Refinement Modal Logic
Laura Bozzelli, Hans van Ditmarsch, Tim French, James Hales, Sophie Pinchinat
February 17, 2012

Abstract

In this paper we present refinement modal logic. A refinement is like a bisimulation, except that from the three relational requirements only ‘atoms’ and ‘back’ need to be satisfied. Our logic contains a new operator ∀ in additional to the standard modalities □ for each agent. The operator ∀ acts as a quantifier over the set of all refinements of a given model. We call it the refinement operator. As a variation on a bisimulation quantifier, it can be seen as a refinement quantifier over a variable not occurring in the formula bound by the operator. The logic combines the simplicity of multi-agent modal logic with some powers of monadic second order quantification. We present a sound and complete axiomatization of multi-agent refinement modal logic. We also present an extension of the logic to the modal μ-calculus, and an axiomatization for the single-agent version of this logic. Examples and applications are also discussed: to software verification and design (the set of agents can also be seen as a set of actions), and to dynamic epistemic logic. We further give detailed results on the complexity of satisfiability, and on succinctness.

Contents

1 Introduction

2 Technical preliminaries

3 Refinement

3.1 Refinement and its basic properties

3.2 Game and logical characterization of refinement

3.3 Refinement as bisimulation plus model restriction

3.4 Refinement and action models

Informática, Universidad Politécnica de Madrid, Spain, laura.bozzelli@fi.upm.es
Logic, University of Sevilla, Spain, hvd@us.es & IMSc, Chennai, India
Computer Science and Software Engineering, University of Western Australia, tim@csse.uwa.edu.au
Computer Science and Software Engineering, University of Western Australia, james@csse.uwa.edu.au
IRISA, University of Rennes, Sophie.Pinchinat@irisa.fr
Modal logic is frequently used for modelling knowledge in multi-agent systems. The semantics of modal logic uses the notion of “possible worlds”, between which an agent is unable to distinguish. In dynamic systems agents acquire new knowledge (say by an announcement, or the execution of some action) that allows agents to distinguish between worlds that they previously could not separate. From the agent’s point of view, what were “possible worlds” become inconceivable. Thus, a future informative event may be modelled by a reduction in the agent’s accessibility relation. In \cite{45} the future event logic is introduced. It augments the multi-agent logic of knowledge with an operation $\forall \varphi$ that stands for “$\varphi$ holds after all informative events” — the diamond version $\exists \varphi$ stands for “there is an informative event after which $\varphi$.” The proposal was a generalization of a so-called arbitrary public announcement logic with an operator for “$\varphi$ holds after all announcements” \cite{7}. The semantics of informative events encompasses action model execution à la Baltag et al \cite{8}: on finite models, it can be easily shown that a model resulting from action model execution is a refinement of the initial model, and for a given refinement of a model we can construct an action model such that the result of its execution is bisimilar to that
refinement. In [46] an axiomatization of the single-agent version of this logic is presented, and also expressivity and complexity results. These questions were visited in both the context of modal logic, and of the modal µ-calculus.

In the original motivation, the main operator ∃ had a rather temporal sense — therefore the ‘future event’ name. However, we have come to realize that the structural transformation that interprets this operator is of much more general use, on many very different kinds of modal logic, namely anywhere where more than a mere model restriction or pruning is required. We have therefore come to call this the refinement operator, and the logic refinement modal logic.

Thus we may consider refinement modal logic to be a more abstract perspective of future event logic [15] applicable to other modal logics. To any other modal logic! This is significant in that it motivates the application of the new operator in many different settings. In logics for games [34,3] or in control theory [39,43], it may correspond to a player discarding some moves; for program logics [24] it may correspond to operational refinement [32]; and for logics for spatial reasoning it may correspond to sub-space projections [33].

Let us give an example. Consider the following structure. The o state is the designated point. The arrows can be associated with a modality.

```
 o --------> . --------> . --------> .
```

E.g., ◊◊◊☐⊥ is true in the point. From the point of view of the modal language, this structure is essentially the same structure (it is bisimilar) as

```
 . <------> . <------> o <------> . <------> . <------> .
```

This one also satisfies ◊◊◊☐⊥ and any other modal formula for that matter. A more radical structural transformation would be to consider submodels, such as

```
 o --------> . --------> .
```

A distinguishing formula between the two is ◊◊☐⊥, which is true here and false above. Can we consider other ‘submodel-like’ transformations that are neither bisimilar structures nor strict submodels? Yes, we can. Consider

```
 . <------> o <------> . <------> .
```

It is neither a submodel of the initial structure, nor is it bisimilar. It satisfies the formula ◊◊◊☐⊥ ∧ ◊◊◊☐⊥ that certainly is false in any submodel. We call this structure a refinement (or ‘a refinement of the initial structure’), and the original structure a simulation of the latter. Now note that if we consider the three requirements ‘atoms’, ‘forth’, and ‘back’ of a bisimulation, that ‘atoms’ and ‘back’ are satisfied but not ‘forth’, e.g., from the length-three path in the
original structure the last arrow has no image. There seems to be still some ‘submodel-like’ relation with the original structure. Look at its bisimilar duplicate (the one with seven states). The last structure is a submodel of that copy. Such a relation always holds: a refinement of a given structure can always be seen as the model restriction of a bisimilar copy of the given structure. This work deals with the semantic operation of refinement, as in this example, in full generality, and also applied to the multi-agent case.

Previous works [16, 30] employed a notion of refinement. In [30] it was shown that model restrictions were not sufficient to simulate informative events, and they introduced refinement trees for this purpose — a precursor of the dynamic epistemic logics developed later [47].

In order to abstract from a particular implementation, a entire theory of modal specifications has been developed [37, 41], which relies on a refinement preorder, known as modal refinement. Modal specifications are deterministic automata equipped with transitions of two types: may and must. Informally, a must-transition is available in every component that implements the modal specification, while a may-transition need not be. Its definition is close to our definition of refinement (as it is some kind of submodel quantifier), but the two notions are incomparable. Although may and must correspond to different modalities, there is no way to associate may and must with different (and independent) agents, because must is a subtype of may.

We incorporate implicit quantification over informative events directly into the language using, again, a notion of refinement; in our case a refinement is the inverse of simulation [4]. The work is closely related to some recent work on bisimulation quantified modal logics [15, 18]. The refinement operator, seen as refinement quantifier, is weaker than a bisimulation quantifier [45], as it is only based on simulations rather than bisimulations, and as it only allows us to vary the interpretation of a propositional variable that does not occur in the formula bound by it. Bisimulation quantified modal logic has previously been axiomatized by providing a provably correct translation to the modal $\mu$-calculus [14]. This is reputedly a very complicated one. The axiomatization for the refinement operator, in stark contrast, is quite simple and elegant.

Overview of the paper Section 2 gives a wide overview of our technical apparatus: modal logic, cover logic, modal $\mu$-calculus, and bisimulation quantified logic. Section 3 introduces the semantic operation of refinement. This includes a game and (modal) logical characterization. Then, in Section 4, we introduce two logics with a refinement quantifier that is interpreted with the refinement relation: refinement modal logic and refinement $\mu$-calculus. Section 5 contains the axiomatization of that refinement modal logic and the completeness proof. We demonstrate that is is equally expressive as modal logic. We mention results for model classes $KD45$ and $S5$. Section 6 gives the axiomatization of refinement $\mu$-calculus. Again, we have a reduction here, to standard $\mu$-calculus. In Section 7 we show that, although the use of refinement quantification does not change the
expressive power of the logics, they do make each logic exponentially more succinct. We give a non-elementary complexity bound for refinement modal $\mu$-calculus.

## 2 Technical preliminaries

Throughout the paper we assume a finite set of agents $A$ and a countable set of propositional variables $P$ as background parameters when defining the structures and the logics. Agents are named $a, b, a', b', \ldots$, and propositional variables are $p, q, r, p', p'', p_1, p_2, \ldots$. Agent $a$ is assumed female, and $b$ male.

### Structures

A model $M = (S, R, V)$ consists of a domain $S$ of (factual) states (or worlds), an accessibility function $R : A \to \mathcal{P}(S \times S)$, and a valuation $V : P \to \mathcal{P}(S)$. States are $s, t, u, v, s', \ldots, s_1, \ldots$. For $s \in S$, $M_s$ is a pointed model. For $R(a)$ we write $R_a$; accessibility function $R$ can be seen as a set of accessibility relations $R_a$, and $V$ as a set of valuations $V(p)$. Given two states $s, s'$ in the domain, $R_a(s, s')$ means that in state $s$ agent $a$ considers $s'$ a possibility. We will also use a relation $R_a$ simply as a set of pairs $\subseteq S \times S$, and use the abbreviation $sR_a = \{ t \in S \mid (s, t) \in R_a \}$. As we will be often required to discuss several models at once, we will use the convention that $M = (S^M, R^M, V^M)$, $N = (S^N, R^N, V^N)$, etc. The class of all models (given parameter sets of agents $A$ and propositional variables $P$) is denoted $\mathcal{K}$. The class of all models where for all agents the accessibility relation is reflexive, transitive and symmetric is denoted $S5$, and the model class with a serial, transitive and euclidean accessibility relation is denoted $KD45$.

### Multi-agent modal logic

The language $\mathcal{L}$ of multi-agent modal logic is inductively defined as

$$\varphi ::= p \mid \neg \varphi \mid (\varphi \land \varphi) \mid \square_a \varphi$$

where $a \in A$ and $p \in P$. Without the construct $\square_a \varphi$ we get the language $L_a$ of propositional logic. Standard abbreviations are: $\varphi \lor \psi$ iff $-(\neg \varphi \land \neg \psi)$, $\varphi \rightarrow \psi$ iff $\neg \varphi \lor \psi$, $\top$ iff $p \lor \neg p$, $\bot$ iff $p \land \neg p$, and $\diamond_a \varphi$ iff $\neg \square_a \neg \varphi$. If there is a single agent only ($|A| = 1$), we may write $\Box \varphi$ instead of $\square_a \varphi$. Formula variables are $\varphi, \psi, \chi, \varphi', \ldots, \varphi_1, \ldots$ and for sets of formulas we write $\Phi, \Psi, \ldots$. For a finite set $\Phi$ of $\mathcal{L}$ formulas we let the cover operator $\nabla_a \Phi$ be an abbreviation for $\square_a \bigvee_{\varphi \in \Phi} \varphi \land \bigwedge_{\varphi \in \Phi} \diamond_a \varphi$; we note $\bigvee_{\varphi \in \emptyset} \varphi$ is always false, whilst $\bigwedge_{\varphi \in \emptyset} \varphi$ is always true.

Given a finite set of formulae $\Psi = \{ \psi_1, \ldots, \psi_n \}$ and a formula $\varphi$ with possible occurrences of a propositional variable $p$. Let $\varphi[\psi \backslash p]$ denote the substitution of all occurrences of $p$ in $\varphi$ by $\psi$. Then $\varphi[\Psi \backslash p]$ abbreviates $\{ \varphi[\psi_1 \backslash p], \ldots, \varphi[\psi_n \backslash p] \}$, and similarly $\bigvee \varphi[\Psi \backslash p]$ stands for $\varphi[\psi_1 \backslash p] \lor \ldots \lor \varphi[\psi_n \backslash p]$ and $\bigwedge \varphi[\Psi \backslash p]$ stands for $\varphi[\psi_1 \backslash p] \land \ldots \land \varphi[\psi_n \backslash p]$. For example, the definition of $\nabla_a \Phi$ is written as $\square_a \bigvee \Phi \land \bigwedge \diamond_a \Phi$.

We now define the semantics of modal logic. Assume an epistemic model $M = (S, R, V)$.
The interpretation of $\varphi \in L$ is defined by induction.

$M_s \models p$ iff $s \in V_p$

$M_s \models \neg \varphi$ iff $M_s \not\models \varphi$

$M_s \models \varphi \land \psi$ iff $M_s \models \varphi$ and $M_s \models \psi$

$M_s \models \Box_a \varphi$ iff for all $t \in S : (s,t) \in R_a$ implies $M_t \models \varphi$

A formula $\varphi$ is valid on a model $M$, notation $M \models \varphi$, iff for all $s \in S$, $M_s \models \varphi$; and $\varphi$ is valid iff $\varphi$ is valid on all $M$ (in the model class $K$, given agents $A$ and basic propositions $P$). The set of validities, i.e., the logic in the stricter sense of the word, is called $K$.

**Cover logic** The cover operator $\nabla$ has also been used as a syntactic primitive in modal logics [14]. It has recently been axiomatized [9]. The language $L_{\nabla}$ of cover logic is defined as

$$\varphi ::= p \mid \neg \varphi \mid (\varphi \land \varphi) \mid \nabla_a \{\varphi, \ldots, \varphi\},$$

where $p \in P$, and $a \in A$. The semantics of $\nabla_a \Phi$ is the obvious one if we recall our introduction by abbreviation of the cover operator:

$M_s \models \nabla_a \Phi$ iff for all $\varphi \in \Phi$ there is a $t \in sR_a$ such that $M_t \models \varphi$, and for all $t \in sR_a$ there is a $\varphi \in \Phi$ such that $M_t \models \varphi$.

The set of validities of cover logic is called $K_{\nabla}$. The conjunction of two cover formulae is again equivalent to a cover formula:

$$\nabla_a \Phi \land \nabla_a \Psi \iff \nabla_a (\Phi \land \Psi \cup (\vee \Phi \land \Psi)).$$

The modal box and diamond are definable as $\Box_a \varphi$ iff $\nabla_a \emptyset \lor \nabla_a \{\varphi\}$, and $\Diamond_a \varphi$ iff $\nabla_a \{\varphi, \top\}$, respectively. Cover logic $K_{\nabla}$ is equally expressive as modal logic $K$ (also in the multi-agent version) [9] [28]. We use cover operators in the presentation of the axioms.

**Modal $\mu$-calculus** For the modal $\mu$-calculus, apart from the set of propositional variables $P$ we have another parameter set $X$ of variables to be used in the fixed-point construction. The language $L^\mu$ of modal $\mu$-calculus is defined as follows.

$$\varphi ::= x \mid p \mid \neg \varphi \mid (\varphi \land \varphi) \mid \Box_a \varphi \mid \mu x. \varphi$$

where $a \in A$, $x \in X$, $p \in P$, and where in $\mu x. \varphi$ the variable $x$ only occurs positively (i.e. in the scope of an even number of negations) in the formula $\varphi$. We will refer to a variable $x$ in an expression $\mu x. \varphi$ as a fixed-point variable. The formula $\nu p. \varphi$ is an abbreviation for $\neg \mu x. \neg \varphi[-x/x]$.

For the semantics of the $\mu$-calculus, the valuation $V$ of propositional variables is extended to include fixed-point variables. We write $V^{[x \rightarrow T]}$ for the operation that changes a given valuation $V$ into one wherein $V(x) = T$ (where $T \subseteq S$) and the valuation of all other fixed-point and propositional variables remains the same. Given a model $M = (S, R, V)$,
we similarly write $M^{[p \rightarrow T]}$ for the model $M = (S, R, V^{[p \rightarrow T]})$. The semantics of $\mu x.\varphi$ (the top-down presentation, not the bottom-up presentation) is now as follows: Let $\varphi \in L^\mu$ and model $M$ be given.

$$M_s \models \mu x.\varphi \text{ iff } s \in \bigcap\{T \subseteq S \mid \{u \mid M_u^{[p \rightarrow T]} \models \varphi\} \subseteq T\}$$

**Disjunctive formula** An important technical definition we require later on is that of a disjunctive formula. A disjunctive $L^\mu$ formula is specified by the following abstract syntax:

$$\varphi ::= x \mid (\varphi \lor \varphi) \mid (\varphi_0 \land \bigwedge_{a \in B} \nabla_a\{\varphi, \ldots, \varphi\}) \mid \mu x.\varphi \mid \nu x.\varphi \quad (1)$$

where $p \in P$, $x \in X$, $\varphi_0 \in L_0$ (propositional logic), and $B \subseteq A$. To get the disjunctive $L$ formula (of modal logic) we omit the clauses containing $\mu$-calculus variables $x$:

$$\varphi ::= (\varphi \lor \varphi) \mid (\varphi_0 \land \bigwedge_{a \in B} \nabla_a\{\varphi, \ldots, \varphi\})$$

If the context of the logic is clear, we simply write disjunctive formula (or df). If $B = \emptyset$, we have that $\bigwedge_{a \in B} \nabla_a\{\varphi_1, \ldots, \varphi_n\} = \top$, as expected.

Every $L^\mu$ formula is equivalent to a disjunctive $L^\mu$ formula \[25\]. \hfill (2a)

Every $L$ formula is equivalent to a disjunctive $L$ formula \[48\]. \hfill (2b)

**Bisimulation quantified modal logic** The language $L_{\forall}$ is defined as

$$\varphi ::= p \mid \neg \varphi \mid (\varphi \land \varphi) \mid \Box_a \varphi \mid \forall p \varphi$$

where $a \in A$ and $p \in P$. We let $\exists p \varphi$ abbreviate $\neg \forall p \neg \varphi$. We write $\forall$ and $\exists$ for the bisimulation quantifiers in order to distinguish them from the refinement quantifiers $\forall$ and $\exists$, to be introduced later. Given an atom $p$ and a formula $\varphi$, the expression $\forall p \varphi$ means that there exists a denotation of propositional variable $p$ such that $\varphi$. It is interpreted as follows (restricted bisimulation $\simeq^p$ is introduced further below in Definition\[1\]).

$$M_s \models \exists p \varphi \text{ iff there is a } N_t \text{ such that } N_t \simeq^p M_s \text{ and } N_t \models \varphi$$

In \[18\] Lemma 2.43 a bisimulation quantifier characterization of fixed points is given. The characterization employs the universal modality $\blacksquare$ which quantifies over all states in the model. (Let $L_{\forall\Box}$ be the language of bisimulation quantified modal logic with $\blacksquare$ as well.) The only crucial clauses in the inductively defined translation $t : L^\mu \rightarrow L_{\forall\Box}$ are those for the fixed-point operators. The atoms $p$ in the translation are required not to occur in $\varphi$.

$$t(\nu x.\varphi) \text{ is equivalent to } \exists p(p \land \blacksquare(p \rightarrow t(\varphi[p/x]))) \quad (3a)$$

$$t(\mu x.\varphi) \text{ is equivalent to } \forall p(\blacksquare(t(\varphi[p/x]) \rightarrow p) \rightarrow p) \quad (3b)$$
The first equation captures the intuition of a greatest fixed point as a least upper bound of the set of states that are prefixed points of $\phi$, whereas the second equation captures a least fixed point as the greatest lower bound of the set of states that are postfixed points of $\phi$. From [13] we know that bisimulation quantifiers are also expressible in the modal $\mu$-calculus, and thus these equivalences also hold in the modal $\mu$-calculus. For more information on the modal $\mu$-calculus, see [14, 48].

3 Refinement

In this section we define the notion of structural refinement, investigate its properties, give a game characterization in (basic) modal logic, and compare refinement to bisimulation and other established semantic notions in the literature.

3.1 Refinement and its basic properties

**Definition 1 (Bisimulation, simulation, refinement)** Let two models $M = (S, R, V)$ and $M' = (S', R', V')$ be given. A non-empty relation $R \subseteq S \times S'$ is a bisimulation if for all $(s, s') \in R$ and $a \in A$:

- atoms $s \in V(p)$ iff $s' \in V'(p)$ for all $p \in P$;
- forth-$a$ for all $t \in S$, if $R_a(s, t)$, then there is a $t' \in S'$ such that $R'_a(s', t')$ and $(t, t') \in R$;
- back-$a$ for all $t' \in S'$, if $R'_a(s', t')$, then there is a $t \in S$ such that $R_a(s, t)$ and $(t, t') \in R$.

We write $M \simeq M'$ ($M$ and $M'$ are bisimilar) iff there is a bisimulation between $M$ and $M'$, and we write $M_s \simeq M'_s$ ($M_s$ and $M'_s$ are bisimilar) iff there is a bisimulation between $M$ and $M'$ linking $s$ and $s'$. A restricted bisimulation $R^p : M_s \simeq^p M'_s$ is a bisimulation that satisfies atoms for all variables except $p$. A total bisimulation is a bisimulation such that all states in the domain and codomain occur in a pair of the relation.

A relation $\exists_B$ that satisfies atoms, back-$a$, and forth-$a$ for every $a \in A \setminus B$, and that satisfies atoms, and back-$b$ for every $b \in B$, is a $B$-refinement, and in that case $M'_s$ is (also called) a $B$-refinement of $M_s$, and we write $M_s \succeq_B M'_s$. An $A$-refinement we call a refinement (plain and simple) and for $\{a\}$-refinement we write $a$-refinement. Dually, we similarly define the $B$-simulation $\mathfrak{S}_B$. We also similarly define restricted refinement and restricted simulation.

The definition of simulation and refinement above varies slightly from the one given by Blackburn et al. [10, p.110]. Here we ensure that simulations and refinements preserve the interpretations (i.e., the truth and falsity) of atoms, whereas [10] has them only preserve the truth of propositional variables in a simulation—and presumably preserve their falsity in a refinement. We prefer to preserve the entire interpretation, as we feel it suits our applications better. For example, in the case where refinement represents information
change, we would not wish basic facts to become false in the process. The changes are supposed to be merely of information, and not factual.

We allow ourselves to overload the term refinement in the following way (as in the definition): if $\mathcal{F}_B : M_s \succeq_B M'_s$, then we call $\mathcal{F}_B$ a refinement but we also call $M'_s$ a refinement of $M_s$. The context will disambiguate. This is similar to the double use of the term simulation.

In an epistemic setting a refinement corresponds to the diminishing uncertainty of agents. This means that there is a potential decrease in the number of states and transitions in a model. On the other hand, the number of states as a consequence of refinement may also increase, because the uncertainty of agents over the extent of decreased uncertainty in other agents may still increase. This is perhaps contrary to the concept of program refinement, where detail is added to a specification. However, in program refinement the added detail requires a more detailed state space (i.e., extra atoms) and as such is more the domain of bisimulation quantifiers, rather than refinement quantification. Still, the consequence of program refinement is a more deterministic system which agrees with the notion of diminishing uncertainty.

**Proposition 2** The relation $\succeq_a$ is reflexive and transitive (a pre-order), and satisfies the Church-Rosser property.  

