma_W : W_{s_1} \times \cdots \times W_{s_n} \to \mathcal{P}(W_s)$ for every symbol $\sigma \in \Sigma_{s_1 \ldots s_n, s}$, defined as $\Gamma \in \sigma_W(\Gamma_1, \ldots, \Gamma_n)$ if and only if for any $\varphi_i \in \Gamma_i$, $1 \leq i \leq n$, $\sigma(\varphi_1, \ldots, \varphi_n) \in \Gamma$; In particular, the interpretation for a constant symbol $\sigma \in \Sigma_{\lambda, s}$ is $\sigma_W = \{\Gamma \in W_s \mid \sigma \in \Gamma\}$.

The carrier set W is not empty, thanks to Lemma 69.

The canonical model has a nontrivial property stated as the next lemma. The proof of the lemma is difficult, so we leave it to the end of the subsection.

Theorem 71 (Existence Lemma)**.** *Let $\mathbb{\Sigma} = (S, \Sigma)$ be a signature and Γ be a witnessed MCS of sort $s \in S$. Given a symbol $\sigma \in \Sigma_{s_1 \ldots s_n, s}$ and patterns $\varphi_1, \ldots, \varphi_n$ of appropriate sorts. If $\sigma(\varphi_1, \ldots, \varphi_n) \in \Gamma$, then there exist n witnessed MCSs $\Gamma_1, \ldots, \Gamma_n$ of appropriate sorts such that $\varphi_i \in \Gamma_i$ for every $1 \leq i \leq n$, and $\Gamma \in \sigma_W(\Gamma_1, \ldots, \Gamma_n)$.*

Definition 72 (Generated Models)**.** Let $\mathbb{\Sigma} = (S, \Sigma)$ be a signature and $W = (\{W_s\}_{s \in S}, \_{}_W)$ be the canonical model. Given a witnessed MCS $\Gamma = \{\Gamma_s\}_{s \in S}$. Define $Y = \{Y_s\}_{s \in S}$ be the smallest sets such that $Y_s \subseteq W_s$ for every sort s, and the following inductive properties are satisfied:

- $\Gamma_s \in Y_s$ for every sort s;
- If $\Delta \in Y_s$ and there exist a symbol $\sigma \in \Sigma_{s_1 \ldots s_n, s}$ and witnessed MCSs $\Delta_1, \ldots, \Delta_n$ of appropriate sorts such that $\Delta \in \sigma_W(\Delta_1, \ldots, \Delta_n)$, then $\Delta_1 \in Y_{s_1}, \ldots, \Delta_n \in Y_{s_n}$.

Let $Y = (Y, \_{}_Y)$ be the model generated from Γ, where

$$\sigma_Y(\Delta_1, \ldots, \Delta_n) = Y_s \cap \sigma_W(\Delta_1, \ldots, \Delta_n) \quad \text{for every } \sigma \in \Sigma_{s_1 \ldots s_n, s} \text{ and } \Delta_1 \in Y_{s_1}, \ldots, \Delta_n \in Y_{s_n}.$$

We give some intuition about the generated model $Y = (Y, \_{}_Y)$. The interpretation σ_Y is just the restriction of the interpretation σ_M on Y. The carrier set Y is defined inductively. Firstly, Y contains Γ. Given a set $\Delta \in Y$. If sets $\Delta_1, \ldots, \Delta_n$ are "generated" from Δ by a symbol σ, meaning that $\Delta \in \sigma_W(\Delta_1, , \ldots, \Delta_n)$, then they are also in Y. Of course, a set Δ is in Y maybe because it is generated from a set Δ' by a symbol σ', while Δ' is generated from a set Δ'' by a symbol σ'', and so on. This generating path keeps going and eventually ends at Γ in finite number of steps. By definition, every member of Y has at least one such generating path, which we formally define as follows.

Definition 73 (Generating Paths)**.** Let $\Gamma = \{\Gamma_s\}_{s \in S}$ be a witnessed MCS and Y be the model generated from Γ. A *generating path* π is either the empty path ϵ, or a sequence of pairs $\langle(\sigma_1, p_1), \ldots, (\sigma_k, p_k)\rangle$ where $\sigma_1, \ldots, \sigma_k$ are symbols (not necessarily distinct) and $p_1, \ldots, p_k$ are natural numbers representing positions. The *generating path relation*, denoted as GP, is a binary relation between witnessed MCSs in Y and generating paths, defined as the smallest relation that satisfies the following conditions:

- $GP(\Gamma_s, \epsilon)$ holds for every sort s;
- If $GP(\Delta, \pi)$ holds for a set $\Delta \in Y_s$ and a generating path π, and there exist a symbol $\sigma \in \Sigma_{s_1 \ldots s_n, s}$ and witnessed MCSs $\Delta_1, \ldots, \Delta_n$ such that $\Delta \in \sigma_W(\Delta_1, \ldots, \Delta_n)$, then $GP(\Delta_i, \langle \pi, (\sigma, i)\rangle)$ holds for every $1 \leq i \leq n$.

We say that Δ has a generating path π in the generated model if $GP(\Delta, \pi)$ holds. It is easy to see that every witnessed MCS in Y has at least one generating path, and if a witnessed MCS of sort s has the empty path ϵ as its generating path, it must be Γ_s itself.

Definition 74 (Symbol Contexts for Generating Paths)**.** Given a generating path π. Define the symbol context C_π inductively as follows. If $\pi = \epsilon$, then C_π is the identity context $\Box$. If $\pi = \langle \pi_0, (\sigma, i)\rangle$ where $\sigma \in \Sigma_{s_1 \ldots s_n, s}$ and $1 \leq i \leq n$, then $C_\pi = C_{\pi_0}[\sigma(\top_{s_1}, \ldots, \top_{s_{i-1}}, \Box, \top_{s_{i+1}}, \ldots, \top_{s_n})]$.

A good intuition about Definition 74 is given as the next lemma.

Lemma 75. *Let Γ be a witnessed MCS and Y be the model generated from Γ. Let $\Delta \in Y$. If Δ has a generating path π, then $C_\pi[\varphi] \in \Gamma$ for any pattern $\varphi \in \Delta$.*

Proof: The proof is by induction on the length of the generating path π. If π is the empty path ϵ, then Δ must be

Γ and C_π is the identity context, and $C_\pi[\varphi] = \varphi \in \Gamma$ for any $\varphi \in \Delta$. Now assume Δ has a generating path $\pi = \langle \pi_0, (\sigma, i)\rangle$ with $\sigma \in \Sigma_{s_1 \dots s_n, s}$. By Definition 73, there exist witnessed MCSs $\Delta_{s_1}, \dots, \Delta_{s_n}, \Delta_s \in Y$ and $1 \leq i \leq n$ such that $\Delta = \Delta_{s_i}$, $\Delta_s \in \sigma_W(\Delta_{s_1}, \dots, \Delta_{s_n})$, and Δ_s has π_0 as its generating path. For every $\varphi \in \Delta = \Delta_i$, since $\top_{s_j} \in \Delta_{s_j}$ for any $j \neq i$, by Definition 70, $\sigma(\top_{s_1}, \dots, \top_{s_{i-1}}, \varphi, \top_{s_{i+1}}, \dots, \top_{s_n}) \in \Delta_s$. By induction hypothesis, $C_{\pi_0}[\sigma(\top_{s_1}, \dots, \top_{s_{i-1}}, \varphi, \top_{s_{i+1}}, \dots, \top_{s_n})] \in \Gamma$, while the latter is exactly $C_\pi[\varphi]$. ∎

Lemma 76 (Singleton Variables)**.** *Let Γ be a witnessed MCS and Y be the model generated from Γ. For every $\Gamma_1, \Gamma_2 \in Y$ of the same sort and every variable x, if $x \in \Gamma_1 \cap \Gamma_2$ then $\Gamma_1 = \Gamma_2$.*

Proof: Let π_i be a generating path of Γ_i for $i = 1, 2$. Assume $\Gamma_1 \neq \Gamma_2$. Then there exists a pattern φ such that $\varphi \in \Gamma_1$ and $\neg\varphi \in \Gamma_2$. Because $x \in \Gamma_1 \cap \Gamma_2$, we know $x \wedge \varphi \in \Gamma_1$ and $x \wedge \neg\varphi \in \Gamma_2$. By Lemma 75, $C_{\pi_1}[x \wedge \varphi], C_{\pi_2}[x \wedge \neg\varphi] \in \Gamma$, and thus $C_{\pi_1}[x \wedge \varphi] \wedge C_{\pi_2}[x \wedge \neg\varphi] \in \Gamma$. On the other hand, $\neg(C_{\pi_1}[x \wedge \varphi] \wedge C_{\pi_2}[x \wedge \neg\varphi])$ is an instance of (Singleton Variable) and thus it is included in Γ. This contradicts the consistency of Γ. ∎

We will establish an important result about generated models in Lemma 79 (the Truth Lemma), which links the semantics and syntax and is essential to the completeness result. Roughly speaking, the lemma says that for any generated model Y and any witnessed MCS $\Delta \in Y$, a pattern φ is in Δ if and only if the interpretation of φ in Y contains Δ. To prove the lemma, it is important to show that every variable is interpreted to a singleton. Lemma 76 ensures that every variable belongs to *at most one* witnessed MCS. To make sure it is interpreted to *exactly one* MCS, we complete our model by adding a dummy element $\star$ to the carrier set, and interpreting all variables which are interpreted to none of the MCSs to the dummy element. This motivates the next definition.

Definition 77 (Completed Models and Completed Valuations)**.** Let $\Gamma = \{\Gamma_s\}_{s \in S}$ be a witnessed MCS and Y be the Γ-generated model. Γ-*completed model*, denoted as $M = (\{M_s\}_{s \in S}, \_M)$, is inductively defined as follows for all sorts $s \in S$:

- $M_s = Y_s$, if every $x{:}s$ belongs at least one MCS in Y_s;
- $M_s = Y_s \cup \{\star_s\}$, otherwise.

We assume $\star_s$ is an entity that is different from any MCSs, and $\star_{s_1} \neq \star_{s_2}$ if $s_1 \neq s_2$. For every $\sigma \in \Sigma_{s_1 \dots s_n, s}$, define its interpretation

$$\sigma_M(\Delta_1, \dots, \Delta_n) = \begin{cases} \emptyset & \text{if some } \Delta_i = \star_{s_i} \\ \sigma_Y(\Delta_1, \dots, \Delta_n) \cup \{\star_s\} & \text{if all } \Delta_j \neq \star_{s_j} \\ & \text{and some } \Delta_i = \Gamma_{s_i} \\ \sigma_{\mathcal{Y}_{\Gamma_0}}(\Delta_1, \dots, \Delta_n) & \text{otherwise} \end{cases}$$

The completed valuation ρ: $\text{Var} \to M$ is defined as

$$\rho(x{:}s) = \begin{cases} \Delta & \text{if } x{:}s \in \Delta \\ \star_s & \text{otherwise} \end{cases}$$

The valuation ρ is a well-defined function, because by Lemma 76, if there are two witnessed MCSs Δ_1 and Δ_2 such that $x \in \Delta_1$ and $x \in \Delta_2$, then $\Delta_1 = \Delta_2$.

Now we come back to prove Lemma 71. We need the following technical lemma.

Lemma 78. *Let $\sigma \in \Sigma_{s_1 \dots s_n, s}$ be a symbol, $\Phi_1, \dots, \Phi_n, \phi$ be patterns of appropriate sorts, and $y_1, \dots, y_n, x$ be variables of appropriate sorts such that $y_1, \dots, y_n$ are distinct, and $y_1, \dots, y_n \notin FV(\phi) \cup \bigcup_{1 \leq i \leq n} FV(\Phi_i)$. Then*

$$\begin{aligned} &\vdash \sigma(\Phi_1, \dots, \Phi_n) \\ &\quad \to \exists y_1, \dots, \exists y_n. \\ &\qquad \sigma(\Phi_1 \wedge (\exists x.\phi \to \phi[y_1/x]), \dots, \Phi_n \wedge (\exists x.\phi \to \phi[y_n/x])) \end{aligned}$$

Proof: Notice that for every $1 \leq i \leq n$,

$$\vdash \exists x.\phi \to \exists y_i.(\phi[y_i/x]).$$

By easy matching logic reasoning,

$$\begin{aligned} &\vdash \sigma(\Phi_1, \dots, \Phi_n) \\ &\quad \to \sigma(\Phi_1 \wedge (\exists x.\phi \to \exists y_1.(\phi[y_1/x])), \\ &\qquad \dots, \\ &\qquad \Phi_n \wedge (\exists x.\phi \to \exists y_n.(\phi[y_n/x]))) \end{aligned}$$

Then use Proposition 43 to move the quantifiers $\exists y_1, \dots, \exists y_n$ to the top. ∎

Now we are ready to prove Lemma 78.

Proof of Lemma 78: Recall that $\Gamma \in \sigma_W(\Gamma_1, \dots, \Gamma_n)$ means for every $\phi_i \in \Gamma_i$, $1 \leq i \leq n$, $\sigma(\phi_1, \dots, \phi_n) \in \Gamma$. The main technique that we will be using here is similar to Lemma 69. We start with the singleton sets $\{\varphi_i\}$ for every $1 \leq i \leq n$ and extend them to witnessed MCSs Γ_i, while this time we also need to make sure the results $\Gamma_1, \dots, \Gamma_n$ satisfy the desired property $\Gamma \in \sigma_W(\Gamma_1, \dots, \Gamma_n)$. Another difference compared to Lemma 69 is that this time we do not extend our set of variables, because our starting point, $\{\varphi_i\}$, contains just one pattern and uses only finitely many variables. Readers will see how these conditions play a role in the upcoming proof.

Enumerate all patterns of sorts $s_1, \dots, s_n$ as follows $\psi_0, \psi_1, \psi_2, \dots \in \bigcup_{1 \leq i \leq n} \text{Pattern}_{s_i}$. Notice that $s_1, \dots, s_n$ do not need to be all distinct. To ease our notation, we define a "choice" operator, denoted as $[\varphi_s]_{s'}$, as follows

$$[\varphi_s]_{s'} = \begin{cases} \varphi_s & \text{if } s = s' \\ \text{nothing} & \text{otherwise} \end{cases}$$

For example, $\varphi_s \wedge [\psi]_s$ means $\varphi_s \wedge \psi$ if ψ also has sort s. Otherwise, it means φ_s. The choice operator propagates with all logic connectives in the natural way. For example, $[\neg\psi]_s = \neg[\psi]_s$.

In the following, we will define a non-decreasing sequence of pattern sets $\Gamma_i^{(0)} \subseteq \Gamma_i^{(1)} \subseteq \Gamma_i^{(2)} \subseteq \dots \subseteq \text{Pattern}_{s_i}$ for each $1 \leq i \leq n$, such that the following conditions are true for all $1 \leq i \leq n$ and $k \geq 0$:

1) If ψ_k has sort s_i, then either ψ_k or $\neg\psi_k$ belongs to $\Gamma_i^{(k+1)}$.

2) If ψ_k has the form $\exists x.\phi_k$ and it belongs to $\Gamma_i^{(k+1)}$, then there exists a variable z such that $(\exists x.\phi_k) \to \phi_k[z/x]$ also belongs to $\Gamma_i^{(k+1)}$.
3) $\Gamma_i^{(k)}$ is finite.
4) Let $\pi_i^{(k)} = \bigwedge \Gamma_i^{(k)}$ for every $1 \le i \le n$. Then $\sigma(\pi_1^{(k)}, \ldots, \pi_n^{(k)}) \in \Gamma$.
5) $\Gamma_i^{(k)}$ is consistent.

Among the above five conditions, condition (2)–(5) are like "safety" properties while condition (1) is like a "liveness" properties. We will eventually let $\Gamma_i = \bigcup_{k\ge 0} \Gamma_i^{(k)}$ and prove that Γ_i has the desired property. Before we present the actual construction, we give some hints on how to prove these conditions hold. Conditions (1)–(3) will be satisfied directly by construction, although we will put a notable effort in satisfying condition (2). Condition (4) will be proved hold by induction on k. Condition (5) is in fact a consequence of condition (4) as shown below. Assume condition (4) holds but condition (5) fails. This means that $\Gamma_i^{(k)}$ is not consistent for some $1 \le i \le n$, so $\vdash \pi_i^{(k)} \to \bot$. By (Framing)

$$\vdash \sigma(\pi_1^{(k)}, \ldots, \pi_i^{(k)}, \ldots, \pi_n^{(k)}) \to \sigma(\pi_1^{(k)}, \ldots, \bot, \ldots, \pi_n^{(k)})$$

Then by Proposition 43 and FOL reasoning,

$$\vdash \sigma(\pi_1^{(k)}, \ldots, \pi_i^{(k)}, \ldots, \pi_n^{(k)}) \to \bot$$

Since $\sigma(\pi_1^{(k)}, \ldots, \pi_i^{(k)}, \ldots, \pi_n^{(k)}) \in \Gamma$ by condition (4), we know $\bot \in \Gamma$ by Proposition 67. And this contradicts the fact that Γ is consistent.

