Reflective Metalogical Frameworks*

David Basin
Department of Computer Science,
ETH Zurich

Manuel Clavel
Facultad de Informática,
Universidad Complutense de Madrid, Spain

José Meseguer
Computer Science Department,
University of Illinois at Urbana-Champaign, USA

April 8, 2003

Abstract

A metalogical framework is a logic with an associated methodology that is used to represent other logics and to reason about their metalogical properties. We propose that logical frameworks can be good metalogical frameworks when their theories always have initial models and they support reflective and parameterized reasoning.

We develop this thesis both abstractly and concretely. Abstractly, we formalize our proposal as a set of requirements and explain how any logic satisfying these requirements can be used for metalogical reasoning. Concretely, we present membership equational logic as a particular metalogic that satisfies these requirements. Using membership equational logic, and its realization in the Maude system, we show how reflection can be used for different, non-trivial kinds of formal metatheoretic reasoning. In particular, one can prove metatheorems that relate theories or establish properties of parameterized classes of theories.

1 Introduction

Metalogical Reasoning

A logical framework is a logic with an associated methodology that is employed for representing and using other logics, theories, and, more generally, formal

*Supported by DARPA through Rome Laboratories Contract F30602-C-0312, by DARPA and NASA through Contract NAS2-98073, by Office of Naval Research Contract N00014-99-C-0198, by National Science Foundation Grant CCR-9900334, and by Spanish CICYT Project TIC2002-01167.

systems. A number of logical frameworks have been proposed and in order to compare them and analyze their effectiveness, it is helpful to distinguish between their intended applications. In particular, we can distinguish between *logical frameworks*, where the emphasis is on reasoning *in* a logic, in the sense of simulating its derivations in the framework logic, and *metalogical frameworks* [6, 9], where the emphasis is on reasoning *about* logics and even about relationships *between* logics. For example, in a logical framework one might establish the provability of some formula ϕ or check that a putative proof of ϕ is an actual proof, whereas in a metalogical framework one might show that some logic, or family of related logics, has some proof-theoretic property, such as cut-elimination. Metalogical frameworks are more powerful, as they include the ability to reason about a logic's entailment relation (or other proof-theoretic or semantic properties), as opposed to merely being adequate to simulate entailment.

Induction plays a central rôle in distinguishing logical frameworks from their metalogical counterparts. In a logical framework, representations of proof rules are used to construct derivations of (object logic) entailments. This approach is taken in logical frameworks like Isabelle [60] and the Edinburgh LF [37]. There, one may formalize logics and theories where induction is present *within* particular theories (e.g., Peano Arithmetic), but induction is not present *over* the theories. That is, the framework does not support induction over the terms and proofs of a theory. In contrast, in a metalogical framework, it is essential to have induction over theories. Standard proof-theoretic arguments usually require induction over the formulae or derivations of the object theory. Inductive reasoning is also important in computer science applications, e.g., reasoning about data types, recursive programs, operational semantics, and the like.

The importance of induction for metareasoning is well understood. Less explored, but also important, is the rôle of parameterization. Many kinds of metatheorems are statements not about a single logic, but about multiple logics (or theories), i.e., families of related formal systems. Perhaps the simplest example of this is the deduction theorem for *minimal logic (of implication)*. This metatheorem states that for all formulae A and B of minimal logic,

$$\text{if } \vdash_{\mathcal{M}[A]} B \text{ then } \vdash_{\mathcal{M}} A \to B\,,$$

where $\vdash_{\mathcal{M}}$ denotes provability in minimal logic and $\vdash_{\mathcal{M}[A]}$ denotes provability in the logic consisting of minimal logic extended by the additional axiom A. Since A is an arbitrary formula, this (standard) statement of the deduction theorem is a statement about the relationship between a family of pairs of logics. Semantically, it constitutes a statement about a family of pairs of models, the initial models corresponding to the proofs in the two respective logics.

A second example, which suggests the wide scope of these concepts, comes from a rather different domain: functional programming. There researchers have developed language mechanisms to formalize polytypic functions [43]. These are functions like *map* and *fold* that operate recursively over inductively defined data types. Polytypic functions are defined to behave uniformly over members of different data types. For example, *map* operates in a similar way over lists, trees, etc. Whereas one would traditionally define a different *map* function over each of these types, in the polytypic setting one instead writes a single function that operates over elements of all of these types. The recent work of [17] on developing a methodology for defining and reasoning about polytypic functions can be seen as an instance of the general methodology proposed in this paper: establishing general facts about polytypic functions amounts to proving inductive properties about families of theories, namely that some desired property holds of the initial algebra of each extension of an inductive data type with the corresponding polytypic function.

Reflective Metalogical Frameworks

The question we address in this paper is how to design a metalogic tailored to support both inductive and parameterized reasoning. Our answer is motivated by the following observation. A logic's syntax and proofs are inductively built from syntax and proof constructors. A logical framework and a metalogical framework can share these as a common basis. However, whereas for a logical framework the application of these constructors suffices to simulate derivations of the object logic, for a metalogical framework the representation must additionally preserve the inductive nature of the syntax and proof constructors. Model-theoretically, this means that a formalization in the metalogic should have an *initial model*, which corresponds to the syntax and proofs of the formalized object logic.

Our proposal is that for some logical frameworks—namely, those that are reflective and whose theories have initial models—we can take the step from a logical framework to a metalogical framework by reflecting at the metalevel the induction principles for the formalized logics. Moreover, when the metalogic has an associated notion of a parameterized theory, which can also be metarepresented and reasoned about, then we can use reflection and induction to prove metatheorems about *families* of theories. We sum this up with the slogan "*logical frameworks with reflection, initiality, and parameterization are metalogical frameworks*."[1]

We proceed in several steps. First, we formalize our proposal by giving three abstract requirements, which are sufficient for such a metalogical framework. The three requirements—initiality, reflection, and reflected parameterized induction—leave open the possibility of different metalogics, including first-order and higher-order ones. Second, we give a nontrivial instance of these requirements by presenting a concrete realization of these ideas using membership equational logic. Although membership equational logic is not the only candidate for a reflective metalogical framework, we believe it is a good one. Membership equational logic is balanced on a point where it is expressive enough

[1]This slogan should not be construed as a *definition* of a metalogical framework. Logics not meeting some of our requirements may still have useful metalogical framework applications. However, we argue in this work that the combination of reflection, initiality, and parameterization makes a logic ideally suited as a metalogical framework.

to naturally formalize different entailment systems, but it is weak enough so that its theories always have initial models. Moreover, membership equational logic has a formal metatheory that can be used for formalizing parameterization and for reflective inductive reasoning. As we will see, together this means that membership equational logic provides an instance of our requirements. Finally, we show how these ideas are concretely applied on examples, including the deduction theorem.

Overall, we see our contributions as both theoretical and practical. Theoretically, our work contributes to answering the question "*What is a metalogical framework?*" by proposing reflective logical frameworks, as defined by our requirements, as a possible answer. Moreover, our work illuminates the interrelationship between logical and metalogical frameworks, and the rôles of induction, reflection, and parameterization as key ingredients for turning a logical framework into a metalogical one. Practically, we provide evidence that membership equational logic, combined with reflection, is an effective metalogical framework that can be used for nontrivial kinds of metatheoretic reasoning.

Organization

Our paper is organized in three parts. In the first part, Sections 2–3, we formalize our abstract requirements for a reflective metalogical framework. In the second part, Sections 4–10, we present membership equational logic as a concrete instance of our requirements and present a case study. Finally, in Sections 11–12, we discuss tradeoffs and limitations, survey related work, and draw conclusions. We also include an appendix, where we relegate some of the technical details.

2 Logics and Theories

As we aim to give general requirements for reflective metalogical frameworks, we begin with background material that provides a general account of logics, theories, reflection, and parameterization. Most of the material is standard [33, 34, 54] and only summarized here, but the material in Sections 2.5 and 2.6 develops some further model-theoretic requirements on quantification and satisfaction needed for our purposes.

2.1 Entailment Systems

We shall assume that logical syntax is given by a *signature* Σ that provides a grammar for building *sentences*. For first-order logic, a typical signature consists of a collection of function and predicate symbols, each with a prescribed number of arguments, which are used to build sentences by means of the usual logical connectives. For our purposes, it is enough to assume that for each logic there is a category **Sign** of possible signatures, and a functor *sen* assigning to each signature Σ the set $sen(\Sigma)$ of its sentences.

For a given signature Σ in **Sign**, *entailment* (also called *provability*) of a sentence $\varphi \in sen(\Sigma)$ from a set of axioms $\Gamma \subseteq sen(\Sigma)$ is a relation $\Gamma \vdash_\Sigma \varphi$ that holds if and only if we can prove φ from the axioms Γ using the rules of the logic. We define this relation relative to a signature and independent of the details of a particular logic or deductive system. In what follows, $|\mathcal{C}|$ denotes the collection of objects of a category $\mathcal{C}$.

Definition 1 [54] *An* entailment system *is a triple* $\mathcal{E} = (\mathbf{Sign}, sen, \vdash)$ *such that*

- **Sign** *is a category whose objects are called* signatures*,*
- $sen : \mathbf{Sign} \longrightarrow \mathbf{Set}$ *is a functor associating to each signature* Σ *a corresponding set of* Σ-sentences*, and*
- $\vdash$ *is a function associating to each* $\Sigma \in |\mathbf{Sign}|$ *a binary relation* $\vdash_\Sigma \subseteq \mathcal{P}(sen(\Sigma)) \times sen(\Sigma)$ *called* Σ-entailment *such that the following properties are satisfied:*
  1. *reflexivity: for any* $\varphi \in sen(\Sigma)$, $\{\varphi\} \vdash_\Sigma \varphi$,
  2. *monotonicity: if* $\Gamma \vdash_\Sigma \varphi$ *and* $\Gamma' \supseteq \Gamma$, *then* $\Gamma' \vdash_\Sigma \varphi$,
  3. *transitivity: if* $\Gamma \vdash_\Sigma \varphi_i$, *for all* $i \in I$, *and* $\Gamma \cup \{\varphi_i \mid i \in I\} \vdash_\Sigma \psi$, *then* $\Gamma \vdash_\Sigma \psi$,
  4. $\vdash$*-translation: if* $\Gamma \vdash_\Sigma \varphi$, *then for any* $H : \Sigma \longrightarrow \Sigma'$ *in* **Sign**, $sen(H)(\Gamma) \vdash_{\Sigma'} sen(H)(\varphi)$, *where* $sen(H)(\Gamma) = \{sen(H)(\varphi) \mid \varphi \in \Gamma\}$, *as is standard.*

Except for the explicit treatment of syntax translations, the axioms are essentially Scott's axioms for a consequence relation [68].

2.2 Theories and Parameterization

Definition 2 [54] *Given an entailment system* $\mathcal{E}$, *its category* **Th** *of* theories[2] *has as objects pairs* $T = (\Sigma, \Gamma)$, *with* Σ *a signature and* $\Gamma \subseteq sen(\Sigma)$. *A* theory morphism *(also called a* theory interpretation*)* $H : (\Sigma, \Gamma) \longrightarrow (\Sigma', \Gamma')$ *is a signature morphism* $H : \Sigma \longrightarrow \Sigma'$ *such that if* $\varphi \in \Gamma$, *then* $\Gamma' \vdash_{\Sigma'} sen(H)(\varphi)$.

By composing with the forgetful functor $sign : \mathbf{Th} \longrightarrow \mathbf{Sign}$, with $sign(\Sigma, \Gamma) = \Sigma$, we can extend the functor $sen : \mathbf{Sign} \longrightarrow \mathbf{Set}$ to a functor $sen : \mathbf{Th} \longrightarrow \mathbf{Set}$, i.e., we define $sen(T) \triangleq sen(sign(T))$.

To make specifications of theories more modular, it is useful to *parameterize* theories. For example, the theory *Vect* of vector spaces is parameterized by the theory *Field* for the field of coefficients. This general notion can then be specialized by instantiating the parameter *Field* with a more concrete instance satisfying the requirements of *Field*, for example with the theory $Field_p$ of fields of characteristic p, or *OrdField*, of ordered fields.

[2] Theories in this axiomatization are sometimes called *theory presentations* in the literature.

As pointed out by Goguen and Burstall [34], a *parameterized theory* can be defined for logics in general as a pair of theories: the *parameter* P and the *body* T, which are related by a theory map $J : P \longrightarrow T$, which is typically a theory inclusion. To *instantiate* such a parameterized theory, the key data needed is a theory morphism $H : P \longrightarrow Q$ from the parameter theory to another theory Q (for example, from *Field* to Field_p or to *OrdField* in the case of *Vect* parameterized over *Field*). The *instantiation* by H is then defined as the pushout commutative diagram

$$\begin{array}{ccc} T & \xrightarrow{H^T} & T[H] \\ {\scriptstyle J}\uparrow & & \uparrow{\scriptstyle J^Q} \\ P & \xrightarrow[H]{} & Q \end{array}$$

in the category **Th** of theories, when such a pushout exists. Being a pushout means that for any other pair of theory maps $H' : T \longrightarrow S$, $J' : Q \longrightarrow S$ to another theory S such that $H' \circ J = J' \circ H$, there exists a unique theory morphism $F : T[H] \longrightarrow S$ such that $F \circ H^T = H'$ and $F \circ J^Q = J'$. In what follows we shall denote a parameterized theory $J : P \longrightarrow T$ by the notation $T[P]$, and, as suggested in the above diagram, for each $H : P \longrightarrow Q$, $T[H]$ will then denote the instance of T defined by H.

2.3 Institutions and Logics

The axiomatization of model theory we employ is that of Goguen and Burstall on *institutions* [33, 34].

Definition 3 [33] *An* institution *is a 4-tuple* $\mathcal{I} = (\mathbf{Sign}, sen, \mathbf{Mod}, \models)$ *where*

- **Sign** *is a category whose objects are called* signatures*,*
- $sen : \mathbf{Sign} \longrightarrow \mathbf{Set}$ *is a functor associating to each signature* Σ *a set of* Σ-sentences*,*
- $\mathbf{Mod} : \mathbf{Sign} \longrightarrow \mathbf{Cat}^{op}$ *is a functor that gives for each signature* Σ *a category whose objects are called* Σ-models*, and*
- $\models$ *is a function associating to each* $\Sigma \in |\mathbf{Sign}|$ *a binary relation* $\models_\Sigma \subseteq |\mathbf{Mod}(\Sigma)| \times sen(\Sigma)$ *called* Σ-satisfaction *satisfying the following* satisfaction condition *for each* $H : \Sigma \longrightarrow \Sigma'$ *in* **Sign***: for all* $M' \in |\mathbf{Mod}(\Sigma')|$ *and all* $\varphi \in sen(\Sigma)$*,*

$$M' \models_{\Sigma'} sen(H)(\varphi) \iff \mathbf{Mod}(H)(M') \models_\Sigma \varphi.$$

The satisfaction condition just requires that, for any syntax translation between two signatures, a model of the second signature satisfies a translated sentence if and only if the translation of this model satisfies the original sentence. Note that **Mod** is a contravariant functor, that is, the translation of models goes “backwards”.

Given a theory $T = (\Sigma, \Gamma)$, we define the category $\mathbf{Mod}(T)$ as the full subcategory of $\mathbf{Mod}(\Sigma)$ determined by those models $M \in |\mathbf{Mod}(\Sigma)|$ that satisfy all the sentences in Γ, i.e., $M \models_\Sigma \varphi$ for each $\varphi \in \Gamma$. We say that the theory T has an initial model if there is a model, denoted $\mathcal{I}(T)$, with $\mathcal{I}(T) \in |\mathbf{Mod}(T)|$, such that for each $M \in |\mathbf{Mod}(T)|$ there is a unique morphism $\mathcal{I}(T) \longrightarrow M$ in $\mathbf{Mod}(T)$, that is, $\mathcal{I}(T)$ is an initial object in the category $\mathbf{Mod}(T)$.

Defining a *logic* is now simple.

Definition 4 [54] *A* logic *is a 5-tuple* $\mathcal{L} = (\mathbf{Sign}, sen, \mathbf{Mod}, \vdash, \models)$ *such that*

- $(\mathbf{Sign}, sen, \vdash)$ *is an entailment system,*
- $(\mathbf{Sign}, sen, \mathbf{Mod}, \models)$ *is an institution,*

and the following soundness condition *is satisfied: for all* $\Sigma \in |\mathbf{Sign}|$, $\Gamma \subseteq sen(\Sigma)$, *and* $\varphi \in sen(\Sigma)$,

$$\Gamma \vdash_\Sigma \varphi \implies \Gamma \models_\Sigma \varphi,$$

where, by definition, the relation $\Gamma \models_\Sigma \varphi$ *holds if and only if* $M \models_\Sigma \varphi$ *holds for any model* M *that satisfies all the sentences in* Γ.

The logic is called complete *if the above implication is in fact an equivalence.*

The following definition of sublogic is a special case of the more general definition of sublogic in [54].

Definition 5 *A logic* $\mathcal{L} = (\mathbf{Sign}, sen, \mathbf{Mod}, \vdash, \models)$ *is a* sublogic *of a logic* $\mathcal{L}' = (\mathbf{Sign}', sen', \mathbf{Mod}', \vdash', \models')$ *iff the following conditions hold:*

- $\mathbf{Sign} \subseteq \mathbf{Sign}'$, *that is,* $\mathbf{Sign}$ *is a subcategory of* $\mathbf{Sign}'$, *and* $\mathbf{Mod}$ *is the restriction of* $\mathbf{Mod}'$ *to the subcategory* $\mathbf{Sign}$, *i.e.,* $\mathbf{Mod}'|_{\mathbf{Sign}} = \mathbf{Mod}$.
- *For each* $\Sigma \in |\mathbf{Sign}|$ *there is an inclusion* $sen(\Sigma) \subseteq sen'(\Sigma)$ *giving rise to a natural transformation* $\alpha : sen \Longrightarrow sen'|_{\mathbf{Sign}}$.
- *For each* $\Sigma \in |\mathbf{Sign}|$, $\Gamma \subseteq sen(\Sigma)$, *and* $\varphi \in sen(\Sigma)$ *we have*

$$\Gamma \vdash_\Sigma \varphi \Longleftrightarrow \Gamma \vdash'_\Sigma \varphi\,.$$

- *For each* $\Sigma \in |\mathbf{Sign}|$, $M \in \mathbf{Mod}(\Sigma)$, *and* $\varphi \in sen(\Sigma)$ *we have*

$$M \models_\Sigma \varphi \Longleftrightarrow M \models'_\Sigma \varphi\,.$$

We use the notation $\mathcal{L} \subseteq \mathcal{L}'$ *to denote that* $\mathcal{L}$ *is a sublogic of* $\mathcal{L}'$, *and call* $\mathcal{L}'$ *a* superlogic *of* $\mathcal{L}$.

2.4 Reflection

For our abstract requirements, we will require the existence of certain theories that can represent, and formalize statements about, other theories. In particular:

Definition 6 [20, 13] *Given an entailment system $\mathcal{E}$ and a nonempty set of theories $\mathcal{C}$ in it, a theory U is $\mathcal{C}$-*universal *if there is an injective function, called a* representation function*,*

$$\overline{(\_ \vdash \_)} : \bigcup_{T \in \mathcal{C}} (\{T\} \times sen(T)) \longrightarrow sen(U)\,,$$

such that for each $T \in \mathcal{C}, \varphi \in sen(T)$,

$$T \vdash_{sign(T)} \varphi \Longleftrightarrow U \vdash_{sign(U)} \overline{T \vdash \varphi}\,.$$

*If, in addition, $U \in \mathcal{C}$, then the entailment system $\mathcal{E}$ is called $\mathcal{C}$-*reflective*. Finally, a* reflective logic *is a logic whose entailment system is $\mathcal{C}$-reflective for $\mathcal{C}$ the class of all finitely presentable theories in the logic.*

2.5 Universal and Existential Quantification

To reason about parametrically specified families of theories it is important that the metalogic used supports quantification so that we can make universal (or existential) statements about parameter instances. Here we proceed semantically and state general model-theoretic conditions for institutions having universal and existential quantification in their sentences. The key idea is to consider a subcategory $\mathbf{Var} \subseteq \mathbf{Sign}$, whose objects stand for sets of variables. In a first-order language, an object $X \in |\mathbf{Var}|$ corresponds to a signature having only constants, but in a higher-order language $X \in |\mathbf{Var}|$ could also involve function and predicate symbols. We further require an operator j mapping each signature Σ and $X \in |\mathbf{Var}|$ to the signature $\Sigma \oplus X$ such that there are two signature morphisms $j_X : X \longrightarrow \Sigma \oplus X$ and $j_\Sigma : \Sigma \longrightarrow \Sigma \oplus X$. Furthermore, if $Z \in |\mathbf{Var}|$ is a coproduct (i.e., the categorical generalization of a disjoint union) of the form $Z = X \oplus Y$, then we require the existence of an isomorphism

$$\Sigma \oplus Z \simeq (\Sigma \oplus X) \oplus Y\,.$$

For example, in first-order logic, $\Sigma \oplus X$ can be defined as the disjoint union $\Sigma + X$ that adds the variables in X as new constants to Σ.

