

Refinement Modal Logic

Laura Bozzelli*, Hans van Ditmarsch[†], Tim French[‡], James Hales[§], Sophie Pinchinat[¶]

March 27, 2014

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 \forall in addition to the standard modalities \square for each agent. The operator \forall acts as a quantifier over the set of all refinements of a given model. As a variation on a bisimulation quantifier, this refinement operator or *refinement quantifier* \forall can be seen as quantifying over a variable not occurring in the formula bound by it. 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
2	Technical preliminaries	5
3	Refinement	8
3.1	Refinement and its basic properties	8
3.2	Game and logical characterization of refinement	10
3.3	Refinement as bisimulation plus model restriction	13
3.4	Refinement and action models	15
3.5	Modal specifications refinement	16

*Informática, Universidad Politécnica de Madrid, Spain, laura.bozzelli@fi.upm.es

[†]LORIA, CNRS – Université de Lorraine, France, hans.van-ditmarsch@loria.fr

[‡]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

4	Refinement modal logic	16
4.1	Syntax and semantics of refinement modal logic	17
4.2	Examples	19
4.3	Refinement quantification is bisimulation quantification plus relativization	22
4.4	Alternating refinement relations	27
5	Axiomatization RML	27
5.1	Soundness	29
5.2	Example derivations	32
5.3	Completeness	33
5.4	The single-agent case	37
5.5	Refinement epistemic logic	37
6	Axiomatization RML^{μ}	38
6.1	Soundness	39
6.2	Completeness	45
7	Complexity	46
7.1	RML ^{μ} is non-elementary	46
7.2	Succinctness	52
8	Conclusions and perspectives	56

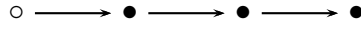
1 Introduction

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 [55] the *future event logic* is introduced. It augments the multi-agent logic of knowledge with an operation $\forall\varphi$ that stands for “ φ holds after all informative events” — the diamond version $\exists\varphi$ stands for “there is an informative event after which φ .” The proposal was a generalization of a so-called arbitrary public announcement logic with an operator for “ φ holds after all announcements” [8]. The semantics of informative events encompasses action model execution à la Baltag et al. [9]: 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 [56] 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 \exists 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 [55] 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 [42, 2] or in control theory [47, 51], it may correspond to a player discarding some moves; for program logics [29] it may correspond to operational refinement [40]; and for logics for spatial reasoning it may correspond to sub-space projections [41].

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



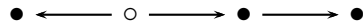
E.g., $\diamond\diamond\diamond\square\perp$ is true in the point. From the point of view of the modal language, this structure is essentially the same structure (it is bisimilar) as



This one also satisfies $\diamond\diamond\diamond\square\perp$ and any other modal formula for that matter. A more radical structural transformation would be to consider submodels, such as



A distinguishing formula between the two is $\diamond\diamond\square\perp$, 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



It is neither a submodel of the initial structure, nor is it bisimilar. It satisfies the formula $\diamond\diamond\square\perp \wedge \diamond\diamond\diamond\square\perp$ 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 [19, 37] employed a notion of refinement. In [37] 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 (for an overview, see [57]). This usage of refinement as a more general operation than model restriction is similar to ours.

In formal methods literature, see e.g. [62], *refinement of datatypes* is considered such that (datatype) C refines A if A simulates C . This usage of refinement as the converse of simulation [1, 11] comes close to ours — in fact, it inspired us to propose a similar notion, although the correspondence is otherwise not very close. A similar usage of refinement as in [62] is found in [3, 4]. In the theory of modal specifications a refinement preorder is used, known as *modal refinement* [45, 49]. Modal specifications are deterministic automata equipped with *may*-transitions and *must*-transitions. A must-transition is available in every component that implements the modal specification, while a may-transition need not be. This is close to our definition of refinement, as it also is some kind of submodel quantifier, but the two notions are incomparable, because ‘must’ is a subtype of ‘may’.

We incorporate implicit quantification over informative events directly into the language using, again, a notion of *refinement*; also in our case a refinement is the converse of simulation. Our work is closely related to some recent work on bisimulation quantified modal logics [17, 22]. The refinement operator, seen as refinement quantifier, is weaker than a bisimulation quantifier [55], 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 μ -calculus [16]. 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 μ -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 μ -calculus. Section 5 contains the axiomatization of that refinement modal logic and the completeness proof. We demonstrate that it is equally expressive as modal logic. We mention results for model classes $\mathcal{KD45}$ and $\mathcal{S5}$. Section 6 gives the axiomatization of refinement μ -calculus. Again, we have a reduction here, to standard μ -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 μ -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', \dots , and propositional variables are $p, q, r, p', p'', p_1, p_2, \dots$. 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 $R : A \rightarrow \mathcal{P}(S \times S)$, and a *valuation* $V : P \rightarrow \mathcal{P}(S)$. States are $s, t, u, v, s', \dots, s_1, \dots$. A pair consisting of a model M (with domain S) and a state $s \in S$ is called a *pointed model*, for which we write M_s . 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 $\mathcal{S5}$, and the model class with a serial, transitive and euclidean accessibility relation is denoted $\mathcal{KD45}$.

The *restriction* M' of a model M , notation $M' \subseteq M$, is a model $M' = (S', R', V')$ such that $S' \subseteq S$, for each $a \in A$, $R'_a = R_a \cap (S' \times S')$, and for each $p \in P$, $V'(p) = V(p) \cap S'$.

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

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \psi) \mid \Box_a\varphi$$

where $a \in A$ and $p \in P$. Without the construct $\Box_a\varphi$ we get the *language* \mathcal{L}_0 of *propositional logic*. Standard abbreviations are: $\varphi \vee \psi$ iff $\neg(\neg\varphi \wedge \neg\psi)$, $\varphi \rightarrow \psi$ iff $\neg\varphi \vee \psi$, \top iff $p \vee \neg p$, \perp iff $p \wedge \neg p$, and $\Diamond_a\varphi$ iff $\neg\Box_a\neg\varphi$. If there is a single agent only ($|A| = 1$), we may write $\Box\varphi$ instead of $\Box_a\varphi$. Formula variables are $\varphi, \psi, \chi, \varphi', \dots, \varphi_1, \dots$ and for sets of formulas we write Φ, Ψ, \dots . For a finite set Φ of \mathcal{L} formulas we let the *cover* operator $\nabla_a\Phi$ be an abbreviation for $\Box_a \bigvee_{\varphi \in \Phi} \varphi \wedge \bigwedge_{\varphi \in \Phi} \Diamond_a\varphi$; we note $\bigvee_{\varphi \in \emptyset} \varphi$ is always false, whilst $\bigwedge_{\varphi \in \emptyset} \varphi$ is always true.

Let a finite set of formulas $\Psi = \{\psi_1, \dots, \psi_n\}$ and a formula φ with possible occurrences of a propositional variable p be given. Let $\varphi[\psi \setminus p]$ denote the substitution of all occurrences of p in φ by ψ . Then $\varphi[\Psi \setminus p]$ abbreviates $\{\varphi[\psi_1 \setminus p], \dots, \varphi[\psi_n \setminus p]\}$, and similarly $\bigvee \varphi[\Psi \setminus p]$ stands for $\varphi[\psi_1 \setminus p] \vee \dots \vee \varphi[\psi_n \setminus p]$ and $\bigwedge \varphi[\Psi \setminus p]$ stands for $\varphi[\psi_1 \setminus p] \wedge \dots \wedge \varphi[\psi_n \setminus p]$. For example, $\Diamond_a\Phi$ abbreviates $\{\Diamond_a\varphi \mid \varphi \in \Phi\}$, and the definition of $\nabla_a\Phi$, above, is then written as $\Box_a \bigvee \Phi \wedge \bigwedge \Diamond_a\Phi$.

We now define the semantics of modal logic. Assume a model $M = (S, R, V)$. The

interpretation of $\varphi \in \mathcal{L}$ is defined by induction.

$$\begin{aligned} M_s &\models p \text{ iff } s \in V_p \\ M_s &\models \neg\varphi \text{ iff } M_s \not\models \varphi \\ M_s &\models \varphi \wedge \psi \text{ iff } M_s \models \varphi \text{ and } M_s \models \psi \\ M_s &\models \Box_a\varphi \text{ iff for all } t \in S : (s, t) \in R_a \text{ implies } M_t \models \varphi \end{aligned}$$

A formula φ is valid on a model M , notation $M \models \varphi$, iff for all $s \in S$, $M_s \models \varphi$; and φ is valid iff φ is valid on all M (in the model class \mathcal{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 \mathbf{K} .

Cover logic The cover operator ∇ has also been used as a syntactic primitive in modal logics [16]. It has recently been axiomatized [10]. The language \mathcal{L}_∇ of cover logic is defined as

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \nabla_a\{\varphi, \dots, \varphi\},$$

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:

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

The set of validities of cover logic is called \mathbf{K}_∇ . The conjunction of two cover formulas is again equivalent to a cover formula:

$$\nabla_a\Phi \wedge \nabla_a\Psi \Leftrightarrow \nabla_a((\Phi \wedge \bigvee \Psi) \cup (\Psi \wedge \bigvee \Phi)).$$

The modal box and diamond are definable as $\Box_a\varphi$ iff $\nabla_a\emptyset \vee \nabla_a\{\varphi\}$, and $\Diamond_a\varphi$ iff $\nabla_a\{\varphi, \top\}$, respectively. Cover logic \mathbf{K}_∇ is equally expressive as modal logic \mathbf{K} (also in the multi-agent version) [10, 34]. We use cover operators in the presentation of the axioms.

Modal μ -calculus For the modal μ -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 \mathcal{L}^μ of modal μ -calculus is defined as follows.

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

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 φ . We will refer to a variable x in an expression $\mu x.\varphi$ as a *fixed-point variable*. The formula $\nu x.\varphi$ is an abbreviation for $\neg\mu x.\neg\varphi[\neg x \setminus x]$. Here, we extend the notion of substitution to modal μ -calculus by ruling out the substitution of bound variables, i.e., to give the crucial clauses: $(\mu x.\varphi)[\psi \setminus x] = \mu x.\varphi$ whereas $(\mu x.\varphi)[\psi \setminus y] = \mu x.\varphi[\psi \setminus y]$.

For the semantics of the μ -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 $M^{[x \mapsto T]}$ for the model $M = (S, R, V^{[x \mapsto T]})$. The semantics of $\mu x.\varphi$ (the top-down presentation, not the bottom-up presentation) is now as follows: Let $\varphi \in \mathcal{L}^\mu$ and model M be given.

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

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

$$\varphi ::= x \mid (\varphi \vee \varphi) \mid (\varphi_0 \wedge \bigwedge_{a \in B} \nabla_a \{\varphi, \dots, \varphi\}) \mid \mu x.\varphi \mid \nu x.\varphi \quad (1)$$

where $x \in X$, $\varphi_0 \in \mathcal{L}_0$ (propositional logic), and $B \subseteq A$. To get the *disjunctive \mathcal{L} formula* (of modal logic) we omit the clauses containing μ -calculus variables x :

$$\varphi ::= (\varphi \vee \varphi) \mid (\varphi_0 \wedge \bigwedge_{a \in B} \nabla_a \{\varphi, \dots, \varphi\}).$$

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

$$\text{Every } \mathcal{L}^\mu \text{ formula is equivalent to a disjunctive } \mathcal{L}^\mu \text{ formula [31].} \quad (2a)$$

$$\text{Every } \mathcal{L} \text{ formula is equivalent to a disjunctive } \mathcal{L} \text{ formula [58].} \quad (2b)$$

Bisimulation quantified modal logic The language $\mathcal{L}_{\tilde{\forall}}$ is defined as

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \Box_a \varphi \mid \tilde{\forall} p \varphi$$

where $a \in A$ and $p \in P$. We let $\tilde{\exists} p \varphi$ abbreviate $\neg \tilde{\forall} p \neg \varphi$. We write $\tilde{\forall}$ and $\tilde{\exists}$ 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 φ , the expression $\tilde{\exists} p \varphi$ means that there exists a denotation of propositional variable p such that φ . It is interpreted as follows (restricted bisimulation \simeq^p is introduced further below in Definition 1):

$$M_s \models \tilde{\forall} p \varphi \text{ iff for all } N_t \text{ such that } N_t \simeq^p M_s, N_t \models \varphi$$

In [22, Lemma 2.43] a bisimulation quantifier characterization of fixed points is given (the details of which are deferred to Section 6 on refinement μ -calculus, where they are pertinent), and from [15] we know that bisimulation quantifiers are also expressible in the modal μ -calculus. For more information on the modal μ -calculus, see [16, 58].

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 $\mathfrak{R} \subseteq S \times S'$ is a bisimulation if for all $(s, s') \in \mathfrak{R}$ 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 $(t, t') \in \mathfrak{R}$;

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 $(t, t') \in \mathfrak{R}$.