Now we are ready to construct the sequence $\Gamma_i^{(0)} \subseteq \Gamma_i^{(1)} \subseteq \Gamma_i^{(2)} \subseteq \ldots$ for $1 \le i \le n$. Let $\Gamma_i^{(0)} = \{\varphi_i\}$ for $1 \le i \le n$. Obviously, $\Gamma_i^{(0)}$ satisfies conditions (3) and (4). Condition (5) follows as a consequence of condition (4). Conditions (1) and (2) are not applicable.

Suppose we have already constructed sets $\Gamma_i^{(k)}$ for every $1 \le i \le n$ and $k \ge 0$, which satisfy the conditions (1)–(5). We show how to construct $\Gamma_i^{(k+1)}$. In order to satisfy condition (1), we should add either ψ_k or $\neg\psi_k$ to $\Gamma_i^{(k)}$, if $\Gamma_i^{(k)}$ has the same sort as ψ_k. Otherwise, we simply let $\Gamma_i^{(k+1)}$ be the same as $\Gamma_i^{(k)}$. The question here is: if $\Gamma_i^{(k)}$ has the same sort as ψ_k, which pattern should we add to $\Gamma_i^{(k)}$, ψ_k or $\neg\psi_k$? Obviously, condition (3) will still hold no matter which one we choose to add, so we just need to make sure that we do not break conditions (2) and (4).

Let us start by satisfying condition (4). Consider pattern $\sigma(\pi_1^{(k)}, \ldots, \pi_n^{(k)})$, which, by condition (4), is in Γ. This tells us that the pattern

$$\sigma(\pi_1^{(k)} \wedge [\psi_k \vee \neg\psi_k]_{s_1}, \ldots, \pi_n^{(k)} \wedge [\psi_k \vee \neg\psi_k]_{s_n})$$

is also in Γ. Recall that $[\_]_s$ is the choice operator, so if ψ_k has sort s_i, then $\pi_i^{(k)} \wedge [\psi_k \vee \neg\psi_k]_{s_i}$ is $\pi_i^{(k)} \wedge (\psi_k \vee \neg\psi_k)$. Otherwise, it is $\pi_i^{(k)}$. Use Proposition 43 and FOL reasoning, and notice that the choice operator propagates with the disjunction $\vee$ and the negation $\neg$, we get

$$\sigma((\pi_1^{(k)} \wedge [\psi_k]_{s_1}) \vee (\pi_1^{(k)} \wedge \neg[\psi_k]_{s_1}), \ldots, (\pi_n^{(k)} \wedge [\psi_k]_{s_n}) \vee (\pi_n^{(k)} \wedge \neg[\psi_k]_{s_n})) \in \Gamma$$

Then we use Proposition 43 again and move all the disjunctions to the top, and we end up with a disjunction of 2^n patterns:

$$\bigvee \sigma(\pi_1^{(k)} \wedge [\neg]_1^{(k)}[\psi_k]_{s_1}, \ldots, \pi_n^{(k)} \wedge [\neg]_n^{(k)}[\psi_k]_{s_n}) \in \Gamma$$

where $[\neg]$ means either nothing or $\neg$. Notice that some $[\psi_k]_{s_i}$'s might be nothing, so some of these 2^n patterns may be the same.

Notice that Γ is an MCS. By proposition 67, among these 2^n patterns there must exists one pattern that is in Γ. We denote *that* pattern as

$$\sigma(\pi_1^{(k)} \wedge [\neg]_1^{(k)}[\psi_k]_{s_1}, \ldots, \pi_n^{(k)} \wedge [\neg]_n^{(k)}[\psi_k]_{s_n})$$

For any $1 \le i \le n$, if $[\neg]_i^{(k)}[\psi_k]_{s_i}$ does not have the form $\exists x.\phi$, we simply define $\Gamma_i^{(k+1)} = \Gamma_i^{(k)} \cup \{[\neg]_i^{(k)}[\psi_k]_{s_i}\}$. If $[\neg]_i^{(k)}[\psi_k]_{s_i}$ does have the form $\exists x.\phi$, we need special effort to satisfy condition (2). Without loss of generality and to ease our notation, let us assume that *for every* $1 \le i \le n$, the pattern $[\neg]_i^{(k)}[\psi_k]_{s_i}$ has the same form $\exists x.\phi$. We are going to find for each index i a variable z_i such that

$$\sigma(\pi_1^{(k)} \wedge \exists x.\phi \wedge (\exists x.\phi \to \phi[z_1/x]), \ldots, \pi_n^{(k)} \wedge \exists x.\phi \wedge (\exists x.\phi \to \phi[z_n/x])) \in \Gamma$$

This will allow us to define $\Gamma_i^{(k+1)} = \Gamma_i^{(k)} \cup \{\exists x.\phi\} \cup \{\exists x.\phi \to \phi[z_i/x]\}$ which satisfies conditions (2) and (4).

We find these variables z_i's by Lemma 78 and the fact that Γ is a witnessed set. Let $\Phi_i \equiv \pi_i^{(k)} \wedge \exists x.\phi$ for $1 \le i \le n$. By construction, $\sigma(\Phi_1, \ldots, \Phi_n) \in \Gamma$. Hence, by Lemma 78 and Proposition 67, for any distinct variables $y_1, \ldots, y_n \notin FV(\phi) \cup \bigcup_{1\le i\le n} FV(\Phi_i)$,

$$\exists y_1 \ldots \exists y_n.\ \sigma(\Phi_1 \wedge (\exists x.\phi \to \phi[y_1/x]), \ldots, \Phi_n \wedge (\exists n.\phi \to \phi[y_n/x])) \in \Gamma$$

The set Γ is a witnessed set, so there exist variables $z_1, \ldots, z_n$ such that

$$\sigma(\Phi_1 \wedge (\exists x.\phi \to \phi[z_1/x]), \ldots, \Phi_n \wedge (\exists x.\phi \to \phi[z_n/x])) \in \Gamma$$

This justifies our construction $\Gamma_i^{(k+1)} = \Gamma_i^{(k)} \cup \{\exists x.\phi\} \cup \{\exists x.\phi \to \phi[z_i/x]\}$.

So far we have proved our construction of the sequences $\Gamma_i^{(0)} \subseteq \Gamma_i^{(1)} \subseteq \Gamma_i^{(2)} \subseteq \ldots$ for $1 \le i \le n$ satisfy the conditions (1)–(5). Let $\Gamma_i = \bigcup_{k\ge 0} \Gamma_i^{(k)}$ for $1 \le i \le n$. By construction, Γ_i is a witnessed MCS. It remains to prove that $\Gamma \in \sigma_W(\Gamma_1, \ldots, \Gamma_n)$. To prove it, assume $\phi_i \in \Gamma_i$ for $1 \le i \le n$. By construction, there exists $K > 0$ such that $\phi_i \in \Gamma_i^{(K)}$ for all $1 \le i \le n$. Therefore, $\vdash \pi_i^{(K)} \to \phi_i$. By

condition (4), $\sigma(\pi_1^{(K)}, \ldots, \pi_n^{(K)}) \in \Gamma$, and thus by (Framing) and Proposition 67, $\sigma(\phi_1, \ldots, \phi_n) \in \Gamma$. ∎

Lemma 79 (Truth Lemma)**.** *Let Γ be a witnessed MCS, M be its completed model, and ρ be the completed valuation. For any witnessed MCS $\Delta \in M$ and any pattern φ such that Δ and φ have the same sort,*

$$\varphi \in \Delta \quad \text{if and only if} \quad \Delta \in \bar{\rho}(\varphi)$$

Proof: The proof is by induction on the structure of φ. If φ is a variable the conclusion follows by Definition 70. If φ has the form $\psi_1 \wedge \psi_2$ or $\neg\psi_1$, the conclusion follows from Proposition 67. If φ has the form $\sigma(\varphi_1, \ldots, \varphi_n)$, the conclusion from left to right is given by Lemma 71. The conclusion from right to left follows from Definition 70.

Now assume φ has the form $\exists x.\psi$. If $\exists x.\psi \in \Delta$, since Δ is a witnessed set, there is a variable y such that $\exists x.\psi \to \psi[y/x] \in \Delta$, and thus $\psi[y/x] \in \Delta$. By induction hypothesis, $\Delta \in \bar{\rho}(\psi[y/x])$, and thus by the semantics of the logic, $\Delta \in \bar{\rho}(\exists x.\psi)$.

Consider the other direction. Assume $\Delta \in \bar{\rho}(\exists x.\psi)$. By definition there exists a witnessed set $\Delta' \in M$ such that $\Delta \in \overline{\rho[\Delta'/x]}(\psi)$. By Definition 77, every element in M (no matter if it is an MCS or $\star$) has a variable that is assigned to it by the completed valuation ρ. Let us assume that variable y is assigned to Δ', i.e., $\rho(y) = \Delta'$. By Lemma 64, $\Delta \in \bar{\rho'}(\psi) = \bar{\rho}(\psi[y/x])$. By induction hypothesis, $\psi[y/x] \in \Delta$. Finally notice that $\vdash \psi[y/x] \to \exists y.\psi[y/x]$. By Proposition 67, $\exists y.\psi[y/x] \in \Delta$, i.e., $\exists x.\psi \in \Delta$. ∎

Theorem 80. *For any consistent set Γ, there is a model M and a valuation ρ such that for all patterns $\varphi \in \Gamma$, $\bar{\rho}(\varphi) \neq \emptyset$.*

Proof: Use Lemma 69 and extend Γ to a witnessed MCS Γ^+. Let M and ρ be the completed model and valuation generated by Γ^+ respectively. By Lemma 79, for all patterns $\varphi \in \Gamma \subseteq \Gamma^+$, we have $\Gamma^+ \in \bar{\rho}(\varphi)$, so $\bar{\rho}(\varphi) \neq \emptyset$. ∎

Now we are ready to prove Theorem 16.

Proof of Theorem 16: Assume the opposite. If $\emptyset \nvdash \varphi$, then $\{\neg\varphi\}$ is consistent by Definition 66. Then there is a model M and an valuation ρ such that $\bar{\rho}(\neg\varphi) \neq \emptyset$, i.e., $\bar{\rho}(\varphi) \neq M$. This contradicts the fact that $\emptyset \vDash \varphi$. ∎

We point out that Lemma 79 in fact gives us the following stronger completeness result of $\mathcal{H}$. In literature, Theorem 16 is called *weak local completeness theorem* while Theorem 81 is called *strong local completeness theorem.*

Theorem 81. *For any set Γ and any pattern φ, $\Gamma \vDash^{loc} \varphi$ implies $\Gamma \Vdash \varphi$, where $\Gamma \vDash^{loc} \varphi$ means that for all models M, all valuations ρ, and all elements $a \in M$, if $a \in \bar{\rho}(\psi)$ for all $\psi \in \Gamma$ then $a \in \bar{\rho}(\varphi)$.*

Proof: Assume the opposite that $\Gamma \nVdash \varphi$, which implies that $\Gamma \cup \{\neg\varphi\}$ is consistent. Extend it to a witnessed MCS Γ^+ and let M, ρ be the completed model and completed valuation generated by Γ^+. By Lemma 79, $\Gamma^+ \in \bar{\rho}(\psi)$ for all $\psi \in \Gamma$, and $\Gamma^+ \in \bar{\rho}(\neg\varphi)$, i.e., $\Gamma^+ \notin \bar{\rho}(\varphi)$. This contradicts with $\Gamma \vDash^{\text{loc}} \varphi$. ∎

Appendix E
Proof of Proposition 20

Proof of Proposition 20: Trivial. Note that MmL coincides with ML on the fragment without μ. ∎

Appendix F
Proof of Proposition 22 and 23

We prove that the theory $\Gamma_{\Sigma}^{\text{term}}$ captures precisely term algebras, up to isomorphism. The proof is mainly a feast of inductive reasoning.

Proof: Let us fix a $\mathbb{\Sigma}^+$-model M such that $M \vDash \Gamma_{\Sigma}^{\text{term}}$. By axiom (Function), the interpretation $c_M : M \times \cdots \times M \to \mathcal{P}(M)$ must be a function, where $c \in \Sigma_{Term \ldots Term\, Term}$, meaning that for all $a_1, \ldots, a_n \in M$, $c_M(a_1, \ldots, a_n)$ contains exactly one element. By abuse of language, we denote *that* element as $c_M(a_1, \ldots, a_n)$ and regard $c_M : M \times \cdots \times M \to M$ as really a function.

To prove the proposition, it suffices to establish an isomorphism between the two algebras $(M, \{c_M\}_{c \in \Sigma})$ and $(T^{\Sigma}_{Term}, \{c_{T^\Sigma}\}_{c \in \Sigma})$.

Let us define a subset $M_0 \subseteq M$ *inductively* as follows (in which we separate the cases of constant constructs from non-constant constructors for clarity):

- $c_M \in M_0$, if $c \in \Sigma_{\lambda, Term}$;
- $c_M(a_1, \ldots, a_n)$, if $c \in \Sigma_{Term \ldots Term, Term}$ and $a_1, \ldots, a_n \in M_0$.

We claim that for all valuation ρ,

$$\bar{\rho}(\mu D. \bigvee_{c \in \Sigma} c(D, \ldots, D)) = M_0.$$

We prove the equation by proving set containment for both directions. Notice that by definition,

$$\bar{\rho}(\mu D. \bigvee_{c \in \Sigma} c(D, \ldots, D)) = \bigcap \{A \subseteq M \mid \bigcup_{c \in \Sigma} c_M(A, \ldots, A) \subseteq A\}.$$

Denote the above set M_1 and we prove $M_0 = M_1$.

(Case $M_0 \subseteq M_1$). Notice that M_0 is defined inductively, so we carry out induction. The base case is $c \in \Sigma_{\lambda, Term}$ and $c_M \in M_0$. We aim to prove $c_M \in M_1$. For that purpose, assume a set $A \subseteq M$ such that $\bigcup_{c \in \Sigma} c_M(A, \ldots, A) \subseteq A$ and try to prove $c_M \in A$. This is trivial, because c_M is in the big-union set on the left. The induction case is $c \in \Sigma_{Term \ldots Term, Term}$ and $a_1, \ldots, a_n \in M_0$ and $c_M(a_1, \ldots, a_n) \in M_0$. We aim to prove $c_M(a_1, \ldots, a_n) \in M_1$. Similarly, we assume a set $A \subseteq M$ such that $\bigcup_{c \in \Sigma} c_M(A, \ldots, A) \subseteq A$ and try to prove $c_M(a_1, \ldots, a_n) \in M_0$. By induction hypothesis, $a_1, \ldots, a_n \in M_1$, which implies that $c_M(a_1, \ldots, a_n)$ is in the big-union on the left, and thus in A. Done.

(Case $M_1 \subseteq M_0$). We just need to prove that M_1 satisfies the condition that $\bigcup_{c \in \Sigma} c_M(M_0, \ldots, M_0) \subseteq M_0$, which follows directly by the construction of M_0.

Hence we conclude that $M_0 = M_1$. By axiom (Inductive Domain), $M_1 = M$ is the total set, and thus $M = M_0$. Note that (Inductive Domain) forces the model M to be an inductive one (i.e., M_0), and thus admits inductive reasoning.

We now define the isomorphism:

$$(M, \{c_M\}_{c\in\Sigma}) \underset{j}{\overset{i}{\rightleftharpoons}} (T^{\Sigma}, \{c_{T^{\Sigma}}\}_{c\in\Sigma})$$

inductively as follows:

- $i(c_M) = c$, for $c \in \Sigma_{\lambda,Term}$;
- $i(c_M(a_1,\ldots,a_n)) = c(i(a_1),\ldots,i(a_n))$, for $c \in \Sigma_{Term\ldots Term,Term}$;
- $j(c) = c_M$, for $c \in \Sigma_{\lambda,Term}$;
- $j(c(t_1,\ldots,t_n)) = c_M(j(t_1),\ldots,j(t_n))$, for $c \in \Sigma_{Term\ldots Term,Term}$.

It is then straightforward to verify that $i \circ j$ and $j \circ i$ are both identity function, by induction. In addition, they are isomorphic to each other. ∎

Proposition 23 is a direct corollary of Theorem 22.

