Modal Logic with Bounded Quantification over Worlds

ROGIER M. VAN EIJK, FRANK S. DE BOER, WIEBE VAN DER HOEK AND JOHN-JULES CH. MEYER, *Institute of Information and Computing Sciences, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands.*
E-mail: {rogier,frankb,wiebe,jj}@cs.uu.nl
Wiebe van der Hoek is also affiliated with: Department of Philosophy, Utrecht University, The Netherlands and Department of Computer Science, University of Liverpool, UK.

Abstract

In this paper[1], we present a logical framework that combines modality with a first-order variable-binding mechanism. The logic, which belongs to the family of hybrid languages, differs from standard first-order modal logics in that quantification is not performed inside the worlds of a model, but the worlds in the model themselves constitute the domain of quantification. The locality principle of modal logic is preserved via the condition that in each world, the domain of quantification is given by a subset of the entire set of worlds in the model. In comparison with standard hybrid languages, the logic covers separate mechanisms for navigation and for variable-binding and formalizes reasoning about the worlds of a model in terms of equational logic. We show that the logic is semantically characterized by a generalization of classical bisimulation, called history-based bisimulation, and study the application of the logic to describe and reason about network topologies.

Keywords: Modal logics, equational logic, bounded quantification over worlds, history-based bisimulation, network topologies, hybrid languages.

1 Introduction

In order to increase the expressiveness of modal logics, during the last decades, a new family of logics has been introduced that combine modal operators with first-order variable-binding mechanisms. Characteristic of these logics, which are referred to as the family of *hybrid languages* [4], is that quantification is not performed inside the worlds of a model like in standard first-order modal logic [7], but instead, the worlds themselves constitute the domain of quantification. In particular, standard hybrid languages, which have originally been developed to increase the expressiveness of tense logics [5], extend modal logic with a collection of *nominals* that are used to label the worlds of a model. These nominals are propositional formulas that are true at exactly one world, and as such are employed as global, unique names for worlds. Further extensions cover operators to quantify over the worlds in a model, to bind variables to the current world, to jump to worlds denoted by a particular nominal or variable, and operators that combine modal, binding and jumping aspects. In this paper, we present a logical framework that belongs to this family of hybrid languages, which, due to its different starting-point and underpinning motivations provides a new perspective on hybrid languages.

[1] This paper is a revised version of [6].

The starting point of the framework is an explicit separation between the mechanisms of navigation and variable-binding. That is, in a particular world of a model, we distinguish between the worlds that are directly *accessible* from it and the worlds over which can be *quantified*. This starting point yields a general framework that can be instantiated in different ways. The framework for instance allows a global domain of quantification, but in particular, also *bounded* forms of quantification where in each world of a model, there is a restricted domain over which is quantified, like for instance the set of worlds that are directly accessible or the worlds that are accessible by following a finite number of successive links. Additionally, in comparison with standard hybrid languages, the logic is tailored to reason about the worlds of a model in terms of *equational logic*.

The paper is organized as follows. In Section 2, we motivate our research in extensions of modal logics by considering network topologies and argue that existing logics such as basic modal logic and graded modal logic are not suited to reason about network topologies. In Section 3, we present the syntax and semantics of our logical framework. Subsequently, in Section 4, we establish a semantic characterization of the logic, which is based on a generalization of the classical notion of bisimulation equivalence. Instead of relating worlds, this new type of bisimulation relates tuples that are comprised of a world together with a sequence of worlds. These additional sequences are employed to represent variable bindings that are generated during the evaluation of formulas. In Section 5, we consider the relation of the framework with the more standard hybrid languages and describe several interesting extensions of the framework that form the subject of future research.

2 Network topologies

An important application of the logical framework presented in this paper is the description of network topologies. Formally, such a network topology is represented by a directed graph whose nodes denote the agents in the system and whose edges make up the accessibility relation, describing what agents know about each other.

DEFINITION 2.1
A *network topology* is a tuple of the form:

$$\mathcal{N} = \langle W, R \rangle,$$

where W is a set of agents and $R \subseteq W \times W$ denotes the *accessibility* relation on W. We use the notation $R(w)$ to denote the set $\{u \in W \mid R(w,u)\}$ of agents that are known by the agent w.

Seen from a logical point of view, these network topologies constitute Kripke models (without a valuation function) that are employed in the semantics of modal logic [11]. This observation naturally leads to a description of network topologies by means of modal logic.

The basic modal language is defined as follows.

DEFINITION 2.2
Formulas φ in the *basic modal language* $\mathcal{L}_0$ are generated using the following BNF-grammar:

$$\varphi ::= true \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \Diamond\varphi.$$

Furthermore, we assume the usual abbreviations $false$ for $\neg true$, $\varphi_1 \vee \varphi_2$ for $\neg(\neg\varphi_1 \wedge \neg\varphi_2)$, $\varphi_1 \rightarrow \varphi_2$ for $\neg\varphi_1 \vee \varphi_2$ and $\varphi_1 \leftrightarrow \varphi_2$ for $\varphi_1 \rightarrow \varphi_2 \wedge \varphi_2 \rightarrow \varphi_1$.

A modal formula is either equal to $true$, the conjunction of two modal formulas, the negation of a modal formula, or the operator $\Diamond$ applied to a modal formula. It is the operator $\Diamond$ that gives the language the modal flavour; it has various readings like for instance the interpretation of expressing *possibility*. The dual $\Box$ of this operator, which is defined as $\neg\Diamond\neg$, can be thought of as denoting *necessity*.

The interpretation of modal formulas is given in the following truth definition.

DEFINITION 2.3
Given a network topology $\mathcal{N} = \langle W, R\rangle$, a world $w \in W$ and a formula $\varphi \in \mathcal{L}_0$, the *truth definition* $\mathcal{N}, w \models \varphi$ is given by:

$$\begin{array}{lcl} \mathcal{N}, w \models true & & \\ \mathcal{N}, w \models \varphi_1 \wedge \varphi_2 & \Leftrightarrow & \mathcal{N}, w \models \varphi_1 \text{ and } \mathcal{N}, w \models \varphi_2 \\ \mathcal{N}, w \models \neg\varphi & \Leftrightarrow & \mathcal{N}, w \not\models \varphi \\ \mathcal{N}, w \models \Diamond\varphi & \Leftrightarrow & \mathcal{N}, v \models \varphi \text{ for some } v \in R(w). \end{array}$$

For instance, $\mathcal{N}, w \models \Box\Diamond true$ expresses that in the network topology $\mathcal{N}$, all acquaintances of the agent w are acquainted to an agent.