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 $\mathfrak{R}^p : M_s \simeq^p M'_s$ 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 \mathfrak{R}_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, we say that M'_s refines M_s for group of agents B , and we write $M_s \succeq_B M'_s$.¹ An A -refinement we call a refinement (plain and simple) and for $\{a\}$ -refinement we write a -refinement.

Dually to refinement, we similarly define B -simulation \mathfrak{R}_B . I.e., a relation \mathfrak{R}_B that satisfies **atoms**, **back- a** , and **forth- a** for every $a \in A \setminus B$, and that satisfies **atoms**, and **forth- b** for every $b \in B$, is a B -simulation.

Restricted refinement and restricted simulation are defined similarly to restricted bisimulation. ⊣

The definition of simulation varies slightly from the one given by Blackburn *et al.* [11, p.110]. Here we ensure that simulations (and refinements) preserve the interpretations (i.e., the truth and falsity) of atoms, whereas [11] 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. Another, inessential, difference with [11] is that in their case **atoms** and **forth** are required for all modalities (in the similarity type), i.e., they consider \mathfrak{R}_B for $B = A$ only.

¹We will overload the meaning of refinement and also say that M'_s is a *refinement* of M_s

If $\mathfrak{R}_B : M_s \succeq_B M'_s$ is a B -refinement, then the converse relation $\mathfrak{R}_B^- ::= \{(s, s') \mid (s', s) \in \mathfrak{R}_B\}$ is a B -simulation, and if M'_s refines M_s then we can also say that M_s simulates M'_s .

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 [40] 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.* \dashv

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 -refinements \mathfrak{R}_1 , and \mathfrak{R}_2 , we can see that their composition, $\{(x, z) \mid \text{there is a } y \text{ for which } (x, y) \in \mathfrak{R}_1, (y, z) \in \mathfrak{R}_2\}$ is also an a -refinement. This is sufficient to demonstrate 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 1 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 $R_a^{N'_t} = \emptyset$ and $R_b^{N'_t} = R_b^M$ for all $b \in A - \{a\}$. It can be seen that N'_t , where $N' = (S^M, R^{N'}, V^M)$ and $t' = s$, satisfies the required properties. \square

An elementary result is the following.

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

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:*

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

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.

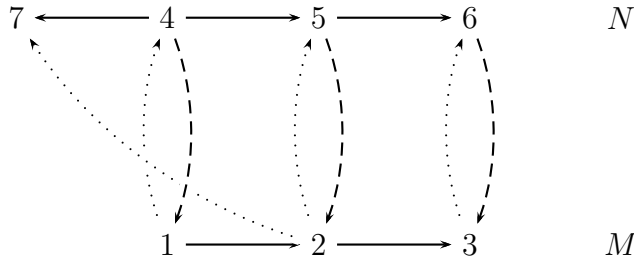


Figure 1: Refinement and simulation, but no bisimulation

3.2 Game and logical characterization of refinement

It is folklore to associate a (infinite duration) two-player game with refinement, in the spirit of [3].

Definition 5 (Refinement game) *Let M_s and N_t be two models. We define a turn-based game $\mathcal{G}_a(M_s, N_t)$ between two players Spoiler and Duplicator (male and female, respectively) by $\mathcal{G}_a(M_s, N_t) = (V, E, (s, t))$ where the set of positions V is partitioned into the positions $V_{\text{Spoiler}} = S^M \times S^N$ of Spoiler and the positions $V_{\text{Duplicator}} = S^M \times [\{\text{forth, back}\} \times (A \cup P)] \times S^N$ 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 1):*

Spoiler's moves		
Move		Name
$((s', t'), (s', (\text{forth}, p), t'))$	whenever $s' \in V^M(p)$	forth-p?
$((s', t'), (s', (\text{back}, p), t'))$	whenever $t' \in V^N(p)$	back-p?
$((s', t'), (s'', (\text{forth}, b), t'))$	whenever $s'' \in R_b^M(s')$	forth-b?
$((s', t'), (s', (\text{back}, b), t''))$	whenever $t'' \in R_b^N(t')$	back-b?
$((s', t'), (s', (\text{back}, a), t''))$	whenever $t'' \in R_a^N(t')$	back-a?
Duplicator's moves		
Move		Name
$((s', (\text{forth}, p), t'), (s', t'))$	whenever $t' \in V^N(p)$	forth-p!
$((s', (\text{back}, p), t'), (s', t'))$	whenever $s' \in V^M(p)$	back-p!
$((s'', (\text{forth}, b), t'), (s'', t''))$	whenever $t'' \in R_b^N(t')$	forth-b!
$((s', (\text{back}, b), t''), (s'', t''))$	whenever $s'' \in R_b^M(s')$	back-b!
$((s', (\text{back}, a), t''), (s'', t''))$	whenever $s'' \in R_a^M(s')$	back-a!

–

In the game $\mathcal{G}_a(M_s, N_t)$, a *play* is a maximal (possibly infinite) sequence of consecutive moves, or equivalently a maximal sequence of adjacent positions in the arena. The play is *winning for Duplicator* if it is infinite or if it is finite and ends in position of Spoiler, otherwise, the play ends in a position of Duplicator and it is winning for Spoiler.

A *strategy of Duplicator* (resp. Spoiler) is a mapping $\sigma : V^*V_{\text{Duplicator}} \rightarrow V$ (resp. $\sigma : V^*V_{\text{Spoiler}} \rightarrow V$) which recommends which moves to choose after each prefix of a play.

A play is an *outcome* of a strategy for **Duplicator** (resp. **Spoiler**) if each time **Duplicator** (resp. **Spoiler**) had to play, she (resp. he) has selected the move recommended by her (resp. his) strategy. A strategy is *winning* if all its outcomes are winning.

Remark 6 *One easily sees that the refinement game of Definition 5 is a particular parity game [38]. Henceforth, according to [35], the refinement game is determined², and memoryless³ strategies suffice.* \dashv

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

Lemma 7 $M_s \succeq_a N_t$ iff **Duplicator** has a winning strategy in $\mathcal{G}_a(M_s, N_t)$. \dashv

Proof Assume **Duplicator** has a winning strategy σ in $\mathcal{G}_a(M_s, N_t)$. By Remark 6 and without loss of generality, this winning σ can be taken to be memoryless. Namely, $\sigma : V_{\text{Duplicator}} \rightarrow V_{\text{Spoiler}}$. Now, define the binary relation $\mathfrak{R}_\sigma \subseteq S^M \times S^N$ as the set of pairs $(s', t') \in V_{\text{Spoiler}}$ that are reachable when **Duplicator** follows her strategy σ . Then it is easy to check that \mathfrak{R}_σ is an a -refinement from M_s to N_t . Also it is not difficult to see that if some a -refinement \mathfrak{R}_a exists from M_s to N_t , then any strategy of **Duplicator** which maintains **Spoiler**'s positions in \mathfrak{R}_a , is winning. Note that by Definition 1 of a refinement, this is always possible for her. \square

We now consider a characterization of the refinement in terms of the logic \mathcal{L}_\forall . Namely, given an agent a , we define the fragment of the a -positive formulas $\mathcal{L}^{a+} \subseteq \mathcal{L}$ by

$$\mathcal{L}^{a+} \ni \varphi ::= p \mid \neg p \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \Box_b \varphi \mid \Diamond_b \varphi \mid \Diamond_a \varphi$$

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

Proposition 8 *For any finitely branching (every state has only finitely many successors) pointed models M_{s_0} and N_{t_0} , and for any agent $a \in A$,*

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

Proof Let us first establish that for every $t \in S^N$ and $s \in S^M$, if **Spoiler** has a winning strategy in $\mathcal{G}_a(M_s, N_t)$, then there exists a formula $\varphi(s, t) \in \mathcal{L}^{a+}$ called a *distinguishing formula for (M_s, N_t)* , for which $N_t \models \varphi(s, t)$ but $M_s \not\models \varphi(s, t)$. Note that if **Spoiler** has a winning strategy in $\mathcal{G}_a(M_s, N_t)$, all plays induced by this strategy have finite length and end in a position where **Duplicator** cannot move. Moreover, by a simple application of König's Lemma (as the game graph $\mathcal{G}_a(M_s, N_t)$ is finitely branching), the length of those plays is bounded.

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 :

²In each position, either **Duplicator** or **Spoiler** has a winning strategy from that position.

³Strategies σ that only take into account the current position in the game, instead of the entire prefix of the game that is currently played.

- if $v = (s, (\text{forth}, p), t)$ (resp. $v = (s, (\text{back}, p), t)$), then there is no move back to (s, t) because $t \notin V^N(p)$ (resp. $s \notin V^M(p)$). A distinguishing formula is $\neg p$ (resp. p).
- if $v = (s', (\text{forth}, b), t)$ (resp. $v = (s, (\text{back}, b), t')$), then $tR_b^N = \emptyset$ (resp. $sR_b^M = \emptyset$). A distinguishing formula is $\Box_b \perp$ (resp. $\Diamond_b \top$). The case $v = (s, (\text{back}, a), t')$ is the same as $(s, (\text{back}, b), t')$. Since forth-a moves are not allowed in the game, position $v = (s', (\text{forth}, a), t)$ is not reachable in the game $\mathcal{G}_a(M_s, N_t)$, so that the formula $\Box_a \perp \notin \mathcal{L}^{a+}$ is not needed.

Assume now that $k > 1$, and pick a winning strategy of **Spoiler** in $\mathcal{G}_a(M_s, N_t)$.

We explore the move from the initial position (s, t) that is 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 $t' \in tR_b^N$, **Spoiler** wins the game $\mathcal{G}_a(M_{s'}, N_{t'})$ 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 $\varphi(s, t) = \Box_b(\bigvee_{t' \in tR_b^N} \varphi(s', t'))$ 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 to $b \neq a$ 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 $\varphi(s, t) = \Diamond_b(\bigwedge_{s' \in sR_b^M} \varphi(s', t'))$ 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 7) and the determinacy of the refinement games (Remark 6), the existence of a winning strategy for **Spoiler** from position (s_0, t_0) is equivalent to $M_{s_0} \not\preceq_a N_{t_0}$; this provides us with the right to left direction of the proposition.

For the other direction, assume $M_s \succeq_a N_t$, and let $\varphi \in \mathcal{L}^{a+}$ with $N_t \models \varphi$. We prove that $M_s \models \varphi$, by induction over the structure of the formula. Basic cases where φ is either p or $\neg p$, but also the cases $\varphi \wedge \psi$ and $\varphi \vee \psi$, are immediate.

Assume $N_t \models \Box_b \varphi$. Then for every $t' \in tR_b^N$, $N_{t'} \models \varphi$. If $tR_b^N = \emptyset$, then by Property **forth-b** of Definition 1 this entails $sR_b^M = \emptyset$ and consequently $M_s \models \Box_b \varphi$ (whatever φ is). Otherwise, $tR_b^N \neq \emptyset$. Take an arbitrary $s' \in sR_b^M$. By Property **forth-b** of Definition 1, there is a $t'_{s'} \in tR_b^M$ 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 $t' \in tR_b^N$ be such that $N_{t'} \models \varphi$. By Property **back-b** of Definition 1, there is some $s' \in sR_b^M$, 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 $b = a$. □

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 **back** for that agent, but not **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 the refined model 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 refined model namely when **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 **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 refined model as a blowup of the initial model of which bits and pieces are cut off.

Example 9 *A simple example is as follows. Consider the structure*

$$\bullet_1 \longrightarrow \bullet_2 \longrightarrow \bullet_3 \longrightarrow \bullet_4$$

and its refinement

$$\bullet_{b'} \longleftarrow \bullet_a \longrightarrow \bullet_b \longrightarrow \bullet_c$$

by way of refinement relation $\mathfrak{R} = \{(1, a), (2, b), (3, c), (2, b')\}$. The arrow $(3, 4)$ has no image in the refined model. 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, and that is because there is yet another arrow from b but no other arrow from b' : arrow $(2, 3)$ has only one image in the refined model. \dashv

The cutting off phase can be described such that the relation to restricted bisimulation becomes 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 the value of that variable. (In other words, it is a restricted bisimulation.) Then, remove arrows to states where that atom is false. The result is a refinement of the original model, except maybe for the value of that variable, as it is now true everywhere — in the example we have chosen a variable that was already initially true everywhere, to simplify the exposition.

Example 10 *Continuing the previous example, consider the following structure bisimilar to the initial model, except for the value of atom p —in the visualization \bullet represents that p is true and \circ represents that p is false.*

$$\circ_{d'} \longleftarrow \circ_{c'} \longleftarrow \bullet_{b'} \longleftarrow \bullet_a \longrightarrow \bullet_b \longrightarrow \bullet_c \longrightarrow \circ_d$$

The relation $\mathfrak{R} = \{(1, a), (2, b), (3, c), (4, d), (2, b'), (3, c'), (4, d')\}$ is a bisimulation, except for the value of p . The refinement from the previous example is a restriction of this structure, namely the result of removing the \circ states and the arrows leading to those states. \dashv

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 unwindings 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.