**Proof** Reflexivity follows from the observation that the identity relation satisfies atoms, and back-$a$ and forth-$a$ for all agents $a$, and therefore also the weaker requirement for refinement. Similarly, given two $a$-simulations $\mathcal{R}_1$, and $\mathcal{R}_2$, we can see that their composition, $\{(x, z) \mid \text{there is a } y \text{ for which } (x, y) \in \mathcal{R}_1, \,(y, z) \in \mathcal{R}_2\}$ is also an $a$-refinement. This is sufficient to construct transitivity. The Church-Rosser property states that if $N_t \succeq_a M_s$ and $N_t \succeq_a M'_{s'}$, then there is some model $N'_{t'}$ such that $M_s \succeq_a N'_{t'}$ and $M'_{s'} \succeq_a N'_{t'}$. From Definition [32] it follows that $M_s$ and $M'_{s'}$ must be bisimilar to one another with respect to $A - \{a\}$. We may therefore construct such a model $N'_{t'}$ by taking $M_s$ (or $M'_{s'}$) and setting $R^N_a = \emptyset$ and $R^N_{b} = R^M_{b}$ for all $b \in A - \{a\}$. It can be seen that $N'_{t'}$ where $N' = (S^M, R^{N'}_{a}, V^M)$ satisfies the required properties. \hfill $\Box$

An elementary result is the following.

**Proposition 3** Let $B = \{a_1, ..., a_n\}$, and $M_s$ and $M_t$ given. Then $M_s(\succeq_{a_1} \circ \cdots \circ \succeq_{a_n})M_t$ iff $M_s \succeq_B M_t$.  

**Example 4** If $N_t \succeq_a M_s$ and $M_s \succeq_a N_t$, it is not necessarily the case that $M_s \simeq_a N_t$. For example, consider the one-agent models $M$ and $N$ where:

- $S^M = \{1, 2, 3\}$, $R^M_a = \{(1, 2), (2, 3)\}$ and $V^M(p) = \emptyset$ for all $p \in P$; and
- $S^N = \{4, 5, 6, 7\}$, $R^N_a = \{(4, 5), (5, 6), (4, 7)\}$ and $V^N(p) = \emptyset$ for all $p \in P$.

These two models are clearly not bisimilar, although $N_4 \succeq_a M_1$ via $\{(4, 1), (5, 2), (6, 3)\}$ and $M_1 \succeq_a N_4$ via $\{(1, 4), (2, 5), (3, 6), (2, 7)\}$. See Figure [7] \hfill $\Box$
Given that the equivalence $M_s \equiv N_t$ defined by $M_s \preceq N_t$ and $M_s \succeq N_t$ is not a bisimulation, an interesting question seems to be what it then represents. It seems to formalize that two structures are only different in resolvable differences in uncertainty (for the agent of the refinement), but not in hard and necessary facts. So the positive formulas (for that agent) should be preserved under this ‘equivalence’ $\equiv$. Such matters will now be addressed.

3.2 Game and logical characterization of refinement

It is folklore to associate a (infinite) two-player game safety game with refinement, in the spirit of [2].

Definition 5 (Refinement game) Let $M_s$ and $N_t$ be two models. We define a turn-based game $G_a(M_s, N_t)$ between two players Spoiler and Duplicator (male and female, respectively) by $G_a(M_s, N_t) = (V, E, (s, t))$ where the set of positions $V$ is partitioned into the positions $V_{\text{Spoiler}} = S^N \times S^M$ of Spoiler and the positions $V_{\text{Duplicator}} = S^N \times \{\text{forth}, \text{back}\} \times (A \cup P) \times S^M$ of Duplicator. Since the initial position $(s, t) \in V_{\text{Spoiler}}$, Spoiler starts. The set of moves $E \subseteq V_{\text{Spoiler}} \times V_{\text{Duplicator}} \cup V_{\text{Duplicator}} \times V_{\text{Spoiler}}$ is the least set such that the following pairs belong to $E$ (we take the convention that $b \neq a$, and for convenience, we name those moves with names similar to the properties of refinement in Definition [7]):
A play in $G_a(M_s, N_t)$ is winning for **Duplicator** iff it is infinite.

Notice that there is no forth-\(a\) move in the game $G_a(M_s, N_t)$, which captures the refinement relation between the structures:

**Lemma 6** $M_s \succeq_a N_t$ iff **Duplicator** has a winning strategy in $G_a(M_s, N_t)$.  

**Proof** Assume **Duplicator** has a winning strategy in $G_a(M_s, N_t)$. Because we are only interested in a safety game, hence it is a regular position, we can assume without loss of generality that this winning strategy is memoryless [21]. Namely, the strategy $\sigma$ of **Duplicator** is a function from $V_{Duplicator} \to V_{Spoiler}$ that tells her how to play. On the basis of $\sigma$, one can define the binary relation $F_\sigma \subseteq S^M \times S^N$ as the set of pairs $(s', t')$ such that, in the game $G_a(M_s, N_t)$, position $(s', t') \in V_{Spoiler}$ is reachable when **Duplicator** follows her strategy $\sigma$. Then it is easy to check that $F_\sigma$ is an $a$-simulation from $M_s$ to $N_t$. Also it is not difficult to see that if some $a$-refinement $\mathcal{F}_a$ exists from $M_s$ to $N_t$, then any strategy of **Duplicator** which consists in maintaining **Spoiler’s** positions in $\mathcal{F}_a$, is winning. Note that this is always possible for her. $
$
We now consider a characterization of the refinement in terms of the logic $L_\forall$. Namely, given an agent $a$, we define the fragment of the $a$-positive formulas $L^{a+} \subseteq L$ by

$$L^{a+} \equiv \varphi ::= p \mid \neg p \mid (\varphi \land \varphi) \mid (\varphi \lor \varphi) \mid \Box_b \varphi \mid \Diamond_b \varphi \mid \Diamond_a \varphi$$

where $b \in A \setminus \{a\}$ and $p \in P$.

**Proposition 7** For any finitely branching (every state has only finitely many successors) models $M$ and $N$, for any $s_0 \in S^M$, for any $t_0 \in S^N$, and for any agent $a \in A$,

$$M_{s_0} \succeq_a N_{t_0} \text{ if, and only if, for every } \varphi \in L^{a+}, N_{t_0} \models \varphi \text{ implies } M_{s_0} \models \varphi.$$  

\[\square\]
Proof Let us first establish that for every $t \in S^N$ and $s \in S^M$, if Spoiler has a winning strategy in $G_a(M_s, N_t)$, then there exists a formula $\varphi(s, t) \in \mathcal{L}^a$ called a distinguishing formula for $(M_s, N_t)$, which satisfies $N_t \models \varphi(s, t)$ but $M_s \not\models \varphi(s, t)$. Note that if Spoiler has a winning strategy in $G_a(M_s, N_t)$, all plays induced by this strategy have finite length and end in a position where Duplicator cannot move.

We reason by induction on $k$, the maximal length of these plays; note that because Spoiler starts, $k > 0$.

If $k = 1$, Spoiler has a winning move from $(s, t)$ to some $v \in V_{\text{Duplicator}}$, where Duplicator is blocked. We reason on the form of $v$:

- if $v = (s, (\text{forth}, p), t)$ (resp. $v = (s, (\text{back}, p), t)$), then there is no move back to $(s, t)$ because $t \not\in V^N(p)$ (resp. $s \not\in V^M(p)$). A distinguishing formula is $\neg p$ (resp. $p$).
- if $v = (s', (\text{forth}, b), t)$ (resp. $v = (s, (\text{back}, b), t')$), then $tR^N_b = \emptyset$ (resp. $sR^M_b = \emptyset$).

A distinguishing formula is $\Box_b \bot$ (resp. $\Diamond_b \top$). Since forth-a moves are not allowed in the game, position $v = (s', (\text{forth}, b), t)$ is not reachable in the game $G_a(M_s, N_t)$, so that the formula $\Box_a \bot \not\in \mathcal{L}^a$ is not needed.

Assume now that $k > 1$, and pick a winning strategy of Spoiler in $G_a(M_s, N_t)$. From initial position $(s, t)$, we explore the move given by this strategy (because $k > 1$, this move cannot be either forth-p?, or back-p?). Three cases remain.

forth-b? The reached position becomes $(s', (\text{forth}, b), t)$, and from there Duplicator loses. That is, for each $t' \in tR^N_b$, Spoiler wins the game $G_a(M_s', N_{t'})$ in at most $k - 2$ steps. By the induction hypothesis, there exists a distinguishing formula $\varphi(s', t') \in \mathcal{L}^a$ for $(M_s', N_{t'})$. It is easy to see that $\varphi(s, t) = \Box_b(\bigvee_{v \in tR^N_b} \varphi(s', t'))$ is a distinguishing formula for $(M_s, N_t)$; notice that since $N$ is finitely branching, the conjunction is finitary.

back-b? This case applies $b \neq a$ and to $b = a$.

The reached position becomes $(s, (\text{back}, b), t')$, and from there Duplicator loses. Using a similar reasoning as for forth-b moves, it is easy to establish that there exists a formula $\varphi(s', t') \in \mathcal{L}^a$, such that $\varphi(s, t) = \Diamond_b(\bigwedge_{v \in tR^M_b} \varphi(s', t'))$ is a distinguishing formula for $(M_s, N_t)$; here, as $M$ is finitely branching, a finitary disjunction is guaranteed.

Now, according to the game characterization of refinement (Lemma 6), and the fact the existence of a winning strategy for Spoiler from position $(s_0, t_0)$ is equivalent to $M_{s_0} \not\models_a N_{t_0}$, we obtain the right to left direction of the proposition. For the other direction, assume $M_s \models_a N_t$, and let $\varphi \in \mathcal{L}^a$ with $N_t \models \varphi$. We prove that $M_s \not\models \varphi$, by induction over the structure of the formula. Basic cases where $\varphi$ is either $p$ or $\neg p$, but also the cases $\varphi \land \psi$ and $\varphi \lor \psi$, are immediate.

Assume $N_t \models \Box_b \varphi$. Then for every $t' \in tR^N_b$, $N_{t'} \models \varphi$. If $tR^N_b = \emptyset$, then by Property forth-b of Definition 1 this entails $sR^M_b = \emptyset$ and consequently $M_s \not\models \Box_b \varphi$ (whatever $\varphi$ is).
Otherwise, $tR^N_b \neq \emptyset$. Take an arbitrary $s' \in sR^M_b$. By Property \textbf{forth-b} of Definition \[1\] there is a $t'_s \in tR^M_b$ with $M_{s'} \succeq_b N_{t'_s}$ and $N_{t'_s} \models \varphi$. By induction hypothesis, $M_s \models \varphi$, which entails $M_s \models \Box_b \varphi$.

Assume $N_t \models \Diamond_b \varphi$, and let $t' \in tR^N_b$ be such that $N_{t'} \models \varphi$. By Property \textbf{back-b} of Definition \[1\] there is some $s' \in sR^M_b$, such that $M_{s'} \succeq_b N_{t'}$. By induction hypothesis, $M_{s'} \models \varphi$ which entails $M_s \models \Diamond_b \varphi$.

Note that the argument still holds if we take $b = a$. \hfill \Box

### 3.3 Refinement as bisimulation plus model restriction

A bisimulation is also a refinement, but refinement allows much more semantic variation. How much more? There is a precise relation. Semantically, a refinement is a bisimulation followed by a model restriction.

An $a$-refinement needs to satisfy \textbf{back} for that agent, but not \textbf{forth}. Let an (‘initial’) model and a refinement of that model be given. For the sake of the exposition we assume that the initial model and its refinement are minimal, i.e., they are bisimulation contractions. Now take an arrow (a pair in the accessibility relation) in that initial model. This arrow may be missing in the refinement, namely when \textbf{forth} is not satisfied for that arrow. On the other hand, any arrow in the refinement should be traceable to an arrow in the initial model – the \textbf{back} condition. There may be several arrows in the refinement that are traceable to the same arrow in the initial model, because the states in which such arrows finish may be non-bisimilar. In other words, we can see the refinement as a blowup of the initial model and then cutting off bits and pieces.

**Example 8** A simple example is as follows. Consider the structure

\[ \bullet_1 \rightarrow \bullet_2 \rightarrow \bullet_3 \rightarrow \bullet_4 \]

and its refinement

\[ \bullet_{t'} \leftarrow \bullet_2 \rightarrow \bullet_b \rightarrow \bullet_c \]

by way of refinement relation $\mathcal{R} = \{(1,a), (2,b), (3,c), (2,b')\}$. The arrow $(3,4)$ has no image in the refinement. On the other hand, the arrow $(1,2)$ has two images, namely $(a,b)$ and $(a, b')$. These two arrows cannot be identified, because $b$ and $b'$ are non-bisimilar: there is yet another arrow from $b$ but no other arrow from $b'$. This reflects that arrow $(2,3)$ has ‘only’ one image in the refined model, not for both images of 2.

There is yet another perspective. This makes the relation to restricted bisimulations clear. When expanding the initial model, the blowing up phase, make a certain propositional variable false in all states of the blowup that you want to prune (that are not in the refinement relation) and make it true in all states that you want to keep. Therefore, the blown up model is bisimilar to the initial model except for that variable. (In other words, it is a restricted bisimulation.) Then, remove arrows to state where that atom is false.
Example 9 Continuing the previous example, consider the following structure bisimilar to the initial model, except for the value of atom $p$—let us say that $\bullet$ represents that $p$ is true and $\circ$ represents that $p$ is false.

\[
\begin{array}{c}
\circ_d & \leftarrow & \circ_c & \leftarrow & \bullet_d & \leftarrow & \bullet_b & \leftarrow & \bullet_c & \leftarrow & \circ_d
\end{array}
\]

The relation $R = \{(1,a),(2,b),(3,c),(4,d),(2,b'),(3,c'),(4,d')\}$ is a bisimulation, except for the value of $p$. We get the refinement from the previous example by removing the $\circ$ states and arrows leading to it.

Winding up, performing an $a$-refinement clearly corresponds to the following operation:

Given a pointed model, first choose a bisimilar pointed model, then remove some pairs from the accessibility relation for $a$ in that model.

Given a propositional variable $q$, this has the same semantic effect as

Given a pointed model, first choose a bisimilar pointed model except for variable $q$, such that $q$ is (only) false in some states that are accessible for $a$, then remove all those pairs from the accessibility relation for $a$.

In other words:

Given a pointed model, first choose a bisimilar pointed model except for variable $q$, then remove all pairs from the accessibility relation for $a$ pointing to states where $q$ is false.

If we do this for all agents at the same time (or if we strictly regard tree unw windings of models only), we can even see the latter operation as follows:

Given a pointed model, first choose a bisimilar pointed model except for variable $q$, then restrict the model to the states where $q$ is true.

Formally, the result is as follows. In the following, given a model $M$ with accessibility relation (set of accessibility relations) $R$, and $R' \subseteq R$, $M|R'$ is the model that is as $M$ but with the accessibility restricted to $R'$. Similarly, $M|p$ is the restriction of $M$ to the states satisfying $p$ (with the corresponding restriction in accessibility relation and valuation).

Proposition 10

- Given $M_s \succeq a N_t$, there is a $N'_t$ (with accessibility function $R'$) and some $R''$ that is as $R'$ except that $R''_a \subseteq R'_a$, such that $M_s \simeq N'_t$ and $N'_t|R'' \simeq N_t$.

- Given $M_s \succeq a N_t$, there is a $N'_t$ (with accessibility function $R'$) and some $p \in P$ such that $M_s \simeq^p N'_t$ and $N'_t|R'' = N_t$, where $R''$ is as $R'$ except that $(u,u') \in R''_a$ iff $N'_u \models p$.

- Given $M_s \succeq a N_t$, there is a $N'_t$ and some $p \in P$ such that $M_s \simeq^p N'_t$ and $N'_t|p = N_t$.\]
Proof We only demonstrate how to construct the proper model $N'_t$ in the first item, and how to value $p$ for use in the second item.

Let an $a$-refinement relation $\mathfrak{R}_a \subseteq S^M \times S^N$ be given (such that $(s,t) \in \mathfrak{R}_a$). We expand the model $N$ and this relation $\mathfrak{R}_a$ as follows to a model $N'$ and a bisimulation $\mathfrak{R} \subseteq S^M \times S^N$. For all $v$ such that $(u,v) \in R^M_a$ for some $u$ and $(u,u') \in \mathfrak{R}_a$ for some $u'$, and for which there is no $v'$ in $N$ such that $(v,v') \in \mathfrak{R}_a$ (in other words, forth is lacking), add $v$ to the model $N$ and also $(u',v)$ to $R^N_a$, and $(v,v)$ to $\mathfrak{R}_a$. The resulting model is $N'_t$ and the resulting relation $\mathfrak{R}$ is a bisimulation. Removing these added pairs again returns $N_t$, so we even have that $M_s \simeq N'_t$ and $N'_t|R'' = N_t$, beyond the proof requirement.

To further satisfy the requirement for $p$ in the second item, we make $p$ false in all such states $v$ with an $a$-image lacking forth, and true anywhere else. □

In Section 4.3 we build upon this semantic result by translating the logic with refinement quantifiers into the logic with bisimulation quantifiers plus relativization of formulae.

### 3.4 Refinement and action models

We recall another important result connecting structural refinement to action model execution [8]. For full details, see [4,5]. An action model $M = (S,R,\text{pre})$ is like a model $M = (S,R,V)$ but with the valuation replaced by a precondition function $\text{pre} : S \to \mathcal{L}$ (for a given language $\mathcal{L}$). The elements of $S$ are called action points. A restricted modal product $(M \otimes M)$ consists of pairs $(s,s)$ such that $M_s \models \text{pre}(s)$, the product of accessibility relations namely such that $((s,s),(t,t)) \in R_a$ iff $(s,t) \in R_a$ and $(s,t) \in R_a$, and keeping the valuation of the state in the pair: $(s,s) \in V(p)$ iff $s \in V(p)$. A pointed action model $M_s$ is an epistemic action.

**Proposition 11** The result of executing an epistemic action in a pointed model is a refinement of that model. Dually, for every refinement of a finite pointed model there is an epistemic action such that the result of its execution in that pointed model is a model bisimilar to the refinement. [4, Prop.4.5]

It is instructive to outline the proof of these results.

Given pointed model $M_s$ and epistemic action $M_s$, the resulting $(M \otimes M)_{(s,s)}$ is a refinement of $M_s$ by way the relation $\mathfrak{R}$ consisting of all pairs $(t,(t,t))$ such that $M_t \models \text{pre}(t)$. Some states of the original model may get lost in the modal product, namely if there is no action whose precondition can be executed there. But all ‘surviving’ (state,action)-pairs simply can be traced back to their first argument: clearly a refinement.

For the other direction, construct an epistemic action $M_{s'}$ that is isomorphic to a given refinement $N_{s'}$ of a model $M_s$, but wherein valuations (determining the value of propositional variables) in states $t \in N$ are replaced by preconditions for action execution of the corresponding action points (also called) $t$. Precondition $\text{pre}(t)$ should be satisfied in exactly those states $s \in M$ such that $(s,t) \in \mathfrak{R}$, where $\mathfrak{R}$ is the refinement relation linking $M_s$ and $N_{s'}$. Now in a finite model, we can single out states (up to bisimilarity) by
a distinguishing formula \([11]\). One then shows that \((M \otimes M, (s, s'))\) can be bisimulation-contracted to \(N_{s'}\). It is unknown if the finiteness restriction can be lifted, because the existence of distinguishing formulae plays a crucial part in the proof.

After introducing refinement modal logic, in Section 4 Example 4.2 presents an action model and its execution in an initial information state, and we will there continue our reflections on the comparison of the frameworks.

### 3.5 Refinement and pruning

Just as refinement is not mere model restriction, it is also immediate to see that refinement is not mere pruning: consider a model \(M\) consisting of a single state \(s\) with an \(a\)-loop. The model \(M'\) with three states \(s_1, s_2, s_3\) such that \(R_a = \{(s_1, s_2), (s_1, s_3), (s_3, s_3)\}\) satisfies \(M \geq_a M'\) but is not bisimilar to any pruning of \(M\).

For refinement and pruning to coincide, one can for example restrict the semantics to the class of deterministic models, that is models such that every accessibility relation \(R_a\) is a functional. This is precisely the classic setting considered in control theory. We refer to Section 4.2 where an example will be given.

Also, pruning plays an important role in game theory, where strategies are in one-to-one correspondence with prunings of the unraveled arena. However, refinement is enough to consider: for example, concerning turn-based 2-player zero-sum games with \(\omega\)-regular winning conditions \([21]\), we have the following: if \(G\) and \(G'\) are two bisimilar arenas, then a player has a winning strategy in \(G\) iff she has winning strategy in \(G'\). Therefore, for a given arena \(G\), the existence of a refinement of \(G\) such that the winning conditions hold is equivalent to determining the existence of a winning strategy in \(G\) itself. This last remark strengthens the relevance of our refinement operator.

### 3.6 Refinement and modal specifications refinement

Modal specifications are classic, convenient, and expressive mathematical objects that represent interfaces of component-based systems \([29, 36, 37, 38, 31, 41]\). Modal specifications are deterministic automata equipped with transitions of two types: may and must. The components that implement such interfaces are deterministic automata; an alternative language-based semantics can therefore be considered, as presented in \([36, 37]\). Informally, a must-transition is available in every component that implements the modal specification, while a may-transition need not be. Modal specifications are interpreted as logical specifications matching the conjunctive \(\nu\)-calculus fragment of the \(\mu\)-calculus \([17]\). In order to abstract from a particular implementation, a entire theory of modal specifications has been developed, which relies on a refinement preorder, known as modal refinement. However, although its definition is close to our definition of refinement, the two notions are incomparable: there is no way to interpret may and must as different agents (agent \(a\) and another agent \(b \neq a\) have clearly independent roles in the semantics of \(a\)-refinement), because must is a subtype of may.
4 Refinement modal logic

In this section we present the refinement modal logic, wherein we add a modal operator that we call a refinement quantifier to the language of multi-agent modal logic, or to the language of the modal \( \mu \)-calculus. From prior publications [45, 46] refinement modal logic is known as ‘future event logic’. In that interpretation different \( \Box_a \) operators stand for different epistemic operators (each describing what an agent knows), and refinement modal logic is then able express what informative events are consistent with a given information state. However, here we take a more general stance.