Proof of Theorem 23: Let us fix a model $M \vDash \Gamma^{\mathbb{N}}$. By Theorem 22, the reduct of M over the sub-signature $\{0 \in \Sigma_{\lambda,Nat}, succ \in \Sigma_{Nat,Nat}\}$ is isomorphic to natural numbers $\mathbb{N}$, under the isomorphism:

$$(M, \{0_M, succ_M\}) \underset{j}{\overset{i}{\rightleftharpoons}} (\mathbb{N}, \{0, s\})$$

where $s(n) = n + 1$ is the successor function on $\mathbb{N}$.

Our aim is to show that the four axioms about *plus* and *mult* force a *unique* interpretation in M. In particular, + and $\times$ obviously give two valid interpretations under the above (i, j)-isomorphism, as they clearly satisfies the axioms. But the uniqueness of the interpretations of *plus* and *mult* is trivial, as the four axioms form a valid *inductive* definition in M. ∎

Appendix G
Properties about Proof System $\mathcal{H}_\mu$

We present and proof some important properties about $\mathcal{H}_\mu$. First of all, we can generalized Lemma 64 to the setting with set variables and μ-binder.

Lemma 82. *$\bar\rho(\varphi[\psi/X]) = \overline{\rho[\bar\rho(\psi)/X]}(\varphi)$ for all $X \in$ SVar.*

Proof: Carry out induction on the structure of φ. The only interesting case is when $\varphi \equiv \mu Z.\varphi_1$. By α-renaming, we can safely assume $Z \notin FV(\psi)$. We have:

$$\begin{aligned}
&\bar\rho((\mu Z.\varphi_1)[\psi/X])\\
&= \bar\rho(\mu Z.(\varphi_1[\psi/X]))\\
&= \bigcap\{A \mid \overline{\rho[A/Z]}(\varphi_1[\psi/X]) \subseteq A\}\\
&= \bigcap\{A \mid \overline{\rho[A/Z][\overline{\rho[A/Z]}(\psi)/X]}(\varphi_1) \subseteq A\}\\
&= \bigcap\{A \mid \overline{\rho[A/Z][\bar\rho(\psi)/X]}(\varphi_1) \subseteq A\}\\
&= \bigcap\{A \mid \overline{\rho[\bar\rho(\psi)/X][A/Z]}(\varphi_1) \subseteq A\}\\
&= \overline{\rho[\bar\rho(\psi)/X]}(\mu Z.\varphi_1)\\
&= \overline{\rho[\bar\rho(\psi)/X]}(\varphi).
\end{aligned}$$

Done. ∎

We prove the soundness theorem.

Proof of Theorem 24: The soundness of all proof rules in $\mathcal{H}$ are proved as in Theorem 13. We just need to prove the soundness of (Set Variable Substitution), (Pre-Fixpoint), and (Knaster-Tarski). Let M be a model.

(Set Variable Substitution). Assume $M \vDash \varphi$. By definition, $\bar\rho(\varphi) = M$ for all ρ. Our goal is to show $M \vDash \varphi[\psi/X]$. Let ρ be any valuation. We have $\bar\rho(\varphi[\psi/X]) = \overline{\rho[\bar\rho(\psi)/X]}(\varphi)$. Note that $\rho[\bar\rho(\psi)/X]$ is just another valuation, so $\overline{\rho[\bar\rho(\psi)/X]}(\varphi) = M$ by assumption.

(Pre-Fixpoint). Let ρ be any valuation. Our goal is to prove $\bar\rho(\varphi[\mu X.\varphi/X] \to \mu X.\varphi) = M$. By definition, $\bar\rho(\varphi[\mu X.\varphi/X]) = \overline{\rho[\bar\rho(\mu X.\varphi)/X]}(\varphi)$, and $\bar\rho(\mu X.\varphi) = \bigcap\{A \mid \overline{\rho[A/X]}(\varphi) \subseteq A\}$. By Knaster-Tarski theorem, $\bar\rho(\mu X.\varphi)$ itself is a fixpoint of $\overline{\rho[A/X]}(\varphi) = A$. Therefore, $\overline{\rho[\bar\rho(\mu X.\varphi)/X]}(\varphi) = \bar\rho(\mu X.\varphi)$. Done.

(Knaster-Tarski). Assume $M \vDash \varphi[\psi/X] \to \psi$. Our goal is to prove $M \vDash \mu X.\varphi \to \psi$. Let ρ be any valuation. We need to prove $\bar\rho(\mu X.\varphi) \subseteq \bar\rho(\psi)$. Note that $\bar\rho(\mu X.\varphi)$ is defined as the least fixpoint of $\overline{\rho[A/X]}(\varphi) = A$. By Knaster-Tarski theorem, it suffices to prove $\bar\rho(\psi)$ is a pre-fixpoint, i.e., $\overline{\rho[\bar\rho(\psi)/X]}(\varphi) \subseteq \bar\rho(\psi)$. This is given by our assumption, $M \vDash \varphi[\psi/X] \to \psi$. This implies that $\bar\rho(\varphi[\psi/X]) \subseteq \bar\rho(\psi)$, i.e., $\overline{\rho[\bar\rho(\psi)/X]}(\varphi) \subseteq \bar\rho(\psi)$. Done. ∎

Lemma 83. $\vdash \mu X.\varphi \leftrightarrow \varphi[\mu X.\varphi/X]$.

Proof: We prove both directions.

(Case "$\to$"). Apply (Knaster-Tarski), and we prove $\vdash \varphi[(\varphi[\mu X.\varphi/X])/X] \to \varphi[\mu X.\varphi/X]$. By Lemma 87, and the fact that φ is positive in X, we just need to prove $\vdash \varphi[\mu X.\varphi/X] \to \varphi$, which is proved by (Pre-Fixpoint).

(Case "$\leftarrow$") is exactly (Pre-Fixpoint). ∎

Lemma 84. *The following propositions hold:*

- *Pre-Fixpoint:* $\vdash \nu X.\varphi \to \varphi[\nu X.\varphi/X]$*;*
- *Knaster-Tarski:* $\vdash \psi \to \varphi[\psi/X]$ *implies* $\vdash \psi \to \nu X.\varphi$.

Proof: These are standard proofs as in modal μ-logic. ∎

Lemma 85. *$\Gamma \vdash \varphi_1 \to \varphi_2$ implies $\Gamma \vdash \mu X.\varphi_1 \to \mu X.\varphi_2$.*

Proof: Use (Knaster-Tarski), and then (Set Variable Substitution). ∎

Lemma 86. *For any context C, we have $\Gamma \vdash \varphi_1 \leftrightarrow \varphi_2$ if and onlyf if $\Gamma \vdash C[\varphi_1] \leftrightarrow C[\varphi_2]$.*

Proof: Carry out induction on the structure of C. Except the case $C \equiv \mu X.C_1$, all other cases have been proved in Proposition 44. While the μ-case is proved by Lemma 85. ∎

Note that Lemma 86 along with Lemma 83 allow us to "unfold" a least fixpoint pattern $\mu X.\varphi$ and replace it, in-place in any context, by $\varphi[\mu X.\varphi/X]$.

Lemma 87. *A context C is positive if it is positive in $\Box$; otherwise, it is negative. Let $\Gamma \vdash \varphi_1 \to \varphi_2$. We have*

$$\begin{aligned}
&\Gamma \vdash C[\varphi_1] \to C[\varphi_2] && \text{if } C \text{ is positive,}\\
&\Gamma \vdash C[\varphi_2] \to C[\varphi_1] && \text{if } C \text{ is negative.}
\end{aligned}$$

Proof: Carry out induction on the structure of C. The cases when C is a propositional/FOL context are trivial. The

case when C is a symbol application is proved by (Framing). The case when C is a μ-binder is proved by Lemma 85. ∎

Lemma 88. *Let ψ be a predicate pattern and C be a context where $\Box$ is not under any μ-binder. We have $\vdash \psi \wedge C[\varphi] \leftrightarrow \psi \wedge C[\psi \wedge \varphi]$ for all φ.*

Proof: Carry out induction on the structure of C. The cases when C is a propositional/FOL context are trivial. The case when C is a symbol application is proved using the fact that predicate patterns propagate through symbols. Since $\Box$ does not occur under any μ-binder, that is all cases. ∎

Lemma 89. *Let ψ be a predicate pattern and φ be a pattern. Let X be a set variable that does not occur under any μ-binder in φ, and $X \notin FV(\psi)$. We have $\vdash \psi \wedge \mu X.\varphi \leftrightarrow \mu X.(\psi \wedge \varphi)$.*

Proof: Note that "$\leftarrow$" is proved by Lemma 85. We only need to prove "$\rightarrow$". By propositional reasoning, the goal becomes $\vdash \mu X.\varphi \rightarrow \psi \rightarrow \mu X.(\psi \wedge \varphi)$ and we apply (Knaster-Tarski). We obtain $\vdash \psi \wedge \varphi[\psi \rightarrow \mu X.(\psi \wedge \varphi)/X] \rightarrow \mu X.(\psi \wedge \varphi)$. By (Pre-Fixpoint), we just need to prove $\vdash \psi \wedge \varphi[\psi \rightarrow \mu X.(\psi \wedge \varphi)/X] \rightarrow \psi \wedge \varphi[\mu X.(\psi \wedge \varphi)/X]$. By Lemma 89, we just need to prove $\vdash \psi \wedge \varphi[\psi \wedge (\psi \rightarrow \mu X.(\psi \wedge \varphi))/X] \rightarrow \psi \wedge \varphi[\mu X.(\psi \wedge \varphi)/X]$, which then by Lemma 87 becomes $\vdash \psi \wedge \varphi[\psi \wedge (\mu X.(\psi \wedge \varphi))/X] \rightarrow \psi \wedge \varphi[\mu X.(\psi \wedge \varphi)/X]$, which then follows by Lemma 89. ∎

We now obtain a version of deduction theorem for $\mathcal{H}_\mu$, which we believe is not in its strongest form, but it is good enough to prove other theorems in this paper.

Theorem 90 (Deduction Theorem of $\mathcal{H}_\mu$)**.** *Let Γ be an axiom set containing definedness axioms and φ,ψ be two patterns. If $\Gamma \cup \{\psi\} \vdash \varphi$ and the proof (1) does not use (Universal Generalization) on free element variables in ψ; (2) does not use (Knaster-Tarski),* unless *set variable X does not occur under any μ-binder in φ and $X \notin FV(\psi)$; (3) does not use (Set Variable Substitution) on free set variables in ψ, then $\Gamma \vdash \lfloor \psi \rfloor \rightarrow \varphi$.*

Proof: Carry out induction on the length of the proof $\Gamma \cup \{\psi\} \vdash \varphi$. (Base Case) and (Induction Case) for (Modus Ponens) and (Universal Generalization) are proved as in Theorem 90. We only need to prove (Induction Case) for (Knaster-Tarski) and (Set Variable Substitution).

(Knaster-Tarski). Suppose $\varphi \equiv \mu X.\varphi_1 \rightarrow \varphi_2$. We should prove that $\Gamma \vdash \lfloor \psi \rfloor \rightarrow (\mu X.\varphi_1 \rightarrow \varphi_2)$, i.e., $\Gamma \vdash \lfloor \psi \rfloor \wedge \mu X.\varphi_1 \rightarrow \varphi_2$. Note that $\lfloor \psi \rfloor$ is a predicate pattern. By Lemma 89, our goal becomes $\Gamma \vdash \mu X.(\lfloor \psi \rfloor \wedge \varphi_1) \rightarrow \varphi_2$. By (Knaster-Tarski), we need to prove $\Gamma \vdash (\lfloor \psi \rfloor \wedge \varphi_1)[\varphi_2/X] \rightarrow \varphi_2$. Note that $X \notin FV(\lfloor \psi \rfloor)$, so the above becomes $\Gamma \vdash \lfloor \psi \rfloor \wedge \varphi_1[\varphi_2/X] \rightarrow \varphi_2$, i.e., $\Gamma \vdash \lfloor \psi \rfloor \rightarrow \varphi_1[\varphi_2/X] \rightarrow \varphi_2$, which is our induction hypothesis.

(Set Variable Substitution). Trivial. Note that $X \notin FV(\psi)$. ∎

Appendix H
Proofs of Proposition 25

Proof of Proposition 25: We refer readers to [1] for some of the proof techniques that we use. Notice that $\varphi(x)$ as well as other formulas are patterns of sort *Pred*. However, the (Inductive Domain) axiom is about the sort *Nat*. Therefore, our first step is to lift φ from *Pred* to *Nat*, using the definedness symbols. In fact, we will use the membership and equality constructs that are defined from the definedness symbols. We define $N = \exists x.x \wedge \lceil \varphi(x) \rceil^{Nat}_{Pred}$, which captures the set of all numbers in which φ holds. One can prove that $x \in N = \lceil \varphi(x) \rceil^{Nat}_{Pred}$.

Since all patterns of sort *Pred* are predicate patterns, we may use the deduction theorem (Theorem 90) and assume $\varphi(0)$ and $\forall x.(\varphi(x) \rightarrow \varphi(succ(x)))$, and to prove $\forall x.\varphi(x)$. Using the equality $x \in N = \lceil \varphi(x) \rceil^{Nat}_{Pred}$, this means that we assume $0 \in N$ and $\forall x.(x \in N \rightarrow succ(x) \in N)$ and prove $\forall x.x \in N$, which implies N by (Membership Elimination).

By (Knaster-Tarski), it suffices to prove only $0 \vee succ(N) \rightarrow N$, which requires to prove $0 \rightarrow N$ and $succ(N) \rightarrow N$. The first is proved by the assumption that $0 \in N$. The second is proved by considering $y \in succ(N) \rightarrow y \in N$, which then becomes $(\exists x.y \in succ(x) \wedge x \in N) \rightarrow y \in N$. By the fact that *succ* is a function, it becomes $x \in N \rightarrow succ(x) \in N$, which is then proved by our second assumption. Done. ∎

Appendix I
Notations and Proofs about Recursive Symbols

Even though we tactically blur the distinction between constant symbol $\sigma \in \Sigma_{\lambda,s_1 \otimes \cdots \otimes s_n \otimes s}$ and n-ary symbol $\sigma \in \Sigma_{s_1 \ldots s_n,s}$, doing so will cause us a lot of trouble in this section, when our aim is to prove such as blur of syntax actually works. Therefore, within this section, we introduce and use a more distinct syntax that distinguishes the two.

We use the following notations (and their meaning):

$\sigma \in \Sigma_{s_1,\ldots,s_n,s}$	
$\alpha_\sigma \in \Sigma_{\lambda,s_1 \otimes \cdots \otimes s_n \otimes s}$	
$\sigma(\varphi_1, \ldots, \varphi_n)$	symbol application
$\alpha_\sigma[\varphi_1, \ldots, \varphi_n]$	projections
$\sigma(x_1, \ldots, x_n) = \alpha_\sigma[x_1, \ldots, x_n]$	recursive symbol
$\alpha_\sigma = \mu\alpha.\exists \vec{x}\langle \vec{x}, \varphi[\alpha/\sigma] \rangle$	definition of α_σ

Before we prove Theorem 29, we introduce a useful lemma that allows us to prove properties about least fixpoint patterns. Recall that rule (Knaster-Tarski) allows us to prove theorems of the form $\Gamma \vdash \mu X.\varphi \rightarrow \psi$. However, in practice, the least fixpoint pattern $\mu X.\varphi$ is not always the only components on the left hand side, but rather stay *within some contexts*. The following lemma is particularly useful in practice, as it allows us to "plug out" the least fixpoint pattern from its context, so that we can apply (Knaster-Tarski). After that, we may "plug it back" into the context.

Lemma 91. *Let $C[\Box]$ be a context such that $\Box$ does not occur under any μ-binder, and*

- $C[\varphi \wedge \psi] = C[\varphi] \wedge \psi$, *for all patterns* φ *and all predicate patterns* ψ*;*
- $C[\exists x.\varphi] = \exists x.C[\varphi]$, *for all* φ *and* $x \notin FV(C[\Box])$.

Then we have that $\Gamma \vdash C[\varphi] \to \psi$ *if and only if* $\Gamma \vdash \varphi \to \exists x.x \wedge \lfloor C[x] \to \psi \rfloor$.

Proof: We prove both directions simultaneously. Note that it is easy to prove that $\Gamma \vdash \varphi = \exists x.(x \wedge (x \in \varphi))$ using rules (Membership) in the proof system $\mathcal{P}$ (see Fig. 3).