Of course the variable q , which has now become true everywhere, may have a value different from its value in the a -refinement. Formally, the result is as follows. First, let M be a model with accessibility relation (set of accessibility relations) R , and let R' be such that for all $a \in A$, $R'_a \subseteq R_a$, then (analogously to a model restriction) $M|R'$ is the model that is the same as M but with the accessibility restricted to R' .

Lemma 11 *Given $M_s \succeq_a N_t$, there is an N'_t (with accessibility function R') and some R'' that is the same as R' except that $R''_a \subseteq R'_a$, such that $M_s \simeq N'_t$ and $N'_t|R'' \simeq N_t$. \dashv*

Proof 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 $\mathfrak{R} \subseteq S^M \times S^{N'}$. Consider $S_-^M := S^M \setminus \mathfrak{R}_a^{-1}(S^N)$ (S_-^M is the set of all states in M that do not have an image in N). Now consider $N' = (S', R', V')$ with domain $S' = S^N \cup S_-^M$, such that for each agent b (including a), $(u', v') \in R'_b$ iff:

- $(u', v') \in R_b^N$, or
- $(u', v') \in R_b^M$, or
- $b = a$, $u' \in S^N$, $v' \in S_-^M$, there is a u such that $(u, u') \in \mathfrak{R}_a$, and $(u, v') \in R_a^M$;

and such that $V' = V^N$ on the S^N part of the domain whereas $V' = V^M$ on the new S_-^M part of the domain. Now define $\mathfrak{R} : S^M \rightarrow S'$ as follows: $(u, u') \in \mathfrak{R}$ iff $(u, u') \in \mathfrak{R}_a$ or $(u \in S_-^M \text{ and } u = u')$. Then \mathfrak{R} is a bisimulation linking M_s and N'_t . If we restrict R'_a to R_a^N , we get N_t back (states in the S_-^M part of N' have become unreachable). We have satisfied the proof requirement that $M_s \simeq N'_t$ and $N'_t|R''_a \simeq N_t$ (for $R''_a = R_a^N$). \square

Lemma 12 *Given $M_s \succeq_a N_t$, there is an N'_t (with accessibility function R') and some $p \in P$ such that $M_s \simeq^p N'_t$ and $N'_t|R'' \simeq^p N_t$, where R'' is the same as R' except that $(u, u') \in R''_a$ iff $N'_{u'} \models p$.* \dashv

Proof To satisfy the requirement for p , we make p false on the S_-^M part of the domain of N' , and true on the S^N part of the domain of N' . (We do not change the value of other propositional letters on N' .) \square

Below, $M|p$ is the restriction of M to the set of states satisfying p .

Proposition 13 *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 $N'_t|p$ is identical to N_t except for maybe the value of p .* \dashv

Proof Clearly, in Lemma 12, $N'_t|R'' \simeq N'_t|p$. The model restriction gets rid of the the S_-^M part of N' , so we now have that $N'_t|p$ is identical to (and not merely bisimilar to) N_t except for maybe the value of p . \square

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 formulas.

3.4 Refinement and action models

We recall another important result connecting structural refinement to action model execution [9]. For full details, see [55]. An *action model* $\mathbf{M} = (\mathbf{S}, \mathbf{R}, \mathbf{pre})$ is like a model $M = (S, R, V)$ but with the valuation replaced by a precondition function $\mathbf{pre} : \mathbf{S} \rightarrow \mathcal{L}$ (for a given language \mathcal{L}). The elements of \mathbf{S} are called *action points*. A restricted modal product $(M \otimes \mathbf{M})$ consists of pairs (s, \mathbf{s}) such that $M_s \models \mathbf{pre}(\mathbf{s})$, the product of accessibility relations namely such that $((s, \mathbf{s}), (t, \mathbf{t})) \in R_a$ iff $(s, t) \in R_a$ and $(\mathbf{s}, \mathbf{t}) \in \mathbf{R}_a$, and keeping the valuation of the state in the pair: $(s, \mathbf{s}) \in V(p)$ iff $s \in V(p)$. A pointed action model $M_{\mathbf{s}}$ is an *epistemic action*.

Proposition 14 [55, Prop. 4, 5] *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.* \dashv

It is instructive to outline the proof of these results.

Given pointed model M_s and epistemic action $M_{\mathbf{s}}$, the resulting $(M \otimes \mathbf{M})_{(s, \mathbf{s})}$ is a refinement of M_s by way the relation \mathfrak{R} consisting of all pairs $(t, (t, \mathbf{t}))$ such that $M_t \models \mathbf{pre}(\mathbf{t})$.

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 $\mathbf{M}_{s'}$ 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 $\mathbf{pre}(t)$ 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 [13]. One then shows that $(M \otimes \mathbf{M}, (s, s'))$ can be bisimulation-contracted to $N_{s'}$. It is unknown if the finiteness restriction can be lifted, because the existence of distinguishing formulas plays a crucial part in the proof.

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 Modal specifications refinement

Modal specifications are classic, convenient, and expressive mathematical objects that represent interfaces of component-based systems [36, 44, 45, 46, 4, 49]. 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 [44, 45]. 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 ν -calculus fragment of the μ -calculus [20]. In order to abstract from a particular implementation, an 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 μ -calculus. From prior publications [55, 56] refinement modal logic is known as ‘future event logic’. In that interpretation different \square_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 refinement modal logic are as follows.

Definition 15 (Languages \mathcal{L}_\forall and \mathcal{L}_\forall^μ) *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*

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \Box_a\varphi \mid \forall_a\varphi$$

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

$$\varphi ::= x \mid p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \Box_a\varphi \mid \forall_a\varphi \mid \mu x.\varphi \quad \dashv$$

We write $\exists_a\varphi$ for $\neg\forall_a\neg\varphi$. For a subset $\{a_1, \dots, a_n\} = B \subseteq A$ of agents we introduce the abbreviation $\exists_B\varphi$ for $\exists_{a_1} \dots \exists_{a_n}\varphi$ (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 $\tilde{\forall}p$ 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 φ 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 16 (Semantics of refinement) *Assume a 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 RML (refinement modal logic) and the set of validities of \mathcal{L}_\forall^μ is the logic RML $^\mu$ (refinement μ -calculus).⁴ \dashv

In other words, $\forall_a\varphi$ is true in a pointed model iff φ 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 [55]).

Proposition 17 (Bisimulation invariance) *Refinement modal logic and refinement μ -calculus are bisimulation invariant.* \dashv

⁴As is usual in the area, we will continue to use the term ‘logic’ in a general sense, beyond that of a set of validities.

Proof Bisimulation invariance is the following property: given $M_s \simeq N_t$ and a formula φ , 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 μx 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. \square

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

Proposition 18 For all agents a, b , $\models \exists_a \exists_b \varphi \leftrightarrow \exists_b \exists_a \varphi$. \dashv

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.) \square

Proposition 19 The following are validities of RML.

1. $\forall_a \varphi \rightarrow \varphi$ (reflexivity)
2. $\forall_a \varphi \rightarrow \forall_a \forall_a \varphi$ (transitivity)
3. $\exists_a \forall_a \varphi \rightarrow \forall_a \exists_a \varphi$ (Church-Rosser)
4. $\exists_a (\varphi \vee \psi) \leftrightarrow (\exists_a \varphi \vee \exists_a \psi)$ and $\forall_a (\varphi \wedge \psi) \leftrightarrow (\forall_a \varphi \wedge \forall_a \psi)$
5. $\exists_a \diamond_a \varphi \leftrightarrow \diamond_a \exists_a \varphi$ \dashv

Proof Note that the first three items directly follow from Proposition 2.

1. The trivial refinement is a refinement.
2. Composition of two refinements is a refinement.
3. Refinement satisfies the Church-Rosser property.
4. This follows from the semantics. Consider the diamond form of the equivalence. The right-to-left direction is trivial. For the left-to-right direction note that if $\varphi \vee \psi$ is true in some refinement of a given model, then φ is true or ψ is true in that refinement, so $\exists_a \varphi$ is true or $\exists_a \psi$ is true in the given model.

5. 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 $M_s \succeq_a M'_{s'}$, $M'_{s'} \models \diamond_a \varphi$, and $M'_{t'} \models \varphi$. Because of **back**, there is a $t \in sR_a$ such that $M_t \succeq_a M'_{t'}$. 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 $M'_{t'}$ be such that $M_t \succeq_a M'_{t'}$, $M_t \models \exists_a \varphi$, and $M'_{t'} \models \varphi$. 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 \mathcal{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 $\exists \square \perp$ is a validity in **RML**: just remove all access. But in refinement epistemic logic, interpreted on $\mathcal{S5}$ models, this is not a validity: seriality of models must be preserved in every refinement. See [56, 27].

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

$$\exists_a (\square_a p \wedge \neg \square_b \square_a p)$$

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 .

On the left, the formula $\exists (\square_a p \wedge \neg \square_b \square_a p)$ is true, because $\square_a p \wedge \neg \square_b \square_a p$ 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 $\square_a p$ is true, whereas agent b also considers another state possible, wherein agent a considers it possible that p is false. Therefore, $\neg \square_b \square_a p$ 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 14). 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

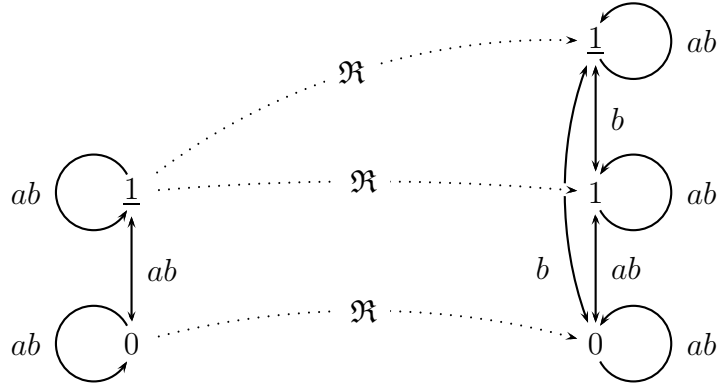


Figure 2: An example of refinement as change of knowledge

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.

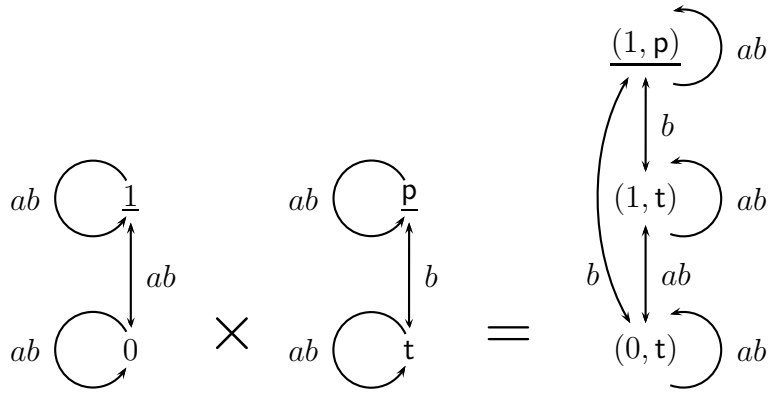


Figure 3: The refinement in Example 4.2.

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 14 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 [23].

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 φ , say in \mathcal{L}^μ , we use refinement quantifiers to express several classic verification/synthesis problems. In this example, we let $\Box\varphi$ stand for $\Box_c\varphi \wedge \Box_u\varphi$.

The *the control problem* [48], known as the question “is there a way to control actions c of the system S so that property φ is guaranteed?”, can be expressed in \mathcal{L}_\forall by wondering whether

$$S \models \exists_c \varphi .$$

The *module checking problem* [33] 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 an abstract action that denotes internal ones, while action u abstracts all external actions, i.e. actions performed by the environment. Also, assume there is an atomic proposition e that distinguishes states where it is the turn of the system to act (thus only action c is available) from states where it is the turn of the environment (thus only action u is available). In this setting, 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 *non-blocking* environments: the property can be captured by the \mathcal{L}^μ -formula $\text{NonBlockingEnv} := \nu x.(e \Rightarrow \Diamond_u \top) \wedge \Box x$, which formally says that it is always the case $(\nu x.(...) \wedge \Box x)$ that whenever in an environment state, there is an outgoing transition from that state $(e \Rightarrow \Diamond_u \top)$. Now, by ‘guarding’ the universal quantification over all u -refinements (i.e. all environments) with the NonBlockingEnv assumption, the statement becomes

$$S \models \forall_u (\text{NonBlockingEnv} \Rightarrow \varphi)$$

The *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

$$S \models \exists_c \forall_u (\text{NonBlockingEnv} \Rightarrow \varphi) .$$

A last example is borrowed from *protocol synthesis problems*. Consider a specification, MUTEX , of a mutual exclusion protocol involving processes $1, 2, \dots, k$, and some property φ specified in \mathcal{L}^μ . Now we may ask if we can find a refinement of MUTEX that satisfies φ 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

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

where **AG** is the CTL-modality, which is defined in \mathcal{L}^μ as $\mathbf{AG}(\psi) \equiv \nu x.\psi \wedge \Box x$, and means 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.