FIGURE 1. Two bisimilar network topologies

The language $\mathcal{L}_0$ is however not expressive enough to describe and reason about network topologies. Consider for instance the network topologies in Figure 1, which from a local perspective denote distinct situations. That is, a logic for network topologies should be able to distinguish between the situation that an agent's circle of acquaintances is comprised of two agents and the situation that this circle consists of only one agent. However, the basic language $\mathcal{L}_0$ lacks the expressive power to distinguish between both network topologies; i.e. there does not exist a formula that is true in the left network and not in the right one. Formally, this follows from the fact that these networks are bisimilar.

Extensions of the basic modal language that deal with numbers of successors are the *graded modal languages* [10]. Rather than one modal operator $\Diamond$ the graded language contains a set $\{\Diamond_n \mid n \geq 0\}$ of operators. A formula of the form $\Diamond_n\varphi$ expresses that there exist more than n accessible worlds in which φ holds. Hence, graded modal logic distinguishes between the network topologies in Figure 1. For instance, the formula $\Diamond_1 true$ is true for the agent in the left network but not for the agent in the right one.

Graded modal languages are still not suitable to describe network topologies. For instance, consider the two networks in Figure 2, which denote a loop and its infinite unfolding, respectively. The left network topology consists of an agent that knows only of itself, while in the second network there is an infinite chain of agents that know of each other. Since it can be shown that graded modal logic does not possess the expressive power to distinguish between these bisimilar networks, this logic is also not adequate to reason about network topologies.

3 Bounded quantification over worlds

Our analysis of the reason why basic modal logic and its extension with graded modalities are not fit to reason about network topologies, is that they lack a mechanism for dealing with

world identities.

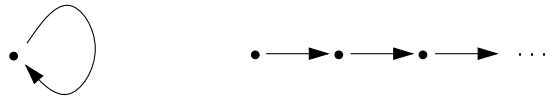

FIGURE 2. Loop and its infinite unfolding

For instance, if we were able to compare the identities of the accessible worlds in Figure 1, and to compare the identity of the current world with that of its successor world in Figure 2, then we would be able to distinguish between these network topologies. This observation naturally leads to an extension of the basic modal logic with a collection of variables to denote the worlds of a model together with an operator to bind them.

DEFINITION 3.1
Given a set Var of variables, terms t and formulas φ of the *extended modal language* $\mathcal{L}_1$ are generated using the following BNF-grammar:

$$\begin{array}{lcl} t & ::= & \mathit{self} \mid x \\ \varphi & ::= & (t_1 = t_2) \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \Diamond\varphi \mid \exists x(\varphi), \end{array}$$

where x ranges over the variables of Var.

We assume the usual abbreviation $\forall x\varphi$ for $\neg\exists x\neg\varphi$. The formula $true$ can be represented by the formula $\mathit{self} = \mathit{self}$. A formula φ is called a *sentence* if it contains no free variables, i.e. all variables x in φ occur in the scope of a quantifier $\exists x$.

Terms of the language $\mathcal{L}_1$ are variables denoting worlds and a special constant self denoting the *current* world. An atomic formula is of the form $t_1 = t_2$, expressing that terms t_1 and t_2 denote the same world. We omit propositional variables here since their treatment is standard and orthogonal to the other logical operators. Additionally, the operator $\exists x$ binds the variable x to some world in the domain of quantification of the current world.

We define the following general models for the extended modal language.

DEFINITION 3.2
A *(generalized) Kripke model* for $\mathcal{L}_1$ is a tuple of the form:

$$\mathcal{M} = \langle W, R, D \rangle,$$

where W is the set of worlds, $R \subseteq W \times W$ denotes the *accessibility* relation on W and $D \subseteq W \times W$ defines the domains of quantification. We write $D(w)$ to denote the domain of quantification $\{u \in W \mid D(w,u)\}$ of the world w.

In addition to an accessibility relation R, a model contains a relation D that defines for each world the set of worlds over which can be quantified. There are various possible instantiations of this domain relation D, like for instance $\{(w,w) \mid w \in W\}$, which only allows variables to be bound to the current world, and the universal relation $\{(w_1, w_2) \mid w_1, w_2 \in W\}$ which enables the binding of variables to any world in the model.

In particular, in the case of *network topologies*, we would like to be able to quantify over precisely the agents that are known. In other words, we assume that network topologies are Kripke models in which the domain relation coincides with the accessibility relation, that is, $R = D$.

Still, there are other possible instantiations. As a final example we mention a domain relation that is given by the transitive closure of the accessibility relation. In this case we are able to quantify over all worlds that are accessible by following one or more links.

Before we define the interpretation of $\mathcal{L}_1$ in terms of Kripke models, we introduce some helpful notation.

DEFINITION 3.3
Given a partial function $f : X \to Y$, we use the notation $f(x) = \perp$ to denote that f is not defined for x. Additionally, the *domain* of f is defined by $dom(f) = \{x \mid f(x) \neq \perp\}$. Its *range* is given by $ran(f) = \{y \in Y \mid \text{ exists } x \in X \text{ with } f(x) = y\}$. Finally, we write $f[y/x]$ to denote the function that behaves like f except on the input x for which it yields the output y.

The interpretation of terms and formulas in $\mathcal{L}_1$ is given via the following truth definition, where we use the notion of an assignment for the interpretation of variables. Such an assignment is a function $s : Var \to W$ of finite domain, which maps variables to worlds in the set W.

DEFINITION 3.4
Given a model $\mathcal{M} = \langle W, R, D\rangle$, the *interpretation* $[\![t]\!]_{w,s}$ of a term t in a world $w \in W$ under an assignment function $s : Var \to W$, is defined by:

$$\begin{array}{lcl} [\![self]\!]_{w,s} & = & w \\ [\![x]\!]_{w,s} & = & s(x). \end{array}$$

The *truth definition* $\mathcal{M}, w, s \models \varphi$ is given by:

$$\begin{array}{lcl} \mathcal{M}, w, s \models (t_1 = t_2) & \Leftrightarrow & [\![t_1]\!]_{w,s} = [\![t_2]\!]_{w,s} \\ \mathcal{M}, w, s \models \varphi_1 \wedge \varphi_2 & \Leftrightarrow & \mathcal{M}, w, s \models \varphi_1 \text{ and } \mathcal{M}, w, s \models \varphi_2 \\ \mathcal{M}, w, s \models \neg\varphi & \Leftrightarrow & \mathcal{M}, w, s \not\models \varphi \\ \mathcal{M}, w, s \models \Diamond\varphi & \Leftrightarrow & \mathcal{M}, v, s \models \varphi \text{ for some } v \in R(w) \\ \mathcal{M}, w, s \models \exists x\varphi & \Leftrightarrow & \mathcal{M}, w, s[v/x] \models \varphi \text{ for some } v \in D(w). \end{array}$$

Additionally, we have $\mathcal{M}, w \models \varphi$ if for all assignments s it holds that $\mathcal{M}, w, s \models \varphi$. Finally, we write $\mathcal{M} \models \varphi$ if $\mathcal{M}, w \models \varphi$ holds for all $w \in W$.