We start with $\Gamma \vdash C[\varphi] \to \psi$. By the mentioned equality, we get $\Gamma \vdash C[\exists x.(x \wedge (x \in \varphi))] \to \psi$. By the properties of C, it becomes $\Gamma \vdash (\exists x.C[x] \wedge x \in \varphi) \to \psi$, which, by FOL reasoning, becomes $\Gamma \vdash x \in \varphi \to (C[x] \to \psi)$. Note that $x \in \varphi$ is a predicate pattern, so the goal is equivalent to $\Gamma \vdash x \in \varphi \to \lfloor C[x] \to \psi \rfloor$.

Now we are almost done. To show the "if" part, we apply (Membership Introduction) on $\Gamma \vdash \varphi \to \exists x.x \wedge \lfloor C[x] \to \psi \rfloor$ and obtain $\Gamma \vdash y \in \varphi \to \exists x.(y \in x) \wedge \lfloor C[x] \to \psi \rfloor$. Note that y is a fresh variable and $y \notin FV(C[x]) \cup FV(\psi)$, so $y \in \lfloor C[x] \to \psi \rfloor = \lfloor C[x] \to \psi \rfloor$. Notice that $y \in x = (y = x)$. And we obtain $\Gamma \vdash y \in \varphi \to \lfloor C[y] \to \psi \rfloor$. Done.

To show the "only if" part, we apply some simple FOL reasoning on $\Gamma \vdash x \in \varphi \to \lfloor C[x] \to \psi \rfloor$ and conclude that $\Gamma \vdash (\exists x.(x \wedge x \in \varphi)) \to \exists x.(x \wedge \lfloor C[x] \to \psi \rfloor)$. Then by the equality $\varphi = \exists x.(x \wedge x \in \varphi)$, we are done. ∎

Note the conditions about the context C in Lemma 91 are important. Many contexts that arise in practice satisfy the conditions. In particular, (nested) symbol contexts satisfy the conditions automatically.

Under the above new notation and the lemma, we are ready to prove Theorem 29.

Proof of Theorem 29: (Pre-Fixpoint). This is proved by simply unfolding α_σ following its definition.

(Knaster-Tarski). We give the following proof that goes backward from conclusion to their sufficient conditions.

$$
\begin{aligned}
& \sigma(x_1, \ldots, x_n) \to \psi \\
\impliedby\ & \alpha_\sigma[x_1, \ldots, x_n] \to \psi \\
\impliedby\ & \alpha \to \exists \alpha.(\alpha \wedge \lfloor \alpha[x_1, \ldots, x_n] \to \psi \rfloor) \\
\impliedby\ & \alpha_\sigma \to \forall \vec{x}.\underbrace{\exists \alpha.(\alpha \wedge \lfloor \alpha[x_1, \ldots, x_n] \to \psi \rfloor)}_{\alpha_0} \\
\impliedby\ & \exists \vec{x}.\langle \vec{x}, \varphi[\forall \vec{x}.\alpha_0/\sigma] \rangle \to \forall \vec{x}.\alpha_0 \\
\impliedby\ & \langle \vec{x}, \varphi[\forall \vec{x}.\alpha_0/\sigma] \rangle \to \alpha_0[z_1/x_1 \ldots z_n/x_n] \\
\impliedby\ & \langle \vec{x}, \varphi[\forall \vec{x}.\alpha_0/\sigma] \rangle \\
& \quad \to \exists \alpha.(\alpha \wedge \lfloor \alpha[z_1, \ldots, z_n] \to \psi[z_1/x_1 \ldots z_n/x_n] \rfloor) \\
\impliedby\ & \langle \vec{x}, \varphi[\forall \vec{x}.\alpha_0/\sigma] \rangle[x_1, \ldots, x_n] \to \psi \\
\impliedby\ & \varphi[\forall \vec{x}.\alpha_0/\sigma] \to \psi \\
\impliedby\ & \varphi[\forall \vec{x}.\alpha_0/\sigma] \to \varphi[\psi/\sigma]
\end{aligned}
$$

Notice that the last step is by $\Gamma \vdash \varphi[\psi/\sigma] \to \psi$.

By the positiveness of φ in σ (see Lemma 87), we just need to prove that for all $\varphi_1, \ldots, \varphi_n$:

$$\Gamma \vdash (\forall \vec{x}.\alpha_0)[\varphi_1, \ldots, \varphi_n] \to \psi[\varphi_1/x_1 \ldots \varphi_n/x_n]$$

By (Key-Value) and definition of α_0, the above becomes

$$\Gamma \vdash z_1 \in \varphi_1 \wedge \cdots \wedge z_n \in \varphi_n \wedge \psi[z_1/x_1 \ldots z_n/x_n] \to \psi[\varphi_1/x_1 \ldots \varphi_n/x_n],$$

which holds by assumption. Done. ∎

What is interesting in the above proof is that we used only (Key-Value) and did not use (Injectivity) and (Product Domain). The last two axioms are used in the proof of Theorem 30, where we need to establish an isomorphism between *models* of LFP and MmL. In there, the two axioms are needed to constrain MmL models.

Appendix J
Proof of Theorem 30

We first show that the theory of products (see Definition 27) capture precisely the product set $M_s \times M_t$. We denote the theory of products as Γ^{product}, consisting of the three axioms (Injectivity), (Key-Value), and (Product Domain).

Lemma 92. *For any signature* $\mathbb{\Sigma}$ *consisting two sorts* s, t *and their product sort* $s \otimes t$*, there exists an isomorphism*

$$M_{s \otimes t} \underset{j}{\overset{i}{\rightleftharpoons}} M_s \times M_t.$$

Under the above isomorphism, we adopt the following abbreviations for all $a \in M_s, b \in M_s, p \in M_s \times M_t$*:*

$$\langle a, b \rangle \equiv (\langle \_, \_ \rangle_{s,t})_M(a, b) \qquad p(v) \equiv (\_(\_)_{s,t})_M(p, v)$$

Then for all $f : M_s \to \mathcal{P}(M_t)$ *and* $\alpha \subseteq \mathcal{P}(M_s \times M_t)$*, we have*

$$f(a) = uncurry(f)(a) \qquad curry(\alpha)(a) = \alpha(a).$$

Proof: By (Product Domain), $M_{s \otimes t} = \bar{\rho}(\exists k \exists v.\langle k, v \rangle) = \cup_{a \in M_s, b \in M_t} \langle a, b \rangle$. Define the (i, j)-isomorphism such that $i(\langle a, b \rangle) = (a, b)$ and $j((a, b)) = \langle a, b \rangle$. Note that i is well-defined because of (Injectivity). Clearly, i, j form an isomorphism between $M_{s \times t}$ and $M_s \times M_t$.

Now we prove the two equations. They are straightforward. Note that $uncurry(f)(a) = \{(a, b) \mid b \in f(a)\}(a) = \{b \mid b \in f(a)\} = f(a)$. Similarly, $curry(\alpha)(a) = \{b \mid (a, b) \in \alpha\} = \alpha(a)$ by definition. Done. ∎

Corollary 93. *For any signature* $\mathbb{\Sigma}$ *containing sorts* $s_1, \ldots, s_n, t$ *and their product sorts* $s_1 \otimes \cdots \otimes s_n \otimes t$*, there exists an isomorphism between* $M_{s_1 \otimes \cdots \otimes s_n \otimes t}$ *and* $M_{s_1} \times \cdots \times M_{s_n} \times M_t$*. And for any function* $f : M_{s_1} \times \cdots \times M_{s_n} \to \mathcal{P}(M_t)$ *and sets* $\alpha \subseteq M_{s_1} \times \cdots \times M_{s_n} \times M_t$*, we have*

$$f(a_1, \ldots, a_n) = uncurry(f)(\alpha)$$
$$curry(\alpha)(a_1, \ldots, a_n) = \alpha(a_1, \ldots, a_n)$$

where we abbreviate $\alpha(a_1, \ldots, a_n) \equiv \alpha(a_1) \ldots (a_n)$ *is a composition of projections.*

We now review the syntax and semantics of LFP, slightly adapted to fit the best with our setting.

Definition 94. Let (S, Σ, Π) be a FOL signature. LFP extends FOL formulas by the following additional rule:

$$\varphi ::= \cdots \mid [\mathsf{lfp}_{R,\vec{x}}\varphi](t_1, \ldots, t_n)$$

where R is an n-ary predicate variable and φ is positive in R. LFP valuations also extend FOL that map every n-ary predicate variable R to and n-ary relation $\rho(R) \subseteq \mathcal{P}(M^n)$.[6] Given a FOL model M and a valuation ρ, LFP extends the semantics of FOL by adding the following valuation rule for least fixpoint formulas:

$$\begin{array}{l} M, \rho \vDash_{\mathsf{LFP}} [\mathsf{lfp}_{R,\vec{x}}\varphi](t_1, \ldots, t_n), \\ \text{if } (\rho(t_1), \ldots, \rho(t_n)) \in \\ \quad \bigcap \{\alpha \subseteq M_{s_1} \times \cdots \times M_{s_n} \mid \text{for all } a_i \in M_{s_i}, 1 \le i \le n, \\ \qquad M, \rho[\alpha/R, \vec{a}/\vec{x}] \vDash_{\mathsf{LFP}} \varphi \text{ implies } (a_1, \ldots, a_n) \in \alpha\} \end{array}$$

LFP formula φ is valid, denoted $\vDash_{\mathsf{LFP}} \varphi$, if $M, \rho \vDash_{\mathsf{LFP}} \varphi$ for all M and ρ.

Proof of Theorem 30: The proof is mainly based on the isomorphism between LFP models and MmL Γ^{LFP}-models. Notice that the (Function) axioms forces symbols in all Γ^{LFP}-models are functions. In addition, the axiom $\forall x{:}Pred\, \forall y{:}Pred.x = y$ forces the carrier set of $Pred$ must be a singleton set, say, $\{\star\}$.

(The "if" direction). We follow the same idea as we prove that ML captures faithfully FOL (see [1]), we construct from an LFP model $(\{M_s^{\mathsf{LFP}}\}_{s\in S}, \Sigma^{\mathsf{LFP}}, \Pi^{\mathsf{LFP}})$ a corresponding MmL Γ^{LFP} model, denoted $(\{M_s^{\mathsf{MmL}}\}_{s\in S} \cup \{M_{Pred}^{\mathsf{MmL}}\}, \Sigma^{\mathsf{MmL}})$ with $M_s^{\mathsf{MmL}} = M_s^{\mathsf{LFP}}$, $M_{Pred}^{\mathsf{MmL}} = \{\star\}$, and Σ^{MmL} defined as in Section II-D consisting of symbols that are all functions. An LFP valuation ρ^{LFP} derives a corresponding MmL valuation ρ^{MmL} such that $\rho^{\mathsf{MmL}}(x) = \rho^{\mathsf{LFP}}(x)$ for all LFP (element) variables x and $\rho^{\mathsf{MmL}}(R) = \rho^{\mathsf{LFP}}(R) \times \{\star\}$. Our goal is to prove that for all LFP formulas φ, we have $M^{\mathsf{LFP}}, \rho^{\mathsf{LFP}} \vDash_{\mathsf{LFP}} \varphi$ if and only if $\overline{\rho^{\mathsf{MmL}}}(\varphi) = \{\star\}$. Firstly, notice that as shown in [1], $\overline{\rho^{\mathsf{MmL}}}(t) = \{\rho^{\mathsf{LFP}}(t)\}$ for all terms t. Therefore, to simplify our notation we uniformly use $\rho(t)$ in both LFP and MmL settings. Carry out induction on the structure of φ. The only additional cases (compared with FOL) are $\varphi \equiv R(t_1, \ldots, t_n)$ and $\varphi \equiv [\mathsf{lfp}_{R,x_1,\ldots,x_n}\psi](t_1, \ldots, t_n)$. The first case is easy, as shown in the following reasoning: $M^{\mathsf{LFP}}, \rho^{\mathsf{LFP}} \vDash R(t_1, \ldots, t_n)$ iff $(\rho(t_1), \ldots, \rho(t_n)) \in \rho^{\mathsf{LFP}}(R)$ iff $(\rho(t_1), \ldots, \rho(t_n), \star) \in \overline{\rho^{\mathsf{MmL}}}(R)$ iff $\overline{\rho^{\mathsf{MmL}}}(R(t_1, \ldots, t_n)) = \{\star\}$. The second case when $\varphi \equiv [\mathsf{lfp}_{R,x_1,\ldots,x_n}\psi](t_1, \ldots, t_n)$ is shown as the following reasoning:

$$\begin{array}{l} M^{\mathsf{LFP}}, \rho^{\mathsf{LFP}} \vDash_{\mathsf{LFP}} [\mathsf{lfp}_{R,x_1,\ldots,x_n}\psi](t_1, \ldots, t_n) \\ \text{iff } (\rho(t_1), \ldots, \rho(t_n)) \in \\ \quad \bigcap \{\alpha \subseteq M_{s_1}^{\mathsf{LFP}} \times \cdots \times M_{s_n}^{\mathsf{LFP}} \mid \text{for all } a_i \in M_{s_i}^{\mathsf{LFP}}, 1 \le i \le n, \\ \qquad M^{\mathsf{LFP}}, \rho^{\mathsf{LFP}}[\alpha/R, \vec{a}/\vec{x}] \vDash_{\mathsf{LFP}} \psi \text{ implies } (a_1, \ldots, a_n) \in \alpha\} \\ \text{iff (by induction hypothesis)} \\ \quad (\rho(t_1), \ldots, \rho(t_n)) \in \\ \quad \bigcap \{\alpha \subseteq M_{s_1}^{\mathsf{MmL}} \times \cdots \times M_{s_n}^{\mathsf{MmL}} \mid \text{for all } a_i \in M_{s_i}^{\mathsf{MmL}}, 1 \le i \le n, \\ \qquad \overline{(\rho[\alpha/R, \vec{a}/\vec{x}])^{\mathsf{MmL}}}(\psi) = \{\star\} \text{ implies } (a_1, \ldots, a_n) \in \alpha\} \\ \text{iff (by definition of } (\rho[\alpha/R, \vec{a}/\vec{x}])^{\mathsf{MmL}}) \\ \quad (\rho(t_1), \ldots, \rho(t_n)) \in \\ \quad \bigcap \{\alpha^+ \subseteq M_{s_1}^{\mathsf{MmL}} \times \cdots \times M_{s_n}^{\mathsf{MmL}} \times \{\star\} \mid \\ \qquad \text{for all } a_i \in M_{s_i}^{\mathsf{MmL}}, 1 \le i \le n, \\ \qquad \overline{\rho^{\mathsf{MmL}}[\alpha^+/R, \vec{a}/\vec{x}]}(\psi) = \{\star\} \text{ implies } (a_1, \ldots, a_n, \star) \in \alpha^+\} \\ \text{iff (by reasoning about sets)} \\ \quad (\rho(t_1), \ldots, \rho(t_n)) \in \\ \quad \bigcap \{\alpha^+ \subseteq M_{s_1}^{\mathsf{MmL}} \times \cdots \times M_{s_n}^{\mathsf{MmL}} \times \{\star\} \mid \\ \qquad \bigcup_{a_i \in M_{s_i}^{\mathsf{MmL}}} (a_1, \ldots, a_n, \overline{\rho^{\mathsf{MmL}}[\alpha^+/R, \vec{a}/\vec{x}]}(\psi)) \subseteq \alpha^+\} \\ \text{iff (by MmL semantics)} \\ \quad (\rho(t_1), \ldots, \rho(t_n)) \in \\ \quad \overline{\rho^{\mathsf{MmL}}}((\mu R\colon s_1 \otimes \ldots \otimes s_n \otimes Pred.\exists x_1 \ldots \exists x_n.\langle x_1, \ldots, x_n, \psi\rangle)), \end{array}$$

and the last statement, by MmL semantics, is equivalent to $\overline{\rho^{\mathsf{MmL}}}([\mathsf{lfp}_{R,x_1,\ldots,x_n}\psi](t_1, \ldots, t_n))$, Done. And now we conclude that $\Gamma^{\mathsf{LFP}} \vDash \varphi$ then $\vDash_{\mathsf{LFP}} \varphi$. Otherwise, there exists an LFP model M^{LFP} and valuation ρ^{LFP} such that $M^{\mathsf{LFP}}, \rho^{\mathsf{LFP}} \nvDash_{\mathsf{LFP}} \varphi$, and this implies that in the Γ^{LFP}-model M^{MmL}, we have $\overline{\rho^{\mathsf{MmL}}}(\varphi) \neq \{\star\}$, meaning that $\Gamma^{\mathsf{LFP}} \nvDash \varphi$.