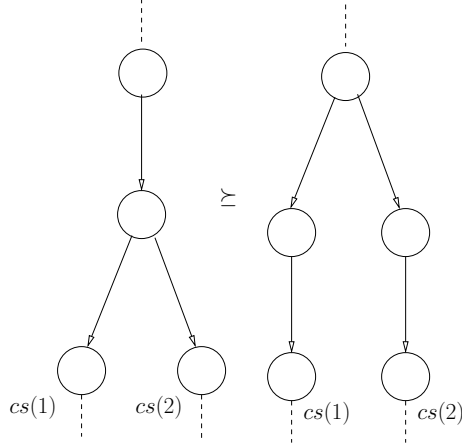


Figure 4: The refinement of MUTEX.

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 φ , followed by a (non-standard) relativization for that agent to that variable, for which we write $\tilde{\exists} q \varphi^{(a,q)}$ (to be defined shortly). For refinement \succeq for the set of all agents (recall that we write \succeq for \succeq_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 φ followed by (standard) relativization to that variable: $\tilde{\exists} q \varphi^q$. 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 [53, 39]). We propose a definition of relativization that may be considered non-standard for several reasons. Firstly, it is relativization not merely to a propositional variable but also for a given agent only. The standard definition is

then the special case of relativization to that variable for all agents (we will prove that consecutive relativization to the same variable for two different agents is commutative, in other words, order independent). Secondly, the relativization that we propose corresponds in the semantics to arrow elimination and not to state elimination (in other words, it does not correspond to submodel restriction). From the modal logical literature, the approach in [39] is arrow-eliminating but that in [53] is state-eliminating.

The arrow-eliminating relativization need only be done in accessible states but not in the actual state (e.g., the relativization of a variable q to a variable p is that same variable q and not $p \wedge q$).

The difference between state-eliminating relativization and arrow-eliminating relativization is similar to the difference between state-eliminating public announcement semantics [43, 9] and arrow-eliminating public announcement semantics [32, 24], in the area of dynamic epistemic logic. As our relativization is with respect to a given agent, we have no option but to use arrow-eliminating relativization.

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 then be used in Section 6.

Definition 20 (Relativization) *Relativization $\bullet^{(a,p)} : \mathcal{L}_{\check{v}} \rightarrow \mathcal{L}_{\check{v}}$ to propositional variable p for agent $a \in A$ is defined as follows.*

$$\begin{aligned}
q^{(a,p)} &= q \\
(\neg\varphi)^{(a,p)} &= \neg\varphi^{(a,p)} \\
(\varphi \wedge \psi)^{(a,p)} &= \varphi^{(a,p)} \wedge \psi^{(a,p)} \\
(\Box_a\varphi)^{(a,p)} &= \Box_a(p \rightarrow \varphi^{(a,p)}) \\
(\Box_b\varphi)^{(a,p)} &= \Box_b\varphi^{(a,p)} && \text{for } b \neq a \\
(\check{\forall}q\varphi)^{(a,p)} &= \check{\forall}q\varphi^{(a,p)} && \text{for } q \neq p \\
(\check{\forall}p\varphi)^{(a,p)} &= \check{\forall}q\varphi[q \setminus p]^{(a,p)} && \text{choose } q \text{ that does not occur in } \varphi \quad \dashv
\end{aligned}$$

Lemma 21 *Let M_s be a model with accessibility function R and $R'_a \subseteq R_a$ such that: $(t, t') \in R'_a$ iff $M_{t'} \models p$. Then $M_s \models \varphi^{(a,p)}$ if and only if $M_s|_{R'_a} \models \varphi$. \dashv*

Proof The proof is by induction on the structure of φ .

- $M_s \models q^{(a,p)} \Leftrightarrow$
 $M_s \models q \Leftrightarrow$ propositional variables do not change value
 $M_s|_{R'_a} \models q$
- The clauses for negation and conjunction are elementary.
- $M_s \models (\Box_a\varphi)^{(a,p)} \Leftrightarrow$
 $M_s \models \Box_a(p \rightarrow \varphi^{(a,p)}) \Leftrightarrow$
for all $t \in sR_a : M_t \models p \rightarrow \varphi^{(a,p)} \Leftrightarrow$
for all $t \in sR_a : M_t \models p$ implies $M_t \models \varphi^{(a,p)} \Leftrightarrow$ I.H.

for all $t \in sR_a : M_t \models p$ implies $M_t|R'_a \models \varphi \Leftrightarrow t \in sR_a$ and $t \models p$ iff $t \in sR'_a$
for all $t \in sR'_a : M_t|R'_a \models \varphi \Leftrightarrow$
 $M_s|R'_a \models \Box_a \varphi$

- $M_s \models (\Box_b \varphi)^{(a,p)} \Leftrightarrow$
 $M_s \models \Box_b \varphi^{(a,p)} \Leftrightarrow$
for all $t \in sR_b : M_t \models \varphi^{(a,p)} \Leftrightarrow$ I.H.
for all $t \in sR_b$ (in M_t) : $M_t|R'_a \models \varphi \Leftrightarrow$ sR_b in M equals sR_b in $M|R'_a$
for all $t \in sR_b$ (in $M_t|R'_a$) : $M_t|R'_a \models \varphi \Leftrightarrow$
 $M_s|R'_a \models \Box_b \varphi$
- For a more natural argument we take the existential quantifier instead of the universal quantifier. First, observe that:

$M_s \models (\exists q \varphi)^{(a,p)} \Leftrightarrow$
 $M_s \models \exists q \varphi^{(a,p)} \Leftrightarrow$
there is an $N_t \simeq^q M_s : N_t \models \varphi^{(a,p)} \Leftrightarrow$ I.H.
there is an $N_t \simeq^q M_s : N_t|R''_a \models \varphi$ where $R''_a \subseteq R_a^N$ s.t. $(u, u') \in R''_a$ iff $M_{u'} \models p$

We also have that, by definition:

$M_s|R'_a \models \exists q \varphi \Leftrightarrow$
there is an $N'_t \simeq^q M_s|R'_a : N'_t \models \varphi$

It remains to show that the two final statements in these chains of equivalences are also equivalent.

From left to right is easy. If $\mathfrak{R} : N_t \simeq^q M_s$, then also $\mathfrak{R} : N_t|R''_a \simeq^q M_s|R'_a$. In N and M we remove all a -arrows to $\neg p$ states; and if it is already a bisimulation, then the forth and back requirements still hold for fewer pairs in the accessibility relation for a . So we can take $N'_t = N_t|R''_a$.

From right to left is not easy. Let us first explain this informally. Given that $N'_t \simeq^q M_s|R'_a$, the part of M that is inaccessible from $M_s|R'_a$ (i.e., not in the s -generated submodel) may not be bisimilar to anything in N' . This is problematic, because we need to transform N'_t to some N_t in a way that establishes a q restricted bisimulation between N_t and all of M_s . Fortunately, the transformation can be an extension of N'_t , wherein we uniformly treat states in that inaccessible part of M and other states of M : we do not need to be economic in our construction. These are the details.

Let $\mathfrak{R} : N'_t \simeq^q M_s|R'_a$ be the restricted bisimulation. To be explicit, let $M = (S, R, V)$ and let $N' = (S', R', V')$. Consider $S^{a\bar{p}} = \{u \in S^M \mid \exists v \in S, (v, u) \in R_a \setminus R'_a\}$. For each $u \in S^{a\bar{p}}$ we need an exact copy M^u of M (let $M^u = (S^u, R^u, V^u)$) in our construction. We now define $N = (S^N, R^N, V^N)$ as follows:

- $S^N = S' \cup \bigcup \{S^u \mid u \in S^{a\bar{p}}\};$
- for all $b \neq a$, $R_b^N = R'_b \cup \bigcup \{R_b^u \mid u \in S^{a\bar{p}}\};$

- $R_a^N = R'_a \cup \bigcup \{R_a^u \mid u \in S^{a\bar{p}}\} \cup \{(v', u) \mid (v', v) \in \mathfrak{R} \text{ and } (v, u) \in R_a \setminus R'_a\}$;
- for all $p \in P$, $V^N(p) = V'(p) \cup \bigcup \{V^u(p) \mid u \in S^{a\bar{p}}\}$.

Now take $t = t'$, and let R''_a as before be restriction of R_a^N to pairs $(u, u') \in R_a^N$ such that u' satisfies p . It is now immediate that $N_t \simeq^q M_s$ and therefore also $N_t \mid R''_a \models \varphi$.

- The other clause for the universal quantifier starts with a renaming operation (that equally applies to the existential quantifier), and then proceeds as in the previous clause.

□

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

Lemma 22 *Let $\varphi \in \mathcal{L}_{\check{\forall}}$. Then $(\varphi^{(a,p)})^{(b,q)} = (\varphi^{(b,q)})^{(a,p)}$.* ←

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

- $((\Box_a \varphi)^{(a,p)})^{(b,q)} \Leftrightarrow$
 $(\Box_a (p \rightarrow \varphi^{(a,p)}))^{(b,q)} \Leftrightarrow$
 $\Box_a (p \rightarrow \varphi^{(a,p)})^{(b,q)} \Leftrightarrow$
 $\Box_a (p^{(b,q)} \rightarrow (\varphi^{(a,p)})^{(b,q)}) \Leftrightarrow$ I.H., and clause for variables
 $\Box_a (p \rightarrow (\varphi^{(b,q)})^{(a,p)}) \Leftrightarrow$
 $(\Box_a \varphi^{(b,q)})^{(a,p)} \Leftrightarrow$
 $((\Box_a \varphi)^{(b,q)})^{(a,p)}$
- $((\check{\forall} p \varphi)^{(a,p)})^{(b,q)} \Leftrightarrow$ choose $r \neq q$ (or else, yet another step)
 $(\check{\forall} r \varphi [r \setminus p]^{(a,p)})^{(b,q)} \Leftrightarrow$
 $\check{\forall} r (\varphi [r \setminus p]^{(a,p)})^{(b,q)} \Leftrightarrow$ I.H.
 $\check{\forall} r (\varphi [r \setminus p]^{(b,q)})^{(a,p)} \Leftrightarrow$ substitution of other variables than q
 $\check{\forall} r (\varphi^{(b,q)} [r \setminus p])^{(a,p)} \Leftrightarrow$
 $(\check{\forall} p \varphi^{(b,q)})^{(a,p)} \Leftrightarrow$
 $((\check{\forall} p \varphi)^{(b,q)})^{(a,p)}$

□

Given Lemma 22, we may view a nesting of relativizations $(\dots (\varphi^{(a_1,p)}) \dots^{(a_n,p)})$ as a relativization $\varphi^{\{a_1, \dots, a_n\}, p}$ for the set of agents $\{a_1, \dots, a_n\}$. Furthermore, for $\varphi^{(A,p)}$ we can write φ^p : the usual relativization for all agents simultaneously.

To make the syntactic correspondence we now introduce a translation.

Definition 23 The translation $t : \mathcal{L}_\forall \rightarrow \mathcal{L}_{\tilde{\forall}}$ is defined by induction on $\varphi \in \mathcal{L}_\forall$. All clauses except $\forall_a \varphi$ are trivial.

$$\begin{aligned}
t(p) &= p \\
t(\neg\varphi) &= \neg t(\varphi) \\
t(\varphi \wedge \psi) &= t(\varphi) \wedge t(\psi) \\
t(\Box_a \varphi) &= \Box_a t(\varphi) \\
t(\forall_a \varphi) &= \tilde{\forall} p t(\varphi)^{(a,p)} \quad \text{where } p \text{ does not occur in } \varphi \quad \dashv
\end{aligned}$$

Example 24

$$\begin{aligned}
t(\exists_a \exists_b r) &= \tilde{\exists} p t(\exists_b r)^{(a,p)} = \\
\tilde{\exists} p (\tilde{\exists} p t(r)^{(b,p)})^{(a,p)} &= \tilde{\exists} p (\tilde{\exists} p r^{(b,p)})^{(a,p)} = \\
\tilde{\exists} p (\tilde{\exists} p r)^{(a,p)} &= \tilde{\exists} p \tilde{\exists} q r^{(a,q)} = \tilde{\exists} p \tilde{\exists} q r \quad \dashv
\end{aligned}$$

Proposition 25 Let $\varphi \in \mathcal{L}_\forall$. Then φ is equivalent to $t(\varphi)$. \(\dashv\)

Proof In the proposition we allowed ourselves a slight abuse of language: it means that, given any M_s , the value of φ in the semantics for refinement modal logic is equivalent to the value of $t(\varphi)$ in the semantics for bisimulation quantified modal logic. The proposition follows from Lemma 12, Lemma 21 and Def. 23. We show the case $\forall_a \varphi$ of the inductive proof—and to suit the intuition we take the existential quantifier \exists_a .

$$\begin{aligned}
M_s &\models \exists_a \varphi \\
\text{iff} & \\
\text{there is an } M'_{s'} &\text{ such that } M_s \succeq_a M'_{s'} \text{ and } M'_{s'} \models \varphi \\
\text{iff (I.H.)} & \\
\text{there is an } M'_{s'} &\text{ such that } M_s \succeq_a M'_{s'} \text{ and } M'_{s'} \models t(\varphi) \\
\text{iff (Lemma 12)} & \\
\text{there is an } N'_{t'} &\text{ with } R''_a \subseteq R'_a \text{ (restr. to } p \text{ true) s.t. } M_s \simeq^p N'_{t'} \text{ and } N'_{t'} | R''_a \models t(\varphi) \\
\text{iff (Lemma 21)} & \\
\text{there is an } N'_{t'} &\text{ such that } M_s \simeq^p N'_{t'} \text{ and } N'_{t'} \models t(\varphi)^{(a,p)} \\
\text{iff} & \\
M_s &\models \tilde{\exists} p t(\varphi)^{(a,p)} \\
\text{iff} & \\
M_s &\models t(\exists_a \varphi). \quad \square
\end{aligned}$$

This corollary makes the characteristic cases of Proposition 25 stand out.