Note the difference in the truth definition between the operators $\Diamond$ and $\exists$ with respect to the point of evaluation: in the truth definition of the former operator there is a shift in perspective, namely, from w to v, whereas in the latter, the point of view w remains fixed. In other words, $\exists$ quantifies over the current domain while the operator $\Diamond$ is used to change the scope of quantification.

Finally, note that the constant $self$ constitutes a *non-rigid designator* [7] in the sense that its denotation differs among the worlds in a model; in particular, in each world the denotation of this designator is the world itself.

The logic $\mathcal{L}_1$ distinguishes between the network topologies in Figures 1 and 2. For instance, the formula $\exists x \exists y \neg(x = y)$ is true in the left network in Figure 1 but not in the right one. Secondly, the formula $\exists x(x = self)$ distinguishes between the two networks in Figure 2. An example of a distinguishing formula that does not contain the constant $self$, is the formula $\exists x \Diamond \exists y(y = x)$, which is true in the left network in Figure 2 but not in the right one.

The general set up of the framework allows us to study the connections between the accessibility relation and the domains of quantification that the language $\mathcal{L}_1$ can express. We consider the property that the domain relation is a contained in the accessibility relation.

OBSERVATION 3.5
For all models $\mathcal{M} = \langle W, R, D\rangle$ and worlds $w \in W$ the following holds:

$$\mathcal{M}, w \models \forall x \Diamond (x = \mathit{self}) \text{ iff } D(w) \subseteq R(w).$$

In Corollary 4.6 below, we prove that there does not exist a formula that expresses $R(w) \subseteq D(w)$, for all w. However, a straightforward refinement of the language is its extension with the inverse operator of $\Diamond$ denoted by $\Diamond^{-1}$, which has a natural interpretation in the context of network topologies: it denotes the *is-known-by* relation.

DEFINITION 3.6
The interpretation of the inverse navigation operator $\Diamond^{-1}$, is defined by:

$$\mathcal{M}, w, s \models \Diamond^{-1}\varphi \Leftrightarrow \mathcal{M}, v, s \models \varphi \text{ for some } v \text{ with } w \in D(v).$$

With this extra operator, we obtain the following result.

OBSERVATION 3.7
Given a model $\mathcal{M} = \langle W, R, D\rangle$ with a reflexive domain relation D, for all worlds $w \in W$, we have $R(w) \subseteq D(w)$ if and only if:

$$\mathcal{M}, w \models \exists x(x = \mathit{self} \wedge \Box(\exists y(y = \mathit{self} \wedge \Diamond^{-1}(x = \mathit{self} \wedge \exists z(z = y))))).$$

As mentioned before, network topologies are Kripke models in which the domain relation coincides with the accessibility relation. Let us consider some properties of network topologies we can express using the language $\mathcal{L}_1$.

EXAMPLE 3.8 (Network topologies)
First of all, the formula

$$\exists x \Diamond (x = \mathit{self})$$

can be thought of expressing 'knowing yourself'. Secondly, the formula

$$\exists x(x = \mathit{self} \wedge \Box \Diamond x = \mathit{self})$$

is true in a world in case all accessible worlds have in turn access to this world. In other words, it expresses 'everyone that I know, knows me'. Additionally, the formula

$$\exists xy(\neg(x = y) \wedge \Diamond(x = \mathit{self} \wedge \neg \Diamond y = \mathit{self}) \wedge \Diamond(y = \mathit{self} \wedge \neg \Diamond x = \mathit{self}))$$

is true in a particular world, in case there are two distinct accessible worlds that are not accessible to one another. Informally, it can be thought of as expressing 'I know two agents that do not know each other'.

Finally, we illustrate that quantification does not commute with modality. Consider the formula

$$\exists x \Box (x = \mathit{self}),$$

which is true in a world in case there is exactly one accessible world, and which can be thought of expressing 'I know of exactly one agent'. On the other hand, the formula

$$\Box \exists x(x = \mathit{self})$$

expresses something different, namely that 'everyone that I know, knows itself'.

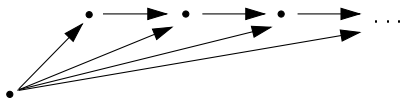

FIGURE 3. Construction of an infinite domain

In the remainder of this section, we consider the finite model property and the decidability of the logic.

LEMMA 3.9
The language $\mathcal{L}_1$ does not satisfy the finite model property.

PROOF. We show that $\mathcal{L}_1$ can enforce infinite domains. Let $P(x, y)$ stand for the following formula:

$$\Diamond(x = \mathit{self} \wedge \Diamond y = \mathit{self}),$$

which expresses that the world x is accessible from the current world, and from x the world y is accessible. Let φ denote the conjunction of the following formulas $\exists x(\mathit{true})$, which reflects a nonempty domain, $\forall x(\neg P(x, x))$ expressing the irreflexivity of the relation P, the formula $\forall x \forall y \forall z((P(x, y) \wedge P(y, z)) \rightarrow P(x, z))$ denoting transitivity, and $\forall x \exists y(P(x, y))$ expressing seriality. It is not difficult to see that if this formula is true in a particular world then its domain of quantification must be infinite (see Figure 3). ∎

In the above proof, the use of the constant self is not essential. The language $\mathcal{L}_1$ without this constant does not satisfy the finite model property either. This can be shown in a similar manner using the following definition of the formula $P(x, y)$:

$$\Diamond(\exists u(u = x) \wedge \Diamond \exists u(u = y)).$$

Thus, $P(x, y)$ expresses that y is in the domain of a world that can be accessed from some accessible world (with respect to the current world) that has x in its domain.

The crux in the proof of Lemma 3.9 is the construction of an infinite *domain of quantification*. It is still an open issue for future research whether $\mathcal{L}_1$ satisfies the finite model property when we restrict to Kripke models in which the domains of quantification are finite.

In [2], it is shown that the hybrid language consisting of the basic modal language extended with variables and an operator $\downarrow x$ to bind the variable x to the current world, has an undecidable validity problem. Since a formula $\downarrow x(\varphi)$ can be modelled by $\exists x(x = \mathit{self} \wedge \varphi)$ in our language, we derive that the language $\mathcal{L}_1$ is undecidable. This also implies that the language $\mathcal{L}_1$ is not part of the *guarded fragment* of first-order logic [1], since this fragment has a decidable validity problem.