(The "only if" part). Notice the axiom $\forall x{:}Pred\, \forall y{:}Pred.x = y$ forces that $M_{Pred} = \{\star\}$ must be a singleton set, which ensures that the above translation from an LFP model M^{LFP} to an MmL model M^{MmL} can go *backward*. Specifically, for every MmL (function) symbol $f \in \Sigma^{\mathsf{MmL}}_{s_1\ldots s_n,s}$, we construct from its interpretation $f_{M^{\mathsf{MmL}}}\colon M_{s_1} \times \cdots \times M_{s_n} \to \mathcal{P}(M_s)$, the corresponding LFP function $f_{M^{\mathsf{LFP}}}\colon M_{s_1} \times \cdots \times M_{s_n} \to M_s$ such that $f_{M^{\mathsf{MmL}}}(a_1, \ldots, a_n) = \{f_{M^{\mathsf{LFP}}}(a_1, \ldots, a_n)\}$. Similarly, for every MmL (function) symbol $\pi \in \Sigma^{\mathsf{MmL}}_{s_1\ldots s_n,Pred}$, we construct from its interpretation $\pi_{M^{\mathsf{MmL}}}\colon M_{s_1} \times \cdots \times M_{s_n} \to \{\emptyset, \{\star\}\}$, the corresponding LFP predicate $\pi_{M^{\mathsf{LFP}}} \subseteq M_{s_1} \times \cdots \times M_{s_n}$, such that $\pi_{M^{\mathsf{LFP}}} \subseteq M_{s_1} \times \cdots \times M_{s_n} = \{(a_1, \ldots, a_n) \mid \pi_{M^{\mathsf{MmL}}}(a_1, \ldots, a_n) = \{\star\}\}$. Then we carry out the same reasoning as in the "if" part, and we are done. ∎

Appendix K
Proof of Theorem 31

Theorem 31 shows that our definition of modal μ-logic in MmL is faithful. We have shown a proof sketch in the main

[6] This is where we are different from the classic LFP. In classic LFP, formulas cannot contain predicate variables that occur free. And the semantics of predicate variables, which is needed when we define the semantics of $[\mathsf{lfp}_{R,x_1,\ldots,x_n}]$, are given by an extended model M' that takes R as an n-ary predicate symbol and interprets it as a relation $\alpha \subseteq M_{s_1} \times \cdots \times M_{s_n}$. Here, we allow predicate variables to occur free in a formula, and we extend valuations to give them semantics, instead of modifying the model. This slightly modified presentation is obviously the same as the classic one, but fits better in our setting and looks more similar and uniform to MmL.

paper. We give the complete detailed proof in this subsection. The main purpose is to give an example, as the proofs of the corresponding theorems for LTL/CTL/DL have similar forms.

Lemma 95. $\vdash_\mu \varphi$ *implies* $\Gamma^\mu \vdash \varphi$.

Proof: We need to prove that all modal μ-logic proof rules are provable in matching μ-logic. Recall that modal μ-logic contains all propositional tautologies and (Modus Ponens), plus the following four rules:

$$\text{(K)}\quad \circ(\varphi_1 \to \varphi_2) \to (\circ\varphi_1 \to \circ\varphi_2) \qquad \text{(N)}\quad \frac{\varphi}{\circ\varphi}$$

$$(\mu_1)\quad \varphi[(\mu X.\varphi)/X] \to \mu X.\varphi \qquad (\mu_2)\quad \frac{\varphi[\psi/X] \to \psi}{\mu X.\varphi \to \psi}$$

Notice that (K) and (N) are proved by Proposition 12, and (μ_1) and (μ_2) are exactly (Pre-Fixpoint) and (Knaster-Tarski). ∎

Lemma 96. *For all* $\mathbb{S} = (S, R)$ *and all valuations* V: PVar $\to \mathcal{P}(S)$*, we have* $s \in [\![\varphi]\!]^{\mathbb{S}}_V$ *if and only if* $s \in \bar{V}(\varphi)$.

Proof: Carry out structural induction on φ.

(Case $\varphi \equiv X$). We have $[\![X]\!]^{\mathbb{S}}_V = V(X) = \bar{V}(X)$. Proved.

(Case $\varphi \equiv \varphi_1 \wedge \varphi_2$). We have $[\![\varphi_1 \wedge \varphi_2]\!]^{\mathbb{S}}_V = [\![\varphi_1]\!]^{\mathbb{S}}_V \cap [\![\varphi_2]\!]^{\mathbb{S}}_V = \bar{V}(\varphi_1) \wedge \bar{V}(\varphi_2) = \bar{V}(\varphi_1 \wedge \varphi_2)$. Proved.

(Case $\varphi \equiv \neg\varphi_1$). We have $[\![\neg\varphi_1]\!]^{\mathbb{S}}_V = S \setminus [\![\varphi_1]\!]^{\mathbb{S}}_V = S \setminus \bar{V}(\varphi_1) = S \setminus (S \setminus \bar{V}(\neg\varphi_1)) = \bar{V}(\neg\varphi_1)$. Proved.

(Case $\varphi \equiv \circ\varphi_1$). By Proposition 32, we have $[\![\circ\varphi_1]\!]^{\mathbb{S}}_V = \{s \in S \mid s\ R\ t \text{ implies } t \in [\![\varphi_1]\!]^{\mathbb{S}}_V \text{ for all } t \in S\} = \{s \in S \mid s \in \bar{V}(\circ\varphi_1)\} = \bar{V}(\circ\varphi_1)$. Proved.

(Case $\varphi \equiv \mu X.\varphi_1$). We have $[\![\mu X.\varphi_1]\!]^{\mathbb{S}}_V = \bigcap\{A \subseteq S \mid [\![\varphi_1]\!]^{\mathbb{S}}_{V[A/X]} \subseteq A\} = \bar{V}(\mu X.\varphi_1)$. Proved.

Induction is finished and lemma is proved. ∎

Corollary 97. $\Gamma^\mu \vDash \varphi$ *implies* $\vDash_\mu \varphi$.

Proof: Assume the opposite. Then there exist $\mathbb{S} = (S, R)$, ρ: PVar $\to \mathcal{P}(S)$, and $s \in S$ such that $s \notin [\![\varphi]\!]^{\mathbb{S}}_V$. By Lemma 96, $s \notin \bar{V}(\varphi)$. Since $\mathbb{S} \vDash \Gamma^\mu$, we have $\Gamma^\mu \nvdash \varphi$. Contradiction. ∎

Now we have completed the proof of Theorem 31, where (2) $\Longrightarrow$ (3) is given by Lemma 95, and (5) $\Longrightarrow$ (6) is given by Corollary 97.

Appendix L
Proof of Proposition 32

Proof of Proposition 32: We simply apply definition. Recall that $s \in \bullet_{\mathbb{S}}(t)$ iff $s\ R\ t$.

(Case "$\bullet$"). $s \in \bar{\rho}(\bullet\varphi)$ iff there exists $t \in \bar{\rho}(\varphi)$ such that $s \in \bullet_{\mathbb{S}}(t)$ iff there exists t such that $s\ R\ t$ and $t \in \bar{\rho}(\varphi)$.

(Case "$\circ$"). $s \in \bar{\rho}(\circ\varphi)$ iff $s \in \bar{\rho}(\neg\bullet\neg\varphi)$ iff $s \notin \bar{\rho}(\bullet\neg\varphi)$ iff (use (Case "$\bullet$")) for all t, $t \in \bar{\rho}(\neg\varphi)$ implies $s \notin \bullet_{\mathbb{S}}(t)$ iff for all t, $s \in \bullet_{\mathbb{S}}(t)$ implies $t \in \bar{\rho}(\varphi)$ iff for all t, $s\ R\ t$ implies $t \in \bar{\rho}(\varphi)$.

(Case "$\Diamond$"). Note that $\bar{\rho}(\Diamond\varphi) = \bigcap\{A \subseteq S \mid \overline{\rho[A/X]}(\varphi \vee \bullet X) \subseteq A\} = \bigcap\{A \subseteq S \mid \bar{\rho}(\varphi) \cup \bullet_{\mathbb{S}}(A) \subseteq A\}$. On the other hand, $\{s \in S \mid \exists t \in S \text{ such that } t \in \bar{\rho}(\varphi) \text{ and } s\ R^*\ t\} = \{s \in S \mid \exists t \in S, \exists n \geq 0 \text{ such that } t \in \bar{\rho}(\varphi) \text{ and } s\ R^n\ t\} = \{s \in S \mid \exists n \geq 0 \text{ such that } s \in \bullet^n_{\mathbb{S}}(\bar{\rho}(\varphi))\} = \bigcup_{n\geq 0} \bullet^n_{\mathbb{S}}(\bar{\rho}(\varphi))$. Therefore, we just need to prove the two sets:

$$(\eta) \equiv \bigcap\{A \subseteq S \mid \bar{\rho}(\varphi) \cup \bullet_{\mathbb{S}}(A) \subseteq A\}$$
$$(\xi) \equiv \bigcup_{n\geq 0} \bullet^n_{\mathbb{S}}(\bar{\rho}(\varphi))$$

are equal. This can be directly proved by Knaster-Tarski theorem.

(Case "$\Box$"). Similar to (Case "$\Diamond$").

(Case "$\varphi_1\ U\ \varphi_2$"). As in (Case "$\Diamond$"), we define two sets:

$$(\eta) \equiv \bar{\rho}(\varphi_1\ U\ \varphi_2) = \bigcap\{A \subseteq S \mid \bar{\rho}(\varphi_2) \cup (\bar{\rho}(\varphi_1) \cap \bullet_{\mathbb{S}}(A))) \subseteq A\}$$
$$(\xi) \equiv \{s \in S \mid \text{exist } n \geq 0 \text{ and } t_1, \ldots, t_n \in S \text{ such that } s\ R\ t_1\ R \ldots R\ t_n, \text{ and } s, t_1, \ldots, t_{n-1} \in \bar{\rho}(\varphi_1), t_n \in \bar{\rho}(\varphi_2)\}$$

and then use Knaster-Tarski theorem to prove them equal.

(Case "WF"). Again, we define two sets:

$$(\eta) \equiv \bar{\rho}(\mu X.\circ X) = \bigcap\{A \subseteq S \mid (S \setminus A) \subseteq \bullet_{\mathbb{S}}(S \setminus A)\}$$
$$(\xi) \equiv \{s \in S \mid s \text{ has no infintie path}\}$$

and then use Knaster-Tarski theorem to prove them equal. ∎

Appendix M
Proof of Theorem 33

As a review, we formally define the semantics of infinite-trace LTL and present in Fig. 4 its sound and complete proof system. There are different notions of semantics of infinite-trace LTL. We here review the one that fits best in our setting.

Let us first formally define some characteristic subclasses of transition systems.

Definition 98. A transition system $\mathbb{S} = (S, R)$ is:

- *well-founded* if for all $s \in S$, there is no infinite path from s;
- *non-terminating*, if for all $s \in S$ there is $t \in S$ such that $s\ R\ t$.
- *linear*, if for all $s \in S$ and $t_1, t_2 \in S$ such that $s\ R\ t_1$ and $s\ R\ t_2$, then $t_1 = t_2$.

Definition 99. Infinite-trace LTL formulas φ is interpreted over a transition system $\mathbb{S} = (S, R)$ that is *non-terminating* and *linear*. We use s_k to denote the unique state such that $sRs_1Rs_2R \ldots Rs_k$, for $k \geq 0$. When $k = 0$, we let $s_0 = s$. Given a valuation V: PVar $\to \mathcal{P}(S)$, semantics of infinite-trace LTL is inductively defined for all $s \in S$ and φ as follows:

- $s \vDash_{\mathsf{infLTL}} X$ if $s \in V(X)$;
- $s \vDash_{\mathsf{infLTL}} \varphi_1 \wedge \varphi_2$ if $s \vDash_{\mathsf{infLTL}} \varphi_1$ and $s \vDash_{\mathsf{infLTL}} \varphi_2$;
- $s \vDash_{\mathsf{infLTL}} \neg\varphi$ if $s \nvDash_{\mathsf{infLTL}} \varphi$;
- $s \vDash_{\mathsf{infLTL}} \circ\varphi$ if $s_1 \vDash_{\mathsf{infLTL}} \varphi$;
- $s \vDash_{\mathsf{infLTL}} \varphi_1\ U\ \varphi_2$ if exists $k \geq 0$ such that $s_k \vDash_{\mathsf{infLTL}} \varphi_2$ and for all $0 \leq i < k$, $s_i \vDash_{\mathsf{infLTL}} \varphi_1$.

Lemma 100. $\vdash_{\mathsf{infLTL}} \varphi$ *implies* $\Gamma^{\mathsf{infLTL}} \vdash \varphi$.

Proof: We just need to prove that all proof rules in Fig. 4 can be proved in Γ^{infLTL}.

(Taut) and (MP). Trivial.

(Taut)	φ, if φ is a propositional tautology
(MP)	$\dfrac{\varphi_1 \quad \varphi_1 \to \varphi_2}{\varphi_2}$
($K_\circ$)	$\circ(\varphi_1 \to \varphi_2) \to (\circ\varphi_1 \to \circ\varphi_2)$
($N_\circ$)	$\dfrac{\varphi}{\circ\varphi}$
($K_\Box$)	$\Box(\varphi_1 \to \varphi_2) \to (\Box\varphi_1 \to \Box\varphi_2)$
($N_\Box$)	$\dfrac{\varphi}{\Box\varphi}$
(Fun)	$\circ\varphi \leftrightarrow \neg(\circ\neg\varphi)$
(U_1)	$(\varphi_1 \mathbin{U} \varphi_2) \to \Diamond\varphi_2$
(U_2)	$(\varphi_1 \mathbin{U} \varphi_2) \leftrightarrow (\varphi_2 \vee (\varphi_1 \wedge \circ(\varphi_1 \mathbin{U} \varphi_2)))$
(Ind)	$\Box(\varphi \to \circ\varphi) \to (\varphi \to \Box\varphi)$

Fig. 4. Infinite-trace LTL proof system

(Taut)	φ, if φ is a propositional tautology
(MP)	$\dfrac{\varphi_1 \quad \varphi_1 \to \varphi_2}{\varphi_2}$
($K_\circ$)	$\circ(\varphi_1 \to \varphi_2) \to (\circ\varphi_1 \to \circ\varphi_2)$
($N_\circ$)	$\dfrac{\varphi}{\circ\varphi}$
($K_\Box$)	$\Box(\varphi_1 \to \varphi_2) \to (\Box\varphi_1 \to \Box\varphi_2)$
($N_\Box$)	$\dfrac{\varphi}{\Box\varphi}$
($\neg\circ$)	$\neg\circ\varphi \to \circ\neg\varphi$
(coInd)	$\dfrac{\circ\varphi \to \varphi}{\varphi}$
(Fix)	$(\varphi_1 \mathbin{U_w} \varphi_2) \leftrightarrow (\varphi_2 \vee (\varphi_1 \wedge \circ(\varphi_1 \mathbin{U_w} \varphi_2)))$

Fig. 5. Finite-trace LTL proof system

($K_\circ$) and ($N_\circ$). By Proposition 12.

($K_\Box$) and ($N_\Box$). Proved by applying (Knaster-Tarski) first, followed by simple propositional and modal logic reasoning.

(Fun, "$\to$"). Proved from axiom (Inf) $\bullet\top$ and simple modal logic reasoning.

(Fun, "$\leftarrow$"). Exactly axiom (Lin).

(U_1). By (Knaster-Tarski) followed by propositional reasoning.

(U_2). By definition of $\varphi_1 \mathbin{U} \varphi_2$ as a least fixpoint and (Fun).

(Ind). By (Knaster-Tarski). ∎

Lemma 101. *$s \vDash_{\mathsf{infLTL}} \varphi$ if and only if $s \in \bar{V}(\varphi)$.*

Proof: We make two interesting observations. Firstly, it suffices to prove merely the "only if" part. The "if" part follows by considering the "only if" part on $\neg\varphi$.

Secondly, the definition of "$s \vDash_{\mathsf{infLTL}} \varphi$" is an *inductive* one, meaning that "$\vDash_{\mathsf{infLTL}}$" is the least relation that satisfies the five conditions in Definition 99. To show that "$s \vDash_{\mathsf{infLTL}} \varphi$ implies $s \in \bar{V}(\varphi)$", it suffices to show that $s \in \bar{V}(\varphi)$ satisfies the same conditions. This is easily followed by Proposition 32.

Note how interesting that this lemma is proved by applying Knaster-Tarski theorem in the meta-level. ∎

Corollary 102. *$\Gamma^{\mathsf{infLTL}} \vDash \varphi$ implies $\vDash_{\mathsf{infLTL}} \varphi$.*

Proof: Assume the opposite and there exists a transition system $\mathbb{S} = (S, R)$ that is linear and non-terminating, a valuation V, and a state $s \in S$ such that $s \nvDash_{\mathsf{infLTL}} \varphi$. By Lemma 101, $s \notin \bar{V}(\varphi)$, meaning that $\mathbb{S} \nvDash \varphi$. Since $\mathbb{S}$ is non-terminating and linear, the axioms (Inf) and (Lin) hold in $\mathbb{S}$, and thus $\Gamma^{\mathsf{infLTL}} \nvDash \varphi$. Contradiction. ∎

Now we are ready to prove Theorem 33.