Corollary 26 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 $\tilde{\exists} p \varphi^{(a,p)}$;
- *refinement is bisimulation quantification plus relativization:*
 $\exists \varphi$ is equivalent to $\tilde{\exists} p \varphi^p$. \(\dashv\)

In the logic of public announcements, the latter is written as: $\exists \varphi$ is equivalent to $\tilde{\exists} p \langle p! \rangle \varphi$.

4.4 Alternating refinement relations

Alternating transition systems (ATS) were introduced [3] to model multi-agent 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 requiring ‘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 \mathcal{S}' *a-refines* an ATS \mathcal{S} , every formula true in \mathcal{S}' is also true in \mathcal{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] 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 \mathcal{L}_\forall , as the formula $\exists_a(\Box_b p \wedge \Diamond_a(\exists_b \Box_a p))$ 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 **RML** for the logic 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 φ is a theorem, and p occurs in φ , and ψ is any formula, then $\varphi[\psi \setminus p]$ 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 RML is therefore not a normal modal logic.

Definition 27 (Axiomatization RML) *The axiomatization RML consists of all sub-*

stitution instances of the axioms

$$\begin{array}{ll}
\mathbf{Prop} & \text{All tautologies of propositional logic} \\
\mathbf{K} & \Box_a(\varphi \rightarrow \psi) \rightarrow \Box_a\varphi \rightarrow \Box_a\psi \\
\mathbf{R} & \forall_a(\varphi \rightarrow \psi) \rightarrow \forall_a\varphi \rightarrow \forall_a\psi \\
\mathbf{RProp} & \forall_a p \leftrightarrow p \text{ and } \forall_a \neg p \leftrightarrow \neg p \\
\mathbf{RK} & \exists_a \nabla_a \Phi \leftrightarrow \bigwedge \diamond_a \exists_a \Phi \\
\mathbf{RKmulti} & \exists_a \nabla_b \Phi \leftrightarrow \nabla_b \exists_a \Phi \quad \text{where } a \neq b \\
\mathbf{RKconj} & \exists_a \bigwedge_{b \in B} \nabla_b \Phi^b \leftrightarrow \bigwedge_{b \in B} \exists_a \nabla_b \Phi^b
\end{array}$$

and the rules

$$\begin{array}{ll}
\mathbf{MP} & \text{From } \varphi \rightarrow \psi \text{ and } \varphi \text{ infer } \psi \\
\mathbf{NecK} & \text{From } \varphi \text{ infer } \Box_a \varphi \\
\mathbf{NecR} & \text{From } \varphi \text{ infer } \forall_a \varphi
\end{array}$$

where $a, b \in A$, $p \in P$, and $B \subseteq A$. If φ is derivable, we write $\vdash \varphi$, and φ is called a theorem, as usual. The well-known axiomatization \mathbf{K} for the logic \mathbf{K} consists of the axioms \mathbf{Prop} , \mathbf{K} , and the rules \mathbf{MP} and \mathbf{NecK} . \dashv

In the definition, given $\Phi = \{\varphi_1, \dots, \varphi_n\}$, note that $\exists_a \nabla_a \Phi \leftrightarrow \bigwedge \diamond_a \exists_a \Phi$ stands for $\exists_a \nabla_a \Phi \leftrightarrow \bigwedge_{\varphi \in \Phi} \diamond_a \exists_a \varphi$ (see the technical preliminaries) and so for $\exists_a \nabla_a \{\varphi_1, \dots, \varphi_n\} \leftrightarrow \bigwedge \diamond_a \exists_a \varphi_1 \wedge \dots \wedge \bigwedge \diamond_a \exists_a \varphi_n$. The axiomatization \mathbf{RML} is surprisingly simple given the complexity of the semantic definition of the refinement operator \forall ; 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 19), the corresponding modal axioms are not required. These properties are schematically derivable. First, we demonstrate soundness of \mathbf{RML} .

Given the definitions of \Box and \diamond in terms of cover, it may be instructive to see how the \mathbf{RK} axiom works as a reduction principle for $\exists \Box \varphi$ and $\exists \diamond \varphi$ —note that we need both, as there is no principle for $\exists \neg \varphi$. For simplicity we do not label the operators with agents. We get (for the second equivalence, see also Prop. 33, later):

$$\begin{array}{ll}
& \exists \Box \varphi \leftrightarrow \exists (\nabla \{\varphi\} \vee \nabla \emptyset) \\
(\text{use } \mathbf{R}, \mathbf{Prop} \text{ and } \mathbf{MP}) & \leftrightarrow \exists \nabla \{\varphi\} \vee \exists \nabla \emptyset \\
& (\text{use } \mathbf{RK}) \leftrightarrow \exists \nabla \{\varphi\} \vee \bigwedge \diamond \exists \emptyset \\
(\text{empty conjunction is true}) & \leftrightarrow \exists \nabla \{\varphi\} \vee \top \\
& \leftrightarrow \top
\end{array}$$

and

$$\begin{array}{ll}
& \exists \diamond \varphi \leftrightarrow \exists \nabla \{\varphi, \top\} \\
(\text{use } \mathbf{RK}) & \leftrightarrow \diamond \exists \varphi \wedge \diamond \exists \top \\
& \leftrightarrow \diamond \exists \varphi
\end{array}$$

One may wonder why we did not choose $\exists\Box\varphi \leftrightarrow \top$ and $\exists\Diamond\varphi \leftrightarrow \Diamond\exists\varphi$ (we recall Proposition 19) 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 $\Phi = \{\varphi\}$, $\Phi = \emptyset$, and $\Phi = \{\varphi, \top\}$, but any finite set of formulas.

5.1 Soundness

Theorem 28 *The axiomatization **RML** is sound for RML.* ⊣

Proof As all models of \mathcal{L}_\forall 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 $M_s \models \forall_a(\varphi \rightarrow \psi)$, and $M_s \models \forall_a\varphi$. Then for every N_t , where $N_t \preceq_a M_s$, we have $N_t \models \varphi \rightarrow \psi$, and also $N_t \models \varphi$. From $N_t \models \varphi \rightarrow \psi$ 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. $\models p \leftrightarrow \forall_a p$. Similarly, for $\models \neg p \leftrightarrow \forall_a \neg p$, using that $s \notin V^M(p)$ if and only if $t \notin V^N(p)$.

RK

Suppose M_s is a model, where $M = (S, R, V)$, such that for some set Φ , $M_s \models \exists_a \nabla_a \Phi$. Therefore, there is a model $N_t \preceq_a M_s$ such that $N_t \models \nabla_a \Phi$ —where $N = (S^\Phi, R^\Phi, V^\Phi)$. Expanding the definition, we have that for every $\varphi \in \Phi$ there is some $u \in tR_a^\Phi$ such that $N_u \models \varphi$. Also, because of **back**, for every such $u \in tR_a^\Phi$ 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 $M_s \models \bigwedge \Diamond_a \exists_a \Phi$.

Conversely, suppose that $M_s \models \bigwedge \Diamond_a \exists_a \Phi$. 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 $N_{u^\varphi}^\varphi \preceq_a M_{t^\varphi}$, where $N^\varphi = (S^\varphi, R^\varphi, V^\varphi)$, such that $N_{u^\varphi}^\varphi \models \varphi$. Without loss of generality, we may assume that for all $\varphi, \varphi' \in \Phi$ the models N^φ and $N^{\varphi'}$ are disjoint.

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

$$\begin{aligned}
S^\Phi &= \{s'\} \cup S \cup \bigcup_{\varphi \in \Phi} S^\varphi \\
R_a^\Phi &= \{(s', u^\varphi) \mid \varphi \in \Phi\} \cup R_a \cup \bigcup_{\varphi \in \Phi} R_a^\varphi \\
R_b^\Phi &= \{(s', t) \mid (s, t) \in R_b\} \cup R_b \cup \bigcup_{\varphi \in \Phi} R_b^\varphi && \text{for } b \neq a \\
V^\Phi(p) &= \overline{\{s'\}} \cup V(p) \cup \bigcup_{\varphi \in \Phi} V^\varphi(p) && \text{for } p \in P
\end{aligned}$$

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

We can see that $M_s \succeq_a M_{s'}^\Phi$, via the relation $\mathcal{R}^\Phi = \{(s, s')\} \cup \mathcal{I} \cup \bigcup_{\varphi \in \Phi} \mathcal{R}^\varphi$ where \mathcal{I} is the identity on S and each \mathcal{R}^φ is the refinement relation corresponding to $M_{t^\varphi} \succeq_a N_{u^\varphi}^\varphi$ (see also [25]). Furthermore, for each $t \in s'R_a^\Phi$ it is clear that $M_t^\Phi \simeq N_{u^\varphi}^\varphi$ for some φ , and thus $M_t^\Phi \models \varphi$, and so $M_t^\Phi \models \bigvee \Phi$. Therefore $M_{s'}^\Phi \models \Box_a \bigvee \Phi$. Finally, for each $\varphi \in \Phi$ there is some $u^\varphi \in s'R_a^\Phi$ where $M_{u^\varphi}^\Phi \models \varphi$, so for each $\varphi \in \Phi$ we have $M_s^\Phi \models \Diamond_a \varphi$, so we have $M_{s'}^\Phi \models \bigwedge \Diamond_a \Phi$. Combined, $M_{s'}^\Phi \models \Box_a \bigvee \Phi$ and $M_{s'}^\Phi \models \bigwedge \Diamond_a \Phi$ state that $M_{s'}^\Phi \models \nabla_a \Phi$, and therefore $M_s \models \exists_a \nabla_a \Phi$.

RKmulti

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 $M'_u \preceq_a M_v$. 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 $M_s \models \bigwedge \Diamond_b \exists_a \Phi$. 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 $M_s \models \Box_b \bigvee \exists_a \Phi$, so together we get $M_s \models \nabla_b \exists_a \Phi$.

For the converse direction, suppose that $M_s \models \nabla_b \exists_a \Phi$. From the definition of ∇_b it follows that $M_s \models \bigwedge \Diamond_b \exists_a \Phi$. We now proceed in a similar way as in the case **RK**. From $M_s \models \bigwedge \Diamond_b \exists_a \Phi$ it follows that for every $\varphi \in \Phi$ there is some $t^\varphi \in sR_b$ such that $M_{t^\varphi} \models \exists_a \varphi$. Thus, for each $\varphi \in \Phi$, there is some model $N_{u^\varphi}^\varphi \preceq_a M_{t^\varphi}$, where $N^\varphi = (S^\varphi, R^\varphi, V^\varphi)$, such that $N_{u^\varphi}^\varphi \models \varphi$. Define the model $M^\Phi = (S^\Phi, R^\Phi, V^\Phi)$ similar to the case **RK**, except that: the roles of a and b have been swapped, and the accessibility relation for all agents c different from a and b is defined as that for a .

$$\begin{aligned}
S^\Phi &= \{s'\} \cup S \cup \bigcup_{\varphi \in \Phi} S^\varphi \\
R_b^\Phi &= \{(s', u^\varphi) \mid \varphi \in \Phi\} \cup R_b \cup \bigcup_{\varphi \in \Phi} R_b^\varphi \\
R_c^\Phi &= \{(s', t) \mid (s, t) \in R_c\} \cup R_c \cup \bigcup_{\varphi \in \Phi} R_c^\varphi && \text{for } c \neq b \\
V^\Phi(p) &= \overline{\{s'\}} \cup V(p) \cup \bigcup_{\varphi \in \Phi} V^\varphi(p) && \text{for } p \in P
\end{aligned}$$

where $\overline{\{s'\}} = \{s'\}$ if $s \in V(p)$ and else $\overline{\{s'\}} = \emptyset$ (R_c^Φ also defines R_a^Φ , namely for $c = a$).