We can extend the above notation to theories $T = (\Sigma, \Gamma)$ by defining $T \oplus X = (\Sigma \oplus X, \Gamma)$. Assume that we have an institution satisfying the above requirements. Then given a signature Σ and $X \in |\mathbf{Var}|$, we define the set $form_X(\Sigma)$ (and its extension $form_X(T)$) of Σ-*formulae with variables* in X as the set $sen(\Sigma \oplus X)$ of sentences in the extended signature $\Sigma \oplus X$. We then say that an institution $\mathcal{I}$ with the above requirements has *universal* quantification if for each $\Sigma \in |\mathbf{Sign}|$ and $X \in |\mathbf{Var}|$ there is a function $\forall X.\_ : form_X(\Sigma) \longrightarrow sen(\Sigma)$

(respectively $\exists X.\_ : \mathit{form}_X(\Sigma) \longrightarrow \mathit{sen}(\Sigma)$) such that for each $A \in |\mathbf{Mod}(\Sigma)|$ and $\phi \in \mathit{form}_X(\Sigma)$ we have

$$A \models_\Sigma \forall X.\, \phi \iff \forall A' \in |\mathbf{Mod}(\Sigma \oplus X)|.\, (\mathbf{Mod}(j_\Sigma)(A') = A \implies (A' \models_{\Sigma \oplus X} \phi))\,.$$

Similarly an institution has *existential* quantification when

$$A \models_\Sigma \exists X.\, \phi \iff \exists A' \in |\mathbf{Mod}(\Sigma \oplus X)|.\, (\mathbf{Mod}(j_\Sigma)(A') = A \wedge (A' \models_{\Sigma \oplus X} \phi))\,.$$

2.6 Tarskian Semantics

In our abstract requirements for logics $\mathcal{L}$ having good properties as metalogical frameworks we will also find it useful to assume that $\mathcal{L}$ is a sublogic of a logic $\mathcal{S}$ whose underlying institution has a Tarskian semantics.

We say that an institution $(\mathbf{Sign}, \mathit{sen}, \mathbf{Mod}, \models)$ has a *Tarskian semantics* if for each $\Sigma \in |\mathbf{Sign}|$ the set $\mathit{sen}(\Sigma)$ has two operations, $\neg : \mathit{sen}(\Sigma) \longrightarrow \mathit{sen}(\Sigma)$, and $\_ \wedge \_ : \mathit{sen}(\Sigma) \times \mathit{sen}(\Sigma) \longrightarrow \mathit{sen}(\Sigma)$, such that, for all $\phi, \psi \in \mathit{sen}(\Sigma)$ and $M \in |\mathbf{Mod}(\Sigma)|$ we have

- $M \models_\Sigma \neg\phi \iff M \not\models_\Sigma \phi$, and
- $M \models_\Sigma \phi \wedge \psi \iff M \models_\Sigma \phi$ and $M \models_\Sigma \psi$.

As usual, we can define other Boolean connectives in terms of $\neg$ and $\wedge$. Hence, given any Boolean expression $b(x_1, \ldots, x_n)$ and sentences $\phi_1, \ldots, \phi_n \in \mathit{sen}(\Sigma)$, for all $M \in \mathbf{Mod}(\Sigma)$ we also have

- $M \models_\Sigma b(\phi_1, \ldots, \phi_n) \iff b(M \models_\Sigma \phi_1, \ldots, M \models_\Sigma \phi_n)$.

3 Abstract Requirements

In this section we motivate and formalize our abstract requirements for reflective metalogical frameworks. These requirements provide a formal account of our slogan "logical frameworks with reflection, initiality, and parameterization are metalogical frameworks."

3.1 Motivation

To motivate our abstract requirements for reflective metalogical frameworks we return to the deduction theorem, which will be the running example in this paper. Full details for this example will be given later in Section 10; for the moment it suffices to point out that the minimal logic of implication can be specified by a Horn theory $\mathcal{M} = (\Sigma_\mathcal{M}, \Gamma_\mathcal{M})$ with two unary predicates, $\_ : \mathit{Formula}$ and $\_ : \mathit{Theorem}$, classifying which expressions are formulae and theorems. In fact, the theorems of $\mathcal{M}$ are exactly the expressions t such that $t : \mathit{Theorem}$ holds in the *initial model* $\mathcal{I}(\mathcal{M})$ of the Horn theory $\mathcal{M}$. We can then understand the deduction theorem as a statement involving two parameterized theories extending $\mathcal{M}$ and having the same parameter theory P. The theory P is of the form

$$P = (\Sigma_{\mathcal{M}} \oplus \{A, B\}, \Gamma_{\mathcal{M}} \cup \{A : \mathit{Formula}, B : \mathit{Formula}\})\,,$$

i.e., we add to $\mathcal{M}$ the variables A and B as constants, which we will call *parameters*, as well as the axioms stating that A and B are formulae.

The two parameterized theories $T_1[P]$ and $T_2[P]$ involved in the deduction theorem are, respectively:

1. the extension of P by the extra axiom $A : \mathit{Theorem}$, with $J_1 : P \longrightarrow T_1$ the obvious theory inclusion (this captures the idea of assuming A as an extra axiom, but A for the moment is a parameter), and
2. the trivial extension of P by itself, that is, we do not add anything else. So J_2 is the identity map $1_P : P \longrightarrow P$.

Under this view, the deduction theorem becomes a theorem about a family of theory instantiations for both $T_1[P]$ and $T_2[P]$, namely those associated to the family $\mathcal{V}$ of all theory morphisms $H : P \longrightarrow (\mathcal{M} \oplus \{A, B\}) \cup E$ such that H is the identity signature morphism, and $E = \{A = H(A), B = H(B)\}$ is a set of axioms that assign non-parameterized expressions (i.e., ground expressions that do not contain A and B) $H(A)$ and $H(B)$ to the parameters A and B in P. Moreover, since the axioms of P must be satisfied by such an H, we must also have that $(\mathcal{M} \oplus \{A, B\}) \cup E \vdash H(A) : \mathit{Formula}$ and $(\mathcal{M} \oplus \{A, B\}) \cup E \vdash H(B) : \mathit{Formula}$. That is, the family $\mathcal{V}$ formalizes all the different ways of assigning two formulae in minimal logic to the parameters A and B. The deduction theorem can then be expressed as a metatheoretic statement relating the initial models $\mathcal{I}(T_1[H])$ and $\mathcal{I}(T_2[H])$ of the different instantiations $T_1[H]$ and $T_2[H]$ for all such H, as follows:

$$\forall H \in \mathcal{V}.\, \mathcal{I}(T_1[H]) \models H(B) : \mathit{Theorem} \Longrightarrow \mathcal{I}(T_2[H]) \models H(A \to B) : \mathit{Theorem}\,. \quad (1)$$

Taking stock, the deduction theorem makes a semantic statement about a family of pairs of parameterized theories, namely all parameter instances $H \in \mathcal{V}$ of the two parameterized theories $T_1[H]$ and $T_2[H]$. Its formal statement presupposes a metatheory in which parameterized theories are "first class objects," i.e., we can build sentences from them, and its proof requires the ability to reason inductively about such theories and their relationships. Hence, the key idea is to require a class $\mathcal{C}$ of theories in our logic $\mathcal{L}$ such that there is a universal theory U (so we can formalize and reason about other theories) and all theories $T \in \mathcal{C}$ have initial models (so inductive consequence is well-defined). Moreover, when conducting proofs, we want not only the possibility of having induction principles to reason about truth in such initial models $\mathcal{I}(T)$, but also a way of extending U to a theory with *reflected parameterized induction principles* (in the sense made precise in Section 3.2 below), so as to be able to reason metatheoretically about properties of families of parameterized theories.

3.2 Requirements

By formalizing the above ideas for a general logic, we arrive at our abstract requirements for reflective metalogical frameworks. In order to specify these requirements, we first define two notions, related to theory morphisms introduced in Section 2.2. Let $T[P]$ be a parameterized theory where the signature of P is of the form $\Sigma \oplus V$, for V a signature whose symbols stand for parameters. Then, given a family of theory morphisms $\mathcal{V} = \{H_i : P \longrightarrow S_i\}_{i\in I}$, we define the *restriction of* $\mathcal{V}$ *to* Σ as the family

$$\mathcal{V}|_{\Sigma} \triangleq \{H_i \circ j_{\Sigma} : \Sigma \longrightarrow \Sigma'_i\}_{i\in I} ,$$

where Σ'_i is the signature of S_i. Similarly, given a signature map $K : \Sigma \longrightarrow \Sigma'$, we define the *extension of* K *to* $\mathcal{V}$ as the family

$$Ext^{V}_{\mathcal{V}}(K) \triangleq \{H \in \mathcal{V} \mid H \circ j_{\Sigma} = K\} .$$

We now state our three abstract requirements for a logic $\mathcal{L}$ to be a metalogical framework.

1) Reflection. $\mathcal{L}$ has a class $\mathcal{C}$ of finitely presentable theories such that there is a theory U in $\mathcal{C}$ that is $\mathcal{C}$-*universal*.

2) Initiality. Each $T \in \mathcal{C}$ has an initial model $\mathcal{I}(T) \in |\mathbf{Mod}(T)|$.

3) Reflected parameterized induction. This requirement has three parts.

(i) $\mathcal{L}$ has a superlogic $\mathcal{S}$ which has a Tarskian semantics and admits both universal and existential quantification.

(ii) For $P = (\Sigma \oplus V, \Gamma)$ a parameter theory in $\mathcal{C}$, $\mathcal{P} = \{T_i[P]\}_{i\in I}$ a family of parameterized theories in $\mathcal{C}$ such that $P \in \mathcal{P}$ (i.e, for some $i \in I$, $T_i[P] = P[P] = P$), and $\mathcal{V} = \{H_j : P \longrightarrow S_j\}_{j\in J}$ a family of theory morphisms with the S_j theories in $\mathcal{C}$ of the form $S_j = (\Sigma \oplus V, \Gamma_j)$, and with each H_j the identity signature morphism, there is a theory extension $U \subseteq Ind(U)$ in $\mathcal{S}$, a formula $\overline{\Gamma}^{\mathcal{T}} \in form_V(Ind(U))$ representing the axioms Γ of P, and a representation function

$$\overline{(\_)}^{\mathcal{V}} : \coprod_{i\in I} sen(T_i[P]) \longrightarrow form_V(Ind(U)) ,^{3}$$

where $\coprod_{i\in I} sen(T_i[P])$ is the disjoint union of all the sets of sentences for all theories in $\mathcal{P}$. For $\phi \in sen(T_i[P])$ we denote by $\phi^{T_i[P]}$ the corresponding copy of ϕ in the i-th component of such a disjoint union.

[3] Note that objects in V stand for parameters when occurring in sentences in $sen(T_i[P])$, while they stand for variables, and therefore can be quantified over when occurring in formulae in $form_V(Ind(U))$.

(iii) Under the same conditions of (ii), if V can be decomposed as a coproduct $V = V_1 \oplus \ldots \oplus V_n$, then for all finite multisets of theories $\{T_1[P], \ldots, T_k[P]\}$ in $\mathcal{P}$, and all finite multisets of sentences $\{\phi_1, \ldots, \phi_k\}$ with $\phi_l \in \mathit{sen}(T_l[P])$, $1 \leq l \leq k$, if

$$\mathit{Ind}(U) \vdash_{\mathcal{S}} Q_1 V_1.\, Q_2 V_2. \cdots Q_n V_n.\, (\overline{\Gamma}^{\mathcal{T}} \implies b(\overline{\phi_1^{T_1[P]}}^{\mathcal{V}}, \ldots, \overline{\phi_k^{T_k[P]}}^{\mathcal{V}}))\,, \quad (2)$$

then

$$Q_1(K_1 \in \mathcal{V}|_{\Sigma \,\oplus\, V_1}).\, Q_2(K_2 \in \mathit{Ext}^{V_2}_{\mathcal{V}|_{\Sigma \,\oplus\, V_1 \,\oplus\, V_2}}(K_1)).\, \ldots\, Q_n(K_n \in \mathit{Ext}^{V_n}_{\mathcal{V}}(K_{n-1})).$$
$$b(\mathcal{I}(T_1[K_n]) \models_{\mathcal{L}} \mathit{sen}(K_n)(\phi_1), \ldots, \mathcal{I}(T_k[K_n]) \models_{\mathcal{L}} \mathit{sen}(K_n)(\phi_k))\,, \quad (3)$$

where $\vdash_{\mathcal{S}}$ and $\models_{\mathcal{L}}$ denote, respectively, the entailment relation in $\mathcal{S}$ and the satisfaction relation in $\mathcal{L}$, each Q_i is either $\forall$ or $\exists$, and b is a Boolean expression in k arguments such that $b(\overline{\phi_1^{T_1[P]}}^{\mathcal{V}}, \ldots, \overline{\phi_k^{T_k[P]}}^{\mathcal{V}}) \in \mathit{form}_V(\mathit{Ind}(U))$.

Although the formalization of reflected parameterized induction looks complex, the idea behind it is straightforward. Condition (3i) expresses a relationship between the framework logic $\mathcal{L}$, which is used to represent families of logics as families of parameterized theories, and a superlogic $\mathcal{S}$, which includes a theory $\mathit{Ind}(U)$, that can be used to reason about families of parameterized theories in $\mathcal{L}$. For simplicity, we require the relationship between $\mathcal{S}$ and $\mathcal{L}$ to be the superlogic relation, but one could generalize this in various directions. For example, $\mathcal{S}$ could be a superlogic in the more general sense of [54], or even just a logic $\mathcal{S}$ together with a conservative map of logics mapping $\mathcal{L}$ to $\mathcal{S}$, as in [54]. Similarly, one could investigate weaker versions of the Tarskian semantics requirement for $\mathcal{S}$.

To reason in $\mathit{Ind}(U)$ about families of parameterized theories in $\mathcal{L}$, we reflect parameterization over sentences as quantification over the $\overline{(\_)}^{\mathcal{V}}$-representation of those sentences, where $\overline{(\_)}^{\mathcal{V}}$ is the representation function required in (3ii). Here, the idea is that parameters are represented as variables in $\mathit{Ind}(U)$. Since, by the definition of theory morphisms, all theory instantiations $\beta : P \longrightarrow Q$ must be such that Q satisfies the axioms Γ of the parameter theory P, the quantification over the $\overline{(\_)}^{\mathcal{V}}$-representation of sentences is conditional on the satisfaction of the $\overline{(\_)}^{\mathcal{T}}$-representation of those axioms. Finally, (3iii) requires that $\mathit{Ind}(U)$ is a theory formalizing parameterized induction in the sense that (2) implies (3).

Note that the metatheoretic statement (1) of the deduction theorem is a special case of (3) where V is not decomposed, $Q_1 = \forall$, and b is an implication. Hence, if we can formalize this in a logic $\mathcal{L}$ meeting our requirements, we can reduce the problem of establishing (1) to establishing the corresponding instance of (2). In Section 9 we will examine these requirements in detail in the context of membership equational logic (using many-kinded first-order logic with equality as its superlogic) and in Section 10 we will give a concrete example of inductive meta-reasoning about families of parameterized theories using this form of reflected parameterized induction.

Since an unparameterized theory can be viewed as a special case of a parameterized theory, the above principle of reflected parameterized induction can be specialized to a principle by which we can reason about (finite) families of unparameterized theories. The idea is that, to reason in $\mathit{Ind}(U)$ about families of unparameterized theories in $\mathcal{L}$, we reflect sentences using their $\overline{(\_)}^{\mathcal{V}}$-representations. But, since sentences are not parameterized, we must neither quantify over parameters nor make the sentences' $\overline{(\_)}^{\mathcal{V}}$-representations conditional on the satisfaction of the axioms of a parameter theory. More formally, whenever the category **Th** of theories for $\mathcal{L}$ has an initial object $\emptyset$ such that $\emptyset \in \mathcal{C} \cap |\mathbf{Var}|$, then we can view each theory $T \in \mathcal{C}$ as a theory parameterized over $\emptyset$, i.e., as $T = T[\emptyset]$, so that for $P = \emptyset$, $\mathcal{P} = \mathcal{C}$, and $\mathcal{V} = \{1_\emptyset : \emptyset \longrightarrow \emptyset\}$, we have $T[1_\emptyset] = T$ for each $T \in \mathcal{C}$. Hence we obtain as a special case the following requirement:

3') Reflected Induction. For $\{T_1, \ldots, T_k\}$ a finite multiset of theories in $\mathcal{C}$, and $\phi_l \in \mathit{sen}(T_l)$, for $1 \leq l \leq k$, and each Boolean expression b in k arguments, such that

$$b(\overline{\phi_1^{T_1}}^{\mathcal{V}}, \ldots \overline{\phi_k^{T_k}}^{\mathcal{V}}) \in \mathit{sen}(\mathit{Ind}(U))\,,$$

if

$$\mathit{Ind}(U) \vdash_{\mathcal{S}} b(\overline{\phi_1^{T_1}}^{\mathcal{V}}, \ldots, \overline{\phi_k^{T_k}}^{\mathcal{V}})\,,$$

then

$$b(\mathcal{I}(T_1) \models_{\mathcal{L}} \phi_1, \ldots, \mathcal{I}(T_n) \models_{\mathcal{L}} \phi_n)\,.$$

In particular, for $k = 1$ and $b(\phi) = \phi$, this provides a reflective alternative to usual approaches to induction in which the theory T itself is augmented with inductive reasoning principles. Namely, for all $T \in \mathcal{C}$, and $\phi \in \mathit{sen}(T)$,

$$\text{if } \mathit{Ind}(U) \vdash_{\mathcal{S}} \overline{\phi^T}^{\mathcal{V}} \text{ then } \mathcal{I}(T) \models_{\mathcal{L}} \phi\,.$$

4 Background on Membership Equational Logic

In the following sections we will show how membership equational logic can serve as a concrete instance of our abstract requirements. In this section we review standard background material on membership equational logic and the Maude language. We postpone discussion of the reflective aspects to Section 8.

4.1 Membership Equational Logic

Membership equational logic is an expressive version of equational logic. A full account of its syntax and semantics can be found in [10, 56]. Here we define the basic notions needed in this paper.

A *signature* in membership equational logic is a triple $\Omega = (K, \Sigma, S)$, with K a set of *kinds*, Σ a K-kinded signature $\Sigma = \{\Sigma_{w,k}\}_{(w,k) \in K^* \times K}$, and $S = \{S_k\}_{k \in K}$ a pairwise disjoint K-kinded family of sets. We call S_k the set of *sorts* of kind k. The pair (K, Σ) is what is usually called a many-sorted signature of function symbols; however we call the elements of K kinds because each kind k now has a set S_k of associated *sorts*, which in the models will be interpreted as subsets of the carrier for the kind. As usual, we denote by T_Σ the K-kinded algebra of ground Σ-terms, and by $T_\Sigma(X)$ the algebra of Σ-terms on the K-kinded set of variables X.

The atomic formulae of membership equational logic are either *equations* $t = t'$, where t and t' are Σ-terms of the same kind, or *membership assertions* of the form $t : s$, where the term t has kind k and $s \in S_k$. Sentences are Horn clauses on these atomic formulae, i.e., sentences of the form

$$\forall(x_1, \ldots, x_m).\, A_1 \wedge \ldots \wedge A_n \Longrightarrow A_0 \,,$$

where each A_i is either an equation or a membership assertion, and each x_j is a K-kinded variable. A theory in membership equational logic is a pair (Ω, E), where E is a finite set of sentences in membership equational logic over the signature Ω.

For example, Figure 1 gives a set of membership equational axioms specifying minimal logic of implication (by formalizing its presentation as a Hilbert system). Here we assume that there is one kind, `Expression`, and that `SentConstant`, `Formula`, and `Theorem` are sorts belonging to this kind. These sorts formalize the notion of an expression being a sentential constant, a formula, or a theorem. Implication, $\rightarrow$, is an infix binary symbol that takes two terms of kind `Expression` and returns a term of the same kind. Finally, A, B, and C are variables of the kind `Expression`. We will consider this example in more detail in Section 7.

The proof theory of membership equational logic is developed in [10]. For the purposes of this paper it is sufficient to observe that membership equational logic is a sublogic of many-kinded first-order logic with equality (namely, the many-kinded Horn clause fragment obtained by requiring that all predicate symbols other than equality are unary) and first-order calculi can be used to establish the provability of a formula ϕ relative to a membership equational theory (Ω, E), i.e., $(\Omega, E) \vdash \phi$, for ϕ a first-order sentence in the language of Ω.

We employ standard semantic concepts from many-sorted logic. Given a signature $\Omega = (K, \Sigma, S)$, an Ω-*algebra* is a many-kinded Σ-algebra A (a K-indexed-set $\{A_k\}_{k \in K}$ together with a collection of appropriately kinded functions interpreting the function symbols in Σ) and an assignment that associates to each sort $s \in S_k$ a subset $A_s \subseteq A_k$. Hence, sorts can be thought of as unary predicates that semantically denote subsets of the appropriate kind. An algebra A and a (kind-respecting) valuation σ, assigning to variables of kind k values in A_k, satisfy an equation $t = t'$ iff $\sigma(t) = \sigma(t')$, where we overload notation by identifying σ with its unique homomorphic extension to Σ-terms. We write $A, \sigma \models t = t'$ to denote such a satisfaction. Similarly, $A, \sigma \models t : s$ holds iff $\sigma(t) \in A_s$.