Proof of Theorem 33: Use Lemma 100 and Corollary 102, as well as the soundness of MmL proof system and the completeness of infinite-trace LTL proof system. ∎

Appendix N
Proof of Theorem 34

We review the semantics of finite-trace LTL as well as its sound and complete proof system presented in Fig. 5.

The following definition is adapted from [10] to fit best in our setting.

Definition 103. Finite-trace LTL formulas φ is interpreted over a transition system $\mathbb{S} = (S, R)$ that is *well-founded* and *linear*. One can show that $S = \{s_1, \ldots, s_n\}$ must be finite, and the transition relation of $\mathbb{S}$ must be of the linear structure $s_1 \mathbin{R} \ldots \mathbin{R} s_n$. Given a valuation $V\colon \mathrm{PVar} \to \mathcal{P}(S)$, semantics of infinite-trace LTL is inductively defined for all $s_i \in S$ and φ as follows:

- $s_i \vDash_{\mathsf{finLTL}} X$ if $s_i \in V(X)$;
- $s_i \vDash_{\mathsf{finLTL}} \varphi_1 \wedge \varphi_2$ if $s_i \vDash_{\mathsf{finLTL}} \varphi_1$ and $s_i \vDash_{\mathsf{finLTL}} \varphi_2$;
- $s_i \vDash_{\mathsf{finLTL}} \neg\varphi$ if $s_i \nvDash_{\mathsf{finLTL}} \varphi$;
- $s_i \vDash_{\mathsf{finLTL}} \circ\varphi$ if $s_i = s_n$ or $s_{i+1} \vDash_{\mathsf{finLTL}} \varphi$;
- $s_i \vDash_{\mathsf{finLTL}} \varphi_1 \mathbin{U_w} \varphi_2$ if either $s_j \vDash_{\mathsf{finLTL}} \varphi_1$ for all $j \geq i$, or there exists $i \leq k \leq n$ such that $s_k \vDash_{\mathsf{finLTL}} \varphi_2$ and for all $i \leq j < k$, $s_j \vDash_{\mathsf{finLTL}} \varphi_1$.

Lemma 104. *$\vDash_{\mathsf{finLTL}} \varphi$ implies $\Gamma^{\mathsf{finLTL}} \vdash \varphi$.*

Proof: We just need to prove all proof rules in Fig. 5 can be proved by axioms (Fin) and (Lin) in MmL. We skip the ones that have shown in Lemma 100.

($\neg\circ$). Proved by axiom (Lin).

(coInd). Use axiom (Fin) $\mu X.\circ X$ and to prove $\Gamma^{\mathsf{finLTL}} \vdash \mu X.\circ X \to \varphi$ by (Knaster-Tarski).

(Fix). By definition of $\varphi_1 \mathbin{U_w} \varphi_2$ as a least fixpoint. ∎

Lemma 105. *$s \vDash_{\mathsf{finLTL}} \varphi$ if and only if $s \in \bar{V}(\varphi)$.*

Proof: As in Lemma 101, we just need to prove the "only if" part, by showing that $s \in \bar{V}(\varphi)$ satisfies the five conditions in Definition 103. This is easily followed by Proposition 32. The case $\varphi_1 \mathbin{U_w} \varphi_2$ shall be proved by directly applying MmL semantics. ∎

Corollary 106. *$\Gamma^{\mathsf{finLTL}} \vDash \varphi$ implies $\vDash^{\mathsf{finLTL}} \varphi$.*

Proof: Assume the opposite and use Lemma 105. ∎

Now we can prove Theorem 34.

Proof of Theorem 34: Use Lemma 104 and Corollary 106, as well as the soundness of MmL proof system and the completeness of finite-trace LTL proof system. ∎

(Taut)	φ, if φ is a propositional tautology
(MP)	$\dfrac{\varphi_1 \quad \varphi_1 \to \varphi_2}{\varphi_2}$
(CTL_1)	$\mathsf{EX}(\varphi_1 \vee \varphi_2) \leftrightarrow \mathsf{EX}\varphi_1 \vee \mathsf{EX}\varphi_2$
(CTL_2)	$\mathsf{AX}\varphi \leftrightarrow \neg(\mathsf{EX}\neg\varphi)$
(CTL_3)	$\varphi_1 \,\mathsf{EU}\, \varphi_2 \leftrightarrow \varphi_2 \vee (\varphi_1 \wedge \mathsf{EX}(\varphi_1 \,\mathsf{EU}\, \varphi_2))$
(CTL_4)	$\varphi_1 \,\mathsf{AU}\, \varphi_2 \leftrightarrow \varphi_2 \vee (\varphi_1 \wedge \mathsf{AX}(\varphi_1 \,\mathsf{AU}\, \varphi_2))$
(CTL_5)	$\mathsf{EX}\mathit{true} \wedge \mathsf{AX}\mathit{true}$
(CTL_6)	$\mathsf{AG}(\varphi_3 \to (\neg\varphi_2 \wedge \mathsf{EX}\varphi_3)) \to (\varphi_3 \to \neg(\varphi_1 \,\mathsf{AU}\, \varphi_2))$
(CTL_7)	$\mathsf{AG}(\varphi_3 \to (\neg\varphi_2 \wedge (\varphi_1 \to \mathsf{AX}\varphi_3))) \to (\varphi_3 \to \neg(\varphi_1 \,\mathsf{EU}\, \varphi_2))$
(CTL_8)	$\mathsf{AG}(\varphi_1 \to \varphi_2) \to (\mathsf{EX}\varphi_1 \to \mathsf{EX}\varphi_2)$

Fig. 6. CTL proof system

(Taut)	φ, if φ is a propositional tautology
(MP)	$\dfrac{\varphi_1 \quad \varphi_1 \to \varphi_2}{\varphi_2}$
(DL_1)	$[\alpha](\varphi_1 \to \varphi_2) \to ([\alpha]\varphi_1 \to [\alpha]\varphi_2)$
(DL_2)	$[\alpha](\varphi_1 \wedge \varphi_2) \leftrightarrow ([\alpha]\varphi_1 \wedge [\alpha]\varphi_2)$
(DL_3)	$[\alpha \cup \beta]\varphi \leftrightarrow [\alpha]\varphi \wedge [\beta]\varphi$
(DL_4)	$[\alpha\,;\beta]\varphi \leftrightarrow [\alpha][\beta]\varphi$
(DL_5)	$[\psi?]\varphi \leftrightarrow (\psi \to \varphi)$
(DL_6)	$\varphi \wedge [\alpha][\alpha^*]\varphi \leftrightarrow [\alpha^*]\varphi$
(DL_7)	$\varphi \wedge [\alpha^*](\varphi \to [\alpha]\varphi) \to [\alpha^*]\varphi$
(Gen)	$\dfrac{\varphi}{[\alpha]\varphi}$

Fig. 7. Dynamic logic proof system

Appendix O
Proof of Theorem 35

We review the semantics of CTL as well as its sound and complete proof system presented in Fig. 6.

Definition 107. CTL formulas are interpreted on a transition system $\mathbb{S} = (S, R)$ that is non-terminating, and a valuation $V\colon \mathrm{PVar} \to \mathcal{P}(S)$. We call an (infinite) sequence of states $s_0 s_1 \ldots$ a *path* if $s_i \, R \, s_{i+1}$ for all $i \geq 0$. CTL semantics is defined inductively for all $s_0 \in S$ and φ as follows:

- $s_0 \vDash_{\mathsf{CTL}} X$ if $s_0 \in V(X)$;
- $s_0 \vDash_{\mathsf{CTL}} \varphi_1 \wedge \varphi_2$ if $s_0 \vDash_{\mathsf{CTL}} \varphi_1$ and $s_0 \vDash_{\mathsf{CTL}} \varphi_2$;
- $s_0 \vDash_{\mathsf{CTL}} \neg\varphi$ if $s_0 \nvDash_{\mathsf{CTL}} \varphi$;
- $s_0 \vDash_{\mathsf{CTL}} \mathsf{EX}\varphi$ if there exists s_1 such that $s_0 \, R \, s_1$, $s_1 \vDash_{\mathsf{CTL}} \varphi$;
- $s_0 \vDash_{\mathsf{CTL}} \mathsf{AX}\varphi$ if for all s_1 such that $s_0 \, R \, s_1$, $s_1 \vDash_{\mathsf{CTL}} \varphi$;
- $s_0 \vDash_{\mathsf{CTL}} \varphi_1 \,\mathsf{EU}\, \varphi_2$ if there exists a path $s_0 s_1 \ldots$ and $k \geq 0$ such that $s_k \vDash_{\mathsf{CTL}} \varphi_2$, and $s_0, \ldots, s_{k-1} \vDash_{\mathsf{CTL}} \varphi_1$;
- $s_0 \vDash_{\mathsf{CTL}} \varphi_1 \,\mathsf{AU}\, \varphi_2$ if for all paths $s_0 s_1 \ldots$ there exists $k \geq 0$ such that $s_k \vDash_{\mathsf{CTL}} \varphi_2$, and $s_0, \ldots, s_{k-1} \vDash_{\mathsf{CTL}} \varphi_1$;.

We write $\vDash_{\mathsf{CTL}} \varphi$ if for all $\mathbb{S} = (S, R)$, all valuations ρ, and all $s \in S$, $s \vDash_{\mathsf{CTL}} \varphi$.

Lemma 108. *$\vdash_{CTL} \varphi$ implies $\Gamma^{CTL} \vdash \varphi$.*

Proof: We just need to prove all CTL rules from the axiom (Inf) in MmL. We skip the first 7 rules as they are simple. The rest 3 rules can be proved by applying (Knaster-Tarski) and use properties in Properties 115. ∎

Lemma 109. *$s \vDash_{CTL} \varphi$ if and only if $s \in \bar{V}(\varphi)$.*

Proof: As in Lemma 101, we just need to prove the "only if" part by showing that $s \in \bar{V}(\varphi)$ satisfies all 7 conditions in Definition 107. The first 5 of them are simple. We show the last two ones about "EU" and "AU".

(Case EU). Assume there exists a path $s_0 s_1 \ldots$ and $k \geq 0$ such that $s_k \in \bar{V}(\varphi_2)$ and $s_0, \ldots, s_{k-1} \in \bar{V}(\varphi_1)$. Our goal is to show $s_0 \in \bar{V}(\varphi_1 \,\mathsf{EU}\, \varphi_2)$. By semantics of MmL, $\bar{V}(\varphi_1 \,\mathsf{EU}\, \varphi_2) = \bar{V}(\mu X.\varphi_2 \vee (\varphi_1 \wedge \bullet X)) = \bigcap\{A \subseteq S \mid \bar{V}(\varphi_2) \cup (\bar{V}(\varphi_1) \cap \bullet_{\mathbb{S}}(A)) \subseteq A\}$. Therefore, it suffices to prove that $s_0 \in A$ for all $A \subseteq S$ such that $\bar{V}(\varphi_2) \subseteq A$ and $\bar{V}(\varphi_1) \cap \bullet_{\mathbb{S}}(A) \subseteq A$. This is easy, $s_k \in \bar{V}(\varphi_2) \subseteq A$ implies $s_{k-1} \in \bullet_{\mathbb{S}}(s_k)$. Also, $s_{k-1} \in \bar{V}(\varphi_1)$ by assumption. Then $s_{k-1} \in \bar{V}(\varphi_1) \cap \bullet_{\mathbb{S}}(s_k) \subseteq A$. Repeat this procedure for k times and we obtain $s_0 \in A$. Done.

(Case AU). Let us denote $\circ_{\mathbb{S}}(A) = \{s \in S \mid \text{for all } t \in S \text{ such that } s \, R \, t, t \in A\}$ to be the "interpretation" of "all-path next $\circ$" in $\mathbb{S}$. Prove by contradiction. Assume the opposite statement that $s_0 \notin \bar{V}(\varphi_1 \,\mathsf{AU}\, \varphi_2) = \bar{V}(\mu X.\varphi_2 \vee (\varphi_1 \wedge \circ X)) = \bigcap\{A \subseteq S \mid \bar{V}(\varphi_2) \cup (\bar{V}(\varphi_1) \cap \circ_{\mathbb{S}}(A)) \subseteq A\}$. This means that there exists $A \subseteq S$ such that $\bar{V}(\varphi_2) \subseteq A$ and $\bar{V}(\varphi_1) \cap \circ_{\mathbb{S}}(A) \subseteq A$, and $s_0 \notin A$. This is going to cause contradiction. Firstly by $\bar{V}(\varphi_2) \subseteq A$, $s_0 \notin \bar{V}(\varphi_2)$, which implies that $s_0 \in \bar{V}(\neg\varphi_2)$. Secondly by $\bar{V}(\varphi_1) \cap \circ_{\mathbb{S}}(A) \subseteq A$, we know that $(S \setminus A) \subseteq \bar{V}(\neg\varphi_1) \cup \bullet_{\mathbb{S}}(S \setminus A)$. Since $s_0 \notin A$, we know either $s_0 \in \bar{V}(\neg\varphi_1)$ or $s_0 \in \bullet_{\mathbb{S}}(S \setminus A)$. If it is the first case, then we have a contradiction as any path starting from s_0 contradicts with the condition. If it is the second case, then there exists a state, say s_1, such that $s_0 \, R \, s_1$ and $s_1 \notin A$, which also implies $s_1 \notin \bar{V}(\varphi_2)$. Repeat this process and obtain a sequence of state $s_0 s_1 \ldots$. If the sequence is finite, say $s_0 s_1 \ldots s_n$, then by construction $s_0, \ldots, s_n \notin \bar{V}(\varphi_2)$ and $s_n \in \bar{V}(\neg\varphi_1)$, which is a contradiction to the condition. If the sequence is infinite, then by construction $s_0 s_1 \ldots$ satisfies that $s_0, s_1, \ldots \notin \bar{V}(\varphi_2)$, which also contradicts to the condition. Done. ∎

Corollary 110. *$\Gamma^{CTL} \vDash \varphi$ implies $\vDash_{CTL} \varphi$.*

Proof: Use Lemma 109 and prove by contradiction. Note that it is easy to verify that $\mathbb{S} \vDash \Gamma^{\mathsf{CTL}}$ if $\mathbb{S}$ is non-terminating. ∎

Now we are ready to prove Theorem 35.

Proof of Theorem 35: Use Lemma 108 and Corollary 110, as well as soundness of MmL and completeness of CTL. ∎

Appendix P
Proof of Theorem 36

We review the semantics of DL as well as its sound and complete proof system presented in Fig. 7.

Definition 111. Let $\mathbb{S} = (S, \{R_a\}_{a \in \mathrm{APgm}})$ be an APgm-labeled transition system where $R_a \in S \times S$ is the transition relation for atomic program a. Let $V\colon \mathrm{PVar} \to \mathcal{P}(S)$ be a valuation. DL semantics is inductively defined as follows where state

formulas are evaluated to subsets of S and program formulas are evaluated to relations of S:

- $[\![p]\!]^{\mathbb{S}}_V = V(p)$;
- $[\![\varphi_1 \wedge \varphi_2]\!]^{\mathbb{S}}_V = [\![\varphi_1]\!]^{\mathbb{S}}_V \cap [\![\varphi_2]\!]^{\mathbb{S}}_V$;
- $[\![\neg\varphi]\!]^{\mathbb{S}}_V = S \setminus [\![\varphi]\!]^{\mathbb{S}}_V$;
- $[\![[\alpha]\varphi]\!]^{\mathbb{S}}_V = \{s \in S \mid \text{for all } t \in S \text{ such that } (s,t) \in [\![\alpha]\!]^{\mathbb{S}}_V, \text{we have } t \in [\![\varphi]\!]^{\mathbb{S}}_V\}$;
- $[\![a]\!] = R_a$ for $a \in \text{APGM}$;
- $[\![\alpha_1 ; \alpha_2]\!]^{\mathbb{S}}_V = [\![\alpha_1]\!]^{\mathbb{S}}_V \circ [\![\alpha_2]\!]^{\mathbb{S}}_V$;
- $[\![\alpha_1 \cup \alpha_2]\!]^{\mathbb{S}}_V = [\![\alpha_1]\!]^{\mathbb{S}}_V \cup [\![\varphi_2]\!]^{\mathbb{S}}_V$;
- $[\![\alpha^*]\!]^{\mathbb{S}}_V = ([\![\alpha]\!]^{\mathbb{S}}_V)^*$;
- $[\![\varphi?]\!]^{\mathbb{S}}_V = \{(s,s) \mid s \in [\![\varphi]\!]^{\mathbb{S}}_V\}$.

where "$R_1 \circ R_2$" is the *composition* of two relations R_1, R_2 defined as $R_1 \circ R_2 = \{(s_1, s_3) \mid$ there exists s_2 such that $(s_1, s_2) \in R_1$ and $(s_2, s_3) \in R_2\}$. We write $\vDash_{\mathsf{DL}} \varphi$ if $[\![\varphi]\!]^{\mathbb{S}}_V = S$ for all $\mathbb{S}$ and V.