We can see that $M_s \succeq_a M_{s'}^\Phi$, via the relation $\mathcal{R}^\Phi = \{(s, s')\} \cup \mathcal{I} \cup \bigcup_{\varphi \in \Phi} \mathcal{R}^\varphi$ where \mathcal{I} is the identity on S and each \mathcal{R}^φ is the refinement relation corresponding to $M_{t^\varphi} \succeq_a N_{u^\varphi}^\varphi$ (see also [25]). Furthermore, for each $t \in s'R_b^\Phi$ it is clear that $M_t^\Phi \simeq N_{u^\varphi}^\varphi$ for some φ , and thus $M_t^\Phi \models \varphi$, and so $M_t^\Phi \models \bigvee \Phi$. Therefore $M_{s'}^\Phi \models \Box_b \bigvee \Phi$. Finally, for each $\varphi \in \Phi$ there is some $u^\varphi \in s'R_b^\Phi$ where $M_{u^\varphi}^\Phi \models \varphi$, so for each $\varphi \in \Phi$ we have $M_s^\Phi \models \Diamond_b \varphi$, so we have $M_{s'}^\Phi \models \bigwedge \Diamond_b \Phi$. Combined, $M_{s'}^\Phi \models \Box_b \bigvee \Phi$ and $M_{s'}^\Phi \models \bigwedge \Diamond_b \Phi$ state that $M_{s'}^\Phi \models \nabla_b \Phi$, and therefore $M_s \models \exists_a \nabla_b \Phi$.

RKconj

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

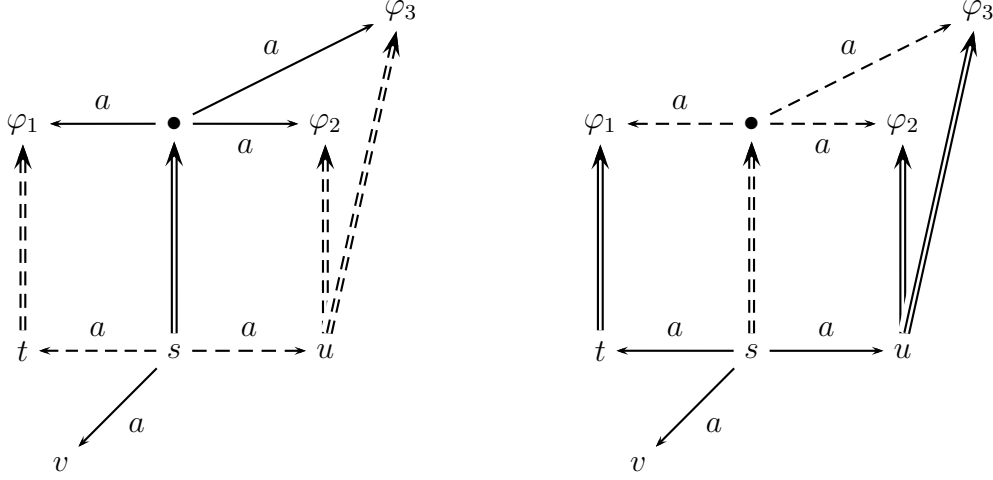


Figure 5: The interaction between refinement and modality involved in axiom **RK**.

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 **RKmulti** 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 Ψ 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, it follows that $N_t \models \bigwedge_{b \in B} \nabla_b \Phi^b$.

NecR

If φ is a validity, then it is satisfied by every model, so for any model M_s , φ 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 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 φ_2 and φ_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

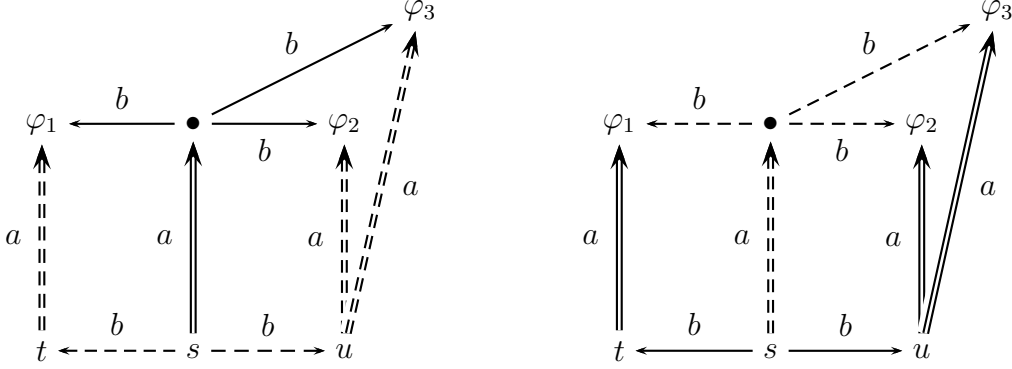


Figure 6: The interaction between refinement and modality involved in axiom **RKmulti**.

forth. Therefore, on the right side of the equivalence in axiom **RK** we only have $\bigwedge \diamond_a \exists_a \Phi$ and we cannot guarantee that $\Box_a \bigvee \exists_a \Phi$ 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 ∇_b and \exists_a simply commute. This in contrast to the axiom **RK** where on the right-hand side a construct $\Box_a \bigvee \exists_a \Phi$ is ‘missing’, so to speak. If it had been $\Box_a \bigvee \exists_a \Phi \wedge \bigwedge \diamond_a \exists_a \Phi$, then we would have had $\nabla_a \exists_a \Phi$, as in **RKmulti** but with $a = b$.

The axioms **RK** and **RKmulti** are different, because in an a -refinement the condition **forth** is not required, whereas for other agents b **forth** is required. Given some refinement wherein we have a cover of Φ , so that at least one of Φ 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 $\bigwedge \diamond_a \exists_a \Phi$, but we do not have $\Box_a \bigvee \exists_a \Phi$. 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

In these examples we also use ‘substitution of equivalents’, see Proposition 32, ahead.

Example 29 $\vdash \diamond_a \top \rightarrow \exists_a (\diamond_a \top \wedge (\Box_a p \vee \Box_a \neg p))$ \dashv

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 out the truth about p : if true, announce p to the agent (and announcement is a model restriction, and therefore a refinement), after which p is known by the agent to be true, 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 suffices to derive the equivalent $\diamond_a \top \rightarrow \exists_a(\nabla_a\{p\} \vee \nabla_a\{\neg p\})$. In some cases several deductions have been combined into single statements, but this is restricted to cases of well-known modal theorems.

$\vdash \diamond_a \top \leftrightarrow \diamond_a(p \vee \neg p)$	Prop, NecK, K
$\vdash \diamond_a(p \vee \neg p) \leftrightarrow (\diamond_a p \vee \diamond_a \neg p)$	Prop, NecK, K
$\vdash \diamond_a p \rightarrow \exists_a \nabla_a\{p\}$	See below
$\vdash \diamond_a \neg p \rightarrow \exists_a \nabla_a\{\neg p\}$	See below
$\vdash \diamond_a p \rightarrow \exists_a(\nabla_a\{p\} \vee \nabla_a\{\neg p\})$	Prop, NecR, R
$\vdash \diamond_a \neg p \rightarrow \exists_a(\nabla_a\{p\} \vee \nabla_a\{\neg p\})$	Prop, NecR, R
$\vdash \diamond_a \top \rightarrow \exists_a(\nabla_a\{p\} \vee \nabla_a\{\neg p\})$	Prop, MP

Lines 3 and 4 of the derivation require the following derivation, where φ is a propositional formula (i.e., $\varphi \in \mathcal{L}_0$).

$\vdash \varphi \leftrightarrow \exists_a \varphi$	Proposition 34, ahead
$\vdash \diamond_a \varphi \leftrightarrow \diamond_a \exists_a \varphi$	Prop, NecK, K
$\vdash \diamond_a \varphi \leftrightarrow \exists_a \nabla_a\{\varphi\}$	RK [$\Phi = \{\varphi\}$]

Example 30 $\vdash (\diamond_a p \wedge \diamond_b p \wedge \diamond_a \neg p \wedge \diamond_b \neg p) \rightarrow \exists_a(\Box_a p \wedge \neg \Box_b p)$ ⊢

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 . This theorem formalizes that. In the following, let φ be $(\diamond_a p \wedge \diamond_b p \wedge \diamond_a \neg p \wedge \diamond_b \neg p)$.

$\vdash \varphi \rightarrow \diamond_a p \wedge \diamond_b \neg p$	Prop
$\vdash \varphi \rightarrow \diamond_a p \wedge \nabla_b\{\neg p, \top\}$	Definition of ∇
$\vdash \varphi \rightarrow \diamond_a \neg \neg p \wedge \nabla_b\{\neg \neg \neg p, \neg \neg \top\}$	Prop
$\vdash \varphi \rightarrow \diamond_a \neg \forall_a \neg p \wedge \nabla_b\{\neg \forall_a \neg \neg p, \neg \forall_a \neg \top\}$	RProp
$\vdash \varphi \rightarrow \diamond_a \exists_a p \wedge \nabla_b\{\exists_a \neg p, \exists_a \top\}$	Definition of \exists
$\vdash \varphi \rightarrow \exists_a \nabla_a\{p\} \wedge \nabla_b\{\exists_a \neg p, \exists_a \top\}$	RK
$\vdash \varphi \rightarrow \exists_a \nabla_a\{p\} \wedge \exists_a \nabla_b\{\neg p, \top\}$	RKmulti
$\vdash \varphi \rightarrow \exists_a(\nabla_a\{p\} \wedge \nabla_b\{\neg p, \top\})$	RKconj
$\vdash \varphi \rightarrow \exists_a(\Box_a p \wedge \diamond_a p \wedge \diamond_b \neg p)$	Definition of ∇
$\vdash \varphi \rightarrow \exists_a(\Box_a p \wedge \diamond_b \neg p)$	Prop
$\vdash \varphi \rightarrow \exists_a(\Box_a p \wedge \neg \Box_b p)$	Definition of \diamond

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 **RML** we can observe that all axioms involving refinement operators \exists are equivalences, except for **R**; however, $\exists_a(\varphi \vee \psi) \leftrightarrow \exists_a\varphi \vee \exists_a\psi$ 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 φ 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 φ 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 **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 results, after which we apply the reduction argument and demonstrate completeness.

Definition 31 (Substitution of equivalents) *An axiomatization satisfies substitution of equivalents if the following holds. Let $\varphi_1, \varphi_2, \varphi_3 \in \mathcal{L}$ and $p \in P$. If $\vdash \varphi_1 \leftrightarrow \varphi_2$ then $\vdash \varphi_3[\varphi_2 \setminus p] \leftrightarrow \varphi_3[\varphi_1 \setminus p]$.* \dashv

Proposition 32 *The axiomatization **RML** satisfies substitution of equivalents.* \dashv

Proof This can be shown by induction on φ_3 . All cases are standard. The case $\Box_a\varphi$ is shown by using an inductive hypothesis $\vdash \varphi[\varphi_2 \setminus p] \leftrightarrow \varphi[\varphi_1 \setminus p]$ and then successively applying **NecK**, **K**, and some elementary tautologies and applications of **MP**. (The required pattern is: from $\vdash x \rightarrow y$, to $\vdash \Box(x \rightarrow y)$, to $\vdash \Box x \rightarrow \Box y$. Then, similarly, for the other direction of the equivalence $x \leftrightarrow y$. Then, some more propositional steps to wind it up.) Whereas the case $\forall_a\varphi$ is shown with the same inductive hypothesis but applying **NecR** and **R** instead of **NecK** and **K**. \square

Proposition 33

1. $\vdash \forall_a(\varphi \wedge \psi) \leftrightarrow \forall_a\varphi \wedge \forall_a\psi$
2. $\vdash \exists_a(\varphi \vee \psi) \leftrightarrow \exists_a\varphi \vee \exists_a\psi$
3. $\vdash \exists_a(\varphi \wedge \psi) \rightarrow \exists_a\varphi \wedge \exists_a\psi$ \dashv

Proof Item 1. can be easily derived from **R**, **NecR** and **MP**, similarly to the way that in modal logic we derive $\vdash \Box(\varphi \wedge \psi) \leftrightarrow \Box\varphi \wedge \Box\psi$. Item 2. is the dual of item 1. and requires mere propositional reasoning. Item 3. can be derived using the tautologies $\varphi \wedge \psi \rightarrow \varphi$ and $\varphi \wedge \psi \rightarrow \psi$, respectively, propositional reasoning, and **R**. (Alternatively, for Item 3., we can think of deriving its dual, with the crucial steps in the derivation that $\varphi \rightarrow \varphi \vee \psi$ is a tautology, from which with **R** and **MP** we get $\forall\varphi \rightarrow \forall(\varphi \vee \psi)$.) \square

Proposition 34

1. $\vdash \forall_a \varphi \leftrightarrow \varphi$ for all propositional φ .
2. $\vdash \exists_a \varphi \leftrightarrow \varphi$ for all propositional φ .

Proof We show $\vdash \forall_a \varphi \leftrightarrow \varphi$ for all propositional φ .⁵ The proof of $\vdash \exists_a \varphi \leftrightarrow \varphi$ for all propositional φ is similar. For convenience in the proof we omit the agent label and write \forall .