We list some relevant validities and semantic properties, and also relate the logic to well-known logical frameworks such as bisimulation quantified modal logic (by way of relativization), and dynamic epistemic logics.

4.1 Syntax and semantics of refinement modal logic

The syntax and the semantics of future event logic are as follows.

**Definition 12 (Languages \( \mathcal{L}_\forall \) and \( \mathcal{L}_\forall^\mu \))** Given a finite set of agents \( A \) and a countable set of propositional atoms \( P \), the language \( \mathcal{L}_\forall \) of refinement modal logic is inductively defined as

\[
\varphi ::= p \mid \neg \varphi \mid (\varphi \land \varphi) \mid \Box_a \varphi \mid \forall_a \varphi
\]

where \( a \in A \) and \( p \in P \). Similarly, the language \( \mathcal{L}_\forall^\mu \) of refinement \( \mu \)-calculus has an extra inductive clause \( \mu_x.\varphi \), where \( X \) is the set of variables and \( x \in X \).

\[
\varphi ::= x \mid p \mid \neg \varphi \mid (\varphi \land \varphi) \mid \Box_a \varphi \mid \forall_a \varphi \mid \mu_x.\varphi
\]

We write \( \exists_a \varphi \) for \( \neg \forall_a \neg \varphi \). For a subset \{\( a_1, \ldots, a_n \)\} = \( B \subseteq A \) of agents we introduce the abbreviation \( \exists_B \varphi \) for \( \exists_{a_1} \ldots \exists_{a_n} \varphi \) (in any order), where we write \( \exists \varphi \) for \( \exists_A \varphi \), and similarly for \( \forall_B \) and \( \forall \). (So in the single-agent version we are also entitled to write \( \forall \) and \( \exists \).)

Note the two differences between bisimulation quantifiers \( \forall p \) and the refinement quantifier \( \forall \). The former we write with a ‘tilde’-symbol over the quantifier. The latter (and also \( \forall_a \)) has no variable. A refinement quantifier can be seen as implicitly quantifying over a variable, namely over a variable that does not occur in the formula \( \varphi \) that it binds (nor should it occur in a formula of which \( \exists \varphi \) is a subformula). Section 4.3 will relate bisimulation quantification to the refinement operator.

**Definition 13 (Semantics of refinement)** Assume an epistemic model \( M = (S, R, V) \).

\[
M_s \models \forall_a \varphi \text{ iff for all } M'_s : M_s \succeq_a M'_s \text{ implies } M'_s \models \varphi
\]

The set of validities of \( \mathcal{L}_\forall \) is the logic \( \text{RML} \) (refinement modal logic) and the set of validities of \( \mathcal{L}_\forall^\mu \) is the logic \( \text{RML}^\mu \) (refinement \( \mu \)-calculus). As is usual in the area, we will continue to use the term ‘logic’ in a general sense, beyond that of a set of validities.
In other words, $\forall_a \varphi$ is true in a pointed model iff $\varphi$ is true in all its $a$-refinements. Typical model operations that produce an $a$-refinement are: blowing up the model (to a bisimilar model) such as adding copies that are indistinguishable from the current model and one another, and removing pairs of the accessibility relation for the agent $a$ (or, alternatively worded: removing states accessible only by agent $a$). In the final part of this section we relate these semantics to the well-known frameworks action model logic and bisimulation quantified logic (and see also \[45\]).

**Proposition 14 (Bisimulation invariance)** Refinement modal logic and refinement $\mu$-calculus are bisimulation invariant. ⊣

**Proof** Bisimulation invariance is the following property: given $M_s \simeq N_t$ and a formula $\varphi$, then $M_s \models \varphi$ iff $N_t \models \varphi$. If the logic has operators beyond the standard modalities $\Box_a$, this property does not automatically follow from bisimilarity.

For refinement modal logic bisimulation invariance is straightforward, noting that $\Box_a$ is bisimulation invariant, and that $\mu_x$ is bisimulation invariant. The new operator $\forall_a$ is bisimulation invariant, because $a$-refinement is transitive and bisimulation is just a specific type of $a$-refinement. Formally, let $M_s \simeq N_t$, and $M_s \models \forall_a \varphi$, we have to prove that $N_t \models \forall_a \varphi$. Let $O_u$ be arbitrary such that $N_t \succeq_a O_u$. From $M_s \simeq N_t$ follows $M_s \succeq_a N_t$. From $M_s \succeq_a N_t$ and $N_t \succeq_a O_u$ follows by Proposition 2 that $M_s \succeq_a O_u$. From $M_s \models \forall_a \varphi$ and $M_s \succeq_a O_u$ follows $O_u \models \varphi$. As $O_u$ was arbitrary, we therefore conclude $N_t \models \forall_a \varphi$.

The reverse direction is symmetric. □

The following result justifies our notation $\exists_B$ for sets of agents.

**Proposition 15** For all agents $a, b$, $M_s \models \exists_a \exists_b \varphi \leftrightarrow \exists_b \exists_a \varphi$. ⊣

**Proof** Let $M_s$ be given and let $M_t$ and $M_u$ be such that $M_s \succeq_a M_t$ and $M_t \succeq_b M_u$. We have that $M_s(\succeq_a \circ \succeq_b) M_u$ iff $M_s(\succeq_{(a,b)}) M_u$ iff $M_s(\succeq_b \circ \succeq_a) M_u$. (See Proposition 3) □

**Proposition 16** The following are validities of RML.

- $\forall_a \varphi \rightarrow \varphi$ (reflexivity)
- $\forall_a \varphi \rightarrow \forall_a \forall_a \varphi$ (transitivity)
- $\exists_a \forall_a \varphi \rightarrow \forall_a \exists_a \varphi$ (Church-Rosser)
- $\exists_a \bigcirc_a \varphi \leftrightarrow \bigcirc_a \exists_a \varphi$

**Proof** The first three items directly follow from Proposition 2. The trivial refinement is an $a$-refinement; composition of two refinements is a refinement; and indeed it satisfies the Church-Rosser property.
For the fourth, from left to right: let $M_s$ be such that $M_s \models \exists_a \Diamond_a \varphi$, and let $M'_s$ and $t' \in s'R_a$ be such that $M_s \geq_a M'_s$, $M'_s \models \Diamond_a \varphi$, and $M'_s \models \varphi$. Because of back, there is a $t \in sR_a$ such that $M_t \geq_a M'_t$. Therefore $M_t \models \exists_a \varphi$ and thus $M_s \models \Diamond_a \exists_a \varphi$.

From right to left: let $M_s$ be such that $M_s \models \Diamond_a \exists_a \varphi$, and let $t \in sR_a$ and $M'_s$ be such that $M_t \models \exists_a \varphi$ and $M'_s \models \varphi$. Consider the model $N$ with point $s$ that is the disjoint union of $M$ and $M'$ except that: all outgoing $a$-arrows from $s$ in $M$ are removed (all pairs $(s,t) \in R_a$), a new $a$-arrow links $s$ to $t'$ in $M'$ (add $(s,t')$ to the new $R_a$). Then $N_s$ is an $a$-refinement of $M_s$ that, obviously, satisfies $\Diamond_a \varphi$; so $M_s$ satisfies $\exists_a \Diamond_a \varphi$. (This construction is typical for refinement modal logic semantics. It will reappear in various more complex forms later, e.g., in the soundness proof of the axiomatization RML.)

The semantics of refinement modal logic is with respect to the class $K$ of all models (for a given set of agents and atoms). If we restrict the semantics to a specific model class only, we get a very different logic. For example $\exists \Box \bot$ is a validity in RML: just remove all access. But in refinement epistemic logic, interpreted on $S5$ models, this is not a validity: seriality of models must be preserved in every refinement. See [46, 23].

4.2 Examples

Change of knowledge  Given are two agents that are uncertain about the value of a fact $p$, and where this is common knowledge, and where $p$ is true. Both accessibility relations are equivalence relations, so the epistemic operators model the agents’ knowledge. An informative event is possible after which $a$ knows that $p$ but $b$ does not know that; this is expressed by (where $\exists$ is $\exists_{\{a,b\}}$)

$$\exists((\Box_a p \land \neg \Box_b \Box_a p))$$

In Figure 2 the initial state of information is on the left, and its refinement validating the postcondition is on the right. In the visualization the actual states are underlined. If states are accessible for both $a$ and $b$ we have labelled the (single) arrow with $ab$.

Figure 2: An example of refinement as change of knowledge

On the left, the formula $\exists((\Box_a p \land \neg \Box_b \Box_a p))$ is true, because $\Box_a p \land \neg \Box_b \Box_a p$ is true on the right. On the right, in the actual state there is no alternative for agent $a$ (only the actual
state itself is considered possible by \( a \), so \( \Box_a p \) is true, whereas agent \( b \) also considers another state possible, wherein agent \( a \) considers it possible that \( p \) is false. Therefore, \( \neg \Box_b \Box_a p \) is also true in the actual state on the right.

The model on the right in the figure is neither an \( a \)-refinement of the model on the left, nor a \( b \)-refinement of it, but an \( \{a, b\} \)-refinement.

Recalling Section 3.4 on action models, a refinement of a pointed model can also be obtained by executing an epistemic action (Proposition [11]). Therefore, we should be able to see the refinement in this example as produced by an epistemic action. This is indeed the case. The epistemic action consists of two action points \( t \) and \( p \), they can be distinguished by agent \( a \) but not by agent \( b \). What really happens is \( p \); it has precondition \( p \). Agent \( b \) cannot distinguish this from \( t \) with precondition \( \top \).

The execution of this action is depicted in Figure 3. The point of the structure is the one with precondition \( p \): in fact, \( a \) is learning that \( p \), but \( b \) is uncertain between that action and the ‘trivial’ action wherein nothing is learnt. The trivial action has precondition \( \top \). It can be executed in both states of the initial model. The actual action can only be executed in the state where \( p \) is true. Therefore, the resulting structure is the refinement with three states.

![Figure 3: The refinement in Example 4.2.](image)

Action models can also be added as primitives to the multi-agent modal logical language and are then interpreted with a dynamic modal operator — similar to automata-PDL. To get a well-defined logical language the set of action model frames needs to be enumerable and therefore such action models must be finite. Thus we get action model logic. We now recall the result in Proposition [11] that on finite models every refinement corresponds to the execution of an action model and vice versa (where the action model constructed from a given refinement may be infinite), but that it is unknown if that finiteness restriction can be lifted. If that result can be generalized, that would be of interest, as that would suggest that refinement modal logic is equally expressive as action model logic with quantification over action models. If these logics were equally expressive, action model logic with quantification would be decidable—a surprising fact, given that public announcement logic with quantification over public announcements (singleton action models) is undecidable [19].
Software verification and design  Consider a class of discrete-event systems, whose elements represent devices that interact with an environment. Each device is described by means of actions \( c \) and \( u \), respectively called ‘controllable’ and ‘uncontrollable’ actions. Given an expected property described by some formula \( \varphi \), say in \( \mathcal{L}^\mu \), we use refinement quantifiers to express several classic verification/synthesis problems.

The \textit{the control problem} [40], know as the question “is there a way to control actions \( c \) of the system \( S \) so that property \( \varphi \) is guaranteed?”, can be expressed in \( \mathcal{L}_\forall \) by wondering whether

\[
S \models \exists_c \varphi .
\]

The \textit{module checking problem} [27] is the problem of determining whether an open system satisfies a given property. In other words, whether the property holds when the system is composed with an arbitrary environment. Let us say that action \( c \) is internal, while action \( u \) comes from the environment. We answer positively to the module checking problem iff \( S \models \forall_u \varphi \). As arbitrary environments are too permissive, we may force hypotheses such as restricting to \textit{non-blocking} environments. By ‘guarding’ the universal quantification over all \( u \)-refinements (i.e. all environments) with the \textbf{NonBlocking} assumption, the statement becomes

\[
S \models \forall_u (\text{NonBlocking} \Rightarrow \varphi)
\]

where \textbf{NonBlocking} is easily expressible in \( \mathcal{L}^\mu \) as \( \equiv \nu x. \Diamond u \top \land \Box x \).

The \textit{generalized control problem} is the combination of the two previous problems, by questioning the existence of a control such that the controlled system satisfies the property in all possible environments. This is expressed by wondering whether

\[
S \models \exists_c \forall_u (\text{NonBlocking} \Rightarrow \varphi) .
\]

A last example is borrowed from \textit{protocol synthesis problems}. Consider a specification, \textbf{MUTEX}, of a mutual exclusion protocol involving processes \( 1, 2, \ldots k \), and some property \( \varphi \) specified in \( \mathcal{L}^\mu \). Now we may ask if we can find a refinement of \textbf{MUTEX} that satisfies \( \varphi \) but also such that if process \( i \) is in the critical section (\( cs_i \)) at time \( n + 1 \), then this is known at time \( n \). This is expressed as

\[
\text{MUTEX} \models \exists [\text{AG}(\Diamond cs_i \Rightarrow \Box cs_i) \land \varphi]
\]

where \textbf{AG} is the CTL-modality, which rewrites in \( \mathcal{L}^\mu \) as \( \text{AG}(\psi) \equiv \nu x. \psi \land \Box x \) and meaning that this is true at any time. The refinement consists in moving the nondeterministic choices forward, so that a fork at time \( n \) becomes a fork at time \( n - 1 \) with each branch having a single successor at time \( n \), as depicted in Figure[4].

4.3 Refinement quantification is bisimulation quantification plus relativization

In Section[3.3] we presented a semantic perspective of refinement as bisimulation followed by model restriction, or, alternatively and equivalently, as a restricted bisimulation, namely
except for some propositional variable, followed by a model restriction to that variable. We now lift this result to a corresponding syntactic, logical, perspective of the refinement quantifier as a bisimulation quantifier followed by relativization.

More precisely, in this section we will show that a refinement formula $\exists_a \varphi$ is equivalent to a bisimulation quantification over a variable not occurring in $\varphi$, followed by a (non-standard) relativization for that agent to that variable, for which we write $\exists q \varphi^{(a,q)}$ (to be defined shortly). For refinement $\preceq$ for the set of all agents (recall that we write $\preceq$ for $\preceq_A$, and $\exists$ for $\exists_A$) we can expand this perspective to even more familiar ground: a refinement formula $\exists \varphi$ is equivalent to a bisimulation quantification over a variable not in $\varphi$ followed by (standard) relativization to that variable: $\exists q \varphi^q$. These results immediately clarify in what sense the refinement modality constitutes ‘implicit’ quantification, namely over a variable not occurring in the formula bound by it.

For the syntactic correspondence we first introduce the notion of relativization (for settings in modal logic, see [44, 31]). We propose a non-standard definition of relativization — non-standard in two ways. Firstly, it is relativization not merely to a propositional variable but also to an agent. This variation is inessential, but it matches our framework and purposes, and the standard definition is then a special case. Secondly, the relativization is arrow-eliminating and not state-eliminating (it is not mere domain restriction). This simplifies our approach, as the relativization need only be done in accessible states but not in the actual state (e.g., the relativization to $p$ of another variable $q$ is that same variable and not $p \land q$). That variation is similar to the (also inessential) difference, in the area of dynamic epistemic logic, between state eliminating public announcement and arrow eliminating public announcement. Given our purpose to translate refinement modal logic into bisimulation quantified modal logic, we also expand the definition of relativization to include quantifiers. This definition will also be used in Section 6.

**Definition 17 (Relativization)** Relativization to propositional variable $p$ for agent $a \in \mathcal{A}$.
A is defined as follows.

\[ q^{(a,p)} = q \]
\[ (\neg \varphi)^{(a,p)} = \neg \varphi^{(a,p)} \]
\[ (\varphi \land \psi)^{(a,p)} = \varphi^{(a,p)} \land \psi^{(a,p)} \]
\[ (\square_a \varphi)^{(a,p)} = \square_a (p \to \varphi^{(a,p)}) \]
\[ (\square_b \varphi)^{(a,p)} = \square_b \varphi^{(a,p)} \quad \text{for } b \neq a \]
\[ (\forall p \varphi)^{(a,p)} = \forall q \varphi[q \setminus p]^{(a,p)} \quad \text{where } q \text{ does not occur in } \varphi \]
\[ (\forall q \varphi)^{(a,p)} = \forall q \varphi^{(a,p)} \quad \text{for } q \neq p \]

We now have the obvious

**Lemma 18** Given model \( M_s \) with accessibility function \( R \) and \( R_a' \subseteq R_a \) such that: if \((t, t') \in R_a'\) then \( M_{t'} \models p \). Then \( M_s \models \varphi^{(a,p)} \) if and only if \( M_s|R_a' \models \varphi \).

**Proof** The proof is by induction on the structure of \( \varphi \).

- \( M_s \models q^{(a,p)} \iff \)
  \( M_s \models q \iff \) propositional variables do not change value
  \( M_s|R_a' \models q \)

- \( M_s \models (\neg \varphi)^{(a,p)} \iff \)
  \( M_s \models \neg \varphi^{(a,p)} \iff \)
  \( M_s|R_a' \not\models \varphi \iff \) I.H.
  \( M_s|R_a' \not\models \varphi \iff \)
  \( M_s|R_a' \models \neg \varphi \)

- \( M_s \models (\varphi \land \psi)^{(a,p)} \iff \)
  \( M_s \models \varphi^{(a,p)} \land \psi^{(a,p)} \iff \)
  \( M_s \models \varphi^{(a,p)} \) and \( M_s \models \psi^{(a,p)} \iff \) I.H.
  \( M_s|R_a' \models \varphi \) and \( M_s|R_a' \models \psi \iff \)
  \( M_s|R_a' \models \varphi \land \psi \)

- \( M_s \models (\square_a \varphi)^{(a,p)} \iff \)
  \( M_s \models \square_a (p \to \varphi^{(a,p)}) \iff \)
  \( \text{for all } t \in R_a : M_t \models p \to \varphi^{(a,p)} \iff \)
  \( \text{for all } t \in R_a : M_t \models p \) implies \( M_t \models \varphi^{(a,p)} \iff \) I.H.
  \( \text{for all } t \in R_a : M_t \models p \) implies \( M_t|R_a' \models \varphi \iff \)
  \( t \in sR_a \) and \( t \models p \) iff \( t \in sR_a' \)
  \( \text{for all } t \in R_a' : M_t|R_a' \models \varphi \iff \)
  \( M_s|R_a'| \models \varphi \)

- \( M_s \models (\square_b \varphi)^{(a,p)} \iff \)
  \( M_s \models \square_b \varphi^{(a,p)} \iff \)
  \( \text{for all } t \in R_b : M_t \models \varphi^{(a,p)} \iff \) I.H.
  \( \text{for all } t \in R_b \) (in \( M_t \)) : \( M_t|R_a' \models \varphi \iff \)
  \( sR_b \) in \( M \) equals \( sR_b \) in \( M|R_a' \)
  \( \text{for all } t \in R_b \) (in \( M_t|R_a' \)) : \( M_t|R_a' \models \varphi \iff \)
  \( M_s|R_a' \models \square_a \varphi \)
• \( M_s \models (\forall q \varphi)^{(a,p)} \Leftrightarrow \)
  \( M_s \models \forall q \varphi^{(a,p)} \Leftrightarrow \)
  for all \( N_t \succapprox N_t \models \varphi^{(a,p)} \Leftrightarrow \) I.H.
  for all \( N_t \succapprox N_t|R'_a \models \varphi \Leftrightarrow \)
  for all \( N_t' \succapprox N_t'|R'_a : N_t' \models \varphi \Leftrightarrow \)
  \( M_s|R'_a \models \forall q \varphi \)

\((*)\): The equivalence holds, because the bisimulation variation outside the \( M_s|R'_a \)
part of \( M_s \) does not affect the truth of a formula only evaluated on \( M_s|R'_a \). In other
words, for all \( N_t \) there is a \( N_t' \) such that \( N_t|R'_a \simeq N_t' \) and also, for all \( N_t' \)
there is a \( N_t \) that ‘expands’ \( N_t' \): such that, again, \( N_t|R'_a \simeq N_t' \).

• The other clause for the universal quantifier starts with a renaming operation, and
then proceeds as in the previous clause.

\( \square \)

Agent relativization relates as expected to the standard notion of relativization (to the
set of all agents simultaneously). This is because relativization to different variables for
different agents is commutative.

**Lemma 19** Let \( \varphi \in \mathcal{L}_{\bar{q}} \). Then \((\varphi^{(a,p)})^{(b,q)} = (\varphi^{(b,q)})^{(a,p)}\).

**Proof** By induction on the structure of \( \varphi \). The non-trivial cases are \( \Box_a \varphi \), \( \Box_b \varphi \) (follows
dually), \( \forall p \varphi \), and \( \forall q \varphi \) (also follows dually). Note that \( (a,p) \)-relativization distributes over
implication.