Lemma 112. *$\vdash_{DL} \varphi$ implies $\Gamma^{DL} \vdash \varphi$.*

Proof: We just need to prove that all proof rules in Fig. 7 can be proved in Γ^{DL}. First of all, rules (DL_3) to (DL_6) follow from (syntactic sugar) definitions. Rules (Taut) and (MP) are trivial, We only prove (DL_1), (DL_2), (DL_7), and (Gen).

Notice that $[\alpha]\varphi$ is defined a syntactic sugar based on the structure of α. Therefore, we carry out structure induction on α. We should be careful to prevent circular reasoning. Our proving strategy is to prove (Gen) first, and then prove (DL_1) and (DL_2) simultaneously, and finally prove (DL_7).

(Gen). Carry out induction on α. All cases are trivial. Notice the case when $\alpha \equiv \beta^*$ is proved by proving $\Gamma^{\mathsf{DL}} \vdash \varphi \to [\alpha^*]\varphi$ using (Knaster-Tarski). After simplification, the goal becomes $\Gamma^{\mathsf{DL}} \vdash \varphi \to [\beta]\varphi$. This is proved by applying induction hypothesis, which shows $\Gamma^{\mathsf{DL}} \vdash [\beta]\varphi$.

(DL_1) and (DL_2). We prove both rules simultaneously by induction on α. We discuss only interesting cases and skip the trivial ones. (DL_1, $\alpha \equiv \beta_1 ; \beta_2$) is proved from induction hypothesis, by applying (Gen) on $[\beta_1]$. ($\text{DL}_1, \alpha \equiv \beta^*$) is proved by applying (Knaster-Tarski), following by applying (DL_2, "$\to$") on $[\beta]$. (DL_2, $\alpha \equiv \beta^*$, "$\to$") is proved by (Knaster-Tarski). (DL_2, $\alpha \equiv \beta^*$, "$\leftarrow$") is proved by (Knaster-Tarski), followed by (DL_2) on $[\beta]$.

(DL_7) is proved directly by (Knaster-Tarski), followed by (DL_2, "$\leftarrow$") on $[\alpha]$.

■

We now connect the semantics of DL with the semantics of MmL. First of all, we show that the transition system $\mathbb{S} = (S, \{R_a\}_{a \in \text{APGM}})$ can be regarded as a $\mathbb{\Sigma}^{\mathsf{LTS}}$-model, where S is the carrier set of *State* and APGM (the set of atomic programs) is the carrier set of *Pgm*. The "one-path next $\bullet \in \Sigma_{Pgm\,State,State}$ is interpreted *according to DL semantics* for all $t \in S$ and $a \in \text{APGM}$:

$$\bullet_{\mathbb{S}}(a,t) = \{s \in S \mid (s,t) \in R_a\}.$$

In addition, valuation V: PVar $\to \mathcal{P}(S)$ can be regarded as a matching μ-logic valuation (where we safely ignore the valuations of element variables because they do not appear in DL syntax).

Lemma 113. *Under the above notations, $[\![\varphi]\!]^{\mathbb{S}}_V = \bar{V}(\varphi)$.*

Proof: As in Lemma 101, we just need to prove that $[\![\varphi]\!]^{\mathbb{S}}_V \subseteq \bar{V}(\varphi)$ by showing that $\bar{V}(\varphi)$ satisfies the conditions in Definition 111. The only interesting case is to show $\bar{V}([\alpha]\varphi) = \{s \in S \mid \text{for all } t \in S, (s,t) \in [\![\alpha]\!]^{\mathbb{S}}_V \text{ implies } t \in \bar{V}(\varphi)\}$. We prove it by carrying out structural induction on the DL program formula α. The case when $\alpha \equiv a$ for $a \in \text{APGM}$ is easy. The cases when $\alpha \equiv \beta_1 ; \beta_2$, $\alpha \equiv \beta_1 \cup \beta_2$, and $\alpha \equiv \psi?$ follows directly by basic analysis about sets and using definition of the semantics of DL program formulas. The interesting case is when $\alpha \equiv \beta^*$. In this case we should prove $\bar{V}([\beta^*]\varphi) = \bar{V}(\nu X.\varphi \wedge [\beta]X) = \bigcup\{A \mid A \subseteq \bar{V}(\varphi) \cap \overline{V[A/X]}([\beta]X)\} = \bigcup\{A \mid A \subseteq \bar{V}(\varphi) \cap \{s \mid \text{for all } t, (s,t) \in [\![\beta]\!]^{\mathbb{S}}_V \text{ implies } t \in S\}\} \stackrel{?}{=} \{s \mid \text{for all } t, (s,t) \in [\![\beta^*]\!]^{\mathbb{S}}_V \text{ implies } t \in \bar{V}(\varphi)\}$ We denote the left-hand side of "$\stackrel{?}{=}$" as (η) and the right-hand side as (ξ).

To prove $(\eta) = (\xi)$, we prove containment from both directions.

(Case $(\eta) \subseteq (\xi)$). This is proved by considering an $s \in (\eta)$ and show $s \in (\xi)$. By construction of (η), there exists $A \subseteq S$ such that $A \subseteq \bar{V}(\varphi) \cap \{s \mid \text{for all } t, (s,t) \in [\![\beta]\!]^{\mathbb{S}}_V \text{ implies } t \in A\}$, and that $s \in A$. In order to prove $s \in (\xi)$, we assume $t \in S$ such that $(s,t) \in ([\![\beta]\!]^{\mathbb{S}}_V)^*$ and try to prove $t \in \bar{V}(\varphi)$. By definition, there exists $k \geq 0$ and $s_0, \ldots, s_k$ such that $s = s_0$, $t = s_k$, and $(s_i, s_{i+1}) \in [\![\beta]\!]^{\mathbb{S}}_V$ for all $0 \leq i < k$. By induction and the property of A, and that $s_0 \in A$, we can prove that $s_0, s_1, \ldots, s_k \in \bar{V}(\varphi)$, and thus $t \in \bar{V}(\varphi)$. Done.

(Case $(\xi) \subseteq (\eta)$). Notice that the set η is defined as a greatest fixpoint, so it suffices to show that (ξ) satisfies the condition, i.e., to prove that $(\xi) \subseteq \bar{V}(\varphi) \cap \{s \mid \text{for all } t, (s,t) \in [\![\beta]\!]^{\mathbb{S}}_V \text{ implies } t \in (\xi)\}$. This can be easily proved by the definition of (ξ). Done. ■

Corollary 114. *$\Gamma^{\mathsf{DL}} \vDash \varphi$ implies $\vDash_{\mathsf{DL}} \varphi$.*

Proof: Use Lemma 113, and for the sake of contradiction, assume the opposite. Suppose there exists $\mathbb{S} = (S, \{R_a\}_{a \in \text{APGM}})$ and a valuation V and a state s such that $s \notin [\![\varphi]\!]^{\mathbb{S}}_V$. We then know $s \notin \bar{V}(\varphi)$, which implies that $\mathbb{S} \nvDash \varphi$. Obviously $\mathbb{S} \vDash \Gamma^{\mathsf{DL}}$ as the theory Γ^{DL} contains no addition axioms. This means that $\Gamma^{\mathsf{DL}} \nvDash \varphi$. ■

We are ready to prove Theorem 36.

Proof of Theorem 36: Use Lemma 112 and Corollary 114, as well as soundness of MmL and completeness of DL. ■

Appendix Q
Proof of Theorem 39

As a review, we use the following notations:

"one-path next"	$\bullet\varphi$, where $\bullet \in \Sigma_{Cfg,Cfg}$
"all-path next"	$\circ\varphi \equiv \neg\bullet\neg\varphi$
"eventually"	$\Diamond\varphi \equiv \mu X.\varphi \vee \bullet X$
"always"	$\Box\varphi \equiv \nu X.\varphi \wedge \circ X$
"well-founded"	$\mathsf{WF} \equiv \mu X.\circ X$
"weak eventually"	$\Diamond_w\varphi \equiv \nu X.\varphi \vee \bullet X$

Proposition 115. *The following propositions hold:*

1) $\vdash \bullet\bot \leftrightarrow \bot$
2) $\vdash \bullet(\varphi_1 \vee \varphi_2) \leftrightarrow \bullet\varphi_1 \vee \bullet\varphi_2$
3) $\vdash \bullet(\exists x.\varphi) \leftrightarrow \exists x.\bullet\varphi$
4) $\vdash \circ\top \leftrightarrow \top$
5) $\vdash \circ(\varphi_1 \wedge \varphi_2) \leftrightarrow \circ\varphi_1 \wedge \circ\varphi_2$
6) $\vdash \circ(\forall x.\varphi) \leftrightarrow \forall x.\circ\varphi$
7) $\vdash \varphi \rightarrow \Diamond\varphi$ *and* $\vdash \bullet\Diamond\varphi \rightarrow \Diamond\varphi$
8) $\vdash \Box\varphi \rightarrow \varphi$ *and* $\vdash \Box\varphi \rightarrow \circ\Box\varphi$
9) $\vdash \varphi \rightarrow \Diamond_w\varphi$ *and* $\vdash \bullet\Diamond_w\varphi \rightarrow \Diamond_w\varphi$
10) $\Gamma \vdash \varphi_1 \rightarrow \varphi_2$ *implies* $\Gamma \vdash \star\varphi_1 \rightarrow \star\varphi_2$ *where* $\star \in \{\bullet, \circ, \Diamond, \Box, \Diamond_w\}$
11) $\vdash \Diamond\bot \leftrightarrow \bot$
12) $\vdash \Diamond(\varphi_1 \vee \varphi_2) \leftrightarrow \Diamond\varphi_1 \vee \Diamond\varphi_2$
13) $\vdash \Diamond(\exists x.\varphi) \leftrightarrow \exists x.\Diamond\varphi$
14) $\vdash \Box\top \leftrightarrow \top$
15) $\vdash \Box(\varphi_1 \wedge \varphi_2) \leftrightarrow \Box\varphi_1 \wedge \Box\varphi_2$
16) $\vdash \Box(\forall x.\varphi) \leftrightarrow \forall x.\Box\varphi$
17) $\vdash \Box\varphi \leftrightarrow \neg\Diamond\neg\varphi$
18) $\vdash \circ\varphi_1 \wedge \bullet\varphi_2 \rightarrow \bullet(\varphi_1 \wedge \varphi_2)$
19) $\vdash \circ(\varphi_1 \rightarrow \varphi_2) \wedge \bullet\varphi_1 \rightarrow \bullet\varphi_2$
20) $\vdash \Diamond_w\varphi \leftrightarrow (\mathsf{WF} \rightarrow \Diamond\varphi)$
21) $\vdash \Diamond_w(\varphi_1 \vee \varphi_2) \leftrightarrow \Diamond_w\varphi_1 \vee \Diamond_w\varphi_2$
22) $\vdash \Diamond_w(\exists x.\varphi) \leftrightarrow \exists x.\Diamond_w\varphi$
23) $\vdash \star\star\varphi \leftrightarrow \star\varphi$ *where* $\star \in \{\Diamond, \Box, \Diamond_w\}$
24) $\vdash \mathsf{WF} \leftrightarrow \mu X.\circ^k X$ *when* $k \geq 1$
25) $\vdash \mathsf{WF} \leftrightarrow \mu X.\circ\Box X$
26) $\vdash \Box\varphi_1 \wedge \Diamond_w\varphi_2 \rightarrow \Diamond_w(\varphi_1 \wedge \varphi_2)$
27) $\vdash \Box(\varphi_1 \rightarrow \varphi_2) \wedge \varphi_1 \rightarrow \varphi_2$

Proof: We prove them in order.

(1–3) follows from (Propagation), and (Framing).

(4–6) are proved from (1–3) and that $\circ\varphi \equiv \neg\bullet\neg\varphi$.

(7) is proved by (Pre-Fixpoint) that $\vdash \varphi \vee \bullet\Diamond\varphi \rightarrow \Diamond\varphi$.

(8) is proved by (Pre-Fixpoint) that $\vdash \Box\varphi \rightarrow \varphi \wedge \bullet\Box\varphi$.

(9) is proved by (Knaster-Tarski) that $\vdash \varphi \vee \bullet\Diamond_w\varphi \rightarrow \Diamond_w\varphi$.

(10, when $\star$ is $\bullet$) is exactly (Framing).

(10, when $\star$ is $\circ$) is exactly Proposition 12.

(10, when $\star$ is $\Diamond$) requires us to prove $\Gamma \vdash \Diamond\varphi_1 \rightarrow \Diamond\varphi_2$. By (Knaster-Tarski), it suffices to prove $\Gamma \vdash \varphi_1 \vee \bullet\Diamond\varphi_2 \rightarrow \Diamond\varphi_2$, which is proved by (7).

(10, when $\star$ is $\Box$) requires us to prove $\Gamma \vdash \Box\varphi_1 \rightarrow \Box\varphi_2$. By (Knaster-Tarski), it suffices to prove $\Gamma \vdash \Box\varphi_1 \rightarrow \varphi_1 \wedge \bullet\Box\varphi_2$, which is proved by (8).

(10, when $\star$ is $\Diamond_w$) requires us to prove $\Gamma \vdash \Diamond_w\varphi_1 \rightarrow \Diamond_w\varphi_2$. By (Knaster-Tarski), it suffices to prove $\Gamma \vdash \Diamond_w\varphi_1 \rightarrow \varphi_1 \vee \bullet\Diamond_w\varphi_2$, which is proved by (Pre-Fixpoint).

(11, "$\rightarrow$") is proved by (Knaster-Tarski).

(11,"$\leftarrow$") is trivial.

(12, "$\rightarrow$") is proved by (Knaster-Tarski), followed by (2) to propagate "$\bullet$" through "$\vee$", and finished with (7).

(12, "$\leftarrow$") is prove by (10, when $\star$ is $\Diamond$).

(13, "$\rightarrow$") is proved by (Knaster-Tarski), followed by (3) to propagate "$\bullet$" through "$\exists$", and finished with (7).

(13, "$\leftarrow$") is proved by (10, when $\star$ is $\Diamond$).

(14–16) are proved similar to (11–13).

(17, both directions) are proved by (Knaster-Tarski) followed by (Pre-Fixpoint).

(18) is proved by $\circ\varphi \equiv \neg\bullet\neg\varphi$ and (Propagation).

(19) is proved by (18) followed by (10).

(20, "$\rightarrow$") is proved by proving $\vdash \mathsf{WF} \rightarrow (\Diamond_w\varphi \rightarrow \Diamond\varphi)$, which is proved by (Knaster-Tarski) followed by (19).

(20, "$\leftarrow$") is proved by (Knaster-Tarski), followed by (2) to propagate "$\bullet$" through "$\vee$". After some additional propositional reasoning, we obtain two proof goals: $\vdash \Diamond\varphi \rightarrow \varphi \vee \bullet\Diamond\varphi$ and $\vdash \circ\mathsf{WF} \rightarrow \mathsf{WF}$. The former is proved by (Knaster-Tarski) and the latter is exactly (Pre-Fixpoint).

(21, "$\rightarrow$") is proved by applying (20) everywhere followed by (12).

(21, "$\leftarrow$") is proved by (10, when $\star$ is $\Diamond_w$).

(22, "$\rightarrow$") is proved by applying (20) everywhere followed by (13).

(22, "$\leftarrow$") is proved by (10, when $\star$ is $\Diamond_w$).

(23, when $\star$ is $\Diamond$, "$\rightarrow$") is proved by (Knaster-Tarski) followed by (7).

(23, when $\star$ is $\Diamond$, "$\leftarrow$") is proved by (7) and (10).

(23, when $\star$ is $\Box$, "$\rightarrow$") is proved by (8) and (10).

(23, when $\star$ is $\Box$, "$\leftarrow$") is proved by (Knaster-Tarski) followed by (8).