We first show $\vdash \varphi \rightarrow \forall \varphi$. Assume that φ is in disjunctive normal form (i.e., for propositional logic, different from the *disjunctive form*, *df*, often used in this work). Formula φ therefore has the form $\bigvee_{\gamma \in \Gamma} \gamma$, where each formula γ is a conjunction of atoms or their negation, for which we write, slightly abusing the language, $\gamma = \bigwedge_{p \in \gamma} \bar{p}$ — where $\bar{p} = p$ if p is a conjunct of γ and $\bar{p} = \neg p$ if $\neg p$ is a conjunct of γ . We now get the following. We omit trivial steps of chaining implications and applying **MP**. For readability we assume the ‘ $\varphi \rightarrow$ ’ part in some derived formulas.

$$\begin{array}{ll}
\vdash \varphi & \rightarrow \bigvee_{\gamma \in \Gamma} \bigwedge_{p \in \gamma} \bar{p} & \text{DNF of } \varphi, \mathbf{Prop} \\
\vdash \dots & \bigvee_{\gamma \in \Gamma} \bigwedge_{p \in \gamma} \forall \bar{p} & \mathbf{RProp} \\
\vdash \dots & \bigvee_{\gamma \in \Gamma} \forall \bigwedge_{p \in \gamma} \bar{p} & \mathbf{R}, \mathbf{NecR}, \text{ and Prop. 33.1 } (\forall(\varphi \wedge \psi) \leftrightarrow \forall \varphi \wedge \forall \psi) \\
\vdash \dots & \forall \bigvee_{\gamma \in \Gamma} \bigwedge_{p \in \gamma} \bar{p} & \mathbf{R}, \mathbf{NecR}, \text{ and tautology } \varphi \rightarrow \varphi \vee \psi \\
\vdash \varphi & \rightarrow \forall \varphi & \text{DNF of } \varphi
\end{array}$$

For the converse direction we convert φ to the conjunctive normal form for propositional formulas, i.e., φ is equivalent to $\bigwedge_{\gamma \in \Gamma} \bigvee_{p \in \gamma} \bar{p}$ (where we now write $\bar{p} = p$ if p is a *disjunct* — not conjunct — of γ and $\bar{p} = \neg p$ if $\neg p$ is a disjunct of γ).

$$\begin{array}{ll}
\vdash \forall \varphi & \rightarrow \forall \bigwedge_{\gamma \in \Gamma} \bigvee_{p \in \gamma} \bar{p} & \text{CNF of } \varphi, \mathbf{Prop}, \mathbf{NecR}, \mathbf{R} \\
\vdash \dots & \bigwedge_{\gamma \in \Gamma} \forall \bigvee_{p \in \gamma} \bar{p} & \text{Prop. 33.1} \\
\vdash \dots & \bigwedge_{\gamma \in \Gamma} \bigvee_{p \in \gamma} \bar{p} & * \\
\vdash \forall \varphi & \rightarrow \varphi & \text{CNF of } \varphi
\end{array}$$

We show why $*$ holds by outlining the method and giving an example: write the conjunct $\bigvee_{p \in \gamma} \bar{p}$ in implicative fashion, e.g., instead of $p \vee q \vee \neg r \vee s$ we write $\neg p \rightarrow \neg q \rightarrow r \rightarrow s$. Then, applying **NecR** and **R** and **MP** repeatedly, we get first $\forall(\neg p \rightarrow \neg q \rightarrow r \rightarrow s)$ and then $\forall \neg p \rightarrow \forall \neg q \rightarrow \forall r \rightarrow \forall s$. Then, applying **RProp**, we get $\neg p \rightarrow \neg q \rightarrow r \rightarrow s$, in other words, we have $p \vee q \vee \neg r \vee s$ back. \square

Proposition 35 $\vdash (\varphi \wedge \exists_a \psi) \leftrightarrow \exists_a(\varphi \wedge \psi)$ for all propositional φ (and any $\psi \in \mathcal{L}_\forall$). \dashv

⁵Of course we do not have for all $\varphi \in \mathcal{L}_\forall$ that $\vdash \forall_a \varphi \leftrightarrow \varphi$. But we then still have $\vdash \forall \varphi \rightarrow \varphi$, or, dually, $\vdash \varphi \rightarrow \exists \varphi$. This can be easily shown by induction on the disjunctive form structure of a formula.

Proof Proposition 33 demonstrated that $\exists_a(\varphi \wedge \psi) \rightarrow \exists_a\varphi \wedge \exists_a\psi$ from which, using Proposition 34.2, also follows $\varphi \wedge \exists_a\psi$. For the other direction we first derive $(\forall_a\varphi \wedge \exists_a\psi) \rightarrow \exists_a(\varphi \wedge \psi)$ by propositional means and applications of **Nec** and **R**. This goes as follows. For convenience of applying the available axioms, instead of $(\forall_a\varphi \wedge \exists_a\psi) \rightarrow \exists_a(\varphi \wedge \psi)$ use the equivalent form $\forall_a\neg(\varphi \wedge \psi) \rightarrow \forall_a\varphi \rightarrow \forall_a\neg\psi$. Now we observe that $\neg(\varphi \wedge \psi) \rightarrow \varphi \rightarrow \neg\psi$ is a tautology and therefore derivable, applying **NecR** gets us $\forall_a(\neg(\varphi \wedge \psi) \rightarrow \varphi \rightarrow \neg\psi)$ and successively applying **R** gets us $\forall_a\neg(\varphi \wedge \psi) \rightarrow \forall_a\varphi \rightarrow \forall_a\neg\psi$. Then, finally, we use that $\forall_a\varphi \leftrightarrow \varphi$ (Proposition 34.1) and thus get $(\varphi \wedge \exists_a\psi) \rightarrow \exists_a(\varphi \wedge \psi)$. \square

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

Proposition 36 *Every formula of \mathcal{L}_\forall is logically equivalent to a formula of \mathcal{L} .* \dashv

Proof Given a formula $\psi \in \mathcal{L}_\forall$, we prove by induction on the number of the occurrences of \exists_a in ψ (for any $a \in A$) that it is equivalent to an \exists_a -free formula, and therefore to a formula $\varphi \in \mathcal{L}$, the standard modal logic. The base is trivial. Now assume ψ 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 ψ , where φ is \exists_b -free for any $b \in A$ (i.e. choose an innermost \exists_a). Let φ' be a disjunctive formula that is equivalent to φ . We prove by induction on the structure of φ' that $\exists_a\varphi'$ is logically equivalent to a formula χ without \exists_a . There are two cases:

- $\exists_a(\varphi \vee \psi)$;
- $\exists_a(\varphi_0 \wedge \bigwedge_{b \in B} \nabla_b \Phi^b)$ where φ_0 is propositional, $B \subseteq A$, and each Φ^b a set of *dfs*.

In the first case, apply Proposition 33.2, we get $\exists_a\varphi \vee \exists_a\psi$, and then apply induction. In the second case, if $B = \emptyset$ we use that $\exists_a\varphi_0 \leftrightarrow \varphi_0$ (Proposition 34.2). If $B \neq \emptyset$, then from Proposition 35 follows that this is equivalent to $\varphi_0 \wedge \exists_a \bigwedge_{b \in B} \nabla_b \Phi^b$, 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 χ that is equivalent to $\exists\varphi$. Replacing $\exists\varphi'$ by χ in ψ gives a result with one less \exists -operator, to which the (original) induction hypothesis applies. \square

Proposition 37 *Let $\varphi \in \mathcal{L}_\forall$ be given and $\psi \in \mathcal{L}$ be equivalent to φ . If ψ is a theorem in **K**, then φ is a theorem in **RML**.* \dashv

Proof Given a $\varphi \in \mathcal{L}_\forall$, Proposition 36 gives us an equivalent $\psi \in \mathcal{L}$. Assume that ψ is a theorem in **K**. We can extend the derivation of ψ to a derivation of φ by observing that all steps used in Proposition 36 are not merely logical but also provable equivalences — where we also apply Proposition 32 of substitution of equivalents. \square

Theorem 38 *The axiom schema **RML** is sound and complete for the logic RML.* \dashv

Proof The soundness proof is given in Theorem 28, so we are left to show completeness. Suppose that $\varphi \in \mathcal{L}_V$ is valid: $\models \varphi$. Applying Lemma 36 we know that there is some equivalent formula $\psi \in \mathcal{L}$, i.e., not containing any refinement operator. As φ is valid, from that and the validity $\varphi \leftrightarrow \psi$ it follows that ψ 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 ψ is derivable, i.e. it is a theorem. From Proposition 37 it follows that φ is a theorem. \square

5.4 The single-agent case

The axiomatization for the single-agent case is the unlabelled version of **RML**, minus the axioms **RKmulti** and **RKconj**.⁶ The single-agent axiomatization was presented in [56]. The completeness proof there is (slightly) different from the multi-agent case of the proof here. In [56] it is used that every refinement modal logical formula is equivalent to a formula in cover logic with the special syntax $\varphi ::= \perp \mid \top \mid \varphi \vee \varphi \mid p \wedge \varphi \mid \neg p \wedge \varphi \mid \nabla\{\varphi, \dots, \varphi\}$ [10, 34], 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 \mathcal{C} , the semantic interpretation of \forall is given by:

$$M_s \models \forall_a \varphi \text{ iff for all } M'_{s'} \in \mathcal{C} : M_s \succeq_a M'_{s'} \text{ implies } M'_{s'} \models \varphi.$$

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

$$\mathbf{RK} \quad \exists \nabla \Phi \leftrightarrow \bigwedge \diamond \exists \Phi.$$

⁶It is clear that axiom **RKmulti** is not needed in the single-agent case, as this is for different agents. But axiom **RKconj** is also not necessary in the single-agent case. We recall that $\nabla_a \Phi \wedge \nabla_a \Psi$ is equivalent to $\nabla_a((\Phi \wedge \Psi) \cup (\Psi \wedge \Phi))$, see page 6. So, we can assume that there are no conjunctions of cover formulas in the single-agent case.

For example, in $\mathcal{S5}$ we have that $\exists\nabla\{\Box p, \neg\Box p\}$ is inconsistent, but that $\Diamond\exists\Box p \wedge \Diamond\exists\neg\Box p$ 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:

$$\mathbf{RS5} \quad \exists\nabla\Phi \leftrightarrow (\bigvee\Phi \wedge \bigwedge\Diamond\Phi),$$

and, for $\Phi \neq \emptyset$,

$$\mathbf{RKD45} \quad \exists\nabla\Phi \leftrightarrow \bigwedge\Diamond\Phi,$$

where Φ 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 [27].

A study of how various classes of models affect the properties of bisimulation quantified logics is given in [22]. Refinement epistemic logics are investigated in [27, 25]. In [25] a multi-agent *KD45* axiomatization is also reported. (For multi-agent *S5*, see ‘Recent results’ in Section 8.)

6 Axiomatization \mathbf{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 (single-agent) axiomatization **RML** for refinement modal logic.

We recall the definition of modal μ -calculus in the technical introductory Section 2. In [22, 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 $\mathcal{L}_{\tilde{\forall}\blacksquare}$ be the language of bisimulation quantified modal logic with \blacksquare as well. First, observe that this impacts the semantics of bisimulation quantification. For two models to be bisimilar, it must now also be the case that every state in one model is bisimilar to a state in the other.

We can inductively define a truth-preserving translation $t : \mathcal{L}^\mu \mapsto \mathcal{L}_{\tilde{\forall}\blacksquare}$. The crucial clauses are those for the fixed-point operators. The atoms p introduced in the translation are required not to occur in φ .

$$\begin{aligned} t(\nu x.\varphi) &\text{ is equivalent to } \tilde{\exists}p(p \wedge \blacksquare(p \rightarrow t(\varphi[p\backslash x]))) \\ t(\mu x.\varphi) &\text{ is equivalent to } \tilde{\forall}p(\blacksquare(t(\varphi[p\backslash x]) \rightarrow p) \rightarrow p) \end{aligned}$$

The first equation captures the intuition of a greatest fixed point as a least upper bound of the set of states that are postfixes of φ , whereas the second equation captures a least fixed point as the greatest lower bound of the set of states that are prefixes of φ . From [15] we know that bisimulation quantifiers are also expressible in the modal μ -calculus, and thus these equivalences also hold in the modal μ -calculus.

Having these tools for modal μ -calculus at our disposal, let us now apply them in *refinement* modal μ -calculus. In order to demonstrate the soundness of the axiomatization defined below, we need to expand the relativization $\bullet^p : \mathcal{L}_{\check{\nu}} \rightarrow \mathcal{L}_{\check{\nu}}$ (Definition 20), single-agent version, to a version $\bullet^p : \mathcal{L}_{\check{\nu}\blacksquare} \rightarrow \mathcal{L}_{\check{\nu}\blacksquare}$ by including a clause for the universal modality:

$$(\blacksquare\varphi)^p = \blacksquare\varphi^p$$

Employing that expanded relativization we can expand the translation $t : \mathcal{L}_{\check{\nu}} \rightarrow \mathcal{L}_{\check{\nu}}$ (Definition 23) to a translation

$$t : \mathcal{L}_{\check{\nu}}^{\mu} \rightarrow \mathcal{L}_{\check{\nu}\blacksquare}^{\mu}$$

by adding the two clauses above for fixed points (this explains why we also wrote $t(\bullet)$ there). This translation t remains truth-preserving (due to Proposition 25 and [22, Lemma 2.43]). We recall the crucial interaction of the translation and the relativization, namely that $t(\exists\varphi)$ is equivalent to $\exists p t(\varphi)^p$. The translation plays an important role in the soundness proof: axioms are shown to be sound by showing that their translations are valid.