\( ((\Box_a \varphi)^{(a,p)})^{(b,q)} \Leftrightarrow ((\Box_a (p \rightarrow \varphi^{(a,p)}))^{(b,q)} \Leftrightarrow \)
\( ((\Box_a p)^{(b,q)} \rightarrow (\varphi^{(a,p)})^{(b,q)}) \Leftrightarrow \)
\( ((\Box_a (p \rightarrow (\varphi^{(b,q)}))^{(a,p)} \Leftrightarrow \)
\( (\Box_a \varphi^{(b,q)})^{(a,p)} \Leftrightarrow ((\Box_a \varphi)^{(b,q)})^{(a,p)} \)

\( ((\forall p \varphi)^{(a,p)})^{(b,q)} \Leftrightarrow ((\forall p \varphi^{(r \backslash p)})^{(a,p)})^{(b,q)} \Leftrightarrow \)
\( ((\forall r \varphi^{(r \backslash p)})^{(a,p)})^{(b,q)} \Leftrightarrow \)
\( ((\forall r (\varphi^{(r \backslash p)}))^{(a,p)})^{(b,q)} \Leftrightarrow \)
\( ((\forall r (\varphi^{(r \backslash p)}))^{(a,p)})^{(b,q)} \Leftrightarrow \)
\( ((\forall (\varphi^{(r \backslash p)}))^{(a,p)})^{(b,q)} \Leftrightarrow \)
\( ((\forall (\varphi^{(r \backslash p)}))^{(a,p)})^{(b,q)} \Leftrightarrow \)

choose \( r \neq q \) (or else, yet another step)

I.H.

substitution of other variables than \( q \)

\( \square \)
Given Lemma 19, we may view a sequence of relativizations \((\ldots (\varphi(a_1,p))\ldots (a_n,p))\) as a relativization \(\varphi^{(a_1,\ldots,a_n,p)}\) to the set of agents \(\{a_1,\ldots,a_n\}\), and a sequence of relativizations \((\ldots (\varphi(a_1,p_1))\ldots (a_n,p_n))\) as a relativization \(\varphi^{(a_1,\ldots,a_n,q)}\) for some variable \(q\); where it is important to observe that \(q\) is typically not a truth function of \(p_1,\ldots,p_n\) (so, in particular, typically not the conjunction \(p_1 \land \ldots \land p_n\)). Therefore for \(\varphi^{(A,p)}\) we can write \(\varphi^p\): the usual relativization for all agents simultaneously. Almost usual: we have tied relativization to the semantic process of arrow elimination (or, from a tree unwinding perspective: pruning), whereas standard relativization is typically model restriction that is state elimination. If the models satisfy seriality, there is no difference. If the actual state satisfies the relativization atom, there is also no difference. In a related area, dynamic epistemic logic, the two options represent the familiar alternative semantics for the public truthful announcement of \(p\): state elimination [35, 8] versus arrow elimination [20, 26].

To make the syntactic correspondence we now introduce a translation \(t: \mathcal{L}_\forall \to \mathcal{L}_\forall\).

**Definition 20** By induction on \(\varphi \in \mathcal{L}_\forall\). All clauses except \(\forall_a\varphi\) are trivial.

\[
\begin{align*}
    t(p) & = p \\
    t(\neg \varphi) & = \neg t(\varphi) \\
    t(\varphi \land \psi) & = t(\varphi) \land t(\psi) \\
    t(\Box a \varphi) & = \Box_a t(\varphi) \\
    t(\forall_a \varphi) & = \forall_p t(\varphi)^{(a,p)} \quad \text{where \(p\) does not occur in \(\varphi\)}
\end{align*}
\]

**Example 21**

\[
\begin{align*}
    t(\exists_a \exists_b r) & = \exists_p t(\exists_b r)^{(a,p)} \\
    \exists_p(\exists_p t(r)^{(b,p)})^{(a,p)} & = \exists_p(\exists_p r)^{(b,p)}^{(a,p)} \\
    \exists_p(\exists_p r)^{(a,p)} & = \exists_p \exists_q r^{(a,q)} = \exists_p \exists q r
\end{align*}
\]

From Lemma 18 and Definition 20 we now immediately get

**Proposition 22** Let \(\varphi \in \mathcal{L}_\forall\). Then \(\varphi\) is equivalent to \(t(\varphi)\).

We allow ourselves a slight abuse of language here: given any \(M\), the value of \(\varphi\) in the semantics for refinement modal logic is equivalent to the value of \(t(\varphi)\) in that model, in the semantics for bisimulation quantified modal logic. From Proposition 22 follows, to have the characteristic aspect of the translation stand out:

**Corollary 23** Consider \(\exists \varphi\) with \(\varphi \in \mathcal{L}\) (i.e., \(\exists\)-free). Then

- a-refinement is bisimulation quantification plus a-relativization:
  \(\exists_a \varphi\) is equivalent to \(\exists_p \varphi^{(a,p)}\);

- refinement is bisimulation quantification plus relativization:
  \(\exists \varphi\) is equivalent to \(\exists_p \varphi^p\).

In the logic of public announcements, the latter is written as: \(\exists \varphi\) is equivalent to \(\exists p(p!)\varphi\).
4.4 Alternating refinement relations

Alternating transition systems (ATS) were introduced [4] to model multiagent systems, where in each move of the game between the agents of an ATS, the choice of an agent at a state is a set of states and the successor state is determined by considering the intersection of the choices made by all agents. A notion of $a$-alternating refinement was introduced to reflect a refined behavior of agent $a$ while keeping intact the behavior of the others. When restricting to turn-based ATS where only one agent plays at a time (concurrent moves are also allowed in the full setting), $a$-alternating refinement amounts to require ‘forth’ for all $b \in A \setminus \{a\}$ as we do, but ‘back’ just for agent $a$. As a consequence, an $a$-refinement is a particular $a$-alternating refinement. A logical characterization of $a$-alternating refinement has been proposed (it essentially relies on the modality $\exists a$ combined with the linear time temporal logic LTL) in the sense that if an ATS $S'$ $a$-refines an ATS $S$, every formula true in $S'$ is also true in $S$. Notice however that the operator $\exists a$ has a more restricted semantics than the one we propose, since the quantification does not range over all possible refinements of the structure but only over refinements obtained by pruning the unraveling of the ATS. Soon after, the more general setting of alternating-time temporal logics [2, 3] considered universal and existential quantifications over $a$-refinements, for arbitrary $a$, combined with LTL formulas. It is worthwhile noticing that the quantifiers still range over particular refinements, and always in the original structure. As a consequence, the language cannot express the ability to nest refinements for different agents. This is easily done in our language $L^\forall$, as the formula $\exists a \left( \Box b \top \land \Diamond a \left( \exists a \Box a b \top \right) \right)$ exemplifies. This formula tells us that one of the choices that $a$ can make, results in $b$ knowing $p$ and $a$ contemplating a subsequent choice by $b$ that makes her to get to know $p$ as well.

5 Axiomatization RML

Here we present the axiomatization $\text{RML}$ for the logic $\text{RML}$. We show the axioms and rules to be sound, we give example derivations, and this is followed by the completeness proof.

The axiomatization presented is a substitution schema, since the substitution rule is not valid. The substitution rule says that: if $\varphi$ is a theorem, and $p$ occurs in $\varphi$, and $\psi$ is any formula, then $\varphi[\psi \backslash p]$ is a theorem. Note that for all atomic propositions $p$, $p \rightarrow \forall p$ is valid, but the same is not true for an arbitrary formula, e.g. $\Diamond a \top \rightarrow \forall \Diamond a \top$ is not valid, because after the maximal refinement there is no accessible state, so that $\Diamond a \top$ is then false even if it was true before. The logic $\text{RML}$ is therefore not a normal modal logic.

Definition 24 (Axiomatization RML) The axiomatization $\text{RML}$ consists of all sub-
stitution instances of the axioms

| Prop | All tautologies of propositional logic |
|------|--------------------------------------|
| K    | □a(φ → ψ) → □aφ → □aψ              |
| R    | ∀a(φ → ψ) → ∀aφ → ∀aψ              |
| RProp | ∀a(p ↔ ¬p) ← p and ∀a¬p ↔ ¬p         |
| RK   | ∃a ▽a Φ ↔ ▽a ∃a Φ                  |
| RKmulti | ∃a ▽a Φ ↔ ▽a ∃a Φ                |
| RKconj | ∃a ∩b∈B ▽b Φb ↔ ∩b∈B ∃a ▽b Φb        |

and the rules

| MP   | From φ → ψ and φ infer ψ |
| NecK | From φ infer □aφ          |
| NecR | From φ infer ∀aφ          |

where a, b ∈ A, p ∈ P, and B ⊆ A. If φ is derivable, we write ⊢ φ, and φ is called a theorem, as usual. The well-known axiomatization K for the logic K consists of the axioms Prop, K, and the rules MP and NecK.

In the definition, given Φ = {φ1, ..., φn}, note that ∃a ▽a Φ ↔ ∩φ∈Φ ▽a ∃a φ (see the technical preliminaries) and so for ∃a ▽a {φ1, ..., φn} ↔ ▽a ∃a φ1 ∧ ... ∧ ▽a ∃a φn. The axiomatization RML is surprisingly simple given the complexity of the semantic definition of the refinement operator ∀; and given the well-known complexity of axiomatizations for logics involving bisimulation quantifiers instead of this single refinement quantifier. We note that while refinement is reflexive, transitive and satisfies the Church-Rosser property (Proposition 2 and Proposition 16), the corresponding modal axioms are not required. These properties are schematically derivable. First, we demonstrate soundness of RML.

*Given the definitions of □ and ▽ in terms of cover, it may be instructive to see how the RK axiom works as a reduction principle for ∃φ and ∃φ — note that we need both, as there is no principle for ∃¬φ. For simplicity we do not label the operators with agents. We get:

\[ \exists a \Box \phi \leftrightarrow \exists a (\bigvee \phi \vee \bigwedge \phi) \]

\[ \leftrightarrow \exists a \bigvee \phi \vee \exists a \bigwedge \phi \]

\[ \leftrightarrow \exists a \bigvee \phi \vee \exists a \perp \]

\[ \leftrightarrow \top \]

and

\[ \exists a \Diamond \phi \leftrightarrow \exists a \bigvee \phi, \top \]

\[ \leftrightarrow \bigvee \exists a \phi \vee \bigwedge \exists a \perp \]

\[ \leftrightarrow \bigvee \exists a \phi \]

One may wonder why we did not choose ∃a φ ↔ ⊤ and ∃a φ ↔ ∃a φ (we recall Proposition 16 as primitives in the axiomatization, as, after all, these are very simple axioms. They are of course valid, but the axiomatization would not be complete. The axiom RK is much more powerful, as this not merely allows Φ = {φ}, Φ = ∅, and Φ = {φ, ⊤}, but any finite set of formulas.*
5.1 Soundness

Theorem 25 The axiomatization RML is sound for RML.

Proof As all models of \( \mathcal{L}_\phi \) are models of \( \mathcal{L} \), the schemas Prop, K and the rule MP and NecK are all sound. We deal with the remaining schemas and rules below.

**R**

Suppose that \( M_s \) is a model such that \( M_s \models \forall_a (\varphi \to \psi) \), and \( M_s \models \forall_a \varphi \). Then for every \( N_t \), where \( N_t \preceq_a M_s \), we have \( N_t \models \varphi \to \psi \), and also \( N_t \models \varphi \). From \( N_t \models \varphi \to \psi \) and \( N_t \models \varphi \) follows \( N_t \models \psi \). As \( N_t \) was arbitrary model such that \( N_t \preceq_a M_s \), from that and \( N_t \models \psi \) follows \( M_s \models \forall_a \psi \).

**RProp**

Let \( M_s \) and \( N_t \) be given such that \( N_t \preceq_a M_s \). By Definition [1] for the semantics of refinement, we have that \( s \in V^M(p) \) if and only if \( t \in V^N(p) \). Therefore \( M_s \models p \iff N_t \models p \), for every \( M_s \) and \( N_t \) with \( N_t \preceq_a M_s \). Therefore \( M_s \models p \iff M_s \models \forall_a p \) for every \( M_s \), i.e. \( \models p \iff \forall_a p \). Similarly, for \( \models \neg p \iff \forall_a \neg p \), using that \( s \not\in V^M(p) \) if and only if \( t \not\in V^N(p) \).

**RK**

Suppose \( M_s \) is a model, where \( M = (S, R, V) \), such that for some set \( \Phi \), \( M_s \models \bigwedge \diamond_a \exists_a \Phi \). Therefore, for every \( \varphi \in \Phi \) there is some \( t^\varphi \in sR_a \) such that \( M_{t^\varphi} \models \exists_a \varphi \). Thus, for each \( \varphi \in \Phi \), there is some model \( N^\varphi_{u^\varphi} \preceq_a M_{t^\varphi} \), where \( N^\varphi = (S^\varphi, R^\varphi, V^\varphi) \), such that \( N^\varphi_{u^\varphi} \models \varphi \). Without loss of generality, we may assume that for all \( \varphi, \varphi' \in \Phi \) the models \( N^\varphi \) and \( N^{\varphi'} \) are disjoint.

We construct the model \( M^\Phi = (S^\Phi, R^\Phi, V^\Phi) \) such that:

\[
\begin{align*}
S^\Phi & = \{s'\} \cup S \cup \bigcup_{\varphi \in \Phi} S^\varphi \\
R^\Phi_a & = \{(s', u^\varphi) \mid \varphi \in \Phi\} \cup R_a \cup \bigcup_{\varphi \in \Phi} R^\varphi_a \\
R^\Phi_b & = \{(s', t) \mid (s, t) \in R_b \} \cup R_b \cup \bigcup_{\varphi \in \Phi} R^\varphi_b \\
V^\Phi(p) & = \{s'\} \cup V(p) \cup \bigcup_{\varphi \in \Phi} V^\varphi(p) \\
\end{align*}
\]

for \( b \neq a \) and \( p \in P \).

where \( \{s'\} = \{s'\} \) if \( s \in V(p) \) and else \( \{s'\} = \emptyset \).

We can see that \( M_s \preceq_a M^\Phi_s \), via the relation \( R^\varphi = \{(s, s')\} \cup I \cup \bigcup_{\varphi \in \Phi} R^\varphi \) where \( I \) is the identity on \( S \) and each \( R^\varphi \) is the refinement relation corresponding to \( M_{t^\varphi} \preceq_a N^\varphi_{u^\varphi} \) (see also [22]). Furthermore, for each \( t \in s'R^\Phi_a \) it is clear that \( M^\Phi_t \models N^\varphi_{u^\varphi} \) for some \( \varphi \), and thus \( M^\Phi_t \models \bigvee \Phi \). Therefore \( M^\Phi_s \models \bigwedge \diamond_a \bigvee \Phi \). Finally, for each \( \varphi \in \Phi \) there is some \( u^\varphi \in s'R^\Phi_a \) where \( M^\Phi_{u^\varphi} \models \varphi \), so for each \( \varphi \in \Phi \) we have \( M^\Phi_s \models \diamond_a \varphi \), so we have \( M^\Phi_s \models \bigwedge \diamond_a \Phi \). Combined, \( M^\Phi_s \models \diamond_a \bigvee \Phi \) and \( M^\Phi_s \models \bigwedge \diamond_a \Phi \) state that \( M^\Phi_s \models \bigwedge \diamond_a \Phi \), and therefore \( M_s \models \bigwedge \diamond_a \Phi \).

Conversely, suppose that \( M_s \models \bigwedge \diamond_a \Phi \). Therefore, there is a model \( N_t \preceq_a M_s \) such that \( N_t \models \bigwedge \diamond_a \Phi \)—where \( N = (S^\Phi, R^\Phi, V^\Phi) \). Expanding the definition, we have that for every \( \varphi \in \Phi \) there is some \( u \in tR^\Phi_a \) such that \( N_u \models \varphi \). Also, because of back, for every such \( u \in tR^\Phi_a \) there is some \( v \in sR_a \) such that \( N_u \preceq_a M_v \). Combining these statements
we have that for every \( \varphi \in \Phi \) there is some \( v \in sR_a \) such that \( M_v \models \exists_a \varphi \), and thus \( M_s \models \bigwedge \Diamond_a \exists_a \Phi \).

**RKmulti**

The direction \( \nabla_b \exists_a \Phi \to \exists_a \nabla_b \Phi \) is proved as in the case **RK**. Note that our assumption is now even stronger, as \( \nabla_b \exists_a \Phi \) entails \( \bigwedge \Diamond_b \exists_a \Phi \).

Conversely, suppose that \( M_s \models \exists_a \nabla_b \Phi \). Therefore, there is a model \( M'_t \preceq_a M_s \) such that \( M'_t \models \nabla_b \Phi \)—let the accessibility relation for agent \( b \) in \( M' \) be \( R'_b \). Expanding the definition, we have that for every \( \varphi \in \Phi \) there is some \( u \in tR'_b \) such that \( M'_u \models \varphi \). Also, because of **back**, for every such \( u \in tR'_b \) there is some \( v \in sR_b \) such that \( M'_v \preceq_a M_s \).

Combining these statements we have that for every \( \varphi \in \Phi \) there is some \( v \in sR_b \) such that \( M_v \models \exists_a \varphi \), and thus \( M_s \models \bigwedge \Diamond_a \exists_a \Phi \). However, as **forth** also holds for agent \( b \), the \( v \in sR_b \) we could construct above are also all the states \( v \) accessible from \( s \). Therefore we also have \( M_s \models \square_b \bigvee \exists_a \Phi \), so together we get \( M_s \models \nabla_b \exists_a \Phi \).

**RKconj**

The direction \( \exists_a \bigwedge_{b \in B} \nabla_b \Phi^b \to \bigwedge_{b \in B} \exists_a \nabla_b \Phi^b \) is merely a more complex form of pattern \( \exists_a (\varphi \land \psi) \to (\exists_a \varphi \land \exists_a \psi) \) which is derivable similar to \( \Diamond_a (\varphi \land \psi) \to \Diamond_a \varphi \land \Diamond_a \psi \) in the modal logic \( K \), using the axiom **R** in place of \( K \).

For the other direction, suppose that \( M_s \) is such that \( M_s \models \bigwedge_{b \in B} \exists_a \nabla_b \Phi^b \), where \( B \subseteq A \). We need to show that \( M_s \models \exists_a \bigwedge_{b \in B} \nabla_b \Phi^b \). To do this we follow the same strategy as for proving **RK**: we construct an \( a \)-refinement \( N_t \) of \( M_s \), and show that \( N_t \models \bigwedge_{b \in B} \nabla_b \Phi^b \).

We begin by constructing the model \( N_t \). Suppose that \( a \in B \). Then we have \( M_s \models \exists_a \nabla_a \Phi^a \), and by **RK** this implies that \( M_s \models \bigwedge \Diamond_a \exists_a \Phi^a \). We also have that for every \( b \in B - \{a\} \), \( M_s \models \exists_a \nabla_b \Phi^b \), and by **RComm** this implies that \( M_s \models \nabla_b \exists_a \Phi^b \), and by the definition of the cover operator, this implies that \( M_s \models \bigwedge \Diamond_b \exists_a \Phi^b \). Hence for every \( b \in B \) and \( \varphi \in \Phi^b \), we have that \( \Diamond_b \exists_a \varphi \). (In other words, for some big set of formulas \( \Psi \) we have that \( M_s \models \bigwedge \Diamond_b \exists_a \Psi \).) At this stage it suffices to refer to the very similar construction in the soundness proof for axiom **RK**, from which similarly to there follows \( N_t \models \bigwedge_{b \in B} \nabla_b \Phi^b \).

**NecR**

If \( \varphi \) is a validity, then it is satisfied by every model, so for any model \( M_s \), \( \varphi \) is satisfied by every model \( N_t \preceq_a M_s \), and hence every model \( M_s \) satisfies \( \forall_a \varphi \). \( \square \)

The soundness of axiom **RK** is visualized in Figure 5. It depicts the interaction between refinement and modality involved in this axiom \( \exists_a \nabla_a \Phi \leftrightarrow \bigwedge \Diamond_a \exists_a \Phi \), for the case that \( \Phi = \{ \varphi_1, \varphi_2, \varphi_3 \} \). The single lines are modal accessibility, and the double lines are the refinement relations. The solid lines are given, and the dashed lines are required. Accessibility relations for other agents than \( a \) are omitted. The picture on the left depicts the implication from left to right in the axiom, and the picture on the right depicts the implication from right to left. Note that the states satisfying \( \varphi_2 \) and \( \varphi_3 \) have the same origin \( u \) in \( M \)—the typical sort of duplication (resulting in non-bisimilar states) allowed when having **back** but not **forth**. Apart from \( u \) and \( t \), state \( s \) in \( M \) has yet another accessible state \( v \), that does not occur in the refinement relation: the other typical sort of thing when having **back** but not... 29
forth. Therefore, on the right side of the equivalence in axiom RK we only have $\land \Diamond_a \exists_a \Phi$ and we cannot guarantee that $\Box_a \lor \exists_a \Phi$ also follows from the left-hand side.

The axiom RKmulti, defined as $\exists_a \nabla_b \Phi \leftrightarrow \nabla_b \exists_a \Phi$ for $a \neq b$, says that refinement with respect to one agent does not interact with the modalities (the uncertainty, say) for another agent: the operators $\nabla_b$ and $\exists_a$ simply commute. This in contrast the axiom RK where on the right-hand side a construct $\Box_a \lor \exists_a \Phi$ is 'missing', so to speak. If it had been $\Box_a \lor \exists_a \Phi \land \land \Diamond_a \exists_a \Phi$, then we would have had $\nabla_a \exists_a \Phi$, as in RK. The difference between RK and RKmulti is because in the former there is no forth requirement for $a$ in refinement: given some refinement wherein we have a cover of $\Phi$, so that at least one of $\Phi$ is necessary (the $\exists_a \nabla_a \Phi$ bit), for each of the covered states we can trace an origin before the refinement, because of back. But there may be more originally accessible states, so whatever holds in those origins, although it is all possible, is not necessary. So we have $\land \Diamond_a \exists_a \Phi$, but we do not have $\Box_a \lor \exists_a \Phi$. In contrast, when the agents are different, back and forth must hold for agent $b$ in a refinement $\succeq_a$ witnessing the operator $\exists_a$: for an $a$-refinement, back and forth must hold for all agents $b \neq a$. Figure 6 should further clarify the issue—compare this to Figure 5. The main difference between the figures is that there cannot now be yet another state $v$ accessible from $s$ but not ‘covered’ as the origin of one of the refined states. In Figure 5 what holds in $t$ and $u$ is not necessary for $a$, but in Figure 6 what holds in $t$ and $u$ is necessary for $b$.