Note that an Ω-algebra is nothing but a K-kinded first-order model with function symbols Σ and an alphabet of unary predicates $\{S_k\}_{k \in K}$. Therefore,

$$\forall A.\, A\,\texttt{:SentConstant} \Longrightarrow A\,\texttt{:Formula},$$
$$\forall(A,B).\, A\,\texttt{:Formula} \wedge B\,\texttt{:Formula} \Longrightarrow A{\rightarrow}B\,\texttt{:Formula},$$
$$\forall(A,B).\, A\,\texttt{:Formula} \wedge B\,\texttt{:Formula} \Longrightarrow A{\rightarrow}(B{\rightarrow}A)\,\texttt{:Theorem},$$
$$\forall(A,B,C).\, A\,\texttt{:Formula} \wedge B\,\texttt{:Formula} \wedge C\,\texttt{:Formula} \Longrightarrow (A{\rightarrow}B){\rightarrow}((A{\rightarrow}(B{\rightarrow}C)){\rightarrow}(A{\rightarrow}C))\,\texttt{:Theorem},$$
$$\forall(A,B).\, A\,\texttt{:Formula} \wedge B\,\texttt{:Formula} \wedge (A{\rightarrow}B)\,\texttt{:Theorem} \wedge A\,\texttt{:Theorem} \Longrightarrow B\,\texttt{:Theorem}$$

Figure 1: Membership equational axioms for minimal logic.

the satisfaction relation can be extended to Horn and to first-order formulae ϕ over these atomic formulae in the standard way. We write $A \models \phi$ when the formula ϕ is satisfied for all valuations σ, and then say that A is a model of ϕ. Similarly, a theory (Ω, E) in membership equational logic is simply a Horn theory for the associated signature, when Ω is viewed as a first-order K-kinded signature. As usual, for ϕ a first-order sentence in the language of Ω, we write $(\Omega, E) \models \phi$ when all the models of the set E of sentences are also models of ϕ.

Theories in membership equational logic have initial models [56]. Given a theory (Ω, E), we denote its initial model by $T_{\Omega/E}$. In particular, when $E = \emptyset$ we obtain the term algebra T_Ω, and for X a K-kinded set of variables the free algebra $T_\Omega(X)$. This provides the basis for reasoning by induction, as is explained in Section 6. We write $(\Omega, E) \models\!\!\simeq \phi$ to denote that the initial model of the membership equational theory (Ω, E) is also a model of ϕ, that is, that the satisfaction relation $T_{\Omega/E} \models \phi$ holds.

4.2 The Maude System

The Maude system [14, 18] implements rewriting logic [55, 50] and membership equational logic, and has been designed with the explicit aims of supporting executable specification and reflective computation.[4] Theories are specified in Maude by modules of two kinds: *functional modules* and *system modules*. Maude's functional modules are theories in membership equational logic. Equations in Maude's functional modules are assumed to be Church-Rosser and terminating; they are executed by the Maude rewrite engine according to the rewriting techniques and operational semantics developed in [10]. Maude's

[4]Rewriting logic is a simple logic whose sentences are sequents of the form $t \longrightarrow t'$, with t and t' Ω-terms on a given signature Ω. Theories in rewriting logic are triples (Ω, E, R), with Ω a signature of function symbols, E a set of Ω-equations, and R a collection of (possibly conditional) Ω-rewrite rules. The inference rules of rewriting logic allow the derivation of all rewrites possible in a given theory. Since a rewrite theory (Ω, E, R) has an underlying equational theory (Ω, E), rewriting logic is parameterized by the choice of the equational logic. An attractive choice in terms of expressiveness is membership equational logic. This is the choice made in Maude.

```
fmod MINIMAL is
kind Expression[SentConstant Formula Theorem] .
op _→_ : Expression Expression -> Expression .
vars A B C : Expression .
cmb A : Formula if A : SentConstant .
cmb A → (B → A):  Theorem if A : Formula and B : Formula .
cmb (A → B) → ((A →(B → C)) → (A → C)): Theorem
    if A : Formula and B : Formula and C : Formula .
cmb B : Theorem if A : Formula and B : Formula
    and A : Theorem and (A→B) : Theorem .
endfm
```

Figure 2: The theory MINIMAL.

system modules are rewrite theories. The rules in a system module are not necessarily Church-Rosser or terminating.

The semantics of a functional (respectively, system) module is *initial*, i.e., such a module denotes the *initial model* in membership equational logic (respectively, rewriting logic) of the specified theory. The syntax for functional modules is of the form `fmod` (Ω, E) `endfm`, with (Ω, E) a membership equational theory meeting the requirements mentioned above. Figure 2 gives the representation in Maude of the membership equational theory presented in Figure 1.[5] The syntax is mostly self-explanatory: `kind` introduces kinds along with their sorts; `op` introduces symbols in the many-kinded signature, where underscores indicate mixfix notation, e.g., $\_\!\longrightarrow\!\_$ is an infix operator; `vars` introduces variables with their kinds; and finally, `mb` and `cmb` precede, respectively, membership axioms and conditional membership axioms in membership equational logic. To simplify the exposition, here and in the rest of the paper we assume that an infinite supply of terms having sort `SentConstant` has already been specified in a previous subtheory. Since their particular representation (as identifiers, numbers or whatever) is immaterial for our purposes, we omit the details.

For convenience, we will henceforth use Maude's syntax to present membership equational theories.

5 Parameterized Membership Equational Theories

In Section 2.2 we used the pushout construction to give a general definition of parameterized theories and their instantiations. Now we specialize this con-

[5] We have taken slight liberties with the representation. Namely, LaTeX symbols are used instead of ASCII characters to improve readability. For increased clarity we have also explicitly named kinds. (The Maude system can infer kind information automatically; kinds are not named but denoted using the name of their sorts enclosed in square brackets. However, we will use Maude's convention for kinds later in the paper.)

struction to define, for each appropriate theory P, a class $\mathcal{P}_P$ of membership equational theories parameterized by P and a class $\mathcal{V}_P$ of theory morphisms that instantiate parameterized theories in $\mathcal{P}_P$. We will use these classes in Section 9 to define reflected parameterized induction for membership equational logic.

Recall that a parameterized theory is given by a theory map from a parameter theory P to a theory T. We consider parameter theories P of the form

$$P = (\Omega \oplus V, E \cup Mb(V)) .$$

P's signature is built from a finite signature $\Omega = (K, \Sigma, S)$ and a finite signature of parameters $V = (K, \{V_{\lambda,k}\}_{k \in K}, S)$ that consists of a pairwise disjoint K-kinded family of constants, which satisfies that, for all $k \in K$, $\Sigma_{\lambda,k} \cap V_{\lambda,k} = \emptyset$.[6] P's axioms consist of a finite set of sentences E over the signature Ω, and a finite set of membership axioms $Mb(V)$ that specify a sort for each v in V. For each $T[P] \in \mathcal{P}_P$ the theory map $P \longrightarrow T$ will be an inclusion. Specifically, the parameterized theories $T[P] \in \mathcal{P}_P$ have the form

$$T[P] = (\Omega \oplus V, E \cup G \cup Mb(V)) ,$$

where G is a finite set of additional axioms (which extend P's axioms).

Recall also that an instance of a parameterized theory is given by a theory map from P to another theory Q. We define $Inst(P)$ as the class of theories

$$Q = (\Omega \oplus V, E \cup Eq(V)) ,$$

where $Eq(V)$ is a finite set of equations of the form $\{v_i = t_i \mid v_i \in V\}$ such that $Eq(V)$ assigns to each constant $v_i \in V$ a *ground* term $t_i \in T_\Omega$, such that $Q \vdash t_i : s_i$, where s_i is the sort assigned to v_i in $Mb(V)$. We then define $\mathcal{V}_P$ as the class of theory morphisms $\beta : P \longrightarrow Q$, such that $Q \in Inst(P)$ and β is the signature identity morphism. Note that the set $\mathcal{V}_P$ is in bijective correspondence with the set $Inst(P)$.

The above defines a notion of instantiation for parameterized theories that, for all $T[P] \in \mathcal{P}_P$ and $\beta \in \mathcal{V}_P$, specializes the pushout construction to

$$\begin{array}{ccc} T[P] & \longrightarrow & T[\beta] \\ \uparrow & & \uparrow \\ P & \xrightarrow[\beta]{} & Q \end{array}$$

where $T[\beta] = (\Omega \oplus V, E \cup G \cup Eq(V))$.

Figures 3 and 4 provide examples using Maude syntax. Figure 3 gives a parameter theory `MINIMAL-P`. Here the signature of parameters V consists of the constants AA and BB of kind `Expression`, and $Mb(V)$ consists of the membership axioms AA:`Formula` and BB:`Formula`. Figure 4 gives a parameterized theory `MINIMAL-DT[MINIMAL-P]` in $\mathcal{P}_{\texttt{MINIMAL-P}}$, which extends `MINIMAL-P` with axioms G, in this case the extra (parametric) axiom AA:`Theorem`.

```
fmod MINIMAL-P is
kind Expression[SentConstant Formula Theorem] .
op AA : -> Expression .
op BB : -> Expression .
op _⟶_ : Expression Expression -> Expression .
vars A B C : Expression .
cmb A : Formula if A : SentConstant .
cmb (A ⟶ B): Formula if A : Formula and B : Formula .
mb AA: Formula .
mb BB: Formula .
cmb A ⟶ (B ⟶ A): Theorem if A : Formula and B : Formula .
cmb (A ⟶ B) ⟶ ((A ⟶(B ⟶ C)) ⟶ (A ⟶ C)): Theorem
    if A : Formula and B : Formula and C : Formula .
cmb B : Theorem if A : Formula and B : Formula
    and A : Theorem and (A ⟶ B): Theorem .
endfm
```

Figure 3: The parameter theory MINIMAL-P.

```
fmod MINIMAL-DT[MINIMAL-P] is
including MINIMAL-P .
mb AA:Theorem .
endfm
```

Figure 4: The parameterized theory MINIMAL-DT[MINIMAL-P].

One of the key ideas behind our use of theory morphisms is the following. Although β is the identity morphism on signatures, it identifies terms in Q, and hence in $T[\beta]$, by adding equations of the form $v_i = t_i$. This has an effect equivalent to mapping constants to terms. More formally, suppose $T[P] \in \mathcal{P}_P$, $P = (\Omega \oplus V, E \cup Mb(V))$, and $\beta \in \mathcal{V}_P$, $\beta : P \longrightarrow Q$. For all terms $t \in T_{\Omega \oplus V}(X)$, considering the equations $Eq(V) = \{v_i = t_i \mid v_i \in V\}$ in Q, we denote by t_β the term in $T_\Omega(X)$ that results from replacing each v_i (if any) in t with the ground term t_i. Note first that for all such t,

$$(t_\beta)_\beta = t_\beta \,. \tag{4}$$

We can extend this notion of term replacement to atomic formulae in the standard way: $(t : s)_\beta \triangleq (t)_\beta : s$ and $(t = t')_\beta \triangleq t_\beta = t'_\beta$. Note then that for all atomic formulae ϕ over the signature of $T[P]$, $\phi = sen(\beta)(\phi)$. But, whenever some v_i occurs in ϕ, $sen(\beta)(\phi) \neq \phi_\beta$. However, in all cases, due to the equations

[6]Note that $\Omega \oplus V$ is not really a coproduct but rather a union, since the kinds of the constants in V are already kinds in Ω, and we do not make disjoint copies of those kinds.

$Eq(V)$,

$$T[\beta] \vdash \phi \iff T[\beta] \models \phi \iff T[\beta] \models sen(\beta)(\phi) \iff T[\beta] \models \phi_\beta \iff T[\beta] \vdash \phi_\beta \,. \tag{5}$$

Finally, note that for all parameter theories $P = (\Omega \oplus V, E \cup Mb(V))$, there is a trivial extension $P[P]$ of P, i.e. where $P[P] = P$.

6 Induction Principles for Membership Equational Theories

Given that membership equational logic *is* a sublogic of Horn logic with equality (indeed, they can be shown to be equivalent [56]) it follows immediately that any theory (Ω, E) has a unique (up to isomorphism) initial model [35]. Hence our second abstract requirement, initiality, is fulfilled. The following is an induction principle for reasoning about properties of sorts, with respect to the initial model. As syntactic sugar, we shall write $\forall x : s.\, \phi(x)$ as shorthand for the formula $\forall x.\, x\!:\!s \Longrightarrow \phi(x)$, for x a variable of kind k and $s \in S_k$. Moreover, for the formula $x\!:\!s \Longrightarrow \phi(x)$, we will say that "$x$ is of sort s (in ϕ)."

Definition 7 (Induction over sort definitions) *Let $T = (\Omega, E)$ be a theory in membership equational logic and let s be a sort in some S_k. Let $C_{[T,s]} = \{C_1, \ldots, C_n\}$ be those sentences in E that specify the sort $s \in S_k$ for k a kind, i.e., those C_i of the form*

$$\forall (x_1, \ldots, x_{p_i}).\, A_1 \wedge \ldots \wedge A_{q_i} \Longrightarrow A_0 \,, \tag{6}$$

where, for some term t of kind k, A_0 is $t\!:\!s$.

For τ a first-order formula with free variable x of sort s over the signature Ω, an induction principle for (Ω, E), with respect to $x\!:\!s$ and $\tau(x)$, *is the formula*

$$\psi_1 \wedge \ldots \wedge \psi_n \Longrightarrow \forall x\!:\!s.\, \tau(x) \,, \tag{7}$$

where, for $1 \leq i \leq n$ and C_i of the form (6), ψ_i is

$$\forall (x_1, \ldots, x_{p_i}).\, [A_1]_\tau \wedge \ldots \wedge [A_{q_i}]_\tau \Longrightarrow [A_0]_\tau \tag{8}$$

and, for $0 \leq j \leq q_i$,

$$[A_j]_\tau \triangleq \begin{cases} \tau(u) & \textit{if } A_j = u\!:\!s\textit{, for } u \textit{ of kind } k \\ A_j & \textit{otherwise}\,. \end{cases}$$

For each membership equational theory (Ω, E), (7) defines an induction schema in many-kinded first-order logic over the signature Ω. As we will see on examples, for C_i, each $[A_j]_\tau$ contributes to either an induction hypothesis (for $j > 0$) or the induction conclusion (for $j = 0$) to the ith premise. Note that for $q_i = 0$, the nullary conjunction in the antecedent of (8) is *true* and the implication can be replaced with the succedent.

In the initial model of a membership equational theory, sorts are interpreted as the smallest sets satisfying the axioms in the theory, and equality is interpreted as the smallest congruence satisfying those axioms. Alternatively, the sets interpreting sorts can be characterized as being inductively generated in stages. This corresponds to the fixedpoint characterization of the least Herbrand model of a collection of Horn clauses [79], and the induction principle we have given formalizes induction over the stages in which the set is inductively defined [2]. By induction over the stages of the inductive definition of a sort s, which amounts to an induction over the proof that some ground term of kind k is of sort s, we can establish that reasoning in the membership equational theory (Ω, E), augmented by (7), is sound.

Theorem 1 (Soundness) *Let (Ω, E) be a membership equational theory. If $ind(\Omega, E) \vdash \tau$, then $(\Omega, E) \models_{\simeq} \tau$, where $ind(\Omega, E)$ is the theory (Ω, E) extended with the induction schema* (7).

As an example, consider the membership equational theory for minimal logic previously given in Figure 1. The above definition gives rise to the following induction principles over the sorts `Formula` and `Theorem` with respect to this theory. For induction over formulae, the corresponding instance of (7) is

$$[\forall A.\,(A\!:\!\texttt{SentConstant} \Longrightarrow \tau(A)) \wedge \ \forall (A,B).(\tau(A) \wedge \tau(B) \Longrightarrow \tau(A{\rightarrow}B))]$$
$$\Longrightarrow$$
$$\forall A\!:\!\texttt{Formula}.\,\tau(A).$$

In the case of `Theorem`, the corresponding induction schema is

$$[\forall (A,B).\,(A\!:\!\texttt{Formula} \wedge \ B\!:\!\texttt{Formula} \Longrightarrow \tau(A{\rightarrow}(B{\rightarrow}A))) \wedge$$
$$\forall (A,B,C).\,(A\!:\!\texttt{Formula} \wedge B\!:\!\texttt{Formula} \ \wedge C\!:\!\texttt{Formula}$$
$$\Longrightarrow \tau((A{\rightarrow}B){\rightarrow}(A{\rightarrow}(B{\rightarrow}C)){\rightarrow}(A{\rightarrow}C))) \wedge$$
$$\forall (A,B).\,(A\!:\!\texttt{Formula} \wedge B\!:\!\texttt{Formula} \wedge \tau(A{\rightarrow}B) \wedge \tau(A) \Longrightarrow \tau(B))]$$
$$\Longrightarrow$$
$$\forall A\!:\!\texttt{Theorem}.\,\tau(A)\,,$$

which formalizes induction over the structure of proofs in minimal logic.

7 Membership Equational Logic as a Logical Framework

We can use membership equational logic to represent theoremhood in a logic as the property that some terms representing formulae have a given sort in a membership equational theory (Ω, E). Conditional membership axioms then directly support the representation of rules as schemas, which is typically used in presenting logics and formal systems. Note that membership equational logic is noncommittal about the structure and properties of the formulae represented by Ω-terms. They are user-definable as an algebraic data type satisfying equational axioms.[7]

Similarly, we can represent theoremhood in a parameterized family of logics as the property that some formula-representing terms have a given sort in a parameterized membership equational theory.[8] The ability to represent parameterized families of logics is important for using membership equational logic also as a metalogical framework, and we will give an example of this in the experimental work reported on in Section 10.

We shall now illustrate the above ideas, using minimal logic as a running example. Representing minimal logic in membership equational logic entails defining a theory T that conservatively represents minimal logic's theoremhood. The formulae of minimal logic correspond to members of the set built from the binary connective $\rightarrow$ (written infix, associating to the right) and sentential constants. Theorems correspond to members of a second set, and are either instances of the standard Hilbert axiom schemata K,

$$A \rightarrow B \rightarrow A\,,$$

or S,

$$(A \rightarrow B) \rightarrow (A \rightarrow B \rightarrow C) \rightarrow (A \rightarrow C)\,,$$

or are generated by applying the rule *modus ponens*,

$$\frac{A \quad A \rightarrow B}{B}\,.$$

Of course, we want our representation to preserve the inductive nature of the set of theorems and proofs in minimal logic. The module `MINIMAL`, presented previously in Figure 2, illustrates one way of representing minimal logic in membership equational logic using this idea. A formula A is a theorem in minimal logic if and only if A is a term of sort `Theorem` in `MINIMAL`.

The module `MINIMAL-PF` in Figure 5 illustrates another alternative. This module represents both the *theorems* and *proofs* in a Hilbert system for minimal logic as members of a sort `Proof`. The axioms and rules manipulate judgments consisting of a formula and its proof tree and having the form $[PS]/A$, with A a formula and $[PS]$ a list of judgements $[PS_1]/A_1, \ldots, [PS_n]/A_n$ containing

[7]This ecumenical neutrality is inherited by rewriting logic and has been applied effectively in its uses as a logical framework. In [16, 49, 48, 57, 76, 77, 50], many examples of logic representations in membership equational logic and in rewriting logic are given, including first-order linear logic, sequent presentations of modal and propositional logics, Horn logic with equality, the lambda calculus, and higher-order pure type systems, among others. In all such examples, representations are direct (syntactically similar to their textbook counterparts).

[8]A sort in a parameterized membership equational theory can be used to represent theoremhood in a family of logics if and only if (1) there is a one–to–one correspondence between logics in the family and instances of the parameterized membership equational theory, and (2) this correspondence is such that theoremhood in a logic in the family can be represented as membership in this sort in the corresponding instance of the parameterized membership equational theory.

```
fmod MINIMAL-PF is
kind Expression[SentConstant Formula] .
kind Proof?[Proof Proofs] .
op _→_ : Expression Expression -> Expression .
op nil : -> Proof? .
op _,_ : Proof? Proof? -> Proof? [assoc id: nil] .
op [_]/_ : Proof? Expression -> Proof? .
vars A B C : Expression .
vars PS QS : Proof? .
cmb A : Formula if A : SentConstants .
cmb (A → B) : Formula if A : Formula and B : Formula .
mb nil : Proofs .
cmb PS : Proofs if PS : Proof .
cmb (PS, QS): Proofs if PS : Proofs and QS : Proofs .
cmb [nil]/A → (B → A) : Proof
    if A : Formula and B : Formula .
cmb [nil]/(A → B) → ((A →(B → C)) → (A → C)) : Proof
    if A : Formula and B : Formula and C : Formula .
cmb [([PS]/A),([QS]/(A → B))]/B : Proof
    if PS : Proofs and QS : Proofs and A : Formula and B : Formula
    and [PS]/A : Proof and  [QS]/(A → B) : Proof .
endfm
```

Figure 5: The module MINIMAL-PF.

antecedent formulae $A_1, \ldots, A_n$ establishing A, and their corresponding proof subtrees $PS_1, \ldots, PS_n$. Lists of judgements are built with the binary constructor _,_ which is declared associative and has nil as its identity element. No subproofs are needed to prove an instance of the two axioms. The rule for *modus ponens* constructs a proof of B from a proof of A and a proof of $A{\rightarrow}B$.

Which representation one chooses depends, in part, on how it is to be used. MINIMAL represents theorems as members of an inductive set, and is well-suited for metatheoretic reasoning about theoremhood. MINIMAL-PF represents minimal logic proofs themselves as formal objects and can be used, for example, to prove metatheorems about the structure of minimal logic proofs.

Finally, to illustrate the use of parameterized modules, consider the task of representing not just minimal logic, but the family of logics that consists of all extensions of minimal logic with a new axiom. A solution to this was given previously in Figure 4, namely, the parameterized theory MINIMAL-DT[MINIMAL-P]. This parameterized theory represents the family that consists of all extensions of minimal logic with a new axiom in the following sense: a formula B is a theorem in minimal logic extended with a new axiom A if and only if B is a term of sort Theorem in MINIMAL-DT[β], where $(AA)_\beta = A$ and $(BB)_\beta = B$.