Moreover, an interesting question arises with respect to the role of the constant self in this result. The language $\mathcal{L}_1$ without this constant self is also not part of the guarded fragment of first-order logic. Yet it is an open issue for future research whether the validity problem of this sublanguage of $\mathcal{L}_1$ is decidable.

4 Semantic characterization

In this section, we study the expressiveness of the language $\mathcal{L}_1$. In particular, we address the issue of what properties the language can express and what properties are beyond its expressive power. The central result is a *semantic characterization* of the language, which

defines the conditions under which two Kripke models satisfy precisely the same formulas of $\mathcal{L}_1$.

For the basic modal language $\mathcal{L}_0$ the semantic characterization is given by the notion of a *bisimulation* [3, 9]. That is, two models satisfy the same basic modal formulas if and only if they are bisimilar. In this paper, we introduce a new notion of bisimulation, called *history-based bisimulation*, which extends classical bisimulation with a mechanism to handle quantifications. Instead of relating worlds, this new type of bisimulation relates tuples that are comprised of a world together with a injective sequence of worlds. These additional sequences are employed to represent variable bindings that are generated during the evaluation of formulas.

Let us first introduce some helpful notation with respect to injective sequences.

DEFINITION 4.1
Given a set of worlds W an *injective sequence* over W, or *sequence* for short, is a function $\bar{v} : I\!N \to W$ of finite domain, which satisfies for all $i, j \in dom(\bar{v})$:

$$\bar{v}(i) = \bar{v}(j) \;\Rightarrow\; i = j.$$

Additionally, ϵ denotes the empty sequence; that is, $\epsilon(i) = \bot$ for all $i \in I\!N$.

Finally, we define:

$$\bar{v} \bullet w = \begin{cases} \bar{v}[w/i] & \text{if } w \notin ran(\bar{v}) \\ \bar{v} & \text{otherwise,} \end{cases}$$

where $i \in I\!N$ is the next index not part of $dom(\bar{v})$. Thus, $\bar{v} \bullet w$ denotes the extension of the sequence $\bar{v}$ with w in case w does not already occur in $\bar{v}$, and denotes $\bar{v}$ itself, otherwise.

An injective sequences $\bar{v} : I\!N \to W$ is an *abstraction* of an assignment $s : Var \to W$ that just contains the information that is needed in the semantic characterization. That is, an assignment s is represented by a sequence that consists of the elements in the range of s in some particular order. This representation thus abstracts from any repetitions of worlds and the particular variable names. From a *computational* point of view, the advantage of sequences in comparison with assignments is that they give rise to a *decidable* semantic characterization (see Observation 4.3).

Next, we introduce the notion of a history-based bisimulation.

DEFINITION 4.2
Given the models $\mathcal{M}_1 = \langle W_1, R_1, D_1 \rangle$ and $\mathcal{M}_2 = \langle W_2, R_2, D_2 \rangle$, a non-empty relation $\sim$ is a *history-based bisimulation*, if $(w_1, \bar{v_1}) \sim (w_2, \bar{v_2})$ implies the following:

(self) $w_1 = \bar{v_1}(i)$ iff $w_2 = \bar{v_2}(i)$, for all $i \in I\!N$,
(bisim) if $u_1 \in R_1(w_1)$ then there exists $u_2 \in R_2(w_2)$ with $(u_1, \bar{v_1}) \sim (u_2, \bar{v_2})$,
(var) if $u_1 \in D_1(w_1)$ then there exists $u_2 \in D_2(w_2)$ with $(w_1, \bar{v_1} \bullet u_1) \sim (w_2, \bar{v_2} \bullet u_2)$

and conditions similar to **(bisim)** and **(var)** from $\mathcal{M}_2$ to $\mathcal{M}_1$.

Additionally, we define $w_1 \sim w_2$ to hold in case $(w_1, \epsilon) \sim (w_2, \epsilon)$.

The reason why the notion of bisimulation is called *history-based* is that the injective sequences record the worlds in the domains of the encountered worlds that have been bound by a particular variable. We could say that they make up a history of landmarks: if during navigation we arrive at a world that has such a landmark in its domain, then in the other model, we should be at a world that has the corresponding landmark in its domain.

It is worth remarking here that the notion of a history-based bisimulation is quite different from the notion of a *history-preserving bisimulation* [8]. The latter is a very strong notion saying that two worlds are history-preserving bisimilar if they are related by a bisimulation and additionally, the respective submodels consisting of the worlds that can reach the world via the accessibility relation, are isomorphic.

If we restrict ourselves to models with a finite number of worlds, the notion of a history-based bisimulation is decidable.

OBSERVATION 4.3
Given models $\mathcal{M}_1$ and $\mathcal{M}_2$ with a *finite* number of worlds, for all worlds w_1 in $\mathcal{M}_1$ and w_2 in $\mathcal{M}_2$, it is *decidable* whether there exists a history-based bisimulation $\sim$ such that $w_1 \sim w_2$.

Note that it is crucial here that injective sequences do not contain repetitions of worlds. This implies that there exists a bound on the number of applications of rule **(var)** that we need to consider.

Before we phrase the semantic characterization of the language $\mathcal{L}_1$ in Theorem 4.5, we define the notion of an image-finite world.

DEFINITION 4.4
Given a model $\mathcal{M} = \langle W, R, D\rangle$, a world $w \in W$ is called *image-finite* if $R(v)$ and $D(v)$ are finite for all v with $(w, v) \in R^*$, where R^* denotes the reflexive, transitive closure of R.

Properly, we do not need the assumption of image-finiteness, as analogous to the proof of the semantic characterization of standard modal logic, we could use ultrafilter extensions [3]. However, for the sake of simplicity we adopt this property here.

THEOREM 4.5 (Semantic characterization)
Given two models $\mathcal{M}_1$ and $\mathcal{M}_2$, for all worlds w_1 from $\mathcal{M}_1$ and w_2 from $\mathcal{M}_2$ the following hold:

(i) if $w_1 \sim w_2$ for some history-based bisimulation $\sim$ then for all *sentences* $\varphi \in \mathcal{L}_1$ we have $\mathcal{M}_1, w_1 \models \varphi \Leftrightarrow \mathcal{M}_2, w_2 \models \varphi$

(ii) if w_1 and w_2 are image-finite and $\mathcal{M}_1, w_1 \models \varphi \Leftrightarrow \mathcal{M}_2, w_2 \models \varphi$ for all *sentences* $\varphi \in \mathcal{L}_1$, then $w_1 \sim w_2$ for some history-based bisimulation $\sim$.