5.2 Example derivations

Example 26 $\vdash \Diamond_a \top \rightarrow \exists_a (\Box_a p \lor \Box_a \neg p)$

In an epistemic setting, where $\Box_a p$ means that the agent knows $p$, and where (in $S5$ models) the condition $\Diamond_a \top$ is always satisfied, this validity expresses that the agent can always find
Figure 6: The interaction between refinement and modality involved in axiom RKmulti.

out the truth about \( p \): if true, announce \( p \) (and announcement is a model restriction, and therefore a refinement), after which \( p \) is known, and if false, announce that \( p \) is false, after which \( p \) is known to be false. This validity is indeed also a theorem of RML. For that, it is more convenient to keep the cover representation. We note that \( \Box_a p \) is in cover notation \( \nabla_a \{p\} \lor \nabla_a \emptyset \). The requirement \( \Diamond_a \top \) or seriality (of consistent belief) rules out the alternative \( \nabla_a \emptyset \). It therefore suffices to derive \( \Diamond_a \top \rightarrow \exists_a (\nabla_a \{p\} \lor \nabla_a \{\neg p\}) \). In some cases several deductions have been combined into single statements, but this is restricted to cases of well-known modal theorems.

\[
\begin{align*}
\vdash \Diamond_a \top & \iff \Diamond_a (p \lor \neg p) & \text{Prop, NecK, K} \\
\vdash \Diamond_a (p \lor \neg p) & \iff (\Diamond_a p \lor \Diamond_a \neg p) & \text{Prop, NecK, K} \\
\vdash \Diamond_a p & \rightarrow \exists_a \nabla_a \{p\} & \text{See below} \\
\vdash \Diamond_a \neg p & \rightarrow \exists_a \nabla_a \{\neg p\} & \text{See below} \\
\vdash \exists_a \Box_a p & \rightarrow \exists_a (\nabla_a \{p\} \lor \nabla_a \{\neg p\}) & \text{Prop, NecR, R} \\
\vdash \exists_a \Box_a \neg p & \rightarrow \exists_a (\nabla_a \{p\} \lor \nabla_a \{\neg p\}) & \text{Prop, NecR, R} \\
\vdash \Diamond_a \top & \rightarrow \exists_a (\nabla_a \{p\} \lor \nabla_a \{\neg p\}) & \text{Prop, MP}
\end{align*}
\]

Lines 3 and 4 of the derivation require the following deduction, which is true for all propositional formulas \( \varphi \):

\[
\begin{align*}
\vdash \varphi & \iff \exists_a \varphi & \text{RProp} \\
\vdash \Diamond_a \varphi & \iff \Diamond_a \exists_a \varphi & \text{Prop, NecK, K} \\
\vdash \exists_a \Diamond_a \varphi & \iff \exists_a \nabla_a \{\varphi\} & \text{RK[Φ = \{\varphi\}]}
\end{align*}
\]

**Example 27** \( \vdash (\Diamond_a p \land \Diamond_b p \land \Diamond_a \neg p \land \Diamond_b \neg p) \rightarrow \exists_a (\Box_a p \land \neg \Box_b p) \)

Consider the informative development described in Example 4.2: given an initial information state wherein agents \( a \) and \( b \) consider either value of \( p \) possible, \( a \) can be informed such that afterwards \( a \) believes that \( p \) but not \( b \). The above formalizes that. (A small difference is that the following says that \( a \) is informed *privately*, it has \( \exists_a \) only; in Example 4.2 we needed a \( \exists \) operator, that is, a stack \( \exists_a \exists_b \).)
Let \( \varphi \) be \((\lozenge_a p \land \lozenge_b p \land \lozenge_a \neg p \land \lozenge_b \neg p)\). In the following we also use ‘substitution of equivalents’, see Lemma 31.

\[
\vdash \varphi \rightarrow \lozenge_a p \land \lozenge_b \neg p \\
\vdash \varphi \rightarrow \lozenge_a p \land \Box_b \{\neg p, \top\} \\
\vdash \varphi \rightarrow \lozenge_a \neg \neg p \land \Box_b \{\neg \forall_a \neg p, \forall_a \top\} \\
\vdash \varphi \rightarrow \lozenge_a \exists_a p \land \Box_b \{\exists_a \neg p, \exists_a \top\} \\
\vdash \varphi \rightarrow \exists_a \Box_a p \land \Box_b \{\exists_a \neg p, \exists_a \top\} \\
\vdash \varphi \rightarrow \exists_a (\Box_a p \land \lozenge_b \{\neg p, \top\}) \\
\vdash \varphi \rightarrow \exists_a (\lozenge_a p \land \neg \Box_b p) \\
\vdash \varphi \rightarrow \exists_a (\lozenge_a p \land \neg \lozenge_b p)
\]

5.3 Completeness

Completeness is shown by a fairly but not altogether straightforward reduction argument: every formula in refinement modal logic is equivalent to a formula in modal logic. So it is a theorem, if its modal logical equivalent is a theorem. In the axiomatization \( \text{RML} \) we can observe that all axioms involving refinement operators \( \exists \) are equivalences, except for \( \text{R} \); however, \( \exists_a (\varphi \lor \psi) \leftrightarrow \exists_a \varphi \lor \exists_a \psi \) is a derivable theorem. This means that by so-called ‘rewriting’ we can push the \( \exists \) operators further inward into a formula, until we reach some expression \( \exists \varphi \) where \( \varphi \) contains no refinement operators. Now we come to the less straightforward part. Because there is a hitch: there is no general way to push a \( \exists \) beyond a negation (or, for that matter, beyond a conjunction). For that, we use another trick, namely that all modal logical formulas are equivalent to formulas in the cover logic syntax, and that all those are equivalent to formulas in disjunctive form (see the introduction) in cover logic. Using that, once we reached some innermost \( \exists \varphi \) where \( \varphi \) contains no refinement operators, we can continue pushing that refinement operator downward until it binds a propositional formula only, and disappears in smoke because of the \( \text{RProp} \) axiom. Then, iterate this. All \( \exists \) operators have disappeared in smoke. We have a formula in modal logic.

For a smooth argument we first give some general semantic and proof theoretic results, after which we apply the reduction argument and demonstrate completeness.

**Proposition 28**

1. \( \vdash \forall_a (\varphi \land \psi) \leftrightarrow \forall_a \varphi \land \forall_a \psi \)
2. \( \vdash \exists_a (\varphi \lor \psi) \leftrightarrow \exists_a \varphi \lor \exists_a \psi \)
3. \( \vdash \exists_a (\varphi \land \psi) \rightarrow \exists_a \varphi \land \exists_a \psi \)

**Proof** Item 1. can be easily derived from \( \text{R} \), \( \text{NecR} \) and \( \text{MP} \), similarly to the way that in modal logic we derive \( \vdash \Box (\varphi \land \psi) \leftrightarrow \Box \varphi \land \Box \psi \). The other derivations are similar. \( \square \)
Proposition 29

1. \( \vdash \forall_a \varphi \iff \varphi \) for all propositional \( \varphi \).

2. \( \vdash \exists_a \varphi \iff \varphi \) for all propositional \( \varphi \).

3. \( \vdash (\varphi \land \exists_a \psi) \iff \exists_a (\varphi \land \psi) \) for all propositional \( \varphi \) (and any \( \psi \in L_\forall \)).

Proof Item 1.: From axiom \texttt{RProp} that \( \forall_a p \iff p \) and \( \forall_a \neg p \iff \neg p \) we can immediately get \( p \iff \exists_a p \) and \( \neg p \iff \exists_a \neg p \) (reverse the implications). Now, use induction on the (propositional) structure of \( \varphi \), using axiom \texttt{RProp} including the diamond version above, axiom \texttt{Prop} for all propositional equivalences, and—if we please ourselves with inductive cases negation and conjunction, the theorem \( \vdash \forall_a (\varphi \land \psi) \iff \forall_a \varphi \land \forall_a \psi \) (Proposition 28). Item 2. is similar to Item 1. For Item 3., Proposition 28 demonstrated that \( \exists_a (\varphi \land \psi) \rightarrow \exists_a (\varphi \land \psi) \) from which, using Item 2., also follows \( \varphi \land \exists_a \psi \). For the other direction we first derive \( (\forall_a \varphi \land \exists_a \psi) \iff \exists_a (\varphi \land \psi) \) by propositional means and applications of \texttt{Nec} and \texttt{R}, and then use that \( \forall_a \varphi \iff \varphi \) (Item 1.).

□

Definition 30 (Substitution of equivalents) An axiomatization satisfies substitution of equivalents if the following holds. Let \( \varphi_1, \varphi_2, \varphi_3 \) be formulas in the logical language. If \( \varphi_1 \) is equivalent to \( \varphi_2 \) and \( \varphi_1 \) is a subformula of \( \varphi_3 \), and \( \varphi_3 \) is a theorem, then \( \varphi_3[\varphi_2 \setminus \varphi_1] \) is also a theorem. □

Proposition 31 The axiomatization \texttt{RML} satisfies substitution of equivalents.

Proof This can be shown by induction on \( \varphi_3 \). For the cases \( \Box_a \varphi \) and \( \forall_a \varphi \), note that if \( \Box_a \varphi \) is a theorem, then \( \varphi \) was already a theorem (and IH.), and that is \( \forall_a \varphi \) is a theorem, \( \varphi \) also was already a theorem (and IH.). □

We now first show that every \( L_\forall \) formula is logically equivalent to a \( L \) formula. We then show that if the latter is a theorem in \texttt{K}, the former is a theorem in \texttt{RML}.

Proposition 32 Every formula of \( L_\forall \) is logically equivalent to a formula of \( L \).

Proof Given a formula \( \psi \in L_\forall \), we prove by induction on the number of the occurrences of \( \exists_a \) in \( \psi \) (for any \( a \in A \)) that it is equivalent to an \( \exists_a \)-free formula, and therefore to a formula \( \varphi \in L \), the standard modal logic. The base is trivial. Now assume \( \psi \) contains \( n + 1 \) occurrences of \( \exists_a \)-operators for some \( a \in A \) (so these may be refinement operators for different agents). Choose a subformula of type \( \exists_a \varphi \) of our given formula \( \psi \), where \( \varphi \) is \( \exists_b \)-free for any \( b \in A \) (i.e. choose an innermost \( \exists_a \)). Let \( \varphi' \) be a disjunctive formula that is equivalent to \( \varphi \). We prove by induction on the structure of \( \varphi' \) that \( \exists_a \varphi' \) is logically equivalent to a formula \( \chi \) without \( \exists_a \). There are two cases:

- \( \exists_a (\varphi \lor \psi) \);
• \( \exists_a(\varphi_0 \land \bigwedge_{b \in B} \nabla_b \Phi^b) \) where \( \varphi_0 \) is propositional, \( B \subseteq A \), and each \( \Phi^b \) a set of dfs.

In the first case, apply Proposition 28.2., we get \( \exists_a \varphi \lor \exists_a \psi \), and then apply induction. In the second case, if \( B = \emptyset \) we use that \( \exists_a \varphi_0 \leftrightarrow \varphi_0 \) (Proposition 29.2). If \( B \neq \emptyset \), then from Proposition 29.3. follows that this is equivalent to \( \varphi_0 \land \bigwedge_{b \in B} \nabla_b \Phi^b \), and we further reduce the right conjunct with one of the axioms RK (if \( B = \{a\} \)), RKmulti (if \( B = \{b\} \) with \( b \neq a \)), or RKconj (if \( |B| > 1 \)), and apply induction again.

Thus we are able to push the refinement operators deeper into the formula until they eventually reach a propositional formula, at which point they disappear and we are left with the required \( \exists \)-free formula \( \chi \) that is equivalent to \( \exists \varphi \). Replacing \( \exists \varphi' \) by \( \chi \) in \( \psi \) gives a result with one less \( \exists \)-operator, to which the (original) induction hypothesis applies. \( \square \)

**Proposition 33** Let \( \varphi \in \mathcal{L}_\forall \) be given and \( \psi \in \mathcal{L} \) be equivalent to \( \varphi \). If \( \psi \) is a theorem in \( K \), then \( \varphi \) is a theorem in \( \text{RML} \).

**Proof** Given a \( \varphi \in \mathcal{L}_\forall \), Proposition 32 gives us an equivalent \( \psi \in \mathcal{L} \). Assume that \( \psi \) is a theorem in \( K \). We can extend the derivation of \( \psi \) to a derivation of \( \varphi \) by observing that all steps used in Proposition 32 are not merely logical but also provable equivalences — where we also apply Proposition 31 of substitution of equivalents. \( \square \)

**Theorem 34** The axiom schema RML is sound and complete for the logic RML.

**Proof** The soundness proof is given in Theorem 25, so we are left to show completeness. Suppose that \( \varphi \in \mathcal{L}_\forall \) is valid: \( \models \varphi \). Applying Lemma 32 we know that there is some equivalent formula \( \psi \in \mathcal{L} \), i.e., not containing any refinement operator. As \( \varphi \) is valid, from that and the validity \( \varphi \leftrightarrow \psi \) it follows that \( \psi \) is also valid in refinement modal logic, and therefore also valid in the logic \( K \) (note that the model class is the same). From the completeness of \( K \) it follows that \( \psi \) is derivable, i.e. it is a theorem. From Proposition 33 it follows that \( \varphi \) is a theorem. \( \square \)

### 5.4 The single-agent case

The axiomatization for the single-agent case is the unlabelled version of \( \text{RML} \), minus the axioms RKmulti and RKconj. The single-agent axiomatization was presented in [46]. The completeness proof there is (slightly) different from the multi-agent case of the proof here. In [46] it is used that every refinement modal logical formula is equivalent to a formula in cover logic with the special syntax \( \varphi ::= \bot | T | \varphi \lor \varphi | p \land \varphi | \neg p \land \varphi | \nabla \{ \varphi, \ldots, \varphi \} \) \([9, 28]\), plus induction on that form. (This syntax is of course very ‘disjunctive formula like’.) That proof was suggested by Yde Venema, as a shorter alternative to the proof with disjunctive forms.
5.5 Refinement epistemic logic

Refinement modal logic RML is presented with respect to the class of all models. As mentioned in Section 4.1, by restricting the class of models that the logic is interpreted over, we may associate different meanings with the modalities. For example, the epistemic logic S5, a.k.a. the logic of knowledge, is interpreted over the model class S5, and the logic of belief KD45 is interpreted over the class KD45. Given any class of models C, the semantic interpretation of ∀ is given by:

\[ M_s \models \forall a \varphi \text{ iff for all } M'_s \in C : M_s \geq_a M'_s \text{ implies } M'_s \models \varphi. \]

Thus we can consider various refinement epistemic logics. Although \( \exists \Box \bot \) is a validity in RML (just remove all access) it is not so in refinement epistemic logic, interpreted on S5 models, because seriality of models must be preserved in every refinement. And therefore it is also not valid in refinement logic of belief.

Our axiomatization RML may not be sound for more restricted model classes. Let us consider the single-agent case, and the axiom

\[ \text{RK} \quad \exists \nabla \Phi \leftrightarrow \bigwedge \Diamond \Phi. \]

For example, in S5 we have that \( \exists \nabla (\Box p, \neg \Box p) \) is inconsistent, but that \( \Diamond \exists \Box p \land \Diamond \exists \neg \Box p \) is consistent: you do not consider an informative development possible after which you both know and don’t know \( p \) at the same time. Therefore, axiom RK is invalid for that class.

The axioms replacing RK in refinement logic of knowledge and refinement logic of belief are, respectively:

\[ \text{RS5} \quad \exists \nabla \Phi \leftrightarrow (\bigvee \Phi \land \bigwedge \Diamond \Phi), \]

and

\[ \text{RKD45} \quad \exists \nabla \Phi \leftrightarrow \bigwedge \Diamond \Phi, \]

where \( \Phi \) is a set of purely propositional formulas. Now if apart from RS5 we also add the usual S5 axioms T, 4, and 5, we have a complete axiomatization for the refinement logic of knowledge. In the case of the refinement logic of belief, we add axioms D (for seriality), 4, and 5 and RKD45 to get a complete axiomatization. For details, see [23].

A study of how various classes of models affect the properties of bisimulation quantified logics is given in [18]. Refinement epistemic logics are investigated in [23, 22]. In [22] a multi-agent KD45 axiomatization is also reported, a multi-agent S5 axiomatization is elusive so far.

6 Axiomatization RML/μ

In this section we give the axiomatization for refinement modal μ-calculus. We restrict ourselves to single-agent refinement modal μ-calculus. The axiomatization is an extension of the axiomatization RML for refinement modal logic. We recall the definition of modal μ-calculus in the technical introductory Section [2]
Definition 35 (axiomatization RML\(\mu\)) The axiomatization RML\(\mu\) is a substitution schema of the axioms and rules of RML (see Section 5), along with the axiom and rule for the modal \(\mu\)-calculus:

\[
F_1 \quad \varphi[\mu x.\varphi \backslash x] \to \mu x.\varphi \\
F_2 \quad \text{From } \varphi[\psi \backslash x] \to \psi \text{ infer } \mu x.\varphi \to \psi
\]

and two new interaction axioms:

\[
R^\mu \quad \forall x.\varphi \leftrightarrow \mu x.\forall \varphi \text{ where } \varphi \text{ is a df} \\
R^\nu \quad \forall x.\varphi \leftrightarrow \nu x.\forall \varphi \text{ where } \varphi \text{ is a df}
\]

We emphasize that the interaction axioms have the important associated condition that the refinement quantification will only commute with a fixed-point operator if the fixed-point formula is a disjunctive formula.

6.1 Soundness

The soundness proofs of Section 5.1 still apply and the soundness of \(F_1\) and \(F_2\) are well known [6], so we are left to show that \(R^\mu\) and \(R^\nu\) are sound. In the proof we use the characterization of refinement quantification in terms of bisimulation quantification and relativization that was established in Proposition 22, and we use the characterization of both fixed points in terms of bisimulation quantification as reported in Section 2. In order to make the construction work, we need to expand the translation \(t: L_\forall \to L_\forall\) (Definition 20) to a translation \(t: L_\forall^\mu \to L_\forall\) by adding the clauses for fixed points from Section 2: \(t(\nu x.\varphi) \iff \exists p (p \land \Box(p \to t(\varphi[p \backslash x])))\) and \(t(\mu x.\varphi) \iff \forall p (\Box(t(\varphi[p \backslash x]) \to p) \to p)\); and we need to expand definition \(\bullet^p: L_\forall \to L_\forall\) of relativization (Definition 17) to include a clause for the universal modality: \((\Box \varphi)^p \leftrightarrow \Box \varphi^p\).

Theorem 36 The axioms \(R^\mu\) and \(R^\nu\) are sound.

Proof The proof consists of two cases, \(R^\mu\) and \(R^\nu\).

Case \(R^\mu\)

It is more convenient in this proof to reason about the axiom in its contrapositive form: \(\exists \nu x.\varphi \leftrightarrow \nu x.\exists \varphi\). The proof demonstrates that \(t(\exists \nu x.\varphi)\) is equivalent to \(t(\nu x.\exists \varphi)\) in bisimulation quantified logic (with the universal modality). Using the translation and
relativization equivalences above we have that, for any \( \varphi \in \mathcal{L}_\psi \):

\[
t(\exists \forall x. \varphi) \iff \exists p \, t(\nu x. \varphi)^p
\]

\[
\iff \exists p (\exists q (q \land \Box (q \rightarrow t(\varphi)))^p
\]

\[
\iff \exists p \exists q (q \land (\Box (q \rightarrow t(\varphi)))^p
\]

\[
\iff \exists q (q \land \exists p (q \rightarrow t(\varphi))^p
\]

\[
\iff \exists q (q \land \Box (q \rightarrow t(\varphi))^p
\]

\[
\iff \exists q (q \land \exists p (q \rightarrow t(\varphi))^p)
\]

\[
\iff \exists q (q \land \Box (q \rightarrow t(\exists \varphi)))
\]

\[
\iff t(\nu x. \exists \varphi)
\]

This proof simply applies known validities of bisimulation quantifiers. Note that line (\(\ast\)) is not an equivalence. The other direction holds if \(\varphi\) is a df. This we now prove:

Let \(\varphi\) be a df, then \(\models \exists q (q \land \exists p (q \rightarrow t(\varphi))^p) \rightarrow \exists q (q \land \exists p (q \rightarrow t(\varphi))^p)\). (4)

Suppose \(M_s\) is any countable model such that \(M_s \models \exists q (q \land \exists p (q \rightarrow t(\varphi))^p)\), where \(\varphi\) is a df. By definition of the bisimulation quantifiers, there exists some model \(N_i = (S^N, R^N, V^N)\) such that \(N_i \approx^q M_s\) and \(N_i \models q \land \exists p (q \rightarrow t(\varphi))^p\). Moreover, as the bisimulation quantified modal logic enjoys the tree-model property, we may assume without loss of generality that \(N_i\) is some tree-like model.

We inductively build a series of models \(N^i = (S^N, R^N, V_i)\) such that \(N_i \approx^p q \approx^q N_i\) and \(V_i\) may differ from \(V^N\) only for the variables \(p\) and \(q\), that is \(V_i(p) = V^N(p)\) for all \(r \notin \{p, q\}\). Moreover, the series of \(V_i\) is such that the sets \(V_i(q)\) and \(V_i(p)\) strictly increase. Its limit yields a valuation \(V_\omega\) such that the model \(N_\omega = (S^N, R^N, V_\omega)\) satisfies \(q \land \Box (q \rightarrow t(\varphi))^p\) at state \(t\), and \(N_i \approx^p q \approx^q N_i\). As a consequence \(M_s \models \exists q (q \land \exists p (q \rightarrow t(\varphi))^p)\).

We now define the series \(\{N^i\}\). We set \(V_0(q) = \{t\}, V_0(p) = \emptyset\). As \(N_i \models \exists p t(\varphi)^p\) and \(\varphi\) is a df, the only case where the valuation of atom \(q\) may influence the interpretation of \(\exists p t(\varphi)^p\) is at a set of states such that all states beyond that set of states are irrelevant to the interpretation of \(\exists p t(\varphi)^p\) at \(t\) (this set of states forms a frontier). This is because in a disjunctive form, if \(q\) is a sub-formula of \(\varphi\), then if \(q\) appears in the scope of a conjunction, it appears within the scope of a modality within that conjunction. Thus, there is a (possibly infinite) set of states \(\{u_0, u_1, \ldots\} \in V^N(q)\) such that the model \(N' = (S^N, R', V')\) with \(V'(q) = \{t, u_0, u_1, \ldots\}, V'(p) = V^N(p)\) for \(p \notin \{q, p\}\) and \(R' = R^N\backslash \{(u_i, s)| s \in S^N, i = 0, 1, \ldots\}\), is such that \(N_i \models t(\varphi)^p\). Consequently the valuation of \(p\) may be restricted to states that are not reachable from any state, \(\{u_0, u_1, \ldots\}\). Let \(S_0 < S^N\) be the set of states reachable from \(t\), but not reachable from \(u_i\) for any \(i\). We define \(N^1\) by setting \(V_1(q) = V'(q), V_1(p) = V'(p) \cap S_0\) and \(V_1(p) = V^N(p)\) for \(p \notin \{q, p\}\). As \(u_0, u_1, \ldots \in V^N(q)\) and \(N_i \models \Box (\exists p (q \rightarrow t(\varphi))^p)\), we have \(N_{u_i} \models q \land \Box (\exists p (q \rightarrow t(\varphi))^p)\) for all \(i\).