8 Reflection in Membership Equational Logic

A universal theory MB-META for membership equational logic is introduced in [22], along with a representation function ($\overline{\_ \vdash \_}$) that encodes pairs consisting of a membership equational theory T and a sentence in it, as sentences in MB-META.[9] Thus, membership equational logic is a reflective logic in the precise sense of Definition 6, and our first abstract requirement for reflective metalogical frameworks, namely, reflection, is fulfilled.

The signature of MB-META contains constructors to represent operations, variables, terms, kinds, sorts, signatures, axioms, and theories. In particular, the signature of MB-META includes the sorts Theory, Signature, AxiomSet, Equation, MembAxiom, Term, and Sort, for terms representing, respectively, theories, signatures, sets of axioms, equations, membership axioms, terms, and sorts. In addition, it contains three Boolean operations

```
op wft_in_ : [Term] [Signature] -> [Bool] .
op _:_in_ : [Term] [Sort] [Theory] -> [Bool] .
op _=_in_ : [Term] [Term] [Theory] -> [Bool] .
```

to represent, respectively, that a term is well-formed in a membership equational signature, and that a membership assertion or an equation hold in a membership equational theory. Note that here, and in what follows, we use Maude's convention for naming kinds: kinds are not named but are denoted using the name of one of their sorts enclosed in square brackets.

The representation function ($\overline{\_ \vdash \_}$) is defined in [22] as follows: for all membership equational theories T, and atomic formulae ϕ over the signature of T,

$$\overline{T \vdash \phi} \triangleq \begin{cases} (\overline{t}\ \texttt{:}\ \overline{s}\ \texttt{in}\ \overline{T}) = \texttt{true} & \text{if } \phi = (t : s) \\ (\overline{t}\ \texttt{=}\ \overline{t'}\ \texttt{in}\ \overline{T}) = \texttt{true} & \text{if } \phi = (t = t')\,, \end{cases}$$

where $\overline{(\_)}$ is a representation function defined recursively over theories, signatures, axioms, and so on. Under this representation function, a membership equational theory $T = (\Omega, E)$, with $E = (\phi_1, \ldots, \phi_n)$, is represented in MB-META by the ground term $\overline{T} = (\overline{\Omega}, \overline{E})$ of sort Theory, where $\overline{\Omega}$ is a term of sort Signature and $\overline{E} = (\overline{\phi}_1 \ldots \overline{\phi}_n)$ is a term of sort AxiomSet[10]; a term t in a membership equational theory T is represented in MB-META by a ground term $\overline{t}$ of sort Term; a sort s is represented by a ground term $\overline{s}$ of sort Sort; an equation $t = t'$ is represented by a ground term (**eq** $\overline{t}$ **=** $\overline{t'}$ **.**) of sort Equation; and, finally, a membership axiom $t : s$ is represented by a ground term (**mb** $\overline{t}$ **:** $\overline{s}$**.**) of sort MembAxiom. Conditional equations and memberships are represented similarly.

The following propositions state the main properties of MB-META as a universal theory:

[9] To avoid unnecessary technical details, we present here a simplified version of the reflective results provided in [22]. Also, for notational convenience, we name differently the universal theory and we use a slightly different syntax for some of the metalevel operations involved in its definition.

[10] Note that although E is a set, its presentation in Maude is as an ordered sequence (of text). Here, and elsewhere, we will assume we can order sets based on their presentation.

Proposition 1 *For all finitely presentable membership equational theories with nonempty kinds* $T = (\Omega, E)$*, with* $\Omega = (K, \Sigma, S)$*, terms* t *in* $(T_\Omega)_k$*, and sorts* s *in* S_k*,*

$$T \vdash t : s \iff \texttt{MB-META} \vdash (\bar{t}\ \textbf{:}\ \bar{s}\ \textbf{in}\ \overline{T}) = \texttt{true}.$$

Similarly, for all terms t*,* t' *in* $T_\Omega(X)$*,*

$$T \vdash t = t' \iff \texttt{MB-META} \vdash (\bar{t}\ \textbf{=}\ \overline{t'}\ \textbf{in}\ \overline{T}) = \texttt{true}\,.$$

Proposition 2 *For all finitely presentable membership equational signatures* Ω *and terms* t *in* T_Ω

$$\texttt{MB-META} \vdash (\textbf{wft}\ \bar{t}\ \textbf{in}\ \overline{\Omega}) = \texttt{true}\,.$$

9 Reflected Parameterized Induction

In this section we show how the induction principles introduced in Section 6 for reasoning in membership equational theories can be uniformly reflected for reasoning, at the metalevel, about families of theories. In doing so, we show how our third abstract requirement for a reflective metalogical framework, namely, reflected parameterized induction, is fulfilled.

We restrict ourselves to classes of the form $\mathcal{P}_P$ of membership equational theories parameterized by P and their associated classes $\mathcal{V}_P$ of theory morphisms as defined in Section 5. First, in Section 9.1, we define a representation function for parameterized theories in $\mathcal{P}_P$. Then, in Section 9.2, we define a representation function for atomic formulae over parameterized membership equational theories. Afterwards, in Section 9.3, we define a representation function for axioms of parameter theories. Finally, in Section 9.4, we show that parameterized induction can be reflected in a theory extension of MB-META.

To simplify our presentation, we omit the full definitions of the representation functions that we use to reflect parameterized induction principles and instead we just state the properties that they satisfy. ([13, 21, 22] can be consulted for a full definition of analogous representation functions.) Despite this simplification, the material presented here is admittedly still rather technical, as a number of such functions are required. As an aid for the reader, Figure 6 provides an overview of the notation used in this section and in the appendix. We also summarize in this figure the representation functions introduced in Section 8.

9.1 Representing Parameterized Theories

We first introduce some preliminary notation. For $P = (\Omega \oplus V, E \cup Mb(V))$ and $T[P]$ a parameterized membership equational theory in $\mathcal{P}_P$, and for all terms $t \in T_{\Omega \oplus V}(X)$, we will denote by $\bar{t}^{[V,X]}$ the reflective representation of t defined in Section 8, except that now parameters $v \in V$ and variables $x \in X$ are replaced by (meta-)variables v and x of the kind `[Term]`.[11] For t a ground term, we

[11] The key difference between $\bar{t}$ and $\bar{t}^{[V,X]}$ is that $\bar{t}$ is a *ground term* of sort `Term`, whereas $\bar{t}^{[V,X]}$ is a term with (meta-)variables of the kind `[Term]`.

$\overline{T}$	The representation function for membership equational theories T; it returns a ground term in `MB-META` of sort `Theory`.
$\overline{T[P]}^{\mathcal{P}}$	The representation function for parameterized theories $T[P]$ in $\mathcal{P}_P$; it returns a term in `MB-META` of the kind `[Theory]` containing (meta-)variables of the kind `[Term]` that correspond to the parameters in P.
$\overline{\phi}^{\mathcal{B}(T[P],X)}$, $\overline{\phi_\beta}^{\mathcal{B}(T[\beta],X)}$	The representation functions for atomic formulae ϕ (respectively ϕ_β, for β a theory morphism in $\mathcal{V}_P$) over parameterized theories $T[P]$ in $\mathcal{P}_P$ (respectively over theories $T[\beta]$), which may contain variables in X and parameters in P (respectively variables in X); it returns an equality in `MB-META` between the constant `true` and a term of the kind `[Bool]` containing (meta-)variables of the kind `[Term]` that correspond to the variables in X and/or the parameters in P occurring in ϕ (respectively in ϕ_β).
$\overline{E \cup Mb(V)}^{\mathcal{C}(P)}$	The representation function for axioms of parameter theories P in $\mathcal{P}_P$; it returns, essentially, the conjunction of the equalities in `MB-META` that result from representing the atomic formulae in $Mb(V)$.

The representation functions below are used to define the functions above

$\overline{\Omega}$, $\overline{E}$, $\overline{t}$, $\overline{s}$	The representation functions for signatures Ω, sets of axioms E, terms t (maybe containing parameters and/or variables), and sorts s of membership equational theories; they returns ground terms in `MB-META` of sort, respectively, `Signature`, `AxiomSet`, `Term`, and `Sort`.
$\overline{t}^{[V,X]}$, $\overline{t}^{[V]}$, $\overline{t}^{[X]}$	The representation functions for terms t that may contain parameters in V and/or variables in X; they return terms in `MB-META` of the kind `[Term]` containing (meta-)variables of the kind `[Term]` that correspond to the parameters in V and/or variables in X occurring in t.
$\overline{t}^{\mathcal{W}(\Omega,V)}$	The representation function for assertions of well-formedness relative to Ω for terms t in membership equational signatures $(\Omega \oplus V)$; it returns an equality in `MB-META` between the constant `true` and a term of the kind `[Bool]` containing (meta-)variables of the kind `[Term]` that correspond to the parameters in P occurring in t.

The representation functions below are used to state the propositions satisfied by the functions above

$\overline{V'}^{[V]}$	The representation function for sets of parameters $V' \subseteq V$; it returns a set of (meta-)variables of the kind `[Term]` in `MB-META`.
$\overline{\beta}$	The representation function for theory morphisms β in $\mathcal{V}_P$, $\beta : P \longrightarrow Q$, $P = (\Omega \oplus V, E \cup Mb(V))$; it returns a ground substitution that assigns to each (meta-)variable in $\overline{V}^{[V]}$ the terms of sort `Term` in `MB-META` that result from representing the term to which the corresponding parameter is instantiated by β.

Figure 6: Representation functions.

shall simply write $\overline{t}^{[V]}$. Similarly, if $t \in T_{\Omega}(X)$ we shall write $\overline{t}^{[X]}$. In addition, we will denote by $\overline{V}^{[V]}$ the set $\overline{V}^{[V]} \triangleq \{\overline{v}^{[V]} \mid v \in V\}$. Finally, for all theory morphisms $\beta \in \mathcal{V}_P$, $\beta : P \longrightarrow Q$, we will denote by $\overline{\beta}$ the ground substitution $\overline{\beta} : \overline{V}^{[V]} \longrightarrow$ `[Term]`, defined as follows: $\overline{\beta}(\overline{v}^{[V]}) \triangleq \overline{t}$, if $v \in V$ and $v_{\beta} = t$.

Proposition 3 *For* $P = (\Omega \oplus V, E \cup Mb(V))$ *a parameter theory in* $\mathcal{P}_P$*, for all parameterized membership equational theories* $T[P] \in \mathcal{P}_P$*, theory morphisms* $\beta \in \mathcal{V}_P$*,* $\beta : P \longrightarrow Q$*, and terms* $t \in T_{\Omega \oplus V}(X)$*, it holds that*

$$\overline{\beta}(\overline{t}^{[V,X]}) = \overline{t_{\beta}}^{[X]} .$$

Proof By structural induction on t. □

We now define a *generic* representation function $\overline{(\_)}^{\mathcal{P}}$ for parameterized membership equational theories. For $P = (\Omega \oplus V, E \cup Mb(V))$, $T[P] \in \mathcal{P}_P$, with $T[P] = (\Omega \oplus V, E \cup G \cup Mb(V))$, and $Mb(V) = \{v_1 : s_1, \ldots, v_n : s_n\}$,

$$\overline{T[P]}^{\mathcal{P}} \triangleq (\overline{\Omega \oplus V}, \overline{E}\ \overline{G}\ (\texttt{eq}\ \overline{v_1} = \overline{v_1}^{[V]}\ \texttt{.} \cdots \texttt{eq}\ \overline{v_n} = \overline{v_n}^{[V]}\ \texttt{.})) .$$

Note that the class $\mathcal{P}_P$ is itself "parameterized" by the choice of P. However, the definition of the representation function $\overline{(\_)}^{\mathcal{P}}$ does not depend on this choice, and, therefore, the representation function $\overline{(\_)}^{\mathcal{P}}$ is generic in P. Note also that $\overline{T[P]}^{\mathcal{P}}$ and $\overline{T[P]}$ are not equal: the former contains (meta-)variables while the latter is a ground term of sort `Theory`. This difference is due to the fact that, in the former the membership axioms for parameters, $Mb(V)$, are not represented but instead they are first transformed into identity equations between parameters that are treated in a special way: the right-hand side of each equation is represented by a (meta-)variable of the kind `[Term]`, with the same name as the corresponding parameter, while the left-hand side is represented as usual.

As an example, consider the parameterized theory `MINIMAL-DT[MINIMAL-P]` given in Figure 4. Note that the terms of sort `Theory` that result from applying the representation functions $\overline{(\_)}$ and $\overline{(\_)}^{\mathcal{P}}$ to this theory are identical except for the fact that the former has as subterms the terms of sort `MembAxiom`

$$\texttt{mb}\ \overline{AA}\ \texttt{:}\ \overline{\texttt{Formula}}\ \texttt{.}\ ,\ \texttt{mb}\ \overline{BB}\ \texttt{:}\ \overline{\texttt{Formula}}\ \texttt{.}\ , \tag{9}$$

which represent the membership axioms for the parameters AA and BB, while the latter has instead the terms of sort `Equation`

$$\texttt{eq}\ \overline{AA} = AA\ \texttt{.}\ ,\ \texttt{eq}\ \overline{BB} = BB\ \texttt{.}\ , \tag{10}$$

where the second occurrence of AA and BB in the terms in (10) are (meta-)variables of the kind `[Term]`.

Proposition 4 *For* $T[P] = (\Omega \oplus V, E \cup G \cup Mb(V))$ *a parameterized theory in* $\mathcal{P}_P$*, for all theory morphisms* $\beta \in \mathcal{V}_P$*,* $\beta : P \longrightarrow Q$*, it holds that*

$$\overline{\beta}(\overline{T[P]}^{\mathcal{P}}) = \overline{T[\beta]} .$$

Proof The only variables in $\overline{T[P]}^{\mathcal{P}}$ appear in its subterms ($\textbf{eq}\ \overline{v}_i = \overline{v}_i^{[V]}\ \textbf{.}$). Therefore, by the definition of substitution application, we have

$$\overline{\beta}(\overline{T[P]}^{\mathcal{P}}) = (\overline{\Omega \oplus V}, \overline{E}\ \overline{G}\ (\textbf{eq}\ \overline{v}_1 = \overline{t}_1\ \textbf{.} \cdots \textbf{eq}\ \overline{v}_1 = \overline{t}_1\ \textbf{.}))\,,$$

which, by the definition of $T[\beta]$, yields the desired result. □

9.2 Representing Atomic Formulae

We now define a *generic* representation function $\overline{(\_)}^{\mathcal{B}(\_,\_)}$ for atomic formulae over parameterized membership equational theories. This function constitutes the representation function $\overline{(\_)}^{\mathcal{V}}$ mentioned in part (ii) of our third abstract requirement for classes of the form $\mathcal{P}_P$ of parameterized theories. For $P = (\Omega \oplus V, E \cup Mb(V))$, for all parameterized theories $T[P] \in \mathcal{P}_P$, with $\Omega = (K, \Sigma, S)$, and membership assertions $t\!:\!s$, with t in $T_{\Omega\oplus V}(X)$ and s in some S_k,

$$\overline{t\!:\!s}^{\mathcal{B}(T[P],X)} \triangleq (\overline{t}^{[V,X]}\ \textbf{:}\ \overline{s}\ \texttt{in}\ \overline{T[P]}^{\mathcal{P}}) = \texttt{true}\,.$$

Similarly, for all equalities $t = t'$, with t, t' in $T_{\Omega\oplus V}(X)$,

$$\overline{t = t'}^{\mathcal{B}(T[P],X)} \triangleq (\overline{t}^{[V,X]}\ \textbf{=}\ \overline{t'}^{[V,X]}\ \texttt{in}\ \overline{T[P]}^{\mathcal{P}}) = \texttt{true}\,.$$

Note that the representation function $\overline{(\_)}^{\mathcal{B}(\_,\_)}$ is generic in P: it is defined for all atomic formulae over all parameterized theories $T[P] \in \mathcal{P}_P$, independently of the choice of P. This representation function satisfies the following proposition, where, as explained in Section 4.1, $\models\!\!\!\simeq$ denotes satisfaction in the initial model of the given theory.

Proposition 5 *For $P = (\Omega \oplus V, E \cup Mb(V))$ a parameter theory in $\mathcal{P}_P$, for all parameterized membership equational theories $T[P] \in \mathcal{P}_P$, atomic formulae ϕ over the signature of $T[P]$, and theory morphisms $\beta \in \mathcal{V}_P$, $\beta : P \longrightarrow Q$, it holds that*

$$\texttt{MB-META} \models\!\!\!\simeq \overline{\beta}(\overline{\phi}^{\mathcal{B}(T[P],\emptyset)}) \Longleftrightarrow T[\beta] \vdash \phi_\beta\,.$$

Proof Let $\phi = t\!:\!s$ (the proof is analogous for $\phi = (t = t')$). First, notice that by the definition of substitution application and Propositions 3 and 4,

$$\begin{aligned}
\overline{\beta}(\overline{t\!:\!s}^{\mathcal{B}(T[P],\emptyset)}) &= (\overline{\beta}(\overline{t}^{[V]}\ \textbf{:}\ \overline{s}\ \texttt{in}\ \overline{T[P]}^{\mathcal{P}}) = \texttt{true}) \\
&= (\overline{\beta}(\overline{t}^{[V]})\ \textbf{:}\ \overline{s}\ \texttt{in}\ \overline{\beta}(\overline{T[P]}^{\mathcal{P}}) = \texttt{true}) \\
&= (\overline{t_\beta}\ \textbf{:}\ \overline{s}\ \texttt{in}\ \overline{T[\beta]} = \texttt{true})\,.
\end{aligned}$$

Thus, since ($\overline{t_\beta}\ \textbf{:}\ \overline{s}\ \texttt{in}\ \overline{T[\beta]}\texttt{= true}$) is a ground atomic formula, we can reduce the problem to proving that

$$\texttt{MB-META} \models (\overline{t_\beta}\ \textbf{:}\ \overline{s}\ \texttt{in}\ \overline{T[\beta]}) = \texttt{true} \Longleftrightarrow T[\beta] \vdash \phi_\beta\,,$$

which, by soundness and completeness of membership equational logic, is equivalent to

$$\texttt{MB-META} \vdash (\overline{t_\beta}\ \textbf{:}\ \overline{s}\ \texttt{in}\ \overline{T[\beta]}) = \texttt{true} \Longleftrightarrow T[\beta] \vdash \phi_\beta\,.$$

Then, by Proposition 1, we obtain the desired result. □

Proposition 6 *For $P = (\Omega \oplus V, E \cup Mb(V))$ a parameter theory in $\mathcal{P}_P$, with $Mb(V) = \{v_1 : s_1, \ldots, v_n : s_n\}$, for all theory morphisms $\beta \in \mathcal{V}_P$, $\beta : P \longrightarrow Q$, for $1 \leq i \leq n$, it holds that*

$$\texttt{MB-META} \models\simeq \overline{\beta}(\overline{v_i : s_i}^{\mathcal{B}(P,\emptyset)}) .$$

Proof First, by the definition of substitution application and Propositions 3 and 4,

$$\begin{aligned} \overline{\beta}(\overline{v_i : s_i}^{\mathcal{B}(P,\emptyset)}) &= (\overline{\beta}(\overline{v_i}^{[V]})\, \texttt{\textbf{:}}\, \overline{s_i}\ \texttt{\textbf{in}}\ \overline{\beta}(\overline{P}^{\mathcal{P}}) = \texttt{true}) \\ &= (\overline{(v_i)_\beta}\, \texttt{\textbf{:}}\, \overline{s_i}\ \texttt{\textbf{in}}\ \overline{Q} = \texttt{true}) . \end{aligned}$$

Recall that, by definition, $Q \vdash (v_i)_\beta : s_i$. Therefore, by Proposition 5 (where $T[P]$ is in this case $P[P] = P$, and $T[\beta]$ is then $P[\beta] = Q$), we obtain the desired result. □

In Appendix A we will also use the following representation function. For $P = (\Omega \oplus V, E \cup Mb(V))$, all parameterized theories $T[P] \in \mathcal{P}_P$, with $\Omega = (K, \Sigma, S)$, all theory morphisms $\beta \in \mathcal{V}_P$, $\beta : P \longrightarrow Q$, and all membership assertions $t : s$, with $t \in T_{\Omega \oplus V}$ and s in some S_k,

$$\overline{(t : s)_\beta}^{\mathcal{B}(T[\beta],X)} \triangleq (\overline{t_\beta}^{[X]}\, \texttt{\textbf{:}}\, \overline{s}\ \texttt{\textbf{in}}\ \overline{T[\beta]}) = \texttt{true} .$$

Similarly, for all equalities $t = t'$, with t, t' in $T_{\Omega \oplus V}(X)$,

$$\overline{(t = t')_\beta}^{\mathcal{B}(T[\beta],X)} \triangleq (\overline{t_\beta}^{[X]}\, \texttt{\textbf{=}}\, \overline{t'_\beta}^{[X]}\ \texttt{\textbf{in}}\ \overline{T[\beta]}) = \texttt{true} .$$

9.3 Representing Axioms of Parameter Theories

Next, we define a representation function $\overline{(\_)}^{\mathcal{C}(\_)}$ for axioms of parameter theories. This function constitutes the representation function $\overline{(\_)}^{\mathcal{T}}$ mentioned in part (ii) of our third abstract requirement for classes of the form $\mathcal{P}_P$ of parameterized theories.