The proof of the semantic characterization is given in the appendix. Let us consider some applications of the result.

FIGURE 4. A confluent and non-confluent model

First of all, consider the confluent model and the non-confluent model in Figure 4. In case their domain relations are equal to the accessibility relations, these two models are related by a history-based bisimulation, implying that the language $\mathcal{L}_1$ cannot distinguish between them. However, note that in case their domain relations are equal to the *transitive closure* of the accessibility relations, the formula $\exists x \Box\Box(x = self)$, which is true in the left network but not in the right one, is an example of a distinguishing formula.

Secondly, the language $\mathcal{L}_1$ cannot express the property that the accessibility relation is contained in the domain relation, as stated in the following result.

COROLLARY 4.6
There does not exist a formula $\varphi \in \mathcal{L}_1$ such that for all models $\mathcal{M} = \langle W, R, D\rangle$ and worlds $w \in W$ we have: $\mathcal{M}, w \models \varphi$ iff $R(w) \subseteq D(w)$.

PROOF. Consider the model $\mathcal{M} = \langle W, R, D\rangle$ and a state $w \in W$ such that

- W is an *infinite* set of worlds,
- R satisfies $(w, v) \in R$ for all $v \in W$,
- D satisfies $(w, v) \in D$ for all $v \in W$.

Additionally, we have a model $\mathcal{M}^\star$ that extends $\mathcal{M}$ with a world $\star$ defined by:

$$\mathcal{M}^\star = \langle W \cup \{\star\}, R \cup \{(w, \star)\}, D\rangle.$$

These two models are related by the following history-based bisimulation $\sim$. For all $u \in W$, sequences $\bar{v}$ over W and $u' \in W \setminus ran(\bar{v})$:

$$\begin{array}{lcl} (u, \bar{v}) & \sim & (u, \bar{v}), \\ (u', \bar{v}) & \sim & (\star, \bar{v}). \end{array}$$

Note that such a world u' exists since $ran(\bar{v})$ is finite while the set W is infinite. Consequently, by Theorem 4.5 we obtain that the language $\mathcal{L}_1$ cannot distinguish between these two models, and as $R(w) \subseteq D(w)$ and $R(w) \cup \{(w, \star)\} \not\subseteq D(w)$, we derive the desired result. ∎

5 Related work and future research

Our framework is closely connected to the work on *hybrid languages*, which also combine modality with first-order variable-binding mechanisms [4]. In particular, hybrid languages extend the basic modal language $\mathcal{L}_0$ with a collection of *nominals* that are used to label worlds in a model. These nominals are propositional formulas that are true at exactly one world in a model, and so to speak are employed as global *unique* names for worlds. Further extensions additionally incorporate operators of the form $@_t$ to *jump* to the world that is denoted by the term t, as well as operators to *bind* variables; e.g. the operator $\downarrow x$ to bind the variable x to the current world and the existential quantifier, which we denote as $\bar{\exists} x$ to distinguish it from the quantifier $\exists x$ from $\mathcal{L}_1$, which quantifies over all worlds of a model.

First of all, the operator $\downarrow x$ to bind the variable x to the current world can be represented in the language $\mathcal{L}_1$ as follows:

$$\downarrow x(\varphi) \;\Leftrightarrow\; \exists x(x = \mathit{self} \wedge \varphi).$$

The operator corresponds to existential quantification in the class of models in which for each world the domain of quantification consists of only the world itself; that is, in the class:

$$\{\mathcal{M} \mid \mathcal{M} \models \exists x(x = \mathit{self} \wedge \forall y(y = x))\}.$$

Additionally, the hybrid quantifier $\bar{\exists} x$ ranges over the *entire* set of worlds in a model. In our framework this operator corresponds with existential quantification in the class of models in which the domain of quantification of each world coincides with the entire set of worlds.

Finally, we mention the hybrid operator $@_t$ that is used to jump to the world denoted by the term t. The truth definition of this operator can be given as follows:

$$\mathcal{M}, w, s \models @_t\varphi \;\Leftrightarrow\; \mathcal{M}, v, s \models \varphi, \text{ where } v = [\![t]\!]_{w,s}.$$

This operator has no counterpart in our framework due to the fact that in each world, it allows moving to worlds that are not necessarily reachable via the accessibility relation. This is in contrast with one of our underlying assumptions that in a world one cannot move to arbitrary worlds but only to those worlds which are accessible.

Let us consider the major differences between our framework and the hybrid languages as described above. First of all, an important characteristic of the hybrid languages is the treatment of terms as formulas. That is, analogous to formulas, a term can be true or false at a world of a model: it is true if its denotation is exactly the current world and is false otherwise. In contrast, our logic is based on a more conventional ontology which distinguishes between terms and formulas; i.e. a term denotes a world and a formula a Boolean value. Consequently, our framework formalizes reasoning about the identities of the worlds of a model directly in terms of *equational logic*.

A second difference with the hybrid languages, is our separation of navigation and variable-binding mechanisms; that is, in our framework, there is one operation for navigating a model and another operation for bounded quantification over worlds. This treatment allows us to study these different mechanisms in isolation as well as to examine their interactions. In contrast, hybrid languages cover operators, such as the operators $\Downarrow x$ and Σx in [4], that embody both navigation and binding aspects.

Thirdly, in the hybrid languages, the interpretation of nominals is *absolute*, which means that each of these constants denotes a unique world in the model. In contrast, our framework allows natural extensions with *relative* constants, which are constants whose interpretation depends on the current world. Such an extension can be used when reasoning about the ambiguities of *names* in, for example, multi-agent topologies; that is, situations in which one agent is known by other agents under different names. Formally, we may extend our language $\mathcal{L}_1$ with a countable set C of names, with typical element c. A term t in the extended language, which is called $\mathcal{L}_2$, is thus either a variable x, the constant $self$, or a name $c \in C$. Formulas are defined as in Definition 3.1 and they are interpreted over the following models:

$$\langle W, R, D, I\rangle,$$

where W is a set of worlds, $R \subseteq W \times W$ denotes the accessibility relation, and I is a total function which assigns to each $w \in W$ an interpretation $I(w)$ of each name $c \in C$, that is, $I(w) \in C \to W$. Furthermore, $D = \{\langle w, I(w)(c)\rangle \mid c \in C\}$. In other words, for each world the domain of quantification is given by the local denotations of the constants.