As \(M_s\) is a countable model, we may assume an enumeration of the worlds (or states) in that model. The induction proceeds by taking the first state \(u_0\) on the frontier and
repeating the process (i.e., finding a valuation $V'$ such that $V'$ makes $q$ true on a frontier $\{v_0, v_1, \ldots\}$, agrees with $V^N$ on the interpretation of all atoms except $q$ and $p$, makes $N'_u \models t(\varphi)^p$ and makes $N_{v_i} \models q \land \Box(\exists p(q \rightarrow t(\varphi)^p))$ for all $i$). We define $V_2$ by taking the union of $V_2(q) = V'(q)$ and $V_2(p) = V_1(p) \cup (V'(p) \cap S_1)$ where $S_1$ is the set of states reachable from $u_0$, but not from $v_i$ for and $i$, and all other atoms have their valuations unchanged. The states $\{v_0, v_1, \ldots\}$ are added to the set of frontier states and the induction continues. The construction is represented in Figure 7.

Figure 7: The inductive step for the construction of $N^\infty$. The formula $t(\varphi)^p$ is independent of any state where $p$ is not true, or any state beyond the frontier defined by $u_0, u_1, \ldots$.

Case $R^\nu$

We also use the contrapositive form of the axiom: $\exists x.\varphi \leftrightarrow \mu x.\exists x.\varphi$. For any $\varphi \in L_\psi$ we have that:

$$
\begin{align*}
t(\exists x.\varphi) & \iff \exists p \ t(\exists x.\varphi)^p \\
& \iff \exists p(\forall q(\Box(t(\varphi) \rightarrow q) \rightarrow q))^p \\
& \iff \exists p\forall q(\Box(t(\varphi)^p \rightarrow q) \rightarrow q) \\
& \iff \forall q\exists p(\Box(t(\varphi)^p \rightarrow q) \rightarrow q) \tag{**} \\
& \iff \forall q\exists p(\Box(t(\varphi)^p \land \neg q) \lor q) \\
& \iff \forall q(\exists p(\Box(t(\varphi)^p \land \neg q) \lor q) \tag{***}) \\
& \iff \forall q(\Box(\exists p t(\varphi)^p \land \neg q) \lor q) \\
& \iff \forall q(\Box(\exists p t(\varphi)^p \rightarrow q) \rightarrow q) \\
& \iff \forall q(\Box(\exists x.\exists x.\varphi \rightarrow q) \rightarrow q) \\
& \iff t(\exists x.\exists x.\varphi)
\end{align*}
$$

The equivalence in (***)) is true because $\Box$ is the existential modality which quantifies over all state in the model. Obviously, the implication in line (**) is only true in one direction
(the usual quantifier swap $\exists \forall \rightarrow \forall \exists$). To prove the other direction in the equivalence $\exists \mu.\varphi \leftrightarrow \mu.\exists \varphi$, we now show directly that $\models \mu.\exists \varphi \rightarrow \exists \mu.\varphi$ in refinement $\mu$-calculus, for $\varphi$ a df (observe that $\mu.\varphi$ is then a df as well).

We use the inductive characterization of $\mu.\exists \varphi$ of \cite{10} which tells that $M_s \models \mu.\exists \varphi$ if and only if $s \in \| \exists \varphi \|_{\tau}$ for some ordinal $\tau$, where we recall the definition of the semantic operation $\| \bullet \|$: $\| \exists \varphi \|_0 = \emptyset$, and $s \in \| \exists \varphi \|_{\tau}$ whenever $M_s^\tau \models \exists \varphi$, where $M^\tau = M[x^\tau]$ with $\sigma = x \mapsto \bigcup_{\tau' < \tau} \| \exists \varphi \|_{\tau'}$.

Suppose $M_s \models \mu.\exists \varphi$. Since $L_\varphi^\mu$ is bisimulation invariant, without loss of generality we may suppose that $M$ is a countable tree-like model. As $M_s$ satisfies $\mu.\exists \varphi$, there must be some least ordinal $\tau$ whereby $s \in \| \exists \varphi \|_{\tau}$. We give a proof by induction over $\tau$ that $s \in \| \exists \varphi \|_{\tau}$ implies $M_s \models \exists \mu.\varphi$. The base case where $\tau = 0$ is trivial. Now consider $M^\tau = M^\sigma$ with $\sigma = x \mapsto \bigcup_{\tau' < \tau} \| \exists \varphi \|_{\tau'}$, then $M_s^\tau \models \exists \varphi$. As $\mu.\varphi$ is a df, we are again in the case where there is a refinement of $M^\tau$ with a frontier such that $x$ may only be true at $s$ or on this frontier, and no point beyond the frontier affects the interpretation of $\varphi$. Formally, there is a set of states $\{u_0, u_1, \ldots \} \in V^\tau(x)$ such that $M_s^\tau \models \exists \varphi$ (i.e., $M_s^\tau \models \exists \mu x.\varphi$), where $M^\tau = (S', R', V')$ with

- $S' \subseteq S^\tau$ is the set of states reachable from $s$, but not from any $u_i$;
- $V'(x) = \{t, u_0, u_1, \ldots \}$, $V'(y) = V^{M^\tau}(y)$ for $y \neq x$; and
- $R' = R^\tau \setminus \{(u_i, t) \mid t \in S^\tau, i = 0, 1, \ldots \}$.

We note that $M_s^\tau$ is a refinement of $M^\tau_s$. Now as for each $i$, $u_i \in \| \exists \varphi \|_j$ for some $j < \tau$, by the inductive hypothesis we may assume there is some model $N^i = (S^i, R^i, V^i)$ where $N^i_{u_i} \models M^\tau_s \models \mu.\varphi$. We may append these models to $M'$, to define $M* = (S*, R*, V*)$ where $S* = S' \cup \bigcup_i S^i$, $R* = R' \cup \bigcup_j R^i \cup \{(t, v_i) \mid (t, u_i) \in R'\}$, and $V*(y) = V'(y) \cup \bigcup_j V^i(y)$ for all $y \in P$. (Notice the similar construction in the soundness proof of axiom RK.) It is clear that $M^*_s$ is a refinement of $M_s$, and by the axiom F1 we can see $M^*_s \models \mu.\varphi$ as required. \hfill $\Box$

We note that the general form of $R^\mu$ is not sound. For example, take $\varphi = \mu z.\diamond(p \rightarrow q) \rightarrow \diamond(\neg p \rightarrow x)$. Then $\forall \mu x.\varphi$ is true if $p$ is true at every immediate successor of the current state, whereas $\mu x.\forall \varphi$ is only true at states with no successor. Likewise $R^\nu$ is not true in the general case, as can be seen by taking $\varphi = p \land \Box(\diamond \top \rightarrow x)$. Then $\nu x.\forall \varphi$ is true if and only if $p$ is true at every reachable state, and $\forall \nu x.\varphi$ is true only if $p$ is true at every state within one step.

6.2 Completeness

The completeness proof of $RML^\mu$ proceeds exactly as for Theorem \cite{34}, replacing the formulas in cover logic with disjunctive formulas, to get a statement similar to that of Proposition \cite{32}

**Proposition 37** Every formula of $L_\nu^\mu$ is equivalent to a formula of $L^\mu$. \hfill $\Box$

39
Proof Given a formula $\psi$, we prove by induction on the number of the occurrences of $\exists$ in $\psi$ that it is equivalent to an $\exists$-free formula, and therefore to a formula in the modal $\mu$-calculus $L^\mu$. The base is trivial. Now assume $\psi$ contains $n + 1$ $\exists$-operators. Choose a subformula of type $\exists \varphi$ of our given formula $\psi$, where $\varphi$ is $\exists$-free (i.e. choose an innermost $\exists$). As $\varphi$ is $\exists$-free, it is semantically equivalent to a formula in disjunctive normal form, and by the completeness of Kozen’s axiom system [49] this equivalence is provable in $RML^\mu$. By NecR and R it follows that $\exists \varphi$ is provably equivalent to some formula $\exists \psi$ where $\psi$ is a disjunctive formula. Thus without loss of generalization, we may assume in the following that $\varphi$ is in disjunctive normal form. We may now proceed by induction over the complexity of $\varphi$, and conclude that $\exists \varphi$ is logically equivalent to a formula $\chi$ without $\exists$. All cases of this induction are as before, we only show the final two, different cases:

- $\exists \mu x. \varphi$ iff $\mu x. \exists \varphi$ (by $R^\nu$ noting that all subformulas of a disjunctive formula are themselves disjunctive); IH.
- $\exists \nu x. \varphi$ iff $\nu x. \exists \varphi$ (by $R^\mu$); IH.

Replacing $\exists \varphi$ by $\chi$ in $\psi$ gives a result with one less $\exists$-operator, to which the (original) induction hypothesis applies. □

Theorem 38 The axiom schema $RML^\mu$ is sound and complete for the logic $RML^\mu$ ⊣

Proof Soundness follows from Theorem 36 and Theorem 25. To see $RML^\mu$ is complete, suppose $\varphi \in L^\mu$ is a valid formula. Then by Lemma 37 $\varphi$ is provably equivalent to some valid formula $\psi \in L^\mu$. As $\psi$ is valid, it must be provable since Prop, K, F1, F2, NecK, and MP give a sound and complete proof system for the modal $\mu$-calculus [49]. A proof of $\varphi$ follows by MP. □

7 Complexity

Decidability for both $L^\nu$ and $L^\mu$ follows from the fact that a computable translation is given in the completeness proofs of Sections 5 and 6; note that the given translations, to $L$ and $L^\mu$ respectively, are recursive and involve transforming formulas into their disjunctive normal forms. Hence they are non-elementary in the size of of the original formula. This non-elementary procedure for $L^\mu$ is optimal as shown in Section 7.1 below.

Unfortunately we were not able to corroborate the lower complexity claims for $RML$ reported in [46]. But towards some indication of a result in that direction, we further establish a doubly exponential succinctness proof for $L^\nu$ in Section 7.2.

7.1 $RML^\mu$ is non-elementary

This section is dedicated to the proof of the following result.

Theorem 39 The satisfiability problem for $RML^\mu$ is non-elementary, even for the single-agent setting. ◀
In the rest of this section, we only consider a single-agent setting.

First, we recall a fragment, written $\text{CTL}^-$, of the standard branching-time logic Computation Tree Logic (CTL) [12], which in turn is a fragment of $\mathcal{L}^\mu$ (see also the example Section 4.2).

$$\text{CTL}^- \ni \varphi ::= T \mid \bot \mid p \mid \neg \varphi \mid \varphi \land \varphi \mid \Box \varphi \mid \Diamond \varphi \mid \text{EF} \varphi \mid \text{AF} \varphi$$

Let $M$ be a model and $s$ be a $M$-state. A path from $s$ is a finite or infinite sequence of states $s = s_0, s_1, \ldots$ s.t. $s_0 = s$ and each $s_{i+1}$ is an successor of $s_i$. Only the semantics of $\text{AF}$ and $\text{EF}$ is recalled (as for other formulas it is clear):

$$M_s \models \text{EF} \varphi \text{ iff there is a maximal path } \pi = s_0, s_1, \ldots \text{ from } s \text{ and } i \geq 0 \text{ such that } M_{s_i} \models \varphi$$
$$M_s \models \text{AF} \varphi \text{ iff for each maximal path } \pi = s_0, s_1, \ldots \text{ from } s, \text{ there is } i \geq 0 \text{ such that } M_{s_i} \models \varphi$$

Directly translating $\text{CTL}^-$ in $\mathcal{L}^\mu$ is routine via the following mapping $\tau : \text{CTL}^- \to \mathcal{L}^\mu$, defined by induction over the formulas:
$$\tau(T) = T, \tau(p) = p, \tau(\neg \varphi) = \neg \tau(\varphi), \tau(\varphi \land \varphi') = \tau(\varphi) \land \tau(\varphi'), \tau(\Box \varphi) = \Box \tau(\varphi), \tau(\Diamond \varphi) = \Diamond \tau(\varphi), \tau(\text{EF} \varphi) = \mu x. \tau(\varphi) \lor \Box x, \tau(\text{AF} \varphi) = \mu x. \tau(\varphi) \lor \Box x.$$

We also use standard abbreviations for the duals $\text{AG} \varphi$ iff $\neg \text{EF} \neg \varphi$ (‘universal always’), and $\text{EG} \varphi$ iff $\neg \text{AF} \neg \varphi$ (‘existential always’). A $\text{CTL}^-$ formula is in positive form if negation is applied only to propositional variables. A $\text{CTL}^-$ formula $\varphi$ is existential if it is in positive form and there are no occurrences of universal modalities (that is $\text{AF}$) and modalities $\Box$. The following can be proved by using Proposition [7] enriched for the case of $\text{EF}$ formulas (with a transfinite induction argument for this fixed-point formula).

**Proposition 40** Let $M_s$ and $N_t$ be two models with $M_s \succeq N_t$. Then for each existential $\text{CTL}^-$ formula $\varphi$, $N_t \models \varphi$ implies $M_s \models \varphi$.

**Definition 41 (Refinement $\text{CTL}^-$)** Refinement $\text{CTL}^- (\text{CTL}^\rightarrow, \text{for short})$ is the extension of $\text{CTL}^-$ with the refinement quantifiers $\exists$ and $\forall$.

**Definition 42 (Refinement Quantifier Alternation Depth)** We first define the alternation length $\ell(\chi)$ of finite sequence $\chi \in \{\exists, \forall\}^*$ of quantifiers, as the number of alternations of existential and universal refinement quantifiers in $\chi$. Formally, $\ell(\epsilon) = 0$, $\ell(Q) = 0$ for every $Q \in \{\exists, \forall\}^*$, and $\ell(QQ'\chi) = \ell(Q'\chi) + 1$ otherwise.

Given a $\mathcal{L}_\nu$ (resp., $\mathcal{L}_\mu^\nu$, resp., $\text{CTL}^-_\nu$) formula $\varphi$, the refinement quantifier alternation depth $\delta(\varphi)$ of $\varphi$ is defined via the standard tree-encoding $T(\varphi)$ of $\varphi$, where each node is labeled by either a modality, or a boolean connective, or a propositional variable. Then, $\delta(\varphi)$ is the maximum of the alternation lengths $\ell(\chi)$ where $\chi$ is the sequence of refinement quantifiers along a maximal path of $T(\varphi)$ from the root.

**Theorem 43** Let the class $C_k = \{\varphi \in \text{CTL}^-_\nu \mid \delta(\varphi) \leq k\}$. The satisfiability problem for $C_k$ is $k$-Expspace-hard, for any $k$.  

41
Theorem 43 is proved by a polynomial-time reduction from satisfiability of Quantified Propositional Temporal Logic (QPTL) [42]. First, we recall the syntax and the semantics of QPTL. The syntax of QPTL formulas $\varphi$ over a countable set $P$ of propositional variables is defined as follows:

$$\varphi ::= p | \neg \varphi | \varphi \land \varphi | \varphi \lor \varphi | X\varphi | F\varphi | \exists p . \varphi$$

where $p \in P$, $X$ is the ‘next’ modality, $F$ is the ‘eventually’ modality, and $\exists$ is the existential quantifier.\footnote{We distinguish (domain) quantifiers $\exists$ and $\forall$ in use here, from the refinement quantifiers $\exists$ and $\forall$, and from the bisimulation quantifiers $\exists$ and $\forall$.}

The semantics is given w.r.t. elements of $(2^P)^\omega$, namely infinite words $w$ over $2^P$. Beforehand, we need some technical notions. Let $w \in (2^P)^\omega$. For each $i \geq 0$, $w(i)$ denotes the $i$th symbol of $w$. Moreover, for each $P' \subseteq P$, we define the equivalence relation $\equiv_{P'}$ over $(2^P)^\omega$: two infinite words $w_1$ and $w_2$ are $\equiv_{P'}$-equivalent whenever their projections onto $P'$ are equal. The projection of an infinite word $w$ onto $P'$, written $\text{proj}(w, P')$, is obtained by removing from each symbol of $w$ all the propositions in $P \setminus P'$. Hence, $w_1 \equiv_{P'} w_2$ iff $\text{proj}(w_1, P') = \text{proj}(w_2, P')$.

Given a QPTL formula $\varphi$, an infinite word $w$ over $2^P$, and a position $h \geq 0$ along $w$, the satisfaction relation $(w, h) \models \varphi$ is inductively defined as follows (we omit the clauses for the boolean connectives):

$$(w, h) \models p \iff p \in w(h)$$

$$(w, h) \models X\varphi \iff (w, h + 1) \models \varphi$$

$$(w, h) \models F\varphi \iff \text{there is } h' \geq h \text{ such that } (w, h') \models \varphi$$

$$(w, h) \models \exists p . \varphi \iff \text{there is } w', w' \equiv_{P' \setminus \{p\}} \text{ w' and } (w', h) \models \varphi$$

We say that the word $w$ satisfies $\varphi$, written $w \models \varphi$, if $(w, 0) \models \varphi$. A QPTL formula $\varphi$ is in positive normal form if it is of the form $Q_1 p_1. Q_2 p_2. \ldots Q_n p_n. \varphi_{n+1}$, where $Q_j \in \{\exists, \forall\}$ for each $1 \leq j \leq n$, and $\varphi_{n+1}$ is a quantification-free QPTL-formula in which negation is applied only to propositional variables.\footnote{Every QPTL formula is constructively equivalent to a formula in positive normal form, with linear size.}

The quantifier alternation depth of $Q_1 p_1. Q_2 p_2. \ldots Q_n p_n. \varphi_{n+1}$ is the number of alternations of (existential and universal) quantifiers in the string $Q_1 Q_2 \ldots Q_n$. The following is a well-known result.

**Theorem 44** [42] Let $k \geq 0$. Then, the satisfiability problem for the class of QPTL formulas in positive normal form whose quantifier alternation depth is $k$ is $k$-ExpSpace-hard.

Note that Theorem 44 holds even if we assume that formulas in positive normal form like $Q_1 p_1. Q_2 p_2. \ldots Q_n p_n. \varphi_{n+1}$ (with $\varphi_{n+1}$ quantification-free) are such that $p_1, \ldots, p_n$ are pairwise distinct, each proposition occurring in $\varphi_{n+1}$ is in $\{p_1, \ldots, p_n\}$, and $Q_n = \forall$.

Theorem 43 directly follows from Theorem 44 and the following theorem, whose proof is given in the rest of this section.
**Theorem 45** For every $\varphi \in \text{QPTL}$, one can construct in time polynomial in the size of $\varphi$ a formula $\tilde{\varphi} \in \text{CTL}_s^*$, such that $\varphi$ is satisfiable if, and only if, $\tilde{\varphi}$ is satisfiable. Moreover, the refinement quantifier alternation depth of $\tilde{\varphi}$, $\delta(\tilde{\varphi})$, is equal to the quantifier alternation depth of $\varphi$.

Before proving Theorem 45, we need additional definitions. Let $P = \{p_1, \ldots, p_n\}$ and $\tilde{P} = P \cup \{p_0, \overline{p}_1, \ldots, \overline{p}_n\}$, where $p_0, \overline{p}_1, \ldots, \overline{p}_n$ are fresh propositional variables (intuitively, $\overline{p}_i$ is used to encode the negation of $p_i$ for each $1 \leq i \leq n$, and $p_0$ is a new variable that will be used to mark a path). For a model $M$ and two states $s$ and $s'$ in $M$, $s'$ is reachable from $s$ if there is a finite path from $s$ leading to $s'$. Let $0 \leq j \leq n$. A pointed model $M_s$ (over $\tilde{P}$) is well-formed w.r.t. $j$ if the following holds:

1. for each state $s'$ of $M$ which is reachable from $s$, there is exactly one proposition $p \in \tilde{P}$ such that $s' \in V^M(p)$ (we say that $s'$ is a $p$-state); moreover, $s$ is a $p_0$-state;
2. each state $s'$ reachable from $s$ which is not a $p_0$-state has no successor;
3. each $p_0$-state $s'$ which is reachable from $s$ satisfies: (i) $s'$ has some $p_0$-successor, (ii) for all $1 \leq i \leq j$, $s'$ has either some $p_i$-successor or some $\overline{p}_i$-successor, where the ‘or’ is exclusive due to 1., and (iii) for all $j + 1 \leq i \leq n$, $s'$ has both a $p_i$-successor and a $\overline{p}_i$-successor.

For each $0 \leq j \leq n$, the following $\text{CTL}^-$ formula $\psi_j$ over $\tilde{P}$ characterizes the set of pointed models which are well-formed w.r.t. $j$:

$$\psi_j := p_0 \land \text{AG}\left\{\left[\bigvee_{p \in \tilde{P}} (p \land \bigwedge_{p' \in \tilde{P} \backslash \{p\}} \neg p')\right] \land \left[\neg p_0 \to \Box \bot\right] \land \right.$$  
$$p_0 \to \left[\Box p_0 \land \bigwedge_{1 \leq i \leq n} (\Diamond p_i \land \Diamond \overline{p}_i) \land \bigwedge_{1 \leq i \leq j} (\Diamond (p_i \lor \overline{p}_i) \land (\Box \neg p_i \lor \Box \neg \overline{p}_i))\right]\right\}$$

In particular, it can be shown that $\psi_0$ enforces the existence of an infinite path labeled with $p_0$ and propositions of $P$ all along.

A pointed model $M_s$ is well-formed if it is well-formed w.r.t. $j$ for some $0 \leq j \leq n$. In this case, we say that $M_s$ is minimal if, additionally, each $p_0$-state which is reachable from $s$ has exactly one $p_0$-successor.

A well-formed pointed model $M_s$ encodes a set of infinite words over $2^P$, written $\text{words}(M_s)$, given by: $w \in \text{words}(M_s)$ iff there is an infinite path $\pi = s_0, s_1, \ldots$ of $M$ from $s$ (note that $\pi$ consists of $p_0$-states) such that for all $h \geq 0$ and $1 \leq j \leq n$, either $p_j \in w(h)$ and $s_h$ has some $p_j$-successor, or $p_j \notin w(h)$ and $s_h$ has some $\overline{p}_j$-successor.