We first define a representation function for assertions of well-formedness relative to an unparameterized membership equational signature. For all memberships equational signatures $(\Omega \oplus V)$, and terms t in $T_{\Omega \oplus V}(X)$,

$$\overline{t}^{\mathcal{W}(\Omega,V)} \triangleq (\texttt{\textbf{wft}}\ \overline{t}^{[V]}\ \texttt{\textbf{in}}\ \overline{\Omega}) = \texttt{true} .$$

Then, for $P = (\Omega \oplus V, E \cup Mb(V))$ in $\mathcal{P}_P$, with $Mb(V) = \{v_1 : s_1, \ldots, v_n : s_n\}$,

$$\overline{E \cup Mb(V)}^{\mathcal{C}(P)} \triangleq (\overline{v_1}^{\mathcal{W}(\Omega,V)} \wedge \cdots \wedge \overline{v_n}^{\mathcal{W}(\Omega,V)} \wedge \overline{v_1\ seps_1}^{\mathcal{B}(P,\emptyset)} \wedge \cdots \wedge \overline{v_n : s_n}^{\mathcal{B}(P,\emptyset)}) .$$

Proposition 7 *For $P = (\Omega \oplus V, E \cup Mb(V))$ a parameter theory in $\mathcal{P}_P$, with $Mb(V) = \{v_1 : s_1, \ldots, v_n : s_n\}$, for all theory morphisms $\beta \in \mathcal{V}_P$, $\beta : P \longrightarrow Q$, for $1 \leq i \leq n$, it holds that*

$$\texttt{MB-META} \models\simeq \overline{\beta}(\overline{(v_i)}^{\mathcal{W}(\Omega,V)}) .$$

Proof By the definition of substitution application and Proposition 3,

$$\begin{aligned}\overline{\beta}(\overline{(v_i)}^{\mathcal{W}(\Omega,V)}) &= (\mathbf{wft}\ \overline{\beta}(\overline{(v_i)}^{[V]})\ \mathbf{in}\ \overline{\Omega} = \mathtt{true}) \\ &= (\mathbf{wft}\ \overline{(v_i)_\beta}\ \mathbf{in}\ \overline{\Omega} = \mathtt{true})\,.\end{aligned}$$

Recall that, by definition, $\overline{(v_i)_\beta}$ is a term in T_Ω. Thus, since $(\mathbf{wft}\ \overline{(v_i)_\beta}\ \mathbf{in}\ \overline{\Omega} = \mathtt{true})$ is a ground atomic formula, we can reduce the problem to proving that,

$$\mathtt{MB\text{-}META} \models (\mathbf{wft}\ \overline{(v_i)_\beta}\ \mathbf{in}\ \overline{\Omega}) = \mathtt{true}\,,$$

which by soundness and completeness of membership equational logic, is equivalent to

$$\mathtt{MB\text{-}META} \vdash (\mathbf{wft}\ \overline{(v_i)_\beta}\ \mathbf{in}\ \overline{\Omega}) = \mathtt{true}\,.$$

By Proposition 2, we obtain the desired result. □

9.4 Reflecting Parameterized Induction

We now show how parameterized induction can be reflected in a theory extension of MB-META. This extension constitutes the theory $Ind(U)$ mentioned in part (ii) of our third abstract requirement. We will use many-kinded first-order logic with equality as the superlogic mentioned in part (i) of the third abstract requirement. First, we introduce an inference rule that reflects induction principles for parameterized membership equational theories. This inference rule is based on the soundness of the induction schema given by (14) in Definition 8 below.

To understand in which way (14) reflects the induction principles for parameterized membership equational theories, the key observation is the following. Let s be a sort in a membership equational theory $T = (\Omega, E)$ defined by the set of sentences $\{C_1, \ldots, C_n\}$ in E. Consider then the set of terms of the form $u = \bar{t}$, for t a term of sort s in T. We can show that this set of terms is inductively defined by a set of sentences over the signature of MB-META that reflect $\{C_1, \ldots, C_n\}$. Hence, to derive the induction schema (14), we can use the set of sentences that reflect $\{C_1, \ldots, C_n\}$, analogously to how we use $\{C_1, \ldots, C_n\}$ to derive the induction schema (7) in Definition 7. Note, in particular, that the induction hypothesis schema given by (15) in Definition 8 is completely analogous to the induction hypothesis schema given by (8) in Definition 7.

Since the induction schema (14) reflects induction principles for parameterized membership equational theories, it can only be used to prove certain first-order formulae over the signature of MB-META, namely, those that formalize relationships among the initial models of instances of parameterized membership equational theories which are parameterized by the same parameter theory. This restriction is formalized in Definition 8 by limiting the applicability of (14) to first-order formulae over the signature of MB-META of the form (13).

Finally, to simplify the soundness proof of the induction schema (14), we impose an extra condition on the first-order formulae of the form (13) to which

(14) can be applied. Namely, that among the parameterized membership equational theories whose initial models are related by the formula, the one that is used to generate the inductive cases must be "equationally generic", in the sense that if an equality holds in all of its instances, then it also holds in the corresponding instances of all the other parameterized theories. This condition is formalized in Definition 8 by requiring that property (11) is satisfied by the parameterized membership equational theories mentioned in the formula to which (14) is applied.[12]

In what follows, for all atomic formulae ϕ over a signature $\Omega \oplus V$, parameters $v \in V$, and terms $t \in T_{\Omega \oplus V}(X)$, we will denote by $(\phi)(v \mapsto t)$ the formula that results from replacing in ϕ each occurrence of v by t. When the parameter v is clear from the context, we shall simply write $\phi(t)$.

Definition 8 (Reflected parameterized induction over sort definitions) *Let $P = (\Omega \oplus V, E \cup Mb(V))$ be a parameter theory in $\mathcal{P}_P$, with $\Omega = (K, \Sigma, S)$, and let $\{T_0[P], T_1[P], \ldots, T_k[P]\}$ be a finite multiset of parameterized theories in $\mathcal{P}_P$ that satisfies the following property: for all theory morphisms β in $\mathcal{V}_P$, $\beta : P \longrightarrow Q$, and terms t and t' in $T_{\Omega \oplus V}(X)$, for $1 \leq l \leq k$,*

$$T_0[\beta] \vdash t = t' \Longrightarrow T_l[\beta] \vdash t = t' \,. \tag{11}$$

Let $T_0[P] = (\Omega \oplus V, E \cup G_0 \cup Mb(V))$, let s be a sort in some S_k, and let $C_{[T_0[P],s]} = \{C_1, \ldots, C_n\}$ be those sentences in $(E \cup G_0)$ specifying the sort s, i.e., those C_i of the form

$$\forall (x_1, \ldots, x_{p_i}).\, A_1 \wedge \ldots \wedge A_{q_i} \Longrightarrow A_0 \,, \tag{12}$$

where, for some term t of kind k, A_0 is $t : s$. Finally, for all sentences C_i in $(E \cup G_0)$ of the form (12), let X_i be the set of variables $\{x_1, \ldots, x_{p_i}\}$, where $X_i \cap (\Sigma \oplus V) = \emptyset$.

Then, for all finite multisets of ground atomic formulae $\{\phi_1, \ldots, \phi_k\}$, with $\phi_l \in sen(T_l[P])$, $1 \leq l \leq k$, parameters $v_i \in V$, Boolean expressions b, and first-order formulae $\tau(\overline{v_i}^{[V]})$, of the form

$$\begin{aligned} &\forall \overline{(V \setminus \{v_i\})}^{[V]}.\, (\overline{E \cup Mb(V)}^{\mathcal{C}(P)} \\ &\quad \Longrightarrow (\overline{v_i : s}^{\mathcal{B}(T_0[P],\emptyset)} \Longrightarrow (b(\overline{\phi_1}^{\mathcal{B}(T_1[P],\emptyset)}, \ldots, \overline{\phi_k}^{\mathcal{B}(T_k[P],\emptyset)})))) \,, \end{aligned} \tag{13}$$

with free variable $\overline{v_i}^{[V]}$ of the kind `[Term]`, *an* induction principle for `MB-META`, with respect to $\overline{v_i : s}^{\mathcal{B}(T_0[P],\emptyset)}$ and $\tau(\overline{v_i}^{[V]})$, *is the formula*

$$[\forall \overline{V}^{[V]}.\, (\overline{E \cup Mb(V)}^{\mathcal{C}(P)} \Longrightarrow (\psi_1 \wedge \ldots \wedge \psi_n))] \Longrightarrow \forall \overline{v_i}^{[V]}.\tau(\overline{v_i}^{[V]}) \,, \tag{14}$$

where, for $1 \leq i \leq n$ and C_i in $(E \cup G_0)$ of the form (12), ψ_i is

$$\forall (\overline{x}_1^{[X_i]}, \ldots, \overline{x}_{p_i}^{[X_i]}).\, [A_1]_\tau \wedge \ldots \wedge [A_{q_i}]_\tau \Longrightarrow [A_0]_\tau \tag{15}$$

[12] In [19] this condition has been relaxed, making it possible to apply (a modified version of) the induction schema (14) to a broader class of metatheorems.

and,[13] *for* $0 \leq j \leq q_i$,

$$[A_j]_\tau \triangleq \begin{cases} b(\overline{\phi_1(v_i \mapsto u)}^{\mathcal{B}(T_1[P],X_i)}, \ldots, \overline{\phi_k(v_i \mapsto u)}^{\mathcal{B}(T_k[P],X_i)}) & \textit{if } A_j = u\!:\!s \\ \overline{A_j}^{\mathcal{B}(T_0[P],X_i)} & \textit{otherwise}. \end{cases}$$

For a given finite multiset $\{T_0[P], T_1[P], \ldots, T_k[P]\}$ of parameterized theories in $\mathcal{P}_P$, the above defines an induction schema given by (14), in many-kinded, first-order logic over the signature of `MB-META`. We prove the soundness of this induction schema in Appendix A. In Section 10 we show how this induction schema can be used to give a metalogical proof of the deduction theorem.

Next, we show that *meta-ind*(`MB-META`), i.e., the theory `MB-META` extended with the induction schema (14), satisfies part (iii) of our third abstract requirement in Section 3.2 for universally quantified inductive theorems over families of parameterized membership equational logics.[14]

By Remark (5), we can state this requirement as follows: for $P = (\Omega \oplus V, E \cup Mb(V))$, for all finite multisets of ground atomic formulae $\phi_l \in sen(T_l[P]), 1 \leq l \leq k$, with $T_l[P] \in \mathcal{P}_P$, whenever we have

$$\begin{aligned} \textit{meta-ind}(\texttt{MB-META}) \vdash \forall \overline{V}^{[V]}.(\overline{E \cup Mb(V)}^{\mathcal{C}(P)} \\ \Longrightarrow b(\overline{\phi_1}^{\mathcal{B}(T_1[P],\emptyset)}, \ldots, \overline{\phi_k}^{\mathcal{B}(T_k[P],\emptyset)})), \end{aligned} \quad (16)$$

then we also have

$$\forall \beta \in \mathcal{V}_P . b(T_1[\beta] \models\!\!\!\simeq (\phi_1)_\beta, \ldots, T_k[\beta] \models\!\!\!\simeq (\phi_k)_\beta) . \quad (17)$$

To prove that (16) implies (17), first note that, by the soundness of the induction schema (14), (16) implies that

$$\texttt{MB-META} \models\!\!\!\simeq \forall \overline{V}^{[V]}.(\overline{E \cup Mb(V)}^{\mathcal{C}(P)} \Longrightarrow b(\overline{\phi_1}^{\mathcal{B}(T_1[P],\emptyset)}, \ldots, \overline{\phi_k}^{\mathcal{B}(T_k[P],\emptyset)})) ,$$

which implies that, for all ground substitutions $h : \overline{V}^{[V]} \longrightarrow$ `[Term]`,

$$\texttt{MB-META} \models\!\!\!\simeq h(\overline{E \cup Mb(V)}^{\mathcal{C}(P)} \Longrightarrow b(\overline{\phi_1}^{\mathcal{B}(T_1[P],\emptyset)}, \ldots, \overline{\phi_k}^{\mathcal{B}(T_k[P],\emptyset)})) ,$$

which, by Propositions 7 and 6, implies, in particular, that for all theory morphisms $\beta \in \mathcal{V}_P$, $\beta : P \longrightarrow Q$,

$$\texttt{MB-META} \models\!\!\!\simeq b(\overline{\beta}(\overline{\phi_1}^{\mathcal{B}(T_1[P],\emptyset)}), \ldots, \overline{\beta}(\overline{\phi_k}^{\mathcal{B}(T_k[P],\emptyset)})) . \quad (18)$$

To complete the proof, notice that, since $(\phi_l)_\beta$ is a ground atomic formula, for $1 \leq l \leq k$, (17) is equivalent to

$$\forall \beta \in \mathcal{V}_P . b(T_1[\beta] \models (\phi_1)_\beta, \ldots, T_k[\beta] \models (\phi_k)_\beta) . \quad (19)$$

[13]Note that in the representation of formulae at the metalevel in (15), we require that object variables are represented using variables instead of constants.

[14]The proof of the corresponding result for the general case of inductive theorems with multiple-quantifiers requires considerable additional technical details, which we omit here.

Now, by the soundness and completeness of membership equational logic, (19) is equivalent to

$$\forall \beta \in \mathcal{V}_P . b(T_1[\beta] \vdash (\phi_1)_\beta, \dots, T_k[\beta] \vdash (\phi_k)_\beta) . \quad (20)$$

Finally, by Proposition 5, (18) implies (20). □

10 The Deduction Theorem for Minimal Logic

In this section we give an example that illustrates how membership equational logic can be used as a reflective metalogical framework. Our example is a standard one in metareasoning, namely, the deduction theorem.

The deduction theorem is interesting for several reasons. To begin with, it is a central metatheorem that holds for Hilbert systems for many logics and justifies proof under temporary assumption in the manner of a natural deduction system. Moreover, although relatively simple, it illustrates some subtle aspects of formal metareasoning. As previously observed, it is a metatheorem that relates different deduction systems: one in which $A \to B$ is proved, and a second (which is obtained from the first by adding the axiom A) in which B is proved, in symbols

$$\text{if } \vdash_{\mathcal{M}[A]} B \text{ then } \vdash_{\mathcal{M}} A \to B .$$

Moreover, since A is an arbitrary formula (as is B), the standard statement of the deduction theorem is actually a statement about the relationship between a *family* of pairs of deduction systems.

10.1 Formalization

Consider the representation of minimal logic in membership equational logic provided by `MINIMAL` in Figure 2. Recall that `MINIMAL` represents minimal logic in the sense that a formula A is a theorem in minimal logic if and only if A is a term of sort `Theorem` in `MINIMAL`. Consider also the parameter theory `MINIMAL-P` introduced in Figure 3, its parameterized extension `MINIMAL-DT[MINIMAL-P]`, introduced in Figure 4, and its trivial extension by itself, which we will also denote `MINIMAL-P`. In addition, let us denote by V_{DT} the set of parameters of `MINIMAL-P`, $V_{\mathrm{DT}} = \{AA, BB\}$, and by X_{DT} its set of variables, $X_{\mathrm{DT}} = \{A, B, C\}$. Recall that `MINIMAL-DT[MINIMAL-P]` represents the family of logics that consists of all extensions of minimal logic with a new axiom in the sense that a formula B is a theorem in minimal logic extended with a new axiom A if and only if B is a term of sort `Theorem` in the corresponding instantiation of `MINIMAL-DT[MINIMAL-P]`. Note that `MINIMAL-P` and `MINIMAL-DT[MINIMAL-P]` are in the class $\mathcal{P}_{\texttt{MINIMAL-P}}$ of theories parameterized by `MINIMAL-P` for $\mathcal{P}_{\texttt{MINIMAL-P}}$ as defined in Section 5. Finally, let `SIG-MINIMAL` denote the signature of `MINIMAL`, and let `AX-MINIMAL-P` denote the axioms of `MINIMAL-P`. Note that `AX-MINIMAL-P` is the union of $Mb(\{AA, BB\})$ and the set of axioms E formed by the three membership axioms formalizing the axiom schemata K and S and the rule *modus ponens*.

The deduction theorem can then be expressed as a metatheoretic statement relating the initial models of all the different instantiations of MINIMAL-DT[MINIMAL-P] and MINIMAL-P as follows:

$$\forall \beta \in \mathcal{V}_{\texttt{MINIMAL-P}}.\, \mathcal{I}(\texttt{MINIMAL-DT[}\beta\texttt{]}) \models \beta(BB)\,\texttt{:Theorem} \Longrightarrow \mathcal{I}(\texttt{MINIMAL-P[}\beta\texttt{]}) \models \beta(AA \rightarrow BB)\,\texttt{:Theorem}\,. \quad (21)$$

Using the results of Section 9 we can formalize the above metatheoretic statement as a theorem about *meta-ind*(MB-META) as follows:

$$\forall(\overline{AA, BB}^{[V_{\mathrm{DT}}]}).\, (\overline{\texttt{AX-MINIMAL-P}}^{\mathcal{C}(\texttt{MINIMAL-P})} \Longrightarrow (\overline{BB\,\texttt{:Theorem}}^{\mathcal{B}(\texttt{MINIMAL-DT[MINIMAL-P]},\emptyset)} \Longrightarrow (\overline{AA \rightarrow BB\,\texttt{:Theorem}}^{\mathcal{B}(\texttt{MINIMAL-P},\emptyset)})))\,. \quad (22)$$

Since *meta-ind*(MB-META) satisfies part (iii) of our third abstract requirement, (22) implies (21).

10.2 Proof of the Deduction Theorem

We sketch here how we prove (22). To begin with, this formula is equivalent to

$$\begin{aligned}\forall(\overline{AA, BB}^{[V_{\mathrm{DT}}]}).\, ((&\textbf{wft}\ \overline{AA}^{[V_{\mathrm{DT}}]}\ \textbf{in}\ \overline{\texttt{SIG-MINIMAL}} = \texttt{true} \wedge \\ &\textbf{wft}\ \overline{BB}^{[V_{\mathrm{DT}}]}\ \textbf{in}\ \overline{\texttt{SIG-MINIMAL}} = \texttt{true} \wedge \\ &\overline{AA}^{[V_{\mathrm{DT}}]}\ \textbf{:}\ \overline{\texttt{Formula}}\ \textbf{in}\ \overline{\texttt{MINIMAL-P}}^{\mathcal{P}} = \texttt{true} \wedge \\ &\overline{BB}^{[V_{\mathrm{DT}}]}\ \textbf{:}\ \overline{\texttt{Formula}}\ \textbf{in}\ \overline{\texttt{MINIMAL-P}}^{\mathcal{P}} = \texttt{true}) \\ \Longrightarrow (&\overline{BB}^{[V_{\mathrm{DT}}]}\ \textbf{:}\ \overline{\texttt{Theorem}}\ \textbf{in}\ \overline{\texttt{MINIMAL-DT[MINIMAL-P]}}^{\mathcal{P}} = \texttt{true} \\ &\Longrightarrow (\overline{AA \rightarrow BB}^{[V_{\mathrm{DT}}]}\ \textbf{:}\ \overline{\texttt{Theorem}}\ \textbf{in}\ \overline{\texttt{MINIMAL-P}}^{\mathcal{P}} = \texttt{true})))\,.\end{aligned} \quad (23)$$

We can now prove (23) mirroring the standard proof of the deduction theorem: induction on the structure of derivations in minimal logic extended with the axiom A.

First, we apply the reflected version of the induction principle for the sort Theorem in the parameterized theory MINIMAL-DT[MINIMAL-P], that is, the corresponding instance of the inference rule (14), where $\mathcal{P}$ is the set formed by the parameterized theories MINIMAL-DT[MINIMAL-P] and MINIMAL-P, and $T_0[P]$ is the parameterized theory MINIMAL-DT[MINIMAL-P].[15] This reduces proving (23) to proving the formula given in Figure 7. Notice that the four conjuncts correspond to the cases involved in proving the deduction theorem by induction over

[15]Since MINIMAL-DT[MINIMAL-P] does not contain any equations, for all theory morphisms β in $\mathcal{V}_P$, the only equations in MINIMAL-DT[β] are equations of the form **eq** $AA = u$ **.** and **eq** $BB = u'$ **.**, with u and u' two arbitrary ground terms in SIG-MINIMAL. Therefore, for all theory morphisms β in $\mathcal{V}_P$ and terms t, t' in MINIMAL-DT[MINIMAL-P],

$$\texttt{MINIMAL-DT[}\beta\texttt{]} \vdash t = t' \Longrightarrow \texttt{MINIMAL-P[}\beta\texttt{]} \vdash t = t'\,.$$

the proof that B is a theorem in minimal logic extended with a new axiom A. The first formalizes the case when B is A. The next two conjuncts formalize the cases where B is either an instance of the K or S axiom schemata. The final conjunct formalizes the case of B being proved by an instance of *modus ponens*. Finally, we can apply the equations in `MB-META` to prove each of the resulting conjuncts. Given our specification of `MB-META`, these proofs mirror the proofs of the corresponding cases in the standard inductive proof of the deduction theorem. For example, the proof of the third conjunct reflects the proof that, for all formulae D, A, B, and C,

$$D \to ((A \to B) \to ((A \to (B \to C)) \to (A \to B))) \tag{24}$$

is a theorem in minimal logic; in particular, this proof mirrors proving (24) by *modus ponens*, using the following instance of the S axiom

$$(A \to B) \to ((A \to (B \to C)) \to (A \to C))$$

and the following instance of the K axiom

$$\begin{aligned}&[(A \to B) \to ((A \to (B \to C)) \to (A \to C))]\\ &\quad \to \ (D \to [(A \to B) \to ((A \to (B \to C)) \to (A \to C))])\,.\end{aligned}$$

11 Discussion and Related Work

In this section, we discuss our general experience in using our approach. We also consider tradeoffs and limitations, and survey related work.