The definition of the truth of a formula φ in the extended language $\mathcal{L}_2$ involves a straightforward adaptation of the truth definition of the language $\mathcal{L}_1$ and is therefore omitted. Instead, we explain here the use of quantification in the description of the ambiguities to which names may give rise. First, we observe that without quantification we cannot describe phenomena like that one agent is known by different agents under different names. For example, given an agent w, we cannot describe the situation that $I(w)(c) = I(w')(c)$, for some $(w, w') \in R$, simply because the modal operators induce a 'context switch', that is, a different interpretation of the names. However this situation can be described using quantifiers simply by the formula:

$$\exists x(x = c \wedge \Diamond(x = c)).$$

So, we bind the value of the constant c to the variable x, and use the fact that the interpretation of the variables is fixed, that is, does not change when 'moving' from one agent to another. This example illustrates the difference with the variable-binding mechanism of *first-order logic*: in first-order logic, the formula $\exists x(x = t \wedge \varphi)$ can be modelled by the substitution $\varphi[t/x]$ of t for x in φ.

Another example of a further extension of our framework concerns a formalization of reasoning about the identities of *objects* as they appear during the execution of an object-oriented program. A constant $c \in C$ is interpreted in this application as a *pointer attribute*. Object structures can be modelled as Kripke models that contain a family of deterministic accessibility relations; one for each pointer attribute.

DEFINITION 5.1
A *deterministic (generalized) Kripke model* is a pair (W, I), where as above, W is a set of worlds and $I \in W \to (C \to W)$.

The set W represents the set of existing objects and I describes the pointer structure. These models support the following natural multi-modal extension $\mathcal{L}_3$ of our basic logic, where each pointer attribute is also used as a *modal operator*:

$$\varphi \quad ::= \quad (t_1 = t_2) \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \langle c\rangle\varphi \mid \exists x(\varphi).$$

Given a model $\mathcal{M} = (W, I)$, an object $w \in W$ and an assignment function $s : Var \to W$, we define the interpretation of constants by:

$$[\![c]\!]_{w,s} = I(w)(c).$$

Additionally, the truth definition is given by:

$$\mathcal{M}, w, s \models \langle c\rangle\varphi \;\Leftrightarrow\; \mathcal{M}, v, s \models \varphi, \text{ where } v = I(w)(c).$$

The component I in (W, I) thus represents both the accessibility and the domain relation.

Consider for instance the model $\mathcal{M}$ in Figure 5, which consists of four objects that are arranged in a ring structure. Each object has two pointers $left$ and $right$ to denote the object on its left and on its right, respectively.

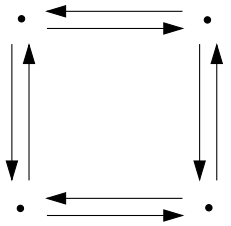

FIGURE 5. A ring structure

In this model, each object is the left neighbour of its right neighbour:

$$\mathcal{M} \models \exists x(x = self \wedge \langle right\rangle(left = x)).$$

Additionally, we have:

$$\mathcal{M} \models \exists x(x = right \wedge \langle left\rangle\langle left\rangle(left = x)),$$

which expresses that for each object, its right neighbour is the same object as the left neighbour of the left neighbour of its left neighbour.

We would like to end our discussion with the conclusion that our approach to the introduction of variable-binding mechanisms into modal logics provides a promising basis for various interesting extensions and applications.

Acknowledgements

The authors wish to thank the anonymous referees for their suggestions and comments.

References

[1] H. Andréka, J. van Benthem, and I. Németi. Modal logics and bounded fragments of predicate logic. *Journal of Philosophical Logic*, **27**, 217–274, 1999.

[2] C. Areces, P. Blackburn, and M. Marx. A road-map on the complexity of hybrid logics. In *Computer Science Logic, Proceedings of CSL'99*, J. Flum and M. Rodríguez-Artalejo, eds. volume 1683 of *Lecture Notes in Computer Science*, pp. 307–321. Springer-Verlag, Heidelberg, 1999.

[3] J.F.A.K van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, Naples, 1983.

[4] P. Blackburn and J. Seligman. Hybrid languages. *Journal of Logic, Language and Information*, **4**, 251–272, 1995.

[5] R. Bull. An approach to tense logic. *Theoria*, **36**:282–306, 1970.

[6] R.M. van Eijk, F.S. de Boer, W. van der Hoek, and J.-J.Ch. Meyer. A modal logic for network topologies. In *Proceedings of the 7th European Workshop on Logics in Artificial Inteligence (JELIA 2000)*, M. Ojeda-Aciego, I.P. de Guzman, G. Brewka, and L.M. Pereira, eds. Volume 1919 of *Lecture Notes in Artificial Intelligence*, pp. 269–283. Springer-Verlag, Heidelberg, 2000.

[7] M. Fitting and R.L. Mendelsohn. *First-Order Modal Logic*. Kluwer Academic Publishers, Dordrecht, 1998.

[8] U. Goltz, R. Kuiper, and W. Penczek. Propositional temporal logics and equivalences. In *Proceedings of Concur'92*, volume 630 of Lecture Notes in Computer Science, pp. 222–236, Springer-Verlag, Berlin, 1992.

[9] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of Association of Computer Machinery*, **32**, 137–162, 1985.

[10] W. van der Hoek. On the semantics of graded modalities. *Journal of Applied Non Classical Logics*, **2**, 81–123, 1992.

[11] G.E. Hughes and M.J. Cresswell. *An Introduction to Modal Logic*. Methuen and Co. Ltd, London, 1968.

Appendix

A Proof of Theorem 4.5

First, we introduce the notion of an assignment-based bisimulation, which is almost similar to the notion of a history-based bisimulation. However, instead of injective sequences this type of bisimulation makes use of assignments.

DEFINITION A.1
Given the models $\mathcal{M}_1 = \langle W_1, R_1, D_1\rangle$ and $\mathcal{M}_2 = \langle W_2, R_2, D_2\rangle$, the worlds $w_1 \in W_1$ and $w_2 \in W_2$, the assignments $s_1 : Var \to W_1$ and $s_2 : Var \to W_2$, the non-empty relation $\sim$ is an *assignment-based bisimulation* if $(w_1, s_1) \sim (w_2, s_2)$ implies the following:

(term) $[\![t_1]\!]_{w_1,s_1} = [\![t_2]\!]_{w_1,s_1}$ iff $[\![t_1]\!]_{w_2,s_2} = [\![t_2]\!]_{w_2,s_2}$, for all terms t_1 and t_2;

(bisim′) if $u_1 \in R_1(w_1)$ then there exists $u_2 \in R_2(w_2)$ with $(u_1, s_1) \sim (u_2, s_2)$;

(var′) if $u_1 \in D_1(w_1)$ then there is $u_2 \in D_2(w_2)$ with $(w_1, s_1[u_1/x]) \sim (w_2, s_2[u_2/x])$, for all $x \notin ran(s_1) \cap ran(s_2)$;

and conditions similar to **(bisim′)** and **(var′)** from $\mathcal{M}_2$ to $\mathcal{M}_1$.