Note that if $M_s$ is well-formed w.r.t. 0, then $\text{words}(M_s) = (2^P)^\omega$. If instead $M_s$ is well-formed w.r.t. $j$ for some $0 < j \leq n$ and $M_s$ is also minimal, then there is an infinite word $u_j \in (2^{\langle p_0,\ldots,p_j \rangle})^\omega$ such that $\text{words}(M_s) = \{w \in (2^P)^\omega \mid \text{proj}(w, \{p_1,\ldots,p_j\}) = u_j\}$. In particular, when $j = n$, $\text{words}(M_s)$ is a singleton.

Also, one can easily see that if $M_s \succeq N_t$ then $\text{words}(M_s) \supseteq \text{words}(N_t)$.  

43
Construction of the $\text{CTL}_\varphi$ formula $\bar{\varphi}$ (in Theorem 45). Pick an QPTL formula $\varphi = Q_1 p_1 . Q_2 p_2 . \cdots . Q_n p_n . \tilde{\varphi}_{n+1}$. For each $1 \leq j \leq n$, we let $\varphi_j = Q_j p_j . Q_{j+1} p_{j+1} . \cdots . Q_n p_n . \tilde{\varphi}_{n+1}$ (note that $\varphi_1$ corresponds to $\varphi$).

First, we construct a $\text{CTL}_\varphi$ formula $\bar{\varphi}_j$ over $\bar{P}$ by using the $\text{CTL}^-$ formulas $\psi_j$ for each $1 \leq j \leq n + 1$. The construction is based on an induction on $n + 1 - j = 0, \ldots, n$ as follows:

**Base case** ($j = n + 1$). Recall that $\tilde{\varphi}_{n+1}$ is a quantification-free QPTL formula in positive normal form over $P$. Let $\Upsilon$ be the following mapping from the set of quantification-free QPTL formulas $\xi$ over $P$ in positive normal form to the set of existential $\text{CTL}^-$ formulas over $\bar{P}$ (it is defined by induction).

- $\Upsilon(p) = \bigcirc p$ and $\Upsilon(\neg p) = \bigcirc \neg p$ for each $p \in P$;
- $\Upsilon (\xi_1 \lor \xi_2) = \Upsilon (\xi_1) \lor \Upsilon(\xi_2)$ and $\Upsilon (\xi_1 \land \xi_2) = \Upsilon (\xi_1) \land \Upsilon(\xi_2)$;
- $\Upsilon (X \xi) = \bigcirc (p_0 \land \Upsilon(\xi))$, $\Upsilon (G \xi) = EF(p_0 \land \Upsilon(\xi))$, and $\Upsilon (F \xi) = EG(p_0 \land \Upsilon(\xi))$. 

Then, $\bar{\varphi}_{n+1} := \Upsilon(\tilde{\varphi}_{n+1})$.

**Induction case** ($1 \leq j \leq n$). Recall $\varphi_j = Q_j p_j . \tilde{\varphi}_{j+1}$.

Then, $\bar{\varphi}_j := \begin{cases} \exists (\psi_j \land \bar{\varphi}_{j+1}) & \text{if } Q_j = \exists \\
 \forall (\psi_j \rightarrow \bar{\varphi}_{j+1}) & \text{if } Q_j = \forall \end{cases}$

Finally, the $\text{CTL}_\varphi$ formula $\bar{\varphi}$ over $\bar{P}$ is given by $\bar{\varphi} := \psi_0 \land \bar{\varphi}_1$.

**Correctness of the construction.** Note that the size of $\bar{\varphi}$ is polynomial in the size of $\varphi$. Moreover, the refinement quantifier alternation depth of $\bar{\varphi}$ is equal to the quantifier alternation depth of $\varphi$. Thus, in order to prove Theorem 45 it remains to show that $\varphi$ is satisfiable iff $\bar{\varphi}$ is satisfiable. For this, we need three preliminary lemmata.

**Lemma 46** Let $M_s$ be a pointed model which is well-formed w.r.t. $n$ and minimal, with $\text{words}(M_s) = \{w\}$. Then, for each quantification-free QPTL formula $\xi$ in positive normal form, $w \models \xi$ if and only if $M_s \models \Upsilon(\xi)$. \hfill $\dashv$

**Proof** Let $\pi = s_0, s_1, \ldots$ be the unique infinite path of $M$ from state $s$ (note that $\pi$ consists of $p_0$-states). Then, by a straightforward structural induction, one can show that for each quantification-free QPTL formula in positive normal form $\xi$, the following holds: for all $h \geq 0$, $M_{s_h} \models \Upsilon(\xi)$ iff $(w, h) \models \xi$. Hence, the result follows. \hfill $\square$

Let $0 \leq j \leq n$ and let $M_s$ be a pointed model which is well-formed w.r.t. $j$. For each $j \leq i \leq n$, an $h$-segment of $M_s$ is a refinement $N_i$ of $M_s$ which is well-formed w.r.t. $h$ and minimal. Note that for each $w \in \text{words}(M_s)$ and $j \leq h \leq n$, by construction, there exists an $h$-segment $N_i$ of $M_s$ such that $w \in \text{words}(N_i)$.

**Lemma 47** Let $1 \leq j \leq n$ and $M_s$ be a pointed model which is well-formed w.r.t. $j - 1$ such that for each $w \in \text{words}(M_s)$, $w \models \varphi_j$. Then, $M_s \models \bar{\varphi}_j$. \hfill $\dashv$
**Proof** The proof is by induction on \( n - j = 0, \ldots, n - 1 \).

**Base case** \((j = n)\). Recall \( \varphi_n = \bigvee_{p_n} \varphi_{n+1} \), where \( \varphi_{n+1} \) is a quantification-free QPTL formula in positive normal form. By construction, \( \overline{\varphi}_n = \forall_\psi \psi_n \rightarrow \Upsilon(\varphi_{n+1}) \). Let \( N_t \) be a refinement of \( M_s \) which satisfies formula \( \psi_n \) (if any). We need to show that \( N_t \models \Upsilon(\varphi_{n+1}) \). By definition of \( \psi_n \), \( N_t \) is well-formed w.r.t. \( n \). Let \( N'_u \) be any \( n \)-segment of \( N_t \), and let \( \text{words}(N'_u) = \{ w \} \). By transitivity, \( N'_u \) is a refinement of \( M_s \), so that \( w \in \text{words}(M_s) \). Thus, by hypothesis, \( w \models \varphi_n = \bigvee_{p_n} \varphi_{n+1} \), which implies \( w \models \varphi_{n+1} \). By Lemma \[\text{Lemma 48}\] it follows that \( N'_u \models \Upsilon(\varphi_{n+1}) \). Since \( N'_u \) is a refinement of \( N_t \) and \( \Upsilon(\varphi_{n+1}) \) is an existential CTL\(^{-} \) formula, by Proposition \[\text{Proposition 40}\] we deduce that \( N_t \models \Upsilon(\varphi_{n+1}) \) as well. Hence, the result holds.

**Induction step** \((1 \leq j \leq n - 1)\). By construction, there are two cases:

1. \( \varphi_j = \exists p_j \varphi_{j+1} \) and \( \overline{\varphi}_j = \exists \psi_j \wedge \overline{\varphi}_{j+1} \): let \( w_0 \in \text{words}(M_s) \). By hypothesis, \( w_0 \models \varphi_j \). Hence, there is infinite word \( \sigma \) such that \( w_0 = p_{j} \) and \( \overline{\varphi}_j \models \varphi_{j+1} \). Since \( M_s \) is well-formed w.r.t. \( j \) and \( w_0 \in \text{words}(M_s) \), it follows that \( \overline{\varphi}_j \models \varphi_{j+1} \). By definition of \( \psi_j \), \( N_s \models \psi_j \). Thus, it suffices to show that \( N_s \models \overline{\varphi}_{j+1} \). Since \( N_s \) is well-formed w.r.t. \( j \) and \( w_0 \in \text{words}(N_s) \), it holds that for each \( w' \in \text{words}(N_s) \), \( w' \equiv_{\{p_j, \ldots, p_n\}} w_0 \). Since every proposition in \( \{p_{j+1}, \ldots, p_n\} \) does not occur free in \( \varphi_{j+1} \) and \( \overline{\varphi}_j \models \varphi_{j+1} \), it follows that for each \( w' \in \text{words}(N_s) \), \( w' \models \varphi_{j+1} \). Thus, by the induction hypothesis, we obtain that \( N_s \models \overline{\varphi}_{j+1} \), and the result holds.

2. \( \varphi_j = \forall p_j \varphi_{j+1} \) and \( \overline{\varphi}_j = \forall \psi_j \rightarrow \overline{\varphi}_{j+1} \): let \( N_t \) be a refinement of \( M_s \) which satisfies formula \( \psi_j \) (if any). We need to show that \( N_t \models \overline{\varphi}_{j+1} \). By definition of \( \psi_j \), \( N_t \) is well-formed w.r.t. \( j \). Thus, by the induction hypothesis it suffices to show that for each \( w \in \text{words}(N_t) \), \( w \models \varphi_{j+1} \). Let \( w \in \text{words}(N_t) \). Since \( N_t \) is a refinement of \( M_s \), it holds that \( w \in \text{words}(M_s) \). Thus, by hypothesis, \( w \models \varphi_{j+1} \). Hence, \( w \models \varphi_{j+1} \), and the result follows.

\[\square\]

**Lemma 48** Let \( 1 \leq j \leq n \) and let \( M_s \) be a pointed model which is well-formed w.r.t. \((j - 1)\) and such that \( M_s \models \overline{\varphi}_j \). Then, there is a \((j - 1)\)-segment \( N_t \) of \( M_s \) such that \( N_t \models \varphi_j \) and for each \( w \in \text{words}(N_t) \), \( w \models \varphi_j \).

**Proof** The proof is by induction on \( n - j = 0, \ldots, n - 1 \). Recall \( \varphi_n = \bigvee_{p_n} \varphi_{n+1} \). Thus, by construction there are two cases:

1. \( \varphi_j = \bigvee_{p_j} \varphi_{j+1} \) and \( \overline{\varphi}_j = \forall \psi_j \rightarrow \overline{\varphi}_{j+1} \): let \( N_t \) be any \((j - 1)\)-segment of \( M_s \). By hypothesis \( M_s \models \overline{\varphi}_j \). Since every refinement of \( N_t \) is also a refinement of \( M_s \), it follows that \( N_t \models \overline{\varphi}_j \). Thus, it suffices to show that for each \( w \in \text{words}(N_t) \), \( w \models \varphi_{j+1} \). Fix \( w \in \text{words}(N_t) \) and let \( w' \) be an infinite word over \( 2^P \) such that \( w' \equiv_{\{p_j, \ldots, p_n\}} w \). Since \( N_t \) is well-formed w.r.t. \( j - 1 \), \( w' \in \text{words}(N_t) \) as well. Let \( N'_u \) be a \( j \)-segment of \( N_t \) such that \( w' \in \text{words}(N'_u) \). By definition of \( \psi_j \), \( N'_u \models \psi_j \). Thus, since \( N_t \models \overline{\varphi}_j \), we deduce that \( N'_u \models \overline{\varphi}_{j+1} \). There are two cases:
Thus, in both cases \( w' \models \varphi_{j+1} \). Since \( w' \) is an arbitrary infinite word over \( 2P \) such that \( w' \equiv_{P \setminus \{p_i\}} w \), we obtain that \( w \models \nabla p_j . \varphi_{j+1} = \varphi_j \), and the result follows.

(2) \( \varphi_j = \exists p_j . \varphi_{j+1} \), \( \bar{\varphi}_j = \exists (\psi_j \land \bar{\varphi}_{j+1}) \), and \( j \leq n - 1 \) (induction step): since \( M_s \models \bar{\varphi}_j \), there is a refinement \( N_t \) of \( M_s \) satisfying both \( \psi_j \) and \( \bar{\varphi}_{j+1} \). By definition of \( \psi_j \), \( N_t \) is well-formed w.r.t. \( j \). Thus, since \( N_t \models \bar{\varphi}_{j+1} \) and \( j \leq n - 1 \), by the induction hypothesis, there is a \( j \)-segment \( N'_u \) of \( N_u \) such that \( N'_u \models \psi_j \), \( N'_u \models \bar{\varphi}_{j+1} \), and for each \( w \in \text{words}(N'_u) \), \( w \models \varphi_{j+1} \). Since \( N_t \) is a refinement of \( M_s \), it easily follows that \( N'_u \) is the refinement of some \( (j - 1) \)-segment \( M'_u \) of \( M_s \). Since \( N'_u \models \psi_j \land \bar{\varphi}_{j+1} \), it holds that \( M'_u \models \bar{\varphi}_j \). Hence, it suffices to show that for each \( w \in \text{words}(M'_u) \), \( w \models \varphi_j \). Let \( w \in \text{words}(M'_u) \). Then, since \( M'_u \) (resp., \( N'_u \)) is minimal and well-formed w.r.t. \( j - 1 \) (resp., \( j \)) and \( N'_u \) is a refinement of \( M'_u \), it follows that there is \( w' \in \text{words}(N'_u) \) such that \( w' \equiv_{P \setminus \{p_i\}} w \). Since \( w' \models \varphi_{j+1} \), we obtain that \( w \models \exists p_j . \varphi_{j+1} = \varphi_j \), and the result follows. \( \square \)

Now, we can prove the correctness of the construction.

**Theorem 49** \( \varphi \) is satisfiable if, and only if, \( \bar{\varphi} \) is satisfiable.

**Proof** First, assume that \( \bar{\varphi} = \psi_0 \land \bar{\varphi}_1 \) is satisfiable. Hence, there is a pointed model \( M_s \) which satisfies both \( \psi_0 \) and \( \bar{\varphi}_1 \). By definition of formula \( \psi_0 \), it follows that \( M_s \) is well-formed w.r.t. \( 0 \). Since \( M_s \models \bar{\varphi}_1 \), by Lemma 45 we deduce that there is an infinite word \( w \) over \( 2P \) such that \( w \models \varphi_1 \). Since \( \varphi = \varphi_1 \), it follows that \( \varphi \) is satisfiable.

Now, assume that \( \varphi \) is satisfiable. Since any proposition in \( P \) does not occur free in \( \varphi \), it follows that for each infinite word \( w \) over \( 2P \), \( w \models \varphi \). Let \( M_s \) be any pointed model which is well-formed w.r.t. \( 0 \). By definition of formula \( \psi_0 \), it holds that \( M_s \models \psi_0 \). Moreover, since \( w \models \varphi \) for each \( w \in \text{words}(M_s) \), and \( \varphi = \varphi_1 \), by Lemma 47 it follows that \( M_s \models \bar{\varphi}_1 \). Therefore, \( M_s \models \psi_0 \land \bar{\varphi}_1 = \bar{\varphi} \). Hence, \( \bar{\varphi} \) is satisfiable. \( \square \)

By using Theorems 44, Theorem 45, and the fact that there exists a linear time translation of \( \text{CTL}^- (\subseteq \text{CTL}_r^-) \) into \( L^\mu_r \) (see page 41), we obtain a proof of Theorem 39's statement, given at the beginning of this section.

### 7.2 Succinctness

In this section we establish the following result.

**Theorem 50** \( \text{RML} \) is doubly exponentially more succinct than \( K \), and \( \text{RML}^\mu \) is doubly exponentially more succinct than modal \( \mu \)-calculus. \( \dashv \)
Theorem \[\text{[54]}\] directly follows from the following result whose proof is given in the rest of this section.

**Proposition 51** There is a finite set $P$ of propositional variables and a family $(\varphi_n)_{n \in \mathbb{N}}$ of one-agent $\mathcal{L}_\forall$ formulas over $P$ such that for each $n \in \mathbb{N}$, $\varphi_n$ has size $O(n^2)$ and refining nesting depth 2, and each equivalent one-agent $\mathcal{L}^\forall$ formula has size at least $2^{2^{O(n)}}$.\[\]

**Construction of the $\mathcal{L}_\forall$ formulas $\varphi_n$ in Proposition 51** let $P = \{l, r, #, 0, 1, a, b\}$. A $n$-configuration is a string on $\{a, b\}$ of length exactly $2^n$. We define a class $\mathcal{C}_n$ of pointed models, where each pointed model in the class encodes in a suitable way a pair of $n$-configurations. Then, we construct the $\mathcal{L}_\forall$ formula $\varphi_n$ in such a way that the following holds: a pointed model $M_s \in \mathcal{C}_n$ satisfies $\varphi_n$ iff the two $n$-configurations encoded by $M_s$ coincide. In order to formally define the class $\mathcal{C}_n$, we need additional definitions. A $n$-block is a pair $bl = (c, i)$ such that $c \in \{a, b\}$ and $1 \leq i \leq 2^n$. We say that $c$ is the content of $bl$ and $i$ is the position of $bl$. Intuitively, $bl$ represent the $i$th bit in the binary encoding of some $n$-configuration. First, we define an encoding of $(c, i)$ by a set $\text{code}(c, i)$ of strings over $2^P$ of length $n + 3$. Since $1 \leq i \leq 2^n$, $i$ can be encoded by a binary string over $\{0, 1\}$ of length exactly $2^n$. Moreover, we keep track for each $1 \leq j \leq 2^n$, of the binary encoding (a string over $\{0, 1\}$ of length $n$), of the position $j$ of the $j$th bit in the binary encoding of $i$. This leads to the following definition. A $n$-sub-block is a string over $2^P$ of length $n + 2$ of the form $sbl = \{#\}, \{b_1\}, \ldots, \{b_n\}, \{B\}$, where $b_1, \ldots, b_n, B \in \{0, 1\}$. The content of $sbl$ is $B$ and the position of $sbl$ is the integer $1 \leq j \leq 2^n$ whose binary encoding is $b_1, \ldots, b_n$. Intuitively, $sbl$ encodes the position and the content $B$ of a bit along the binary encoding of an integer $1 \leq i \leq 2^n$. Then, $\text{code}(c, i)$ is the set of strings over $2^P$ of length $n + 3$ such that

- for each $u \in \text{code}(c, i)$, $u = sbl \cdot \{c\}$, where $sbl$ is a $n$-sub-block whose position $j$ and content $b$ satisfy the following: $b$ is the $j$th bit in the binary encoding of $i$.

- for each $1 \leq j \leq 2^n$, let $B_j$ be the $j$th bit in the binary encoding of $i$ and $sbl_j$ be the $n$-sub-block whose position is $j$ and whose content is $B_j$. Then, $sbl_j \cdot \{c\} \in \text{code}(c, i)$.

Let $M_s$ be a pointed model over $P$. We denote by $\text{Traces}(M_s)$ the set of finite or infinite strings over $2^P$ of the form $(V^M)^{-1}(s_0), (V^M)^{-1}(s_1), \ldots$ such that $s_0, s_1, \ldots$ is a maximal path of $M$ starting from $s$. A pointed model $M_s$ encodes a $n$-block $(c, i)$ if

\[
\text{Traces}(M_s) = \text{code}(c, i) \quad \text{and} \quad M_s \models \bigwedge_{d=0}^{n-1} \Box^d (\Diamond 1 \land \Diamond 0) \in \mathcal{L}
\]

Note that the set of pointed models encoding $(c, i)$ is nonempty. Let $(w_l, w_r)$ be a pair of $n$-configurations. A pointed model $M_s$ encodes the pair $(w_l, w_r)$ if it holds that:

- $s$ has two successors $s_l$ and $s_r$ (called the left successor and right successor of $s$, respectively). Moreover, $(V^M)^{-1}(s) = \emptyset$, $(V^M)^{-1}(s_l) = \{l\}$ and $(V^M)^{-1}(s_r) = \{r\}$;

\[\text{[\text{here, it is not relevant to specify the form of the binary encoding which is used}]}\]
for each \( \text{dir} \in \{l, r\} \), \( s_{\text{dir}} \) has \( 2^{2^n} \) successors \( s_{1, \text{dir}}, \ldots, s_{2^{2^n}, \text{dir}} \). Moreover, for each \( 1 \leq i \leq 2^{2^n} \), \( M_{s_{i, \text{dir}}} \) encodes the \( n \)-block \((c_{i, \text{dir}}, i)\), where \( c_{i, \text{dir}} \) is the \( i \)th symbol of the \( n \)-configuration \( w_{\text{dir}} \).

If additionally \( w_l = w_r \), then we say that \( M_s \) is \textit{well-formed}. The class \( \mathcal{C}_n \) is the class of pointed models \( M_s \) such that \( M_s \) encodes some pair \((w_l, w_r)\) of \( n \)-configurations. In order to define the \( \mathcal{L}_\mathcal{V} \) formula \( \psi_n \) (for each \( n \geq 0 \)), first, we show the following result. Intuitively, Lemma 52 asserts that there is a \( \mathcal{L}_\mathcal{V} \) formula \( \psi_n \) of size \( O(n^2) \) which allows to select for a given pointed model \( M_s \in \mathcal{C}_n \), only the \( n \)-blocks encoded by \( M_s \) having the same position.

\textbf{Lemma 52} For each \( n \geq 0 \), one can construct a one-agent \( \mathcal{L}_\mathcal{V} \) formula \( \psi_n \) of size \( O(n^2) \) and refinement nesting depth 1 satisfying the following for all pairs \((w_l, w_r)\) of \( n \)-configurations: for each \( M_s \in \mathcal{C}_n \) encoding the pair \((w_l, w_r)\) and each refinement \( M'_{s} \) of \( M_s \),

- \( M'_{s} \) satisfies \( \psi_n \) iff there is \( 1 \leq i \leq 2^{2^n} \) such that the set of \#-states (i.e. states whose label is \((\#)\)) \( s'_{\#} \) reachable from \( s' \) is nonempty and for each of such states \( s'_{\#} \), \( M'_{s} \) encodes a \( n \)-block whose position is \( i \) and whose content is either the \( i \)th symbol of \( w_l \) or the \( i \)th symbol of \( w_r \).