11.1 Experience

We have used membership equational logic as a reflective metalogical framework to carry out a number of proofs in formal metatheory based on more sophisticated versions of the deduction theorem for minimal logic. In particular, we have proved results similar to those of Basin and Matthews [7, 8], who have shown how metatheorems that are parameterized by their scope of application can be proved using a theory of parameterized inductive definitions as a metatheory. For example, they present a generalized version of the deduction theorem that can be applied to all extensions of the language and axioms of minimal logic. From their theorem it follows that the deduction theorem holds for the minimal logic of implication and for all propositional extensions of it, but not necessarily for extensions to modal logics (which would require adding new rules, as opposed to new axioms). Although membership equational logic is based on a rather different foundation than those considered in [8], our representation of the object logic is quite similar and—abstracting away from the details involved in moving between levels of representation—the basic structure of the proofs is also similar.

One promising area to apply our results is program transformation and metaprogramming. From a reflective declarative point of view, programs that

$$
\begin{aligned}
&\forall(\overline{AA, BB}^{[V_{DT}]}).\ [((\mathbf{wft}\ \overline{AA}^{[V_{\mathrm{DT}}]}\ \mathbf{in}\ \overline{\texttt{SIG-MINIMAL}} = \texttt{true}) \wedge \\
&\qquad\qquad (\mathbf{wft}\ \overline{BB}^{[V_{\mathrm{DT}}]}\ \mathbf{in}\ \overline{\texttt{SIG-MINIMAL}} = \texttt{true}) \wedge \\
&\qquad\qquad (\overline{AA}^{[V_{\mathrm{DT}}]}\ \mathbf{:}\ \overline{\texttt{Formula}}\ \mathbf{in}\ \overline{\texttt{MINIMAL-P}}^{\mathcal{P}} = \texttt{true}) \wedge \\
&\qquad\qquad (\overline{BB}^{[V_{\mathrm{DT}}]}\ \mathbf{:}\ \overline{\texttt{Formula}}\ \mathbf{in}\ \overline{\texttt{MINIMAL-P}}^{\mathcal{P}} = \texttt{true})) \\
&\Longrightarrow \\
&((\overline{AA \to AA}^{[V_{\mathrm{DT}}]}\ \mathbf{:}\ \overline{\texttt{Theorem}}\ \mathbf{in}\ \overline{\texttt{MINIMAL-P}}^{\mathcal{P}} = \texttt{true}) \\
&\wedge \\
&(\forall(\overline{A, B}^{[X_{\mathrm{DT}}]}). \\
&\quad ((\overline{A}^{[X_{\mathrm{DT}}]}\ \mathbf{:}\ \overline{\texttt{Formula}}\ \mathbf{in}\ \overline{\texttt{MINIMAL-DT[MINIMAL-P]}}^{\mathcal{P}} = \texttt{true}) \wedge \\
&\quad (\overline{B}^{[X_{\mathrm{DT}}]}\ \mathbf{:}\ \overline{\texttt{Formula}}\ \mathbf{in}\ \overline{\texttt{MINIMAL-DT[MINIMAL-P]}}^{\mathcal{P}} = \texttt{true})) \\
&\quad \Longrightarrow \\
&\quad (\overline{(AA \to (A \to (B \to A)))}^{[V_{\mathrm{DT}}, X_{\mathrm{DT}}]}\ \mathbf{:}\ \overline{\texttt{Theorem}}\ \mathbf{in}\ \overline{\texttt{MINIMAL-P}}^{\mathcal{P}} = \texttt{true})) \\
&\wedge \\
&(\forall(\overline{A, B, C}^{[X_{\mathrm{DT}}]}). \\
&\quad ((\overline{A}^{[X_{\mathrm{DT}}]}\ \mathbf{:}\ \overline{\texttt{Formula}}\ \mathbf{in}\ \overline{\texttt{MINIMAL-DT[MINIMAL-P]}}^{\mathcal{P}} = \texttt{true}) \wedge \\
&\quad (\overline{B}^{[X_{\mathrm{DT}}]}\ \mathbf{:}\ \overline{\texttt{Formula}}\ \mathbf{in}\ \overline{\texttt{MINIMAL-DT[MINIMAL-P]}}^{\mathcal{P}} = \texttt{true}) \wedge \\
&\quad (\overline{C}^{[X_{\mathrm{DT}}]}\ \mathbf{:}\ \overline{\texttt{Formula}}\ \mathbf{in}\ \overline{\texttt{MINIMAL-DT[MINIMAL-P]}}^{\mathcal{P}} = \texttt{true})) \\
&\quad \Longrightarrow \\
&\quad (\overline{(AA \to ((A \to B) \to ((A \to (B \to C)) \to (A \to C))))}^{[V_{\mathrm{DT}}, X_{\mathrm{DT}}]} \\
&\quad\ \mathbf{:}\ \overline{\texttt{Theorem}}\ \mathbf{in}\ \overline{\texttt{MINIMAL-P}}^{\mathcal{P}} = \texttt{true})) \\
&\wedge \\
&(\forall(\overline{A, B}^{[X_{\mathrm{DT}}]}). \\
&\quad ((\overline{A}^{[X_{\mathrm{DT}}]}\ \mathbf{:}\ \overline{\texttt{Formula}}\ \mathbf{in}\ \overline{\texttt{MINIMAL-DT[MINIMAL-P]}}^{\mathcal{P}} = \texttt{true}) \wedge \\
&\quad (\overline{B}^{[X_{\mathrm{DT}}]}\ \mathbf{:}\ \overline{\texttt{Formula}}\ \mathbf{in}\ \overline{\texttt{MINIMAL-DT[MINIMAL-P]}}^{\mathcal{P}} = \texttt{true}) \wedge \\
&\quad (\overline{AA \to (A \to B)}^{[V_{\mathrm{DT}}, X_{\mathrm{DT}}]}\ \mathbf{:}\ \overline{\texttt{Theorem}}\ \mathbf{in}\ \overline{\texttt{MINIMAL-P}}^{\mathcal{P}} = \texttt{true}) \wedge \\
&\quad (\overline{AA \to A}^{[V_{\mathrm{DT}}, X_{\mathrm{DT}}]}\ \mathbf{:}\ \overline{\texttt{Theorem}}\ \mathbf{in}\ \overline{\texttt{MINIMAL-P}}^{\mathcal{P}} = \texttt{true})) \\
&\quad \Longrightarrow \\
&\quad (\overline{AA \to B}^{[V_{\mathrm{DT}}, X_{\mathrm{DT}}]}\ \mathbf{:}\ \overline{\texttt{Theorem}}\ \mathbf{in}\ \overline{\texttt{MINIMAL-P}}^{\mathcal{P}} = \texttt{true})))]\,.
\end{aligned}
$$

Figure 7: Goal resulting after induction

transform other programs are first-order functions acting on terms that metarepresent theories, and the properties that they satisfy are metatheorems, as they are understood in this paper. This reflective declarative methodology has been used in [17] to specify polytypic programs like *map* and *cata* in Maude. Accordingly, polytypic programs are specified as metalevel functions that add to a module the equations defining the desired object function by structural induction over the sort definitions. Properties of polytypic programs, like the functoriality of *map*, are then metatheorems that can be proved, as it is shown in [17], using the corresponding reflected induction rule (Definition 14).

Another area of application is formal metareasoning about semantic relations between equational specifications. As it is well-known, equational specifications can be related in different ways, and these relations can be informally formulated as metatheorems of equational logic. The semantic relations between different equational specifications are key conceptual tools in the stepwise specification methodology, and different techniques and criteria have been proposed to metalogically prove them [28]. Using membership equational logic as a reflective metalogical framework, [19] shows that some of these semantic relations can be formalized as theorems of the universal theory of membership equational logic and that they can be logically proved in a way that mirrors their corresponding proofs at the metalogical level.

Here we would also like to comment on our experience in proving these theorems and on the issue of managing proofs that combine reasoning at different levels of reflection. To the working logician or computer scientist, reflective metalogical frameworks may seem complicated and not particularly user-friendly, since there is quite a bit of encoding involved in stating a metatheorem and in carrying out its proof. In particular, reasoning can involve three or more levels (object, meta, meta-meta, etc.).[16]

In our case, we have been able to avoid many of the practical problems of working with a reflective hierarchy by exploiting the reflective capabilities of Maude to build tools and suitable interfaces that hide levels of reflection. As part of our work, we have built an interface—fully specified in Maude—to interact with the `ITP` inductive theorem prover [15, 13]. The `ITP` automatically extracts from a theory the induction principles for reasoning over its sorts (Definition 7), and (in its metaprover extension) the induction rules that correspond to reflecting those induction principles at the metalevel when the task at hand is to prove a metatheorem (Definition 8). Proving an inductive theorem then amounts to computing a strategy at the meta-metalevel, or at the meta-meta-metalevel if the theorem is, as in the case of (22), a metatheorem about the initial model of `MB-META`. Fortunately, the interface we use hides all these levels of encoding from the user. Hence the user can actually abstract away many of the metarepresentation details and focus on the essential structure of proofs of theorems.

[16] Although note that reasoning about a logic encoded as an inductive definition in a logical framework like Isabelle also involves multiple levels, e.g., the framework's metalogic, the theory of inductive definitions, and the object logic. Moreover, there is often an additional language for writing tactics.

11.2 Tradeoffs and Limitations

Our thesis in this paper is that metalogical reasoning based on a metalogic supporting reflection, initiality, and parameterization, offers an interesting and effective possibility for formal metatheory. We have given abstract requirements for a general logic to be a metalogical framework in this sense and have presented membership equational logic as a nontrivial instance of the proposed approach.

We neither prejudge nor preclude other possibilities. One could use, as an alternative instance, stronger higher-order metalogics, provided they also satisfy our requirements, perhaps in some weakened, relativized sense. However, in higher-order settings, both reflection and initiality can be problematic. It is difficult to be more general and still have initial models.[17] There has been some work on reflection for typed higher-order theories like Nuprl [4, 41, 46] and the calculus of constructions [64], but results there are difficult and partial. Difficulties arise in typing a self-interpreter, which is a function that associates metalevel representations with the object level values they denote, in particular reasoning about its domain of termination and the types of values it computes. Combining induction and reflective aspects in higher-order type theories is difficult and, as far as we know, has not yet been done.

There are, of course, tradeoffs involved and the use of a stronger logic, where possible, might ease certain kinds of metatheoretic arguments. The tradeoffs are very similar to those which have been observed when using a logical framework, as opposed to a metalogical framework. There is a wide spectrum of possible logical frameworks with different strengths, weaknesses, and domains where they excel. Although all logical frameworks are (or should be) capable of formalizing, in some way, the same consequence relations (as recursively enumerable sets), there is tension in balancing simplicity, representational flexibility, and generality against strength and specialization. A stronger metalogic provides more structure, but this additional structure makes more representational commitments. These commitments may help in formalizing some logics, e.g., for certain kinds of consequence relations one can directly utilize the properties of intuitionistic (or linear) implication in the metalogic to formalize structural properties of the object logic. However this works best for logics whose consequence relations are close to those of the framework logic, and this lack of flexibility can be problematic when this is not the case (witness the development of increasingly specialized metalogics for representing "difficult" object logics, and the negative results on lack of adequate representations for linear and relevance logics in LF-style frameworks [31]). Similarly, when carrying out metatheoretic reasoning, support for, e.g., higher-order abstract syntax is useful in "internalizing" operations like substitution, equivalence under the renaming of bound variables, and the like. However, this internalization is problematic if

[17] In first-order logic, Horn logic with equality is the most general sublogic whose theories always have initial models [47]. In higher-order logic, the existence of initial models depends crucially on the notion of model adopted (e.g., set theoretic, domain-theoretic, realizability, etc.); initiality can usually be achieved if one adopts a categorical logic approach, but generally *not* in other cases.

one wants to reason, for example, about the names of bound variables.

Membership equational logic provides an alternative way of formalizing operations based on reasoning modulo an equational theory. In this approach, properties of contexts, sequents, substitutions, quantification, etc., can easily be incorporated as equational theories. For example, if one needs a sequent (hypersequent, context, etc.) with particular properties, one can simply state them algebraically. Afterwards, one can use support provided by an implementation of membership equational logic for reasoning modulo equational theories to internalize standard operations (e.g., structural reasoning, substitution and binding operators [75], reduction, etc.) and one can use well-developed equational reasoning techniques and tools to establish important meta-theoretic properties of the relevant structure so axiomatized. There are a number of papers documenting the flexibility and power of this approach in the context of rewriting logic [16, 49, 48, 57, 75, 76, 77].

11.3 Alternative Approaches to Induction

Various approaches have been considered in the past to strengthen logical frameworks so that they can function as metalogical frameworks. All of these differ significantly from our proposal both in their logical basis and in the rôle of reflection in metareasoning.

One approach is to formalize theories in a framework logic supporting some notion of module, where each module is explicitly equipped with its own induction principle. For example, in [6], theories were formalized by collections of parameterized modules (Σ-types) within the Nuprl type theory (a constructive, higher-order logic), and each module included its own induction principle for reasoning about terms or proofs. This approach is powerful and can be used, for example, to relate different theories formalized in this way.

An alternative approach is to formalize theories directly using inductive definitions in a framework logic or framework theory that is strong enough to formalize the corresponding induction principles. A simple example of this is the first-order theory FS_0 of [29], which has been used to carry out experiments in formal metatheory [52]. In FS_0, inductive definitions are terms in the framework theory, which has an induction rule for reasoning about such terms.

Another common choice is to formalize theories as inductive definitions in strong "foundational" framework logics like higher-order logic or set-theory [36, 59], or in a type theory like the calculus of constructions with inductive definitions [58]. In higher-order logic and set theory one can internally develop a theory of inductive definitions, where inductive definitions correspond to terms in the metatheory (e.g., formalized as the least fixedpoint of a monotonic function) and, from the definition, induction principles are formally derived within the framework logic. Alternatively, in the calculus of constructions, given an inductive definition, induction principles are simply added, soundly, to the metalogic. Current research in this area focuses on appropriate induction principles for logics that support higher-order abstract syntax [26, 53, 67].

11.4 Alternative Approaches to Reflection

A formal system can be viewed from a logical viewpoint, or from a computational one. Being reflective makes a formal system particularly flexible and expressive from both viewpoints. Logically, reflection means that the formal system can encode important aspects of its own *metalanguage*, so that, for example, theories, proofs, or provability become expressible at the object level. Computationally, reflection typically means that programs can become data that can be manipulated by other programs (usually called *metaprograms*) and, furthermore, that the computational engine executing the programs can be modified and extended in flexible ways, including ways that can take account of the runtime state of a computation.

In logic, reflection has been vigorously pursued by many researchers since the fundamental work of Gödel and Tarski (see the surveys [72, 73]). In computer science it has been present from the beginning in the form of universal Turing machines. Many researchers have recognized its great importance and usefulness in programming languages [25, 40, 45, 71, 74, 78, 80], in theorem-proving [3, 11, 32, 38, 42, 51, 63, 69, 81], in concurrent and distributed computation, and in many other areas such as compilation, programming environments, operating systems, fault-tolerance, and databases (see [23, 44, 70] for recent snapshots of research in reflection).

The very success and extension of reflective ideas underscores the need for conceptual foundations. This need is real enough, because what we have at present are specific *instances* of reflection, each explained in terms of the particular concepts available for it, such as lambda expressions, Horn clauses, Turing machines, objects and metaobjects, and so on. We need a general theory of reflection capable of unifying and interrelating all the instances.

Metalogical foundations of reflection that make the particular logic of choice an easily changeable parameter can be very useful, because we can then capture in a precise and formalism-independent way the essential features of reflection that intuitively appear to be shared by quite disparate languages and systems. In [20], Clavel and Meseguer proposed metalogical axioms for reflection based on the theory of general logics [54], and explained how reflection in a number of well-known formal systems such as the lambda calculus, Turing machines and rewriting logic, as well as reflective phenomena in declarative programming languages, can be unified and understood in the light of those axioms. The key axiomatic ideas are centered around the notion of a *universal theory*, that is, a theory U that can simulate the metalevel of all other theories in a class $\mathcal{C}$ of theories of interest. In particular, if U is one of the theories in the class $\mathcal{C}$, then U can simulate its own metalevel at the object level, and this process can be iterated *ad infinitum*, giving rise to a "reflective tower." Although the metalogical axioms for reflection proposed in [13, 20] are of course closely related to the reflective ideas at the core of Gödel's incompleteness theorem (see the surveys [72, 73]) the key difference is that in [13, 20] the reflective properties are generalized along several dimensions by: (1) allowing general notions of metarepresentation instead of, say, Gödel numbering; (2) relating general classes of theories instead of just one or a restricted class of theories; and (3) making the underlying logic a *parameter*, instead of relying on standard first- and second-order frameworks. Also, the intention is much more on characterizing *positive* uses of reflection, as opposed to negative impossibility results. In this sense, the uses of set-theory as a universal theory for formalized mathematics are much closer to our goals and intentions than the work on incompleteness. The difference with set theory, which can indeed be viewed as a universal theory in our sense, is that we do not insist on a *single* universal theory for *everything*. Instead, the focus is on investigating and characterizing universal theories for different logics, particularly for computational ones such as equational, Horn, rewriting logic, and lambda calculi. In the context of computational logics, the axioms in [13, 20] are also useful for supporting essential distinctions between reflective *artifacts* such as metacircular interpreters, and the *metalogical properties* that such artifacts may satisfy only in part.

Of course, reflective phenomena exhibit different degrees of reflection, in that some languages may choose to represent only certain metalevel aspects of interest, and may do so in weaker or stronger ways. Thus, in a weaker axiomatization, a universal theory may not belong to the class of theories that it represents, so that the full power of having a reflective tower is lost. For example, the untyped λ-calculus is reflective in the strong sense of expressing its own universal theory and admitting a tower, whereas typed λ-calculi typically have *eval* functions that, although terminating, cannot be typed in the given type discipline; this yields a weaker form of reflection that requires either relaxing the type discipline or putting bounds on the reflective computations (see [63] for a careful treatment of this problem for the calculus of constructions, and [42, 3, 24] for the treatment of reflection in Nuprl's constructive type theory).

Reflective phenomena also differ with respect to which metalevel aspects are represented. For example, theories, the entailment relation, the notion of proof, and inductive reasoning are different metalevel aspects which can be potentially metarepresented in a reflective logic. In a weaker axiomatization, only the entailment relation, holding between the axioms of a theory and the theorems it proves, may be represented. With stronger axioms, the actual proof calculus of the logic, which associates to each object theory a "structure" of proofs that use axioms of that theory as hypotheses, may be represented as well (see [54] for axiomatic definitions of entailment system and proof calculus, and its related notion of proof subcalculus, in the theory of general logics). In the same spirit, Section 3.2 includes an even stronger axiomatization to formalize a different reflective property, namely, the capacity of a logic to represent at the object-level the induction principles for reasoning about its object theories, and uses this formalization to clarify the interrelationship between logical and metalogical frameworks, and the rôles of induction, reflection, and parameterization as key ingredients for turning a logical framework into a metalogical one.

The gradation of strength in the reflective axioms provides formalism-independent criterion to understand and compare the reflective power exhibited by different logics and declarative languages, even when the corresponding formalisms differ greatly from each other. To this end, [13] provides increasingly stronger axiomatizations to formalize the notions of reflective entailment systems (Definition 6 in this paper), reflective proof calculus, and reflective proof subcalculus, and uses those formalizations to axiomatize the notions of reflective declarative languages and of metainterpreters for declarative programming languages. These axiomatizations use the notion of a program that is universal for the set of programs of the language but only relative to the proof subcalculus corresponding to the language's operational semantics[18].

12 Conclusion

We have presented a new approach to metatheoretic reasoning in terms of abstract requirements based on reflection, initiality, and parameterization, and we presented a concrete instance of a metalogic that satisfies these requirements. Our initial theorem proving experiments demonstrate that the machinery for reflective deduction in membership equational logic provides a rich foundation for formalizing and proving metatheorems. These experiments show, for example, that one can prove metatheorems similar to those provable in logical frameworks based on parameterized inductive definitions, and that one has considerable flexibility in moving between theories and proving theorems that relate theories or establish properties of parameterized classes of theories. In essence, we can do this because the requirements that such metatheorems pose on the metatheory—namely, that one can build families of sets using parameterized inductive definitions and that one can reason about their elements by induction—are realizable in membership equational logic using reflection.

There are a number of directions for further work. One concerns generalizing our notion of a parameterized membership equational theory and of its instantiations. Currently we can reason at the metalevel about families of theories that are parameterized by sets of new constants (that stand for parameters), and by sets of new axioms, which may make use of the new constants. For proving other metatheorems it would be useful to develop a more general notion of parameterization, where one could reason at the metalevel about families of theories that are parameterized by arbitrary sets of new sorts, function symbols, and axioms. In particular, this would allow us to prove metatheorems involving the more general *parameterized modules* of Full Maude [14, 27].

Also, our example illustrates how it is possible to carry out proofs similar to those possible in stronger framework logics. However, it would be interesting to have a more formal comparison of the relative strengths of membership equational logic with reflection versus stronger metalogics like higher-order logic or

[18] There is a long tradition of metacircular interpreters and metaprogramming in Prolog, as shown in the papers collected in [1, 12, 61, 30, 5], and the references there. This work strongly suggests that a more declarative variant of Prolog can be made reflective in the sense of Definition 6. A systematic effort to carefully represent metalevel concepts and to give a declarative semantics for Horn logic interpreters using a typed version of the logic has been undertaken by Hill and Lloyd [39]; their work seems a good basis for defining and proving correct a universal theory for Horn logic. A universal theory for Horn logic with equality is discussed in [22].

set theory. For example, it would be quite useful to compare in more detail the ideas presented here with the inductive reasoning principles for higher-order encodings developed by Schürmann in [66, 65] and added to the Twelf system [62], and also to compare experimental results on how meta-theoretic properties requiring inductive reasoning can be established in both approaches. Finally, related to this is the question of how easy it is to reflect induction principles other than structural induction, e.g., induction over an arbitrary, user-definable well-founded order.