For assignment-based bisimulations we have the following characterization result.

LEMMA A.2
Given the models $\mathcal{M}_1 = \langle W_1, R_1, D_1 \rangle$ and $\mathcal{M}_2 = \langle W_2, R_2, D_2 \rangle$, the worlds $w_1 \in W_1$ and $w_2 \in W_2$, the assignments $s_1 : Var \to W_1$ and $s_2 : Var \to W_2$, we have:

(i) if $(w_1, s_1) \sim (w_2, s_2)$ for some assignment-based bisimulation $\sim$ then for all $\varphi \in \mathcal{L}_1$ we have: $\mathcal{M}_1, w_1, s_1 \models \varphi \Leftrightarrow \mathcal{M}_2, w_2, s_2 \models \varphi$;

(ii) if w_1 and w_2 are image finite and $\mathcal{M}_1, w_1, s_1 \models \varphi \Leftrightarrow \mathcal{M}_2, w_2, s_2 \models \varphi$ for all $\varphi \in \mathcal{L}_1$, then $(w_1, s_1) \sim (w_2, s_2)$ for some assignment-based bisimulation $\sim$.

PROOF.

(i) This can be shown by induction on the complexity of φ. We consider the main case: φ is of the form $\exists x \psi$. Suppose $(w_1, s_1) \sim (w_2, s_2)$ and $\mathcal{M}, w_1, s_1 \models \exists x \psi$. Then there exists $u_1 \in D_1(w_1)$ such that $\mathcal{M}, w_1, s_1[u_1/x] \models \psi$ holds. From condition **(var′)** we derive that there exists $u_2 \in D_2(w_2)$ with $(w_1, s_1[u_1/x]) \sim (w_2, s_2[u_2/x])$. Applying the induction hypothesis, we obtain $\mathcal{M}_2, w_2, s_2[u_2/x] \models \psi$ for some $u_2 \in D_2(w_2)$, which yields the desired result $\mathcal{M}_2, w_2, s_2 \models \exists x \psi$. The converse implication can be shown similarly.

(ii) Consider the relation $\sim$ defined by: $(w_1, s_1) \sim (w_2, s_2)$ iff w_1 and w_2 are image finite and $\mathcal{M}_1, w_1, s_1 \models \varphi \Leftrightarrow \mathcal{M}_2, w_2, s_2 \models \varphi$ for all formulas $\varphi \in \mathcal{L}_1$. We claim that $\sim$ is an assignment-based bisimulation. Suppose this is not the case. Then there exist image-finite worlds w_1 and w_2 and assignments s_1 and s_2 with $\mathcal{M}_1, w_1, s_1 \models \varphi \Leftrightarrow \mathcal{M}_2, w_2, s_2 \models \varphi$ for all formulas $\varphi \in \mathcal{L}_1$, but at least one of the conditions of assignment-based bisimulations is not satisfied.

- First of all suppose **(term)** is not satisfied. Then without loss of generality, for some terms t_1 and t_2 we have $[\![t_1]\!]_{w_1,s_1} = [\![t_2]\!]_{w_1,s_1}$ but $[\![t_1]\!]_{w_2,s_2} \neq [\![t_2]\!]_{w_2,s_2}$. In other words, $\mathcal{M}_1, w_1, s_1 \models (t_1 = t_2)$ and $\mathcal{M}_2, w_2, s_2 \not\models (t_1 = t_2)$, which yields a contradiction.
- Next, we assume that condition **(bisim′)** is not satisfied. Consider the set $R_2(w_2) = \{u_1, \ldots, u_k\}$ of worlds that are R_2-accessible from w_2, which is finite because of the image-finiteness of w_2. Suppose that we have a world $v \in R_1(w_1)$ such that for all $1 \leq i \leq k$ there exists a formula $\varphi_i \in \mathcal{L}_1$ with $\mathcal{M}_1, v, s_1 \models \varphi_i$ and $\mathcal{M}_2, u_i, s_2 \not\models \varphi_i$. Let φ be the conjunction $\bigwedge_{1 \leq i \leq k} \varphi_i$. Then we have $\mathcal{M}_1, w_1, s_1 \models \Diamond \varphi$ while $\mathcal{M}_2, w_2, s_2 \not\models \Diamond \varphi$, yielding a contradiction. We conclude that such a world v cannot exist.
- Finally, we suppose that condition **(var′)** is not met. Consider the set $D_2(w_2) = \{u_1, \ldots, u_k\}$, which is finite by the image finiteness condition. Suppose we have a world $v \in D_1(w_1)$ such that for all $1 \leq i \leq k$ there exists a formula $\varphi_i \in \mathcal{L}_1$ with $\mathcal{M}_1, w_1, s_1[v/x] \models \varphi_i$ and $\mathcal{M}_2, w_2, s_2[u_i/x] \not\models \varphi_i$. Subsequently, for the conjunction $\varphi = \bigwedge_{1 \leq i \leq k} \varphi_i$ we obtain $\mathcal{M}_1, w_1, s_1 \models \exists x \varphi$ and $\mathcal{M}_2, w_2, s_2 \not\models \exists x \varphi$, yielding a contradiction.

Hence, we conclude that $\sim$ is an assignment-based bisimulation. ∎

What remains to be done is the translation of these results to the case of history-based bisimulations. First, we consider the connection between sequences and assignments. Recall that we assume sequences to be *injective*, which means that they do not contain any repetitions. We define the relation $\approx$, which relates pairs of sequences with the pairs of assignments they represent.

DEFINITION A.3
For all sequences $\bar{w_1}$ and $\bar{w_2}$ and assignments s_1 and s_2, we define $(\bar{w_1}, \bar{w_2}) \approx (s_1, s_2)$ if and only if the following hold:

(i) $ran(s_1) = ran(\bar{w_1})$ and $ran(s_2) = ran(\bar{w_2})$;

(ii) $s_1(x) = \bar{w_1}(i)$ iff $s_2(x) = \bar{w_2}(i)$, for all $x \in Var$ and $i \in \mathbb{N}$.

We identify the following properties of the relation $\approx$.

PROPOSITION A.4
For all sequences w_1 and w_2, assignments s_1 and s_2, if $(\bar{w_1}, \bar{w_2}) \approx (s_1, s_2)$ then:

(i) $s_1(x) = s_1(y) \Leftrightarrow s_2(x) = s_2(y)$, for all $x, y \in Var$

(ii) $(\bar{w_1} \bullet u_1, \bar{w_2} \bullet u_2) \approx (s_1[u_1/x], s_2[u_2/x])$, for all $x \notin dom(s_1) \cap dom(s_2)$.