\textbf{Proof} The \( \mathcal{L}_\mathcal{V} \) formula \( \psi_n \) is defined as follows:

\[
\psi_n := \xi_n \land \forall (\theta_n \rightarrow \bigvee_{b \in \{0,1\}} \Box^{n+3} b)
\]

where \( \xi_n \) and \( \theta_n \) are \( \mathcal{L} \) formulas defined as follows:

\[
\xi_n := \Diamond \top \land \Box \Diamond \top \land \bigwedge_{d=0}^{n-1} \Box^{d+2} (\Diamond 1 \land \Diamond 0) \land \Box^{n+2} \top \land \Box^{n+3} \top
\]

\[
\theta_n := \Diamond \top \land \Box \Diamond \top \land \Box^{d+2} (b \land \Diamond \top) \land \Box^{n+3} \top
\]

Note that \( \psi_n \) has size \( O(n^2) \) and that \( \delta(\psi_n) = 1 \) (refinement alternation depth). Thus, it remains to prove the second part of the lemma. Fix \( M_s \in \mathcal{C}_n \) encoding some pair \((w_l, w_r)\) of \( n \)-configurations, and let \( M'_{s} \) be a refinement of \( M_s \). By construction, for each \#-state \( s'_{\#} \) reachable from \( s' \) in \( M' \), there is a \#-state \( s_{\#} \) reachable from \( s \) in \( M \) such that \( M'_{s_{\#}} \) is a refinement of \( M_{s_{\#}} \). Moreover, \( M_{s_{\#}} \) encodes some \( n \)-block \((c, i)\), where the content \( c \) is either the \( i \)th symbol of \( w_l \) or the \( i \)th symbol of \( w_r \). Thus, by definition of \( \xi_n \), we obtain the following.

\textbf{Fact 1:} \( M'_{s_{\#}} \) satisfies \( \xi_n \) iff the set of \#-states \( s'_{\#} \) reachable from \( s' \) is nonempty and for each of such states \( s'_{\#} \), \( M'_{s_{\#}} \) encodes some \( n \)-block \((c, i)\), where the content \( c \) is either the \( i \)th symbol of \( w_l \) or the \( i \)th symbol of \( w_r \).

In the second conjunct \( \forall (\theta_n \rightarrow \bigvee_{b \in \{0,1\}} \Box^{n+3} b) \) in definition of \( \psi_n \), the formula \( \theta_n \) intuitively enforces to select the refinements \( M'_{s_{\#}} \) of \( M_{s_{\#}} \) encoding only \( n \)-blocks having the same position. Formally, by definition of \( \eta_n \), we obtain the following.
Fact 2: Let $M'_{s'}$ be a refinement of $M'_{s}$. Then, $M'_{s'}$ satisfies $\theta_n$ iff for all $u, u' \in \text{Traces}(M'_{s'})$, $u, u' \in \text{Traces}(M_{s})$ and the $n$-sub-block in $u$ and the $n$-sub-block in $u'$ have the same position.

Thus, by Fact 2 it follows that the second conjunct $\forall(\theta_n \rightarrow \bigvee_{b \in \{0,1\}} \Box^{n+3}b)$ in definition of $\psi_n$ requires that all the $n$-sub-blocks in $\text{Traces}(M'_{s})$ having the same position have also the same content, i.e., all the $n$-blocks encoded by $M'_{s'}$ have the same position. Thus, by Fact 1 the result follows. □

For each $n \geq 0$, let $\psi_n$ be the $\mathcal{L}_\forall$ formula satisfying the statement of Lemma 52. Then, the one-agent $\mathcal{L}_\forall$ formula $\varphi_n$ is defined as follows:

$$\varphi_n = \forall(\psi_n \rightarrow \bigvee_{c \in \{a,b\}} \Box^{n+4}c)$$

By construction and Lemma 52, we easily obtain the following result.

Lemma 53 For each $n \geq 0$, the $\mathcal{L}_\forall$ formula $\varphi_n$ has size $O(n^2)$ and $\delta(\varphi_n) = 2$ (refinement alternation depth). Moreover, for each $M_s \in \mathcal{C}_n$, $M_s$ satisfies $\varphi_n$ iff $M_s$ is well-formed. □

Proof of Proposition 51: by Lemma 53, in order to complete the proof of Proposition 51, we need to show that for each $n \geq 0$, each one-agent $\mathcal{L}_\forall$ formula equivalent to $\varphi_n$ has size at least $2^{o(n)}$. For this, we use a well-known automata-characterization of (one-agent) $\mathcal{L}_\forall$ in terms of parity symmetric alternating (finite-state) automata (PSAA) which operate on pointed models 50. First, we recall the class of PSAA. We need additional definitions.

A tree $T$ is a prefix closed subset of $\mathbb{N}^*$. The elements of $T$ are called nodes and the empty word $\varepsilon$ is the root of $T$. For $x \in T$, the set of children of $x$ (in $T$) is $\{x \cdot i \in T \mid i \in \mathbb{N}\}$. A path of $T$ is a maximal sequence $\pi = x_0x_1 \ldots$ of $T$-nodes such that $x_0 = \varepsilon$ and for any $i$, $x_{i+1}$ is a child of $x_i$. For an alphabet $\Sigma$, a $\Sigma$-labeled tree is a pair $\langle T, r \rangle$ where $T$ is a tree and $r : T \rightarrow \Sigma$. For a set $X$, $\mathcal{B}_+(X)$ denotes the set of positive boolean formulas over $X$, built from elements in $X$ using $\lor$ and $\land$ (we also allow the formulas $\text{true}$ and $\text{false}$). A subset $Y$ of $X$ satisfies $\theta \in \mathcal{B}_+(X)$ iff the truth assignment that assigns $\text{true}$ to the elements in $Y$ and $\text{false}$ to the elements of $X \setminus Y$ satisfies $\theta$.

A PSAA over $P$ is a tuple $\mathcal{A} = \langle P, Q, q_0, \delta, Acc \rangle$, where $Q$ is a finite set of locations, $q_0 \in Q$ is an initial location, $\delta : Q \times 2^P \rightarrow \mathcal{B}_+((\Box, \Diamond) \times Q)$ is the transition function, and $Acc : Q \rightarrow \mathbb{N}$ is a parity acceptance condition assigning to each location $q \in Q$ an integer (called priority). Intuitively, a target of a move of $\mathcal{A}$ is encoded by an element in $\{\Box, \Diamond\} \times Q$. An atom $(\Diamond, q)$ means that a copy of $\mathcal{A}$ in location $q$ moves to some successor of the current state (of the pointed model in input), while an atom $(\Box, q)$ means that for each successor $s$ of the current state, a copy of $\mathcal{A}$ in location $q$ is sent to state $s$. Formally, for a pointed model $M_{s_0}$ over $P$, a run of $\mathcal{A}$ over $M_{s_0}$ is a $(Q \times S^M)$-labeled tree $\langle T, r \rangle$, where each node of $T$ labeled by $(q, s)$ describes a copy of $\mathcal{A}$ that is in location $q$ and reads the state $s$ of $M$. Moreover, we require that $r(\varepsilon) = (q_0, s_0)$ (initially, $\mathcal{A}$ is in the initial location $q_0$).
location \( q_0 \) reading state \( s_0 \), and for each \( y \in T \) with \( r(y) = (q, s) \), there is a (possibly empty) minimal set \( H \subseteq \{\Box, \Diamond\} \times Q \) satisfying \( \delta(q, (V^M)^{-1}(s)) \) such that the set \( L(y) \) of labels of children of \( y \) in \( T \) is the smallest set satisfying the following: for all atoms \( at \in H \),

- if \( at = (\Diamond, q') \), then for some successor \( s' \) of \( s \) in \( M, (q', s') \in L(y) \);
- if \( at = (\Box, q') \), then for each successor \( s' \) of \( s \) in \( M, (q', s') \in L(y) \).

For an infinite path \( \pi = y_0 y_1 \ldots \) of \( T \), let \( \inf(\pi) \) be the set of locations in \( Q \) that appear in \( r(y_0) r(y_1) \ldots \) infinitely often. The run \( \langle T, r \rangle \) is accepting if for each infinite path \( \pi \) of \( T \), the smallest priority of the locations in \( \inf(\pi) \) is even. The language of \( \mathcal{A} \) is the set of pointed models \( M_s \) over \( P \) such that \( \mathcal{A} \) has an accepting run over \( M_s \). The following is a well-known result.

**Proposition 54 [50]** Given a one-agent \( \mathcal{L}^u \) formula \( \varphi \) over \( P \), one can construct a PSAA \( \mathcal{A}_\varphi \) with \( O(|\varphi|) \) locations whose language is the set of pointed models over \( P \) satisfying \( \varphi \). \( \exists \)

Proposition [51] directly follows from Proposition [54] and the following result.

**Lemma 55** Let \( n \geq 0 \) and \( \mathcal{A}_n \) be a PSAA over \( P \) whose language is the set of pointed models satisfying the \( \mathcal{L}_\varphi \) formula \( \varphi_n \). Then, the number of locations of \( \mathcal{A}_n \) is at least \( 2^{2^n} \). \( \exists \)

**Proof** Let \( n \geq 0 \) and \( \mathcal{A}_n \) as in the statement of the lemma, and \( Q \) be the set of \( \mathcal{A}_n \)-locations. For each \( n \)-configuration \( w \), let \( M^w_s \) be some well-formed pointed model encoding the pair \( (w, w) \), and \( H(w) \) be the set of sets \( Q_l \subseteq Q \) such that there is an accepting run \( \langle T, r \rangle \) of \( \mathcal{A}_n \) over the pointed model \( M^w_s \) so that:

- \( Q_l \) is the set of locations associated with the copies of \( \mathcal{A}_n \) in the run \( \langle T, r \rangle \) which read the left successor \( s_l \) of \( s_w \) in \( M^w \), i.e., \( Q_l = \{ q \in Q \mid \text{for some } x \in T, r(x) = (q, s_l) \} \).

By hypothesis, \( H(w) \neq \emptyset \). Moreover,

**Claim:** for all \( n \)-configurations \( w \) and \( w' \) such that \( w \neq w' \), \( H(w) \cap H(w') = \emptyset \).

**Proof of the claim:** for a model \( M \) and a set \( S' \subseteq S^M \), the restriction of \( M \) to \( S' \) is defined in the obvious way. For \( s \in S^M \), \([M_s] \) denote the restriction of \( M \) to the set of states reachable from \( s \) in \( M \). For all \( n \)-configurations \( w \) and \( \text{dir} \in \{l, r\} \), let \( s_{w,\text{dir}} \) be the \text{dir}-successor of \( s_w \) in \( M^w \). We prove the claim by contradiction. So, assume that there are two distinct \( n \)-configurations \( w \) and \( w' \) such that \( H(w) \cap H(w') \neq \emptyset \). Without loss of generality we can assume that \( M^w \) and \( M^{w'} \) have no states in common. Let \( M^w_{s_w, w'} \) be any pointed model satisfying the following: the successors of \( s_w \) in \( M^w, w' \) are \( s_{w', l} \) and \( s_{w', r} \), and \( [M^w_{s_w, l}] = [M^w_{s_w, l}] \) and \( [M^w_{s_w, r}] = [M^w_{s_w, r}] \). Evidently, \( M^w_{s_w, w'} \) is a pointed model encoding the pair \( (w', w) \). Since \( w \neq w' \), by hypothesis and Lemma [53], \( \mathcal{A}_n \) does not accept \( M^w_{s_w, w'} \). On the other hand, since there is \( Q \in H(w) \cap H(w') \), by definition of the sets \( H(w) \) and \( H(w') \) and the semantics of PSAA, it easily follows that there is an accepting run of \( \mathcal{A}_n \) over \( M^w_{s_w, w'} \), which is a contradiction. Hence, the claim holds.
By the claim above, it follows that for each $n$-configuration $w$, there is $Q_w \in H(w)$ (recall that $H(w) \neq \emptyset$) such that for all $n$-configurations $w'$ distinct from $w$, $Q_w \notin H(w')$. Since the number of distinct $n$-configurations is $2^{2^n}$ and the number of subsets of $Q$ is $2^{|Q|}$, we obtain that $|Q| \geq 2^n$, and the result holds. $\square$

8 Conclusions and perspectives

We conclude that we hope to have established a platform for structural refinement in various modal logics. We established results on axiomatization, complexity, expressivity, and we gave applications to software verification and design, and to dynamic epistemic logics. We clearly established the relation to bisimulation quantified logics: refinement quantification is bisimulation followed by relativization. The multi-agent refinement modal logic and the furthest generalization in the form of refinement $\mu$-calculus are only the beginning. One could think of refinement CTL, refinement PDL, refinement epistemic logics, refinement with further structural restrictions or with protocol restrictions, and so on. Each of these logics may have different axiomatizations and complexities, and equal expressivity as the refinement-less version is certainly not to be expected; e.g., we estimate that refinement modal logic is more expressive than the base modal logic on the $KT$ model class.

A number of perspectives appear both nearby and on the further horizon. We wish to resolve the issue of the complexity of refinement modal logic (we only have resolved the issue for refinement $\mu$-calculus). Complexity of model checking in the logics has not been addressed. Given applications in the logics of knowledge and multi-agent system architecture, the axiomatization refinement epistemic logic, interpreted on the class of multi-$S5$ models, is a coveted price that so far escaped us.

On the further horizon loom the detailed investigation of other refinement logics, mainly refinement PDL and refinement CTL, and the exploration of their applications. The relation of refinement quantification and other form of propositional quantification over information change (quantifying over announcements, quantifying over action models) needs closer investigation, and results to be obtained in that area are unclear.

Acknowledgements

Acknowledgements will be added later.

References

[1] P. Aczel. Non-Well-Founded Sets. CSLI Publications, Stanford, CA, 1988. CSLI Lecture Notes 14.
[2] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. Lecture Notes in Computer Science, 1536:23–60, 1998.

[3] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. Journal of the ACM, 49:672–713, 2002.

[4] Rajeev Alur, Thomas A. Henzinger, Orna Kupferman, and Moshe Y. Vardi. Alternating refinement relations. In International Conference on Concurrency Theory, pages 163–178, 1998.

[5] Adam Antonik, Michael Huth, Kim G. Larsen, Ulrik Nyman, and Andrzej Wasowski. 20 years of modal and mixed specifications. Bulletin of European Association of Theoretical Computer Science, 1(94), 2008.

[6] A. Arnold and D. Niwinski. Rudiments of $\mu$-calculus. North Holland, 2001.

[7] P. Balbiani, A. Baltag, H. van Ditmarsch, A. Herzig, T. Hoshi, and T. De Lima. ‘Knowable’ as ‘known after an announcement’. Review of Symbolic Logic, 1(3):305–334, 2008.

[8] A. Baltag, L.S. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. In I. Gilboa, editor, Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 98), pages 43–56, 1998.

[9] M. Bilkova, A. Palmigiano, and Y. Venema. Proof systems for the coalgebraic cover modality. In Carlos Areces and Robert Goldblatt, editors, Advances in Modal Logic, pages 1–21. College Publications, 2008.

[10] P. Blackburn, M. de Rijke, and Y. Venema. Modal Logic. Cambridge University Press, Cambridge, 2001. Cambridge Tracts in Theoretical Computer Science 53.

[11] M. Browne, E. Clarke, and O. Grumberg. Characterizing Kripke structures in temporal logic. In H. Ehrig, R. Kowalski, G. Levi, and U. Montanari, editors, TAPSOFT ’87, pages 256–270. Springer, 1987. LNCS 249.

[12] E.M. Clarke and E.A. Emerson. Design and Verification of Synchronization Skeletons using Branching Time Temporal Logic. In Proceedings of Workshop on Logic of Programs, LNCS 131, pages 52–71. Springer-Verlag, 1981.

[13] G. d’Agostino and M. Hollenberg. Logical questions concerning the $\mu$-calculus: Interpolation, Lyndon and Los-Tarski. Journal of Symbolic Logic, 65(1):310–332, 2000.

[14] G. d’Agostino and G. Lenzi. An axiomatization of bisimulation quantifiers via the $\mu$-calculus. Theor. Comput. Sci., 338(1-3):64–95, 2005.
[15] G. d’Agostino and G. Lenzi. A note on bisimulation quantifiers and fixed points over transitive frames. *J. Log. Comput.*, 18(4):601–614, 2008.

[16] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge MA, 1995.

[17] Guillaume Feuillade and Sophie Pinchinat. Modal specifications for the control theory of discrete-event systems. *Discrete Event Dynamic Systems*, 17(2):181–205, 2007.

[18] T. French. *Bisimulation quantifiers for modal logic*. PhD thesis, University of Western Australia, 2006.

[19] T. French and H. van Ditmarsch. Undecidability for arbitrary public announcement logic. In C. Areces and R. Goldblatt, editors, *Advances in Modal Logic 7*, pages 23–42, London, 2008. College Publications. Proceedings of the seventh conference “Advances in Modal Logic”.

[20] J.D. Gerbrandy. *Bisimulations on Planet Kripke*. PhD thesis, University of Amsterdam, 1999. ILLC Dissertation Series DS-1999-01.

[21] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.

[22] J. Hales. Refinement quantifiers for logics of belief and knowledge. Honours Thesis, University of Western Australia, 2011.

[23] J. Hales, T. French, and R. Davies. Refinement quantified logics of knowledge. *Electr. Notes Theor. Comput. Sci.*, 278:85–98, 2011.

[24] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, Cambridge MA, 2000. Foundations of Computing Series.

[25] D. Janin and I. Walukiewicz. Results on the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. In *Concurrency Theory, 7th International Conference*, volume 1119 of *LNCS*, pages 263–277. Springer, 1996.

[26] B. Kooi. Expressivity and completeness for public update logics via reduction axioms. *Journal of Applied Non-Classical Logics*, 17(2):231–254, 2007.

[27] O. Kupferman, M. Vardi, and P. Wolper. Module checking. *Information and Computation*, 164(2):322–344, 2001.

[28] C. Kupke, A. Kurz, and Y. Venema. Completeness of the finitary moss logic. In C. Areces and R. Goldblatt, editors, *Advances in Modal Logic 7*, pages 193–217. College Publications, 2008.
[29] Kim G. Larsen, Ulrik Nyman, and Andrzej Wasowski. Modal I/O automata for interface and product line theories. In Proceedings of the 16th European Symposium on Programming (ESOP’07), volume 4421 of Lecture Notes in Computer Science, pages 64–79. Springer, 2007.

[30] A.R. Lomuscio and M.D. Ryan. An algorithmic approach to knowledge evolution. Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM), 13(2), 1998. Special issue on Temporal Logic in Engineering.

[31] J.S. Miller and L.S. Moss. The undecidability of iterated modal relativization. Studia Logica, 79(3):373–407, 2005.

[32] C. Morgan. Programming from Specifications: Second Edition. Prentice Hall International, Hempstead, UK, 1994.

[33] R. Parikh, L.S. Moss, and C. Steinsvold. Topology and epistemic logic. In M. Aiello, I. Pratt-Hartmann, and J. van Benthem, editors, Handbook of Spatial Logics, pages 299–341. Springer Verlag, 2007.

[34] M. Pauly. Logic for social software. PhD thesis, University of Amsterdam, 2001. ILLC Dissertation Series DS-2001-10.

[35] J.A. Plaza. Logics of public communications. In M.L. Emrich, M.S. Pfeifer, M. Hadzikadic, and Z.W. Ras, editors, Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems: Poster Session Program, pages 201–216. Oak Ridge National Laboratory, 1989.

[36] Jean-Baptiste Raclet. Quotient de spécifications pour la réutilisation de composants. PhD thesis, Université de Rennes I, December 2007. (In French).

[37] Jean-Baptiste Raclet. Residual for component specifications. In Proc. of the 4th International Workshop on Formal Aspects of Component Software (FACS’07), volume 215 of Electr. Notes Theor. Comput. Sci., pages 93–110, 2008.

[38] Jean-Baptiste Raclet, Eric Badouel, Albert Benveniste, Benoit Caillaud, and Roberto Passerone. Why are modalities good for interface theories? In Proceedings of the 9th International Conference on Application of Concurrency to System Design (ACSD’09), pages 199–217. IEEE Computer Society Press, 2009.

[39] P. Ramadge and W. Wonham. On the supervisory control of discrete event systems. In Proceedings of the IEEE, pages 81–98, 1989.

[40] Stéphane Riedweg and Sophie Pinchinat. Quantified mu-calculus for control synthesis. In Branislav Rovan and Peter Vojtáš, editors, MFCS, volume 2747 of Lecture Notes in Computer Science, pages 642–651. Springer, 2003.
[41] M. Ryan and P.-Y. Schobbens. Agents and roles: Refinement in alternating-time temporal logic. In Revised Papers from the 8th International Workshop on Intelligent Agents VIII (ATAL ’01), pages 100–114. Springer, 2002.

[42] A.P. Sistla, M.Y. Vardi, and P. Wolper. The Complementation Problem for Buchi Automata with Applications to Temporal Logic. Theoretical Computer Science, 49:217–237, 1987.

[43] J. Tsitsoklis. On the control of discrete event dynamical systems. Mathematics of Control Signals and Systems, 2(2):95–107, 1989.

[44] J. van Benthem. One is a lonely number: on the logic of communication. In Z. Chatzidakis, P. Koepke, and W. Pohlers, editors, Logic colloquium 2002. Lecture Notes in Logic, Vol. 27, pages 96–129. A.K. Peters, 2006.

[45] H. van Ditmarsch and T. French. Simulation and information. In J. Broersen and J.-J. Meyer, editors, Knowledge Representation for Agents and Multi-Agent Systems, pages 51–65. Springer, 2009. LNAI 5605. Presented at LOFT 2008 and KRAMAS 2008.

[46] H. van Ditmarsch, T. French, and S. Pinchinat. Future event logic - axioms and complexity. In L. Beklemishev, V. Goranko, and V. Shehtman, editors, Advances in Modal Logic, Moscow, volume 8, pages 77–99. College Publications, 2010.

[47] H. van Ditmarsch, W. van der Hoek, and B. Kooi. Dynamic Epistemic Logic, volume 337 of Synthese Library. Springer, 2007.

[48] Y. Venema. Lecture notes on the modal $\mu$-calculus. (Draft), 2012.

[49] I. Walukiewicz. Completeness of Kozen’s axiomatisation of the propositional mu-calculus. INFCTRL: Information and Computation (formerly Information and Control), 157, 2000.

[50] T. Wilke. CTL$^+$ is exponentially more succinct than CTL. In Proc. 19th FSTTCS, LNCS 1738, pages 110–121. Springer, 1999.