Acknowledgments

We thank Narciso Martí-Oliet and Miguel Palomino for their careful reading of a draft of this paper and their detailed suggestions for improving the exposition. We also thank the anonymous referees for their helpful comments and urging us to find a more abstract framework for presenting our ideas.

References

[1] H. Abramson and M. Rogers, editors. *Metaprogramming in Logic Programming*. MIT Press, 1989.

[2] P. Aczel. An introduction to inductive definitions. In J. Barwise, editor, *Handbook of Mathematical Logic*, pages 739–782. North-Holland, Amsterdam, 1977.

[3] W. E. Aitken, R. L. Constable, and J. L. Underwood. Metalogical frameworks II: Developing a reflected decision procedure. *Journal of Automated Reasoning*, 22(2(2)):171–221, 1999.

[4] S. F. Allen, R. L. Constable, D. J. Howe, and W. am E. Aitken. The semantics of reflected proof. In *Symposium on Logic in Computer Science*. Computer Society Press of the IEEE, 1990.

[5] K. Apt and F. Turini, editors. *Meta-Logics and Logic Programming*. Logic Programming Series. MIT Press, 1995.

[6] D. Basin and R. Constable. Metalogical frameworks. In G. Huet and G. Plotkin, editors, *Logical Environments*, pages 1–29. Cambridge University Press, 1993.

[7] D. Basin and S. Matthews. Scoped metatheorems. In *Second International Workshop on Rewriting Logic and its Applications*, volume 15 of *Electronic Notes in Theoretical Computer Science*, pages 1–12. Elsevier, 1998.

[8] D. Basin and S. Matthews. Structuring metatheory on inductive definitions. *Information and Computation*, 162(1–2):80–95, October/November 2000.

[9] D. Basin and S. Matthews. Logical frameworks. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, 2nd. Edition*, volume 9, pages 89–164. Kluwer Academic Publishers, Dordrecht, 2002.

[10] A. Bouhoula, J.-P. Jouannaud, and J. Meseguer. Specification and proof in membership equational logic. *Theoretical Computer Science*, 236:35–132, 2000.

[11] R. S. Boyer and J. S. Moore. Metafunctions: Proving them correct and using them efficiently as new proof procedures. In R. S. Boyer and J. S. Moore, editors, *The Correctness Problem in Computer Science*, pages 103–185. Academic Press, 1981.

[12] M. Bruynooghe, editor. *Proc. Second Workshop on Meta-programming in Logic*. K. U. Leuven, 1990.

[13] M. Clavel. *Reflection in Rewriting Logic: Metalogical Foundations and Metaprogramming Applications*. CSLI Publications, 2000.

[14] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and J. F. Quesada. Maude: Specification and programming in rewriting logic. *Theoretical Computer Science*, 285(2):187–244, 2002.

[15] M. Clavel, F. Durán, S. Eker, and J. Meseguer. Building equational proving tools by reflection in rewriting logic. In K. Futatsugi, A. T. Nakagawa, and T. Tamai, editors, *CAFE: An Industrial-Strength Algebraic Formal Method*, pages 1–31. Elsevier, 2000.

[16] M. Clavel, F. Durán, S. Eker, J. Meseguer, and M.-O. Stehr. Maude as a formal meta-tool. In J. Wing and J. Woodcock, editors, *FM'99 — Formal Methods*, volume 1709 of *Lecture Notes in Computer Science*, pages 1684–1703. Springer-Verlag, 1999.

[17] M. Clavel, F. Durán, and N. Martí-Oliet. Polytypic programming in Maude. In K. Futatsugi, editor, *Third International Workshop on Rewriting Logic and its Applications*, volume 36 of *Electronic Notes in Theoretical Computer Science*, pages 339–360. Elsevier, 2000.

[18] M. Clavel, S. Eker, P. Lincoln, and J. Meseguer. Principles of Maude. In J. Meseguer, editor, *First International Workshop on Rewriting Logic and its Applications*, volume 4 of *Electronic Notes in Theoretical Computer Science*, pages 65–89. Elsevier, 1996.

[19] M. Clavel, N. Martí-Oliet, and M. Palomino. Formalizing and proving semantics relations between specifications by reflection. Manuscript, Universidad Complutense de Madrid, `http://www.ucm.es/info/dsip/clavel`, April 2003.

[20] M. Clavel and J. Meseguer. Axiomatizing reflective logics and languages. In G. Kiczales, editor, *Proceedings of Reflection'96*, pages 263–288, San Francisco (California), April 1996. Xerox PARC.

[21] M. Clavel and J. Meseguer. Reflection in conditional rewriting logic. *Theoretical Computer Science*, 285(2):245–288, 2002.

[22] M. Clavel, J. Meseguer, and M. Palomino. Reflection in membership equational logic, many-sorted equational logic, Horn logic with equality, and rewriting logic. In F. Gadducci and U. Montanari, editors, *Fourth International Workshop on Rewriting Logic and its Applications*, volume 71 of *Electronic Notes in Theoretical Computer Science*, pages 63–78. Elsevier, 2002.

[23] P. Cointe, editor. *Proc. Reflection'99*, volume 1616 of *LNCS*. Springer-Verlag, 1999.

[24] R. L. Constable. Using reflection to explain and enhance type theory. In H. Schwichtenberg, editor, *Proof and Computation*, volume 139 of *Computer and System Sciences*, pages 109–144. Springer-Verlag, 1995.

[25] F.-N. Demers and J. Malenfant. Reflection in logic, functional and object-oriented programming: a short comparative study. In *Proc. IJCAI'95 – Workshop on Reflection and Metalevel Architectures and their Applications in AI*, pages 29–38, 1995.

[26] J. Despeyroux, F. Pfenning, and C. Schürmann. Primitive recursion for higher-order abstract syntax. In *Proceedings of the 3rd International Conference on Typed Lambda Calculi and Applications (TLCA'97)*, volume 1210 of *Lecture Notes in Computer Science*, Nancy, France, April 1997. Springer-Verlag.

[27] F. Durán. *A Reflective Module Algebra with Applications to the Maude Language*. PhD thesis, University of Málaga, 1999.

[28] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 1*, volume 6 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1985.

[29] S. Feferman. Finitary inductively presented logics. In *Logic Colloquium '88*. North-Holland, 1988.

[30] L. Fribourg and F. Turini, editors. *Logic Program Synthesis and Transformation–Meta-programming in Logic*, volume 883 of *LNCS*. Springer-Verlag, 1994.

[31] P. Gardner. *Representing logics in type theory*. PhD thesis, University of Edinburgh, 1992. Also published as ECS-LFCS-92-227.

[32] F. Giunchiglia, P. Traverso, A. Cimatti, and P. Pecchiari. A system for multi-level reasoning. In *IMSA'92 – International Workshop on Reflection and Meta-Level Architecture*, pages 190–195. Information-Technology Promotion Agency, Japan, 1992.

[33] J. Goguen and R. Burstall. Introducing institutions. In E. Clarke and D. Kozen, editors, *Logic of Programming Workshop*, volume 164 of *Lecture Notes in Computer Science*, pages 221–256. Springer-Verlag, 1984.

[34] J. Goguen and R. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the ACM*, 39(1):95–146, 1992.

[35] J. Goguen and J. Meseguer. Models and equality for logical programming. In H. Ehrig, G. Levi, R. Kowalski, and U. Montanari, editors, *Proceedings TAPSOFT'87*, volume 250 of *Lecture Notes in Computer Science*, pages 1–22. Springer-Verlag, 1987.

[36] M. Gordon and T. Melham. *Introduction to HOL: A Theorem Proving Environment for Higher Order Logic*. Cambridge University Press, 1993.

[37] R. Harper, F. Honsell, and G. Plotkin. A framework for defining logics. *J. ACM*, 40(1):143–184, January 1993.

[38] J. Harrison. Metatheory and reflection in theorem proving: a survey and critique. Technical Report CRC-053, SRI Cambridge, Cambridge UK, 1995.

[39] P. Hill and J. Lloyd. Analysis of meta-programs. In H. D. Abramson and M. H. Rogers, editors, *Meta-Programming in Logic Programming*, pages 23–52. MIT Press, 1989.

[40] P. Hill and J. Lloyd. *The Gödel Programming Language*. MIT Press, 1994.

[41] D. J. Howe. Computational metatheory in Nuprl. In *9th International Conference On Automated Deduction*, pages 238–257, Argonne, Illinois, 1988.

[42] D. J. Howe. Reflecting the semantics of reflected proof. In P. Aczel, H. Simmons, and S. Wainer, editors, *Proof Theory*, pages 229–250. Cambridge University Press, 1990.

[43] J. Jeuring and P. Jansson. Polytypic programming. In J. Launchbury, E. Meijer, and T. Sheard, editors, *Proc. Second Int. Summer School on Advanced Functional Programming Techniques*, volume 1129 of *LNCS*, pages 68–114. Springer-Verlag, 1996.

[44] G. Kiczales, editor. *Proc. Reflection'96*. Xerox PARC, 1996.

[45] G. Kiczales, J. des Rivieres, and D. G. Bobrow. *The Art of the Metaobject Protocol*. MIT Press, 1991.

[46] T. B. Knoblock and R. L. Constable. Formalized metareasoning in type theory. In *Proc. of the First Annual Symp. on Logic in Computer Science*. IEEE., 1986.

[47] J. J. A. Makowsky. Model theoretic issues in theoretical computer science, part I: Relational data bases and abstract data types. In *Logic Colloquium '82*, pages 303–343, Florence Italy, 1984. North Holland.

[48] N. Martí-Oliet and J. Meseguer. General logics and logical frameworks. In D. Gabbay, editor, *What is a Logical System?*, pages 355–392. Oxford University Press, 1994.

[49] N. Martí-Oliet and J. Meseguer. Rewriting logic as a logical and semantic framework. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, 2nd. Edition*, pages 1–87. Kluwer Academic Publishers, 2002. First published as SRI Tech. Report SRI-CSL-93-05, August 1993.

[50] N. Martí-Oliet and J. Meseguer. Rewriting logic: roadmap and bibliography. *Theoretical Computer Science*, 285:121–154, 2002.

[51] S. Matthews. Reflection in logical systems. In Smith and Yonezawa [70], pages 178–183.

[52] S. Matthews, A. Smaill, and D. Basin. Experience with FS_0 as a framework theory. In G. Huet and G. Plotkin, editors, *Logical Environments*, pages 61–82. Cambridge University Press, 1993.

[53] R. McDowell and D. Miller. A logic for reasoning with higher-order abstract syntax. In *Twelfth Annual IEEE Symposium on Logic in Computer Science*, June 1997.

[54] J. Meseguer. General logics. In H.-D. Ebbinghaus et al., editor, *Logic Colloquium'87*, pages 275–329. North-Holland, 1989.

[55] J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73–155, 1992.

[56] J. Meseguer. Membership algebra as a semantic framework for equational specification. In F. Parisi-Presicce, editor, *Proceedings of WADT'97*, volume 1376 of *Lecture Notes in Computer Science*, pages 18–61. Springer-Verlag, 1998.

[57] J. Meseguer. Research directions in rewriting logic. In U. Berger and H. Schwichtenberg, editors, *Computational Logic, NATO Advanced Study Institute, Marktoberdorf, Germany, July 29 - August 6, 1997*. Springer-Verlag, 1998.

[58] C. Paulin-Mohring. Inductive Definitions in the System Coq — Rules and Properties. In M. Bezem and J.-F. Groote, editors, *Proceedings of the conference Typed Lambda Calculi and Applications*, volume 664 of *Lecture Notes in Computer Science*, 1993. LIP research report 92-49.

[59] L. C. Paulson. A fixedpoint approach to implementing (co)inductive definitions. In *Proceedings of the 12th International Conference on Automated Deduction (CADE-12)*, volume 814 of *Lecture Notes in Artificial Intelligence*, Nancy, France, June 1994. Springer-Verlag.

[60] L. C. Paulson. *Isabelle : a generic theorem prover; with contributions by Tobias Nipkow*, volume 828 of *Lecture Notes in Computer Science*. Springer, Berlin, 1994.

[61] A. Pettorossi, editor. *Proc. Third Workshop on Meta-programming in Logic*, volume 649 of *LNCS*. Springer-Verlag, 1992.

[62] F. Pfenning and C. Schürmann. System description: Twelf — a meta-logical framework for deductive systems. In H. Ganzinger, editor, *Proceedings of the 16th International Conference on Automated Deduction (CADE-16)*, volume 1632 of *LNAI*, pages 202–206. Springer-Verlag, 1999.

[63] H. Rueß. *Formal Meta-Programming in the Calculus of Constructions*. PhD thesis, Universität Ulm, Germany, 1995.

[64] H. Rueß. Computational reflection in the calculus of constructions and its application to theorem proving. In R. Hindley, editor, *Proceedings for the Third International Conference on Typed Lambda Calculus and Applications (TLCA'97)*, Lecture Notes in Computer Science, Nancy, France, April 1997. Springer-Verlag.

[65] C. Schürmann. *Automating the Meta Theory of Deductive Systems*. PhD thesis, Carnegie Mellon University, 2000. Also published as CMU-CS-00-146.

[66] C. Schürmann. A type-theoretic approach to induction with higher-order encodings. In *Proceedings of Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2001)*, volume 2250 of *Lecture Notes in Computer Science*, pages 266–281. Springer, 2001.

[67] C. Schürmann and F. Pfenning. Automated theorem proving in a simple meta-logic for LF. In C. Kirchner and H. Kirchner, editors, *Proceedings of the 15th International Conference on Automated Deduction (CADE-15)*, volume 1421 of *Lecture Notes in Computer Science*, pages 286–300, Lindau, Germany, July 1998. Springer-Verlag.

[68] D. Scott. Completeness and axiomatizability in many-valued logic. In L. H. *et al.*, editor, *Proceedings of the Tarski Symposium*, pages 411–435. American Mathematical Society, 1974.

[69] N. Shankar. *Metamathematics, Machines, and Gödel's Proof.* Cambridge University Press, 1994.

[70] B. Smith and A. Yonezawa, editors. *Proc. IMSA'92 – International Workshop on Reflection and Meta-Level Architecture.* Information-Technology Promotion Agency, Japan, 1992.

[71] B. C. Smith. Reflection and semantics in Lisp. In *Proc. POPL'84*, pages 23–35. ACM, 1984.

[72] C. Smorynski. The incompleteness theorems. In J. Barwise, editor, *Handbook of Mathematical Logic*, pages 821–865. North-Holland, 1977.

[73] R. M. Smullyan. *Diagonalization and Self-Reference.* Oxford University Press, 1994.

[74] G. L. Steele, Jr. and G. J. Sussman. The art of the interpreter or, the modularity complex. Technical Report AIM-453, MIT AI-Lab, 1978.

[75] M.-O. Stehr. CINNI — A generic calculus of explicit substitutions and its application to λ-, ς- and π-calculi. In K. Futatsugi, editor, *Third International Workshop on Rewriting Logic and its Applications*, volume 36 of *Electronic Notes in Theoretical Computer Science*, pages 71–92. Elsevier, 2000.

[76] M.-O. Stehr and J. Meseguer. Pure type systems in rewriting logic. In *Proc. of LFM'99: Workshop on Logical Frameworks and Meta-languages, Paris, France*, 1999. `http://www.cs.bell-labs.com/~felty/LFM99/`.

[77] M.-O. Stehr, P. Naumov, and J. Meseguer. A proof-theoretic approach to the HOL-Nuprl connection with applications to proof translation. Manuscript, SRI International, `http://www.csl.sri.com/~stehr/fi\_eng.html`, February 2000.

[78] V. F. Turchin. The concept of a supercompiler. *ACM Transactions on Programming Languages and Systems*, 8(3):292–325, 1986.

[79] M. van Emden and R. Kowalski. The semantics of predicate logic as a programming language. *J. ACM*, 23:733–42, 1976.

[80] M. Wand and D. P. Friedman. The mystery of the tower revealed. *Lisp and Symbolic Computation*, 1(1):11–38, 1988.

[81] R. W. Weyhrauch. Prolegomena to a theory of mechanized formal reasoning. *Artificial Intelligence*, 13:133–170, 1980.

A Soundness Theorem

In the proof of our main theorem we will use the following two propositions about the Boolean function (**wft** _ **in** _) in MB-META.

Proposition 8 *For all membership equational signatures* Ω *and terms* t *in* $T_\Omega(X)$,

$$\texttt{MB-META} \vdash (\textbf{wft}\ \bar{t}\ \textbf{in}\ \overline{\Omega}) = \texttt{true}\,.$$

Proposition 9 *For all membership equational signatures* Ω *and ground terms* u *in* MB-META*, if*

$$\texttt{MB-META} \vdash (\textbf{wft}\ u\ \textbf{in}\ \overline{\Omega}) = \texttt{true}\,,$$

then there is a ground term t *in* T_Ω *such that,* $\bar{t} = u$.

We now prove two propositions that we will use in our proof of the main theorem.

Proposition 10 *For* $P = (\Omega \oplus V, E \cup Mb(V))$ *a parameter theory in* $\mathcal{P}_P$*, with* $Mb(V) = \{v_1 : s_1, \ldots, v_n : s_n\}$*, for all ground substitutions* h*,* $h : \overline{V}^{[V]} \longrightarrow$ [Term]*, such that*

$$\texttt{MB-META} \models h(\overline{E \cup Mb(V)}^{\mathcal{C}(P)})\,,$$

there is a theory morphism $\beta \in \mathcal{V}_P$*,* $\beta : P \longrightarrow Q$*,* $Q = (\Omega \oplus V, E \cup Eq(V))$*, such that* $\overline{\beta}$ *is the ground substitution* h*.*

Proof First, by the definition of substitution application,

$$\begin{aligned}
&h(\overline{E \cup Mb(V)}^{\mathcal{C}(P)}) \\
&\quad = \ h(\overline{v_1}^{\mathcal{W}(\Omega,V)} \wedge \cdots \wedge \overline{v_n}^{\mathcal{W}(\Omega,V)} \wedge \overline{v_1 : s_1}^{\mathcal{B}(P,\emptyset)} \wedge \cdots \wedge \overline{v_n : s_n}^{\mathcal{B}(P,\emptyset)}) \\
&\quad = \ (h(\overline{v_1}^{\mathcal{W}(\Omega,V)}) \wedge \cdots \wedge h(\overline{v_n}^{\mathcal{W}(\Omega,V)}) \wedge h(\overline{v_1 : s_1}^{\mathcal{B}(P,\emptyset)}) \wedge \cdots \wedge h(\overline{v_n : s_n}^{\mathcal{B}(P,\emptyset)}))\,.
\end{aligned}$$

Now, by the definition of substitution application, for $1 \leq i \leq n$,

$$\begin{aligned}
h(\overline{v_i}^{\mathcal{W}(\Omega,V,))} &= (h(\textbf{wft}\ \overline{v_i}^{[V]}\ \textbf{in}\ \overline{\Omega}) = \texttt{true}) \\
&= (\textbf{wft}\ h(\overline{v_i}^{[V]})\ \textbf{in}\ \overline{\Omega} = \texttt{true})\,.
\end{aligned}$$

Note that, by Proposition 9, using completeness of membership equational logic, and the fact that $h(\overline{v_i}^{\mathcal{W}(\Omega,V,))}$ is a ground atomic formula, if

$$\texttt{MB-META} \models h(\overline{v_i}^{\mathcal{W}(\Omega,V)})$$

then there is a ground term $t_i \in T_\Omega$ such that $\overline{t_i} = h(\overline{v_i}^{[V]})$. Hence, by the definition of substitution application, for $1 \leq i \leq n$,

$$\begin{aligned}
&h(\overline{v_i : s_i}^{\mathcal{B}(P,\emptyset)}) \\
&\quad = \ (h(\overline{v_i}^{[V,\emptyset]}\ \textbf{:}\ \overline{s_i}\ \textbf{in}\ \overline{P}^{\mathcal{P}}) = \texttt{true}) \\
&\quad = \ (h(\overline{v_i}^{[V]})\ \textbf{:}\ \overline{s_i}\ \textbf{in}\ (\overline{\Omega \oplus V}, \overline{E}\ \ \textbf{eq}\ \overline{v}_1\ \textbf{=}\ h(\overline{v_1}^{[V]})\,\textbf{.} \cdots \textbf{eq}\ \overline{v}_n\ \textbf{=}\ h(\overline{v_n}^{[V]})\,\textbf{.}) = \texttt{true}) \\
&\quad = \ (\overline{t_i}\ \textbf{:}\ \overline{s_i}\ \textbf{in}\ (\overline{\Omega \oplus V}, \overline{E}\ \ \textbf{eq}\ \overline{v}_1\ \textbf{=}\ \overline{t_1}\,\textbf{.} \cdots \textbf{eq}\ \overline{v}_n\ \textbf{=}\ \overline{t_n}\,\textbf{.}) = \texttt{true})\,.
\end{aligned}$$

Let $Q = (\Omega \oplus V, E \cup \{v_1 = t_1, \ldots, v_n = t_n\})$. Note that, by Proposition 1, using completeness of membership equational logic, and the fact that $h(\overline{v_i : s_i}^{\mathcal{B}(P,\emptyset)})$ is a ground atomic formula, if

$$\texttt{MB-META} \models h(\overline{v_i : s_i}^{\mathcal{B}(P,\emptyset)})$$

then $Q \vdash t_i : s_i$. Finally, let β be the theory morphism $\beta \in \mathcal{V}_P$, $\beta : P \longrightarrow Q$, such that, for $1 \leq i \leq n$, $(v_i)_\beta = t_i$. By definition, $\overline{\beta}$ is the ground substitution h. □