The following result establishes how to construct history-based bisimulations from assignment-based bisimulations and vice versa.

LEMMA A.5

(i) If $\sim$ is an assignment-based bisimulation then the following relation $\sim_h$ is a history-based bisimulation:

$$(w_1, \bar{v_1}) \sim_h (w_2, \bar{v_2}) \text{ iff } (w_1, s_1) \sim (w_2, s_2) \text{ for some } s_1, s_2 \text{ with } (\bar{v_1}, \bar{v_2}) \approx (s_1, s_2).$$

(ii) If $\sim$ is a history-based bisimulation then the following relation $\sim_a$ is an assignment-based bisimulation:

$$(w_1, s_1) \sim_a (w_2, s_2) \text{ iff } (w_1, \bar{v_1}) \sim (w_2, \bar{v_2}) \text{ for some } \bar{v_1}, \bar{v_2} \text{ with } (\bar{v_1}, \bar{v_2}) \approx (s_1, s_2).$$

PROOF. (i) Suppose $(w_1, \bar{v_1}) \sim_h (w_2, \bar{v_2})$, where $\sim$ is an assignment-based bisimulation. By definition there exist assignments s_1, s_2 with $(\bar{v_1}, \bar{v_2}) \approx (s_1, s_2)$ and $(w_1, s_1) \sim (w_2, s_2)$. We have to prove that all conditions for history-based bisimulations are satisfied.

(self) If we take the constant $self$ for t_1 and a variable x for t_2 in **(term)** we derive $w_1 = s_1(x) \Leftrightarrow w_2 = s_2(x)$, for all x. From the fact $(\bar{v_1}, \bar{v_2}) \approx (s_1, s_2)$ we derive $w_1 = \bar{v_1}(i) \Leftrightarrow w_2 = \bar{v_2}(i)$ for all $i \in I\!N$, which was to be shown.

(bisim) This condition follows immediately from **(bisim′)**.

(var) Consider a state $u_1 \in D_1(w_1)$. By condition **(var′)** there exists $u_2 \in D_2(w_2)$ with $(w_1, s_1[u_1/x]) \sim (w_2, s_2[u_2/x])$, for all $x \notin ran(s_1) \cap ran(s_2)$. Additionally, from $(\bar{v_1}, \bar{v_2}) \approx (s_1, s_2)$ we derive via Proposition A.4(ii) that $(\bar{v_1} \bullet u_1, \bar{v_2} \bullet u_2) \approx (s_1[u_1/x], s_2[u_2/x])$ holds. Consequently, we have $(w_1, \bar{v_1} \bullet u_1) \sim_h (w_2, \bar{v_2} \bullet u_2)$, which was to be shown.

(ii) Suppose $(w_1, s_1) \sim_a (w_2, s_2)$, where $\sim$ is a history-based bisimulation. By definition there exist $\bar{v_1}$ and $\bar{v_2}$ with $(\bar{v_1}, \bar{v_2}) \approx (s_1, s_2)$ and $(w_1, \bar{v_1}) \sim (w_2, \bar{v_2})$. We have to prove that all conditions for assignment-based bisimulations are satisfied.

(term) First, condition **(self)** gives $w_1 = \bar{v_1}(i) \Leftrightarrow w_2 = \bar{v_2}(i)$ for all i. From the fact $(\bar{v_1}, \bar{v_2}) \approx (s_1, s_2)$ we derive $w_1 = s_1(x) \Leftrightarrow w_2 = s_2(x)$, for all $x \in Var$. Secondly, Proposition A.4(i) yields $s_1(x) = s_1(y) \Leftrightarrow s_2(x) = s_2(y)$, for all $x, y \in Var$. Together these two facts yield $[\![t_1]\!]_{w_1,s_1} = [\![t_2]\!]_{w_1,s_1}$ iff $[\![t_1]\!]_{w_2,s_2} = [\![t_2]\!]_{w_2,s_2}$, for all terms t_1 and t_2.

(bisim′) This condition follows immediately from **(bisim)**.

(var′) Consider a state $u_1 \in D_1(w_1)$. By condition **(var)** there exists $u_2 \in D_2(w_2)$ with $(w_1, \bar{v_1} \bullet u_1) \sim (w_2, \bar{v_2} \bullet u_2)$. Additionally, from $(\bar{v_1}, \bar{v_2}) \approx (s_1, s_2)$ we derive via Proposition A.4(ii) that $(\bar{v_1} \bullet u_1, \bar{v_2} \bullet u_2) \approx (s_1[u_1/x], s_2[u_2/x])$ holds for all $x \notin ran(s_1) \cap ran(s_2)$. Consequently, we have $(w_1, s_1[u_1/x]) \sim (w_2, s_2[u_2/x])$ for all $x \notin ran(s_1) \cap ran(s_2)$, which was to be shown.

∎

Finally, we are in position to put all the pieces together.

Proof of Theorem 4.5

(i) Suppose $w_1 \sim w_2$ for some history-based bisimulation $\sim$, which means $(w_1, \epsilon) \sim (w_2, \epsilon)$. Then according to Lemma A.5 we have $(w_1, s) \sim_a (w_2, s)$ for the assignment-based bisimulation $\sim_a$, where s is defined by $s(x) = \bot$ for all $x \in Var$. Then Lemma A.2 yields that for all formulas $\varphi \in \mathcal{L}_1$ we have $\mathcal{M}_1, w_1, s \models \varphi \Leftrightarrow \mathcal{M}_1, w_2, s \models \varphi$. Consequently, for all sentences $\varphi \in \mathcal{L}_1$ we derive $\mathcal{M}_1, w_1 \models \varphi \Leftrightarrow \mathcal{M}_1, w_2 \models \varphi$.

(ii) Suppose w_1 and w_2 are image finite and $\mathcal{M}_1, w_1 \models \varphi \Leftrightarrow \mathcal{M}_2, w_2 \models \varphi$, for all sentences $\varphi \in \mathcal{L}_1$. Since we restrict to sentences we also have $\mathcal{M}_1, w_1, s \models \varphi \Leftrightarrow \mathcal{M}_2, w_2, s \models \varphi$ for all $\varphi \in \mathcal{L}_1$, where s is defined by $s(x) = \bot$ for all $x \in Var$. By Lemma A.2 we have $(w_1, s) \sim (w_2, s)$ for some history-based bisimulation $\sim$. Lemma A.5 then establishes $(w_1, \epsilon) \sim_h (w_2, \epsilon)$, for the history-based bisimulation $\sim_h$, and thus we conclude $w_1 \sim_h w_2$.