(23, when $\star$ is $\Diamond_w$, "$\rightarrow$") is proved by applying (Knaster-Tarski) first. Then we need to prove $\vdash \Diamond_w\Diamond_w\varphi \rightarrow \varphi \vee \bullet\Diamond_w\Diamond_w\varphi$. By (Pre-Fixpoint), we know $\vdash \Diamond_w\Diamond_w\varphi \rightarrow \Diamond_w\varphi \vee \bullet\Diamond_w\Diamond_w\varphi$. Thus, it suffices to prove $\vdash \Diamond_w\varphi \vee \bullet\Diamond_w\Diamond_w\varphi \rightarrow \varphi \vee \bullet\Diamond_w\Diamond_w\varphi$. By propositional reasoning, we just need to prove $\vdash \Diamond_w\varphi \rightarrow \varphi \vee \bullet\Diamond_w\Diamond_w\varphi$. By (Knaster-Tarski), we know $\vdash \Diamond_w\varphi \rightarrow \varphi \vee \bullet\Diamond_w\varphi$, so it suffices to prove $\vdash \varphi \vee \bullet\Diamond_w\varphi \rightarrow \varphi \vee \bullet\Diamond_w\Diamond_w\varphi$. Again by propositional reasoning, it suffices to prove $\vdash \bullet\Diamond_w\varphi \rightarrow \varphi \vee \bullet\Diamond_w\Diamond_w\varphi$, which can be proved by proving $\vdash \bullet\Diamond_w\varphi \rightarrow \bullet\Diamond_w\Diamond_w\varphi$, which is finally proved by (9) and (10).

(23, when $\star$ is $\Diamond_w$, "$\leftarrow$") is proved by (9) and (10).

Note it is sufficient to prove (24) only for the case $k = 1$.

(24, "$\rightarrow$") is proved by applying (Knaster-Tarski) and (Pre-Fixpoint) first. Then we need to prove $\vdash \mu X.\circ\circ X \rightarrow \circ\mu X.\circ\circ X$. Apply (Knaster-Tarski) again, and finished by (Pre-Fixpoint).

(24, "$\leftarrow$") is proved by applying (Knaster-Tarski) followed by (Pre-Fixpoint).

(25, "$\rightarrow$") is proved by applying (Knaster-Tarski) followed by (Pre-Fixpoint). Then we obtain $\vdash \mu X.\circ\Box X \rightarrow \Box\mu X.\circ\Box X$.

Axiom:
$$\frac{\varphi \Rightarrow \varphi' \in A}{A \vdash_C \varphi \Rightarrow \varphi'}$$
Reflexivity:
$$\frac{}{A \vdash_\emptyset \varphi \Rightarrow \varphi}$$
Transitivity:
$$\frac{A \vdash_C \varphi_1 \Rightarrow \varphi_2 \quad A \cup C \vdash \varphi_2 \Rightarrow \varphi_3}{A \vdash_C \varphi_1 \Rightarrow \varphi_3}$$
Logic Framing:
$$\frac{A \vdash_C \varphi \Rightarrow \varphi' \quad \psi \text{ is a FOL formula}}{A \vdash_C \varphi \wedge \psi \Rightarrow \varphi' \wedge \psi}$$
Consequence:
$$\frac{M^{\mathsf{cfg}} \vDash \varphi_1 \rightarrow \varphi_1' \quad A \vdash_C \varphi_1' \Rightarrow \varphi_2' \quad M^{\mathsf{cfg}} \vDash \varphi_2' \rightarrow \varphi_2}{A \vdash_C \varphi_1 \Rightarrow \varphi_2}$$
Case Analysis:
$$\frac{A \vdash_C \varphi_1 \Rightarrow \varphi \quad A \vdash_C \varphi_2 \Rightarrow \varphi}{A \vdash_C \varphi_1 \vee \varphi_2 \Rightarrow \varphi}$$
Abstraction:
$$\frac{A \vdash_C \varphi \Rightarrow \varphi' \quad X \cap FV(\varphi') = \emptyset}{A \vdash_C \exists X.\varphi \Rightarrow \varphi'}$$
Circularity:
$$\frac{A \vdash_{C \cup \{\varphi \Rightarrow \varphi'\}} \varphi \Rightarrow \varphi'}{A \vdash_C \varphi \Rightarrow \varphi'}$$

Fig. 8. Reachability logic proof system

Apply (Knaster-Tarski) on $\Box$, and we obtain $\vdash \mu X.\circ\Box X \rightarrow \circ\Box\mu X.\circ\Box X$, finished by (Pre-Fixpoint).

(25, "$\leftarrow$") is proved by (8), (10), and then apply Lemma 85.

(26) is proved by applying (Knaster-Tarski) firstly. After propositional reasoning, we obtain two goals: $\vdash \Box\varphi_1 \wedge \Diamond_w\varphi_2 \rightarrow \varphi_1 \vee \bullet(\Box\varphi_1 \wedge \Diamond_w\varphi_2)$ and $\vdash \Box\varphi_1 \wedge \Diamond_w\varphi_2 \rightarrow \varphi_2 \vee \bullet(\Box\varphi_1 \wedge \Diamond_w\varphi_2)$. The first goal is easily proved by (8). The second goal is by unfolding "$\Diamond_w\varphi_2$" and "$\Box\varphi_1$", and then use (18).

(27) is proved by (8). ∎

Lemma 116. *$A \vdash_C \varphi_1 \Rightarrow \varphi_2$ implies $\Gamma^{\mathsf{RL}} \vdash \mathrm{RL2MmL}(A \vdash_C \varphi_1 \Rightarrow \varphi_2)$.*

Proof: We need to prove that all reachability logic proof rules in Fig. 8 are provable in matching μ-logic.

(Axiom). We prove for the case when $C \neq \emptyset$. The case when $C = \emptyset$ is the same. Our goal, after translation, is $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \wedge \forall\boxdot C \rightarrow (\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)$. By assumption, $\varphi_1 \Rightarrow \varphi_2 \in A$, and thus we just need to prove $\Gamma^{\mathsf{RL}} \vdash \forall(\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2) \rightarrow (\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)$, which is trivial by FOL reasoning.

(Reflexivity). Notice that $C = \emptyset$ in this rule. Our goal, after translation, is $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \rightarrow (\varphi \rightarrow \Diamond_w\varphi)$, which is true by Proposition 115.

(Transitivity, $C = \emptyset$). Our goal, after translation, is $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \rightarrow (\varphi_1 \rightarrow \Diamond_w\varphi_3)$. Our two assumptions are $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \rightarrow (\varphi_1 \rightarrow \Diamond_w\varphi_2)$ and $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \rightarrow (\varphi_2 \rightarrow \Diamond_w\varphi_3)$. From the latter assumption and Proposition 115, we have $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \rightarrow (\Diamond_w\varphi_2 \rightarrow \Diamond_w\Diamond_w\varphi_3)$, and then by propositional reasoning and the former assumption we have $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \rightarrow (\varphi_1 \rightarrow \Diamond_w\Diamond_w\varphi_3)$. Finally, by Proposition 115 we have $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \rightarrow (\varphi_1 \rightarrow \Diamond_w\varphi_3)$, which is what we want to prove.

(Transitivity, $C \neq \emptyset$). Our goal, after translation, is $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \wedge \forall\circ\boxdot C \rightarrow (\varphi_1 \rightarrow \bullet\Diamond_w\varphi_3)$. Our two assumptions are $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \wedge \forall\circ\boxdot C \rightarrow (\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)$ and $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \wedge \forall\boxdot C \rightarrow (\varphi_2 \rightarrow \Diamond_w\varphi_3)$. From the first assumption, we have $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \wedge \forall\circ\boxdot C \wedge \varphi_1 \rightarrow \forall\boxdot A \wedge \forall\circ\boxdot C \wedge \bullet\Diamond_w\varphi_2$, and thus by propositional reasoning, it suffices to prove that $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \wedge \forall\circ\boxdot C \wedge \bullet\Diamond_w\varphi_2 \rightarrow \bullet\Diamond_w\varphi_3$. From the second assumption and Proposition 115(10), we know that $\Gamma^{\mathsf{RL}} \vdash \bullet\Diamond_w(\forall\boxdot A \wedge \forall\boxdot C \wedge \varphi_2) \rightarrow \bullet\Diamond_w\Diamond_w\varphi_3$, which by Proposition 115(23), implies $\Gamma^{\mathsf{RL}} \vdash \bullet\Diamond_w(\forall\boxdot A \wedge \forall\boxdot C \wedge \varphi_2) \rightarrow \bullet\Diamond_w\varphi_3$. Then, it suffices to prove $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \wedge \forall\circ\boxdot C \wedge \bullet\Diamond_w\varphi_2 \rightarrow \bullet\Diamond_w(\forall\boxdot A \wedge \forall\boxdot C \wedge \varphi_2)$. The rest is easy, since by Proposition 115(8), we just need to prove $\Gamma^{\mathsf{RL}} \vdash \forall\circ\boxdot A \wedge \forall\circ\boxdot C \wedge \bullet\Diamond_w\varphi_2 \rightarrow \bullet\Diamond_w(\forall\boxdot A \wedge \forall\boxdot C \wedge \varphi_2)$, which then by Proposition 115(18) becomes $\Gamma^{\mathsf{RL}} \vdash \bullet(\forall\boxdot A \wedge \forall\boxdot C \wedge \Diamond_w\varphi_2) \rightarrow \bullet\Diamond_w(\forall\boxdot A \wedge \forall\boxdot C \wedge \varphi_2)$, and then by Proposition 115(10) becomes $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \wedge \forall\boxdot C \wedge \Diamond_w\varphi_2 \rightarrow \Diamond_w(\forall\boxdot A \wedge \forall\boxdot C \wedge \varphi_2)$, which is proved by Proposition 115(26).

(Logic Framing). We prove for the case when $C \neq \emptyset$. The case when $C = \emptyset$ is the same. Our goal, after translation, is $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \wedge \forall\circ\boxdot C \rightarrow (\varphi_1 \wedge \psi \rightarrow \bullet\Diamond_w(\varphi_2 \wedge \psi))$. Our assumption is $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \wedge \forall\circ\boxdot C \rightarrow (\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)$. Notice that FOL formula ψ is a predicate pattern, so $\vdash \bullet\Diamond_w(\varphi_2 \wedge \psi) \leftrightarrow (\bullet\Diamond_w\varphi_2) \wedge \psi$, and the rest is by propositional reasoning. The condition that ψ is a FOL formula (and thus a predicate pattern) is crucial to propagate ψ throughout its context.

(Consequence). This is the only rule where axioms in Γ^{RL} is actually used. Again, we prove for the case $C \neq \emptyset$ as the case when $C = \emptyset$ is the same. Our goal, after translation, is $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \wedge \forall\circ\boxdot C \rightarrow (\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)$. Our three assumptions include $M^{\mathsf{cfg}} \vDash \varphi_1 \rightarrow \varphi_1'$, $M^{\mathsf{cfg}} \vDash \varphi_2' \rightarrow \varphi_2$, and $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \wedge \forall\circ\boxdot C \rightarrow (\varphi_1' \rightarrow \bullet\Diamond_w\varphi_2')$. Notice that by definition of Γ^{RL}, we know immediately that $\varphi_1 \rightarrow \varphi_1' \in \Gamma^{\mathsf{RL}}$ and $\varphi_2' \rightarrow \varphi_2 \in \Gamma^{\mathsf{RL}}$. The rest of the proof is simply by Proposition 115(10) and some propositional reasoning.

(Case Analysis). Simply by some propositional reasoning.

(Abstraction). Simply by some FOL reasoning. Notice that $\forall\boxdot A$ and $\forall\boxdot C$ are closed patterns.

(Circularity). We prove for the case when $C \neq \emptyset$, as the case when $C = \emptyset$ is the same. Our goal, after translation, is $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \wedge \forall\circ\boxdot C \rightarrow (\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)$. By FOL reasoning and Proposition 115(20,2,25), the goal becomes $\Gamma^{\mathsf{RL}} \vdash \mu X.\circ\Box X \rightarrow \forall\boxdot A \wedge \forall\circ\boxdot C \rightarrow \forall(\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)$. By (Knaster-Tarski) and some FOL reasoning, it suffices to prove $\Gamma^{\mathsf{RL}} \vdash \circ\Box(\forall\boxdot A \wedge \forall\circ\boxdot C \rightarrow \forall(\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)) \wedge \forall\boxdot A \wedge \forall\circ\boxdot C \rightarrow (\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)$. Our assumption, after translation, is $\Gamma^{\mathsf{RL}} \vdash \forall\boxdot A \wedge \forall\circ\boxdot C \wedge \forall\circ(\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2) \rightarrow (\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)$, so it suffices to prove $\Gamma^{\mathsf{RL}} \circ\Box(\forall\boxdot A \wedge \forall\circ\boxdot C \rightarrow \forall(\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)) \wedge \forall\boxdot A \wedge \forall\circ\boxdot C \rightarrow \forall\boxdot A \wedge \forall\circ\boxdot C \wedge \forall\circ(\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)$, which by some propositional reasoning becomes $\Gamma^{\mathsf{RL}} \vdash \circ\Box(\forall\boxdot A \wedge \forall\circ\boxdot C \rightarrow \forall(\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)) \wedge \forall\boxdot A \wedge \forall\circ\boxdot C \rightarrow \forall\circ(\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)$. By Proposition 115(8), it becomes $\Gamma^{\mathsf{RL}} \vdash \circ\Box(\forall\boxdot A \wedge \forall\circ\boxdot C \rightarrow \forall(\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)) \wedge \circ\forall\boxdot A \wedge \circ\forall\circ\boxdot C \rightarrow \forall\circ(\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)$, and by Proposition 115(5,6,10), it becomes $\Gamma^{\mathsf{RL}} \vdash \Box(\forall\boxdot A \wedge \forall\circ\boxdot C \rightarrow \forall(\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)) \wedge \forall\boxdot A \wedge \forall\circ\boxdot C \rightarrow \forall(\varphi_1 \rightarrow \bullet\Diamond_w\varphi_2)$, which

is proved by Proposition 115(27). ■

Corollary 117. $S \vdash_\emptyset \varphi_1 \Rightarrow \varphi_2$ *implies* $\Gamma^{\mathsf{RL}} \vdash \mathrm{RL2MmL}(S \vdash_\emptyset \varphi_1 \Rightarrow \varphi_2)$.

Proof: Let $A = S$ and $C = \emptyset$ in Lemma 116. ■

Lemma 118. $\Gamma^{\mathsf{RL}} \vDash \mathrm{RL2MmL}(S \vdash_\emptyset \varphi_1 \Rightarrow \varphi_2)$ *implies* $S \vDash_{\mathsf{RL}} \varphi_1 \Rightarrow \varphi_2$.

Proof: Let $\mathbb{S} = (M^{\mathsf{cfg}}_{Cfg}, R)$ be the transition system that is yielded by S. We tactically use the same letter $\mathbb{S}$ to mean the extended $\mathbb{\Sigma}^{\mathsf{RL}}$-model M^{cfg} with $\bullet \in \Sigma_{Cfg,Cfg}$ be interested as the transition relation R. Then $\mathbb{S} \vDash \Gamma^{\mathsf{RL}}$, because all axioms in Γ^{RL} are about only the configuration model M^{cfg} and says nothing about the transition relation R. Since $M^{\mathsf{cfg}} \vDash \Gamma^{\mathsf{cfg}}$ (by definition), then $\mathbb{S} \vDash \Gamma^{\mathsf{cfg}}$. By condition of the lemma, $\mathbb{S} \vDash \mathrm{RL2MmL}(S \vdash_\emptyset \varphi_1 \Rightarrow \varphi_2)$, i.e., $\mathbb{S} \vDash \forall \boxdot S \rightarrow \varphi_1 \rightarrow \Diamond_w \varphi_2$. By construction of $\mathbb{S}$, for all rules $\psi_1 \Rightarrow \psi_2 \in S$, we have $\mathbb{S} \vDash \psi_1 \rightarrow \bullet \psi_2$ (in MmL), which implies $\mathbb{S} \vDash \forall \Box(\psi_1 \rightarrow \Diamond_w \psi_2)$, meaning that $\mathbb{S} \vDash \forall \boxdot S$. Therefore, $\mathbb{S} \vDash \varphi_1 \rightarrow \Diamond_w \varphi_2$ (in MmL), which is exactly the same meaning as $\mathbb{S} \vDash_{\mathsf{RL}} \varphi_1 \Rightarrow \varphi_2$ (in RL). ■

Finally, we are ready to prove Theorem 39.

Proof of Theorem 39: Following the same roadmap as in the proof of Theorem 31, where (2) ⇒ (3) is given by Corollary 117 and (5) ⇒ (6) is given by Lemma 118. The rest is by the sound and (relative) completeness of RL. Notice that technical assumptions of [2] are needed for the completeness result of RL. ■