Proposition 11 *Let $P = (\Omega \oplus V, E \cup Mb(V))$ be a parameter theory in $\mathcal{P}_P$, and let $\{T_0[P], T_1[P], \ldots, T_k[P]\}$ be a finite multiset of parameterized theories in $\mathcal{P}_P$ that satisfies the following property: for all theory morphisms β in $\mathcal{V}_P$, $\beta : P \longrightarrow Q$, and terms t and t' in $T_{\Omega \oplus V}(X)$, for $1 \leq l \leq k$,*

$$T_0[\beta] \vdash t = t' \Longrightarrow T_l[\beta] \vdash t = t' \,. \tag{25}$$

For all ground atomic formulae $\phi \in sen(T_l[P])$, $1 \leq l \leq k$, parameters $v \in V$, and terms t, t' in $T_{\Omega \oplus V}(X)$ of the same kind as v, if

$$T_0[\beta] \vdash t = t' \,, \tag{26}$$

then

$$\texttt{MB-META} \models \overline{(\phi(v \mapsto t))_\beta}^{\mathcal{B}(T_l[\beta],\emptyset)} \Longleftrightarrow \texttt{MB-META} \models \overline{(\phi(v \mapsto t'))_\beta}^{\mathcal{B}(T_l[\beta],\emptyset)} \,. \tag{27}$$

Proof By soundness and completeness of membership equational logic, and the fact that

$$\overline{(\phi(t))_\beta}^{\mathcal{B}(T_l[\beta],\emptyset)} \text{ and } \overline{(\phi(t'))_\beta}^{\mathcal{B}(T_l[\beta],\emptyset)}$$

are both ground atomic formulae, (27) is equivalent to

$$\texttt{MB-META} \vdash \overline{(\phi(t))_\beta}^{\mathcal{B}(T_l[\beta],\emptyset)} \Longleftrightarrow \texttt{MB-META} \vdash \overline{(\phi(t'))_\beta}^{\mathcal{B}(T_l[\beta],\emptyset)} \,.$$

Let $\phi = (u : s)$ (similarly for $\phi = (u = u')$). By definition,

$$\overline{(\phi(t))_\beta}^{\mathcal{B}(T_l[\beta],\emptyset)} = (\overline{(u(t))_\beta} \,\texttt{:}\, \overline{s} \textbf{ in } \overline{T_l[\beta]} = \texttt{true}) \,.$$

Thus, by Proposition 1,

$$\texttt{MB-META} \vdash \overline{(u(t))_\beta} \,\texttt{:}\, \overline{s} \textbf{ in } \overline{T_l[\beta]} = \texttt{true}$$

holds if and only if $T_l[\beta] \vdash (u(t))_\beta : s$ also holds. However, assuming (25) and (26), by Remark (5), this holds if and only if $T_l[\beta] \vdash (u(t'))_\beta : s$ holds as well. But, by Proposition 1, $T_l[\beta] \vdash (u(t'))_\beta : s$ holds if and only if

$$\texttt{MB-META} \vdash \overline{(u(t'))_\beta} \,\texttt{:}\, \overline{s} \textbf{ in } \overline{T_l[\beta]} = \texttt{true} \,,$$

and, by definition,

$$\overline{(\phi(t'))_\beta}^{\mathcal{B}(T_l[\beta],\emptyset)} = (\overline{(u(t'))_\beta} \,\texttt{:}\, \overline{s} \textbf{ in } \overline{T_l[\beta]} = \texttt{true}) \,.$$

□

Theorem 2 *Let* $P = (\Omega \oplus V, E \cup Mb(V))$ *be a parameter theory in* $\mathcal{P}_P$*, with* $\Omega = (K, \Sigma, S)$*, and let* $\{T_0[P], T_1[P], \ldots, T_k[P]\}$ *be a finite multiset of parameterized theories in* $\mathcal{P}_P$ *that satisfies the following property: for all theory morphisms* β *in* $\mathcal{V}_P$*,* $\beta : P \longrightarrow Q$*, and terms* t *and* t' *in* $T_{\Omega \oplus V}(X)$*, for* $1 \leq l \leq k$*,*

$$T_0[\beta] \vdash t = t' \Longrightarrow T_l[\beta] \vdash t = t' \,.$$

Let $T_0[P] = (\Omega \oplus V, E \cup G_0 \cup Mb(V))$*, let* s *be a sort in some* S_k*, and let* $C_{[T_0[P],s]} = \{C_1, \ldots, C_n\}$ *be those sentences in* $(E \cup G_0)$ *specifying the sort* s*, i.e., those* C_i *of the form*

$$\forall (x_1, \ldots, x_{p_i}).\, A_1 \wedge \ldots \wedge A_{q_i} \Longrightarrow A_0 \,, \tag{28}$$

where, for some term t *of kind* k*,* A_0 *is* $t : s$*. Finally, for all sentences* C_i *in* $(E \cup G_0)$ *of the form* (28)*, let* X_i *be the set of variables* $\{x_1, \ldots, x_{p_i}\}$*, where* $X_i \cap (\Sigma \oplus V) = \emptyset$*.*

Then for all finite multisets of ground atomic formulae $\{\phi_1, \ldots, \phi_k\}$ *with* $\phi_l \in sen(T_l[P])$*,* $1 \leq l \leq k$*, parameters* $v_i \in V$*, Boolean expressions* b*, and first-order formulae* $\tau(\overline{v_i}^{[V]})$ *of the form*

$$\forall \overline{(V \setminus \{v_i\})}^{[V]}.\, (\overline{E \cup Mb(V)}^{\mathcal{C}(P)} \\ \Longrightarrow (\overline{v_i : s}^{\mathcal{B}(T_0[P],\emptyset)} \Longrightarrow (b(\overline{\phi_1}^{\mathcal{B}(T_1[P],\emptyset)}, \ldots, \overline{\phi_k}^{\mathcal{B}(T_k[P],\emptyset)})))) \,,$$

with free variable $\overline{v_i}^{[V]}$ *of the kind* `[Term]`*, if*

$$\texttt{MB-META} \models\simeq \forall \overline{V}^{[V]}.\, (\overline{E \cup Mb(V)}^{\mathcal{C}(P)} \implies (\psi_1 \wedge \ldots \wedge \psi_n)) \,, \tag{29}$$

then

$$\texttt{MB-META} \models\simeq \forall \overline{v_i}^{[V]}.\tau(\overline{v_i}^{[V]}) \,, \tag{30}$$

where, for $1 \leq i \leq n$ *and* C_i *in* $(E \cup G_0)$ *of the form* (28)*,* ψ_i *is defined as in Definition* 8*.*

Proof (29) implies that for all ground substitutions h, $h : \overline{V}^{[V]} \longrightarrow$ `[Term]`,

$$\texttt{MB-META} \models\simeq h(\overline{E \cup Mb(V)}^{\mathcal{C}(P)} \implies (\psi_1 \wedge \ldots \wedge \psi_n)) \,.$$

By Propositions 7 and 6, this in turn implies that, for all theory morphisms $\beta \in \mathcal{V}_P$, $\beta : P \longrightarrow Q$,

$$\texttt{MB-META} \models\simeq \overline{\beta}(\psi_1 \wedge \ldots \wedge \psi_n) \,. \tag{31}$$

Observe that by the definition of substitution application, (31) implies that for $1 \leq i \leq n$ and C_i in $(E \cup G_0)$ of the form (28),

$$\texttt{MB-META} \models\simeq \forall (\overline{x}_1^{[X_i]}, \ldots, \overline{x}_{p_i}^{[X_i]}).\, \overline{\beta}([A_1]_\tau) \wedge \ldots \wedge \overline{\beta}([A_{q_i}]_\tau) \Longrightarrow \overline{\beta}([A_0]_\tau) \,.$$

Now notice that proving (30) is equivalent to proving that for all ground substitutions h, $h : \overline{V}^{[V]} \longrightarrow$ `[Term]`, such that

$$\texttt{MB-META} \models\simeq h(\overline{E \cup Mb(V)}^{\mathcal{C}(P)} \;\wedge\; \overline{v_i : s}^{\mathcal{B}(T_0[P],\emptyset)}),$$

it holds that

$$\texttt{MB-META} \models h(b(\overline{\phi_1}^{\mathcal{B}(T_1[P],\emptyset)},\ldots,\overline{\phi_k}^{\mathcal{B}(T_k[P],\emptyset)}))\,.$$

By Proposition 10, this can be reduced to proving that for all theory morphisms $\beta \in \mathcal{V}_P$, $\beta : P \longrightarrow Q$, such that

$$\texttt{MB-META} \models \overline{\beta}(\overline{v_i\!:\!s}^{\mathcal{B}(T_0[P],\emptyset)}), \tag{32}$$

it holds that

$$\texttt{MB-META} \models \overline{\beta}(b(\overline{\phi_1(v_i)}^{\mathcal{B}(T_1[P],\emptyset)},\ldots,\overline{\phi_k(v_i)}^{\mathcal{B}(T_k[P],\emptyset)}))\,. \tag{33}$$

Notice that, by the definition of substitution application, (33) is equivalent to

$$\texttt{MB-META} \models b(\overline{\beta}(\overline{\phi_1(v_i)}^{\mathcal{B}(T_1[P],\emptyset)}),\ldots,\overline{\beta}(\overline{\phi_k(v_i)}^{\mathcal{B}(T_k[P],\emptyset)}))\,, \tag{34}$$

where, for $1 \leq l \leq k$ and $\phi_l = (t_l : s_l)$ (similarly for $\phi_l = (t_l = t'_l)$), by Propositions 3 and 4, and by Remark (4),

$$\begin{array}{rcl}
\overline{\beta}(\overline{\phi_l(v_i)}^{\mathcal{B}(T_l[P],\emptyset)}) & = & \overline{\beta}(\overline{t_l(v_i)\!:\!s_l}^{\mathcal{B}(T_l[P],\emptyset)}) \\
& = & \overline{\beta}(\overline{t_l(v_i)}^{[V]}\ \texttt{:}\ \overline{s_l}\ \texttt{in}\ \overline{T_l[P]}^{\mathcal{P}} = \texttt{true}) \\
& = & (\overline{\beta}(\overline{t_l(v_i)}^{[V]})\ \texttt{:}\ \overline{s_l}\ \texttt{in}\ \overline{\beta}(\overline{T_l[P]}^{\mathcal{P}}) = \texttt{true}) \\
& = & (\overline{(t_l(v_i))_\beta}\ \texttt{:}\ \overline{s_l}\ \texttt{in}\ \overline{T_l[\beta]} = \texttt{true}) \\
& = & (\overline{(t_l((v_i)_\beta))_\beta}\ \texttt{:}\ \overline{s_l}\ \texttt{in}\ \overline{T_l[\beta]} = \texttt{true}) \\
& = & \overline{(t_l((v_i)_\beta))_\beta\!:\!s_l}^{\mathcal{B}(T_l[\beta],\emptyset)} \\
& = & \overline{(\phi_l((v_i)_\beta))_\beta}^{\mathcal{B}(T_l[\beta],\emptyset)}\,.
\end{array}$$

Thus, (34) is equivalent to

$$\texttt{MB-META} \models b(\overline{(\phi_1((v_i)_\beta))_\beta}^{\mathcal{B}(T_1[\beta],\emptyset)},\ldots,\overline{(\phi_k((v_i)_\beta))_\beta}^{\mathcal{B}(T_k[\beta],\emptyset)})\,.$$

Notice also that, by Proposition 5, (32) is equivalent to $T_0[\beta] \vdash (v_i)_\beta\!:\!s$. Recall now that $(v_i)_\beta$ can be any ground term t in T_Ω. Thus, to complete the proof we show that for all terms t in $T_{\Omega\oplus V}(X)$ that have sort s in $T_0[\beta]$, if (31) holds, then

$$\texttt{MB-META} \models b(\overline{(\phi_1(t))_\beta}^{\mathcal{B}(T_1[\beta],\emptyset)},\ldots,\overline{(\phi_k(t))_\beta}^{\mathcal{B}(T_k[\beta],\emptyset)})\,, \tag{35}$$

also holds. We prove this by induction on the proof that t has sort s in $T_0[\beta]$.

Let C_i be a sentence in $T_0[\beta]$ of the form

$$\forall(x_1,\ldots,x_{p_i}).\,A_1 \wedge \ldots \wedge A_{q_i} \Longrightarrow w\!:\!s\,,$$

and let σ be a substitution, $\sigma : X_i \longrightarrow T_\Omega(X)$,[19] such that

[19]Note that we do not have to consider substitutions $\sigma' : X_i \longrightarrow T_{\Omega\oplus V}(X)$, since, for all such substitutions, if $T_0[\beta] \vdash t = \sigma'(w)$, then there is a substitution $\sigma : X_i \longrightarrow T_\Omega(X)$, with $\sigma(x) = (\sigma'(x))_\beta$, for all $x \in X_i$, such that, by Remark (5), $T[\beta] \vdash t = \sigma(w)$, and, more generally, for all atomic formulae ϕ,

$$T_0[\beta] \vdash \sigma'(\phi) \Longleftrightarrow T_0[\beta] \vdash \sigma(\phi)\,.$$

- $T_0[\beta] \vdash t = \sigma(w)$, and
- $T_0[\beta] \vdash \sigma(A_j)$, for $1 \leq j \leq q_i$.

Let $\overline{\sigma}$ be the ground substitution, $\overline{\sigma} : \overline{X_i}^{[X_i]} \longrightarrow$ `[Term]`, where for all $x \in X_i$, $\overline{\sigma}(\overline{x}^{[X_i]}) = \overline{\sigma(x)}$. Note that for all terms $t \in T_{\Omega \oplus V}(X_i)$,

$$\overline{\sigma}(\overline{t}^{[V,X_i]}) = \overline{\sigma(t)}^{[V]} . \tag{36}$$

Since C_i is a sentence in $(E \cup G_0)$, by (31), we have that

$$\texttt{MB-META} \models \overline{\sigma}(\overline{\beta}([A_1]_\tau)) \wedge \ldots \wedge \overline{\sigma}(\overline{\beta}([A_{q_i}]_\tau)) \Longrightarrow \overline{\sigma}(\overline{\beta}([w : s]_\tau)) , \tag{37}$$

where, by the definition of substitution application,

$$\begin{array}{rcl} \overline{\sigma}(\overline{\beta}([w : s]_\tau)) & = & \overline{\sigma}(\overline{\beta}(b(\overline{\phi_1(w)}^{\mathcal{B}(T_1[P],X_i)}, \ldots, \overline{\phi_k(w)}^{\mathcal{B}(T_k[P],X_i)}))) \\ & = & b(\overline{\sigma}(\overline{\beta}(\overline{\phi_1(w)}^{\mathcal{B}(T_1[P],X_i)})), \ldots, \overline{\sigma}(\overline{\beta}(\overline{\phi_k(w)}^{\mathcal{B}(T_k[P],X_i)}))) , \end{array}$$

and, for $1 \leq l \leq k$ and $\phi_l = (t_l : s_l)$ (similarly for $\phi_l = (t_l = t'_l)$), by the definition of substitution application, Propositions 3 and 4, and Remark (36),

$$\begin{array}{rcl} \overline{\sigma}(\overline{\beta}(\overline{\phi_l(w)}^{\mathcal{B}(T_l[P],X_i)})) & = & \overline{\sigma}(\overline{\beta}(\overline{t_l(w) : s_l}^{\mathcal{B}(T_l[P],X_i)})) \\ & = & \overline{\sigma}(\overline{\beta}(\overline{t_l(w)}^{[V,X_i]}\ \texttt{:}\ \overline{s_l}\ \texttt{in}\ \overline{T_l[P]}^{\mathcal{P}} = \texttt{true})) \\ & = & (\overline{\sigma}(\overline{\beta}(\overline{t_l(w)}^{[V,X_i]}))\ \texttt{:}\ \overline{s_l}\ \texttt{in}\ \overline{\beta}(\overline{T_l[P]}^{\mathcal{P}}) = \texttt{true}) \\ & = & (\overline{\sigma}(\overline{(t_l(w))_\beta}^{[X_i]})\ \texttt{:}\ \overline{s_l}\ \texttt{in}\ \overline{T_l[\beta]} = \texttt{true}) \\ & = & (\overline{\sigma((t_l(w))_\beta)}\ \texttt{:}\ \overline{s_l}\ \texttt{in}\ \overline{T_l[\beta]} = \texttt{true}) \\ & = & (\overline{(t_l(\sigma(w)))_\beta}\ \texttt{:}\ \overline{s_l}\ \texttt{in}\ \overline{T_l[\beta]} = \texttt{true})^{20} \\ & = & \overline{(t_l(\sigma(w)))_\beta : s_l}^{\mathcal{B}(T_l[\beta],\emptyset)} \\ & = & \overline{(\phi_l(\sigma(w)))_\beta}^{\mathcal{B}(T_l[\beta],\emptyset)} . \end{array}$$

Notice then that, by Proposition 11, using the initial assumption $T_0[\beta] \vdash t = \sigma(w)$, (35) holds if and only if

$$\texttt{MB-META} \models \overline{\sigma}(\overline{\beta}([w : s]_\tau)) .$$

Therefore, given (37), we can reduce proving (35) to proving that, for $1 \leq j \leq q_i$,

$$\texttt{MB-META} \models \overline{\sigma}(\overline{\beta}([A_j]_\tau)) .$$

We proceed by cases:

[20] By the definition of substitution application, given the assumptions that t_l is a ground term, σ is a substitution, $\sigma : X_i \longrightarrow T_\Omega(X)$, and β is a theory morphism $\beta \in \mathcal{V}$, $\beta : P \longrightarrow Q$. Recall that, for all terms $t \in T_{\Omega \oplus V}(X)$, considering the equations $Eq(V) = \{v_i = t_i \mid v_i \in V\}$ in Q, we denote by t_β the term in $T_\Omega(X)$ that results from replacing each v_i (if any) in t with the ground term t_i in T_Ω.

- $A_j = u\!:\!s$. By the definition of substitution application,

$$\begin{aligned}\overline{\sigma}(\overline{\beta}([u\!:\!s]_\tau)) &= \overline{\sigma}(\overline{\beta}(b(\overline{\phi_1(u)}^{\mathcal{B}(T_1[P],X_i)}, \ldots, \overline{\phi_k(u)}^{\mathcal{B}(T_k[P],X_i)}))) \\ &= b(\overline{\sigma}(\overline{\beta}(\overline{\phi_1(u)}^{\mathcal{B}(T_1[P],X_i)})), \ldots, \overline{\sigma}(\overline{\beta}(\overline{\phi_k(u)}^{\mathcal{B}(T_k[P],X_i)}))),\end{aligned}$$

  and, for $1 \le l \le k$ and $\phi_l = (t_l : s_l)$ (similarly for $\phi_l = (t_l = t'_l)$), by the definition of substitution application, Propositions 3 and 4, and Remark (36),

$$\overline{\sigma}(\overline{\beta}(\overline{\phi_l(u)}^{\mathcal{B}(T_l[P],X_i)})) = \overline{(\phi_l(\sigma(u)))_\beta}^{\mathcal{B}(T_l[\beta],\emptyset)}).$$

  (The proof is analogous to the one above.) From the assumption $T_0[\beta] \vdash \sigma(u)\!:\!s$, we can use the induction hypothesis to obtain the desired result.

- $A_j = u : s'$, $s' \neq s$ (similarly for $A_j = (u = u')$). By the definition of substitution application, Propositions 3 and 4, and Remark (36),

$$\begin{aligned}\overline{\sigma}(\overline{\beta}([u\!:\!s']_\tau)) &= \overline{\sigma}(\overline{\beta}(\overline{u\!:\!s'}^{\mathcal{B}(T_0[P],X_i)})) \\ &= (\overline{\sigma}(\overline{\beta}(\overline{u}^{[V,X_i]})\ \mathbf{:}\ \overline{s'}\ \mathtt{\textbf{in}}\ \overline{\beta}(\overline{T_0[P]}^{\mathcal{P}})) = \mathtt{true}) \\ &= (\overline{\sigma}(\overline{u_\beta}^{[X_i]}\ \mathbf{:}\ \overline{s'}\ \mathtt{\textbf{in}}\ \overline{T_0[\beta]}) = \mathtt{true}) \\ &= (\overline{\sigma(u_\beta)}\ \mathbf{:}\ \overline{s'}\ \mathtt{\textbf{in}}\ \overline{T_0[\beta]} = \mathtt{true}) \\ &= (\overline{(\sigma(u))_\beta}\ \mathbf{:}\ \overline{s'}\ \mathtt{\textbf{in}}\ \overline{T_0[\beta]} = \mathtt{true}) \\ &= (\overline{\beta}(\overline{\sigma(u)}^{[V]})\ \mathbf{:}\ \overline{s'}\ \mathtt{\textbf{in}}\ \overline{\beta}(\overline{T_0[\beta]}^{\mathcal{P}}) = \mathtt{true}) \\ &= \overline{\beta}(\overline{\sigma(u)\!:\!s'}^{\mathcal{B}(T_0[P],\emptyset)}).\end{aligned}$$

  From the assumption $T_0[\beta] \vdash \sigma(u) : s'$, $T_0[\beta] \vdash (\sigma(u))_\beta : s'$ follows by Remark (5). Hence we can apply Proposition 5 to obtain the desired result.

□