Definition 39 (axiomatization \mathbf{RML}^{μ}) *The axiomatization \mathbf{RML}^{μ} is a substitution schema of the (single-agent) axioms and rules of \mathbf{RML} along with the axiom and rule for the modal μ -calculus:*

$$\begin{aligned} \mathbf{F1} \quad & \varphi[\mu x.\varphi \setminus x] \rightarrow \mu x.\varphi \\ \mathbf{F2} \quad & \text{From } \varphi[\psi \setminus x] \rightarrow \psi \text{ infer } \mu x.\varphi \rightarrow \psi \end{aligned}$$

and two new interaction axioms:

$$\begin{aligned} \mathbf{R}^{\mu} \quad & \forall \mu x.\varphi \leftrightarrow \mu x.\forall\varphi \text{ where } \varphi \text{ is a } df \\ \mathbf{R}^{\nu} \quad & \forall \nu x.\varphi \leftrightarrow \nu x.\forall\varphi \text{ where } \varphi \text{ is a } df \end{aligned}$$

For single-agent \mathbf{RML} , see Definition 27 and Section 5.4. We recall that single-agent \mathbf{RML} does not contain the axioms $\mathbf{RKmulti}$ and \mathbf{RKconj} .

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 $\mathbf{F1}$ and $\mathbf{F2}$ are well known [5], so we are left to show that \mathbf{R}^{μ} and \mathbf{R}^{ν} are sound. In the proof we use the characterization of refinement quantification in terms of bisimulation quantification and relativization that was established in Proposition 25. We will also use the characterization of both fixed points in terms of bisimulation quantification as in the previous subsection.

Theorem 40 *The axioms \mathbf{R}^{μ} and \mathbf{R}^{ν} are sound.* —

Proof The proof consists of two cases, \mathbf{R}^μ and \mathbf{R}^ν .

Case \mathbf{R}^μ

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 $\varphi \in \mathcal{L}_V$:

$$\begin{aligned}
t(\exists \nu x. \varphi) &\Leftrightarrow \tilde{\exists} p \, t(\nu x. \varphi)^p \\
&\Leftrightarrow \tilde{\exists} p (\tilde{\exists} q (q \wedge \blacksquare (q \rightarrow t(\varphi[q \setminus x])))^p \\
&\Leftrightarrow \tilde{\exists} p \tilde{\exists} q (q \wedge (\blacksquare (q \rightarrow t(\varphi[q \setminus x])))^p) \\
&\Leftrightarrow \tilde{\exists} p \tilde{\exists} q (q \wedge \blacksquare (q \rightarrow t(\varphi[q \setminus x])^p)) \\
&\Leftrightarrow \tilde{\exists} q \tilde{\exists} p (q \wedge \blacksquare (q \rightarrow t(\varphi[q \setminus x])^p)) \\
&\Leftrightarrow \tilde{\exists} q (q \wedge \tilde{\exists} p \blacksquare (q \rightarrow t(\varphi[q \setminus x])^p)) \\
&\Rightarrow \tilde{\exists} q (q \wedge \blacksquare \tilde{\exists} p (q \rightarrow t(\varphi[q \setminus x])^p)) \quad (*) \\
&\Leftrightarrow \tilde{\exists} q (q \wedge \blacksquare (q \rightarrow \tilde{\exists} p \, t(\varphi[q \setminus x])^p)) \\
&\Leftrightarrow \tilde{\exists} q (q \wedge \blacksquare (q \rightarrow t(\exists \varphi[q \setminus x]))) \\
&\Leftrightarrow t(\nu x. \exists \varphi)
\end{aligned}$$

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

We may assume w.l.o.g. that disjunctive formula $\nu x. \varphi$ contains no free variables, i.e., φ is (also) a disjunctive formula with only the free variable x . We recall that in a disjunctive formula, a conjunction can only be between a purely propositional part and a cover modality part, and that fixed-point variables are not allowed in the propositional part (see Section 2). Importantly, this means that propositional variable q (witnessing fixed-point variable x), that occurs in the formula $(\varphi[q \setminus x])^p$, can only appear in a conjunction if it appears in the scope of a cover operator within that conjunction. This has the following significant consequence:

If $M_s \models \varphi[q \setminus x]$, where φ is a disjunctive formula, then there is a model $N_u \simeq^q M_s$ such that $N_u^ \models \varphi[q \setminus x]$ where N_u^* is the restriction of N_u to states that are not successors of q states.*

That is, whether or not N_u satisfies $\varphi[q \setminus x]$ is invariant to any successors of states in $V^{N^*}(q)$.⁷ To see this, we note that a disjunctive formula $\varphi[q \setminus x]$ is true at M_s only if there is some pointed model N_u that is bisimilar to M_s , and some minimal relation ρ between the states of S^{N_u} and subformulas of $\varphi[q \setminus x]$ such that:

1. $u \rho \varphi[q \setminus x]$;

⁷Throughout this proof we will assume that all models are trees or forests (i.e. every state has at most one predecessor). As every model is bisimilar to a tree, and \mathcal{L}_V^μ and $\mathcal{L}_{\tilde{\forall}\blacksquare}$ are bisimulation invariant, this will not affect the validity of the presented argument.

2. if $v \rho (\psi_1 \vee \psi_2)$, then either $v \rho \psi_1$ or $v \rho \psi_2$ but not both;
3. if $v \rho (\chi \wedge \nabla\Phi)$, then $N_v \models \chi$ and for every successor v' of v there is a unique $\psi \in \Phi$ such that $v' \rho \psi$, and for every $\psi \in \Phi$, there is at least one successor v' of v where $v' \rho \psi$;
4. if $v \rho \sigma y.\psi$, then $v \rho \psi[\sigma y.\psi \setminus y]$, where $\sigma \in \{\nu, \mu\}$.

It is clear that such a relation exists when $N_u \models \varphi[q \setminus x]$. As q is replacing the fixed-point variable x , the minimality of ρ guarantees that if $v \rho q$, then there is no formula $(\chi \wedge \nabla\Phi)$ such that $v \rho (\chi \wedge \nabla\Phi)$. To see this we consider how the formulas q and $\chi \wedge \nabla\Phi$ may have appeared in $\varphi[q \setminus x]$, and specifically what is their smallest common super-formula: if it is a disjunction, then only the disjunct containing q will be considered; and if it is a cover operator, $\nabla\Phi$, we may assume that each disjunctive formula in Φ is related to a unique successor state in N . The smallest common super-formula can not be a fixed-point formula. Consequently, if $v \rho p$ the successors of v do not impact the existence of the relation ρ , and thus do not affect whether or not $N_u \models \varphi[q \setminus x]$.

An explicit construction for N_u can be given via the tableaux of Janin and Walukiewicz [30]. Using their tableaux [30, Def. 3.1], the concept of a marking [30, Def. 3.6] can be adapted to give the required model, N_u . This construction is important for the proof now to follow.

Suppose M_s is any countable model such that $M_s \models \exists q(q \wedge \blacksquare \exists p(q \rightarrow t(\varphi[q \setminus x])^p))$, where φ is a *df*. We would like to build some model M_u^ω such that

- $M_u^\omega \simeq^{p,q} M_s$,
- $M_u^\omega \models q \wedge \blacksquare(q \rightarrow t(\varphi[q \setminus x])^p)$

We inductively build a sequence of (pointed) models $M_u^i = (S^i, R^i, V^i, u)$ such that $M_u^i \simeq^{p,q} M_s$, and furthermore, the models M^i are fixed up to a given set of states.

Definition 41 *Suppose that $M_s = (S, R, V, s)$ is a pointed tree-like model (so for each $t \in S$, there is at most one $t' \in S$ such that $(t', t) \in R$). Let $T \subseteq S - \{s\}$. The model M_s up to T (written $M_s \uparrow T$) is the model (S', R', V, s) where S' is the set of states that are not proper descendants of T and $R' = R \cap (S' \times S')$. \dashv*

Effectively, the model $M_s \uparrow T$ is the model M_s with all the successors of any state in T removed. For each i there will be a set of states $T^i \subset S^i$ such that for all $j > i$, $M^i \uparrow T^i = M^j \uparrow T^i$.

This means we are able to give a well-defined limit for this sequence. At each point of the induction, T^i will represent a frontier of states in the model where we require $q \wedge t(\varphi[q \setminus x])^p$ to be true. Because we are working with disjunctive formulas, we can change the submodels rooted at states in T^i , without affecting the interpretation of $t(\varphi[q \setminus x])^p$ in other parts of the model. This way we are able to find a single model M_s^ω with the required properties.

We now define the sequence of models M^i . For each i we define a model and a set of states $T^i \subseteq S^i$ on which we will extend the construction. The proposition to be shown by inductive proof is

$$\begin{aligned} M_u^i &\simeq^{p,q} M_s \\ \forall u' \in T^i, M_{u'}^i &\models \tilde{\exists}q(q \wedge \blacksquare \tilde{\exists}p(q \rightarrow t(\varphi[q \setminus x]^p))), \\ \forall j < i, \forall u' \in T^j, M_{u'}^i &\models q \wedge t(\varphi[q \setminus x]^p), \text{ and} \\ \forall j < i, N^i \uparrow T^j &= N^j \uparrow T^j. \end{aligned}$$

To define the base case, it is sufficient to let $M^0 = M$ and $T^0 = \{s\}$. It is clear that the induction hypothesis holds here. Now, for the inductive step, assume that the proposition holds for i . For each $u \in T^i$, we have

$$M_u^i \models \tilde{\exists}q(q \wedge \blacksquare \tilde{\exists}p(q \rightarrow t(\varphi[q \setminus x]^p))).$$

Hence, for each $u \in T^i$, there is some (tree-like) $N_{v^u}^u \simeq^{p,q} M_u^i$ such that

$$N_{v^u}^u \models q \wedge t(\varphi[q \setminus x]^p) \wedge \blacksquare \tilde{\exists}p(q \rightarrow t(\varphi[q \setminus x]^p)).$$

We will assume w.l.o.g. that all models N^u for $u \in T^i$ and M^i have disjoint sets of states. As φ is a disjunctive formula, we may further assume that $N_{v^u}^u \models t(\varphi[q \setminus x]^p)$ is invariant to any successors of $V^{N^u}(q) \setminus \{v^u\}$. This allows us (as the induction proceeds) to replace the submodels rooted at v^u without affecting whether $t(\varphi[q \setminus x]^p)$ is satisfied in other parts of the model.

We now append the models $N_{v^u}^u$ (for $u \in T^i$) to the model M^i . Formally, let $M' = M^i \uparrow T^i = (S', R', V')$, and $T' = T^i \cap S'$. Then

$$\begin{aligned} S^{i+1} &= S' \cup \bigcup_{u \in T'} (S^{N^u} - \{v^u\}); \\ R^{i+1} &= R' \cup \bigcup_{u \in T'} (R^{N^u} - \{(v^u, v) \mid v \in S^{N^u}\}) \cup \{(u, v) \mid u \in T', v^u R^{N^u} v\}; \\ \text{for all } r : V^{i+1}(r) &= V'(r) \bigcup_{u \in T'} (V^{N^u}(r) - \{v^u\}). \end{aligned}$$

Finally, we let $T^{i+1} = \bigcup_{u \in T'} (V^{N^u}(q) - \{v^u\})$.

We can see that the proposition to be shown holds for $i+1$ as follows:

- $M_s^{i+1} \simeq^{p,q} M_s$ since, for all $u \in T^i$, $M_u^i \simeq^{p,q} N_{v^u}^u$, and $M_s^i \simeq^{p,q} M_s$ from the induction hypothesis. A $\{p, q\}$ -bisimulation between M_s^{i+1} and M_s can be constructed by composing these bisimulations.⁸
- $\forall v \in T^{i+1}$, $M_v^{i+1} \models \tilde{\exists}q(q \wedge \blacksquare \tilde{\exists}p(q \rightarrow t(\varphi[q \setminus x]^p)))$, since for all $u \in T^{i+1}$, $u \in V^{i+1}(q)$, and $N_{v^u}^u \models q \wedge \blacksquare \tilde{\exists}p(q \rightarrow t(\varphi[q \setminus x]^p))$.

⁸Specifically, let \mathfrak{R}^u be the $\{p, q\}$ -bisimulation between M_u^i and $N_{v^u}^u$, and \mathfrak{R}^i be the $\{p, q\}$ -bisimulation between M_s^i and M_s . We define the $\{p, q\}$ -bisimulation \mathfrak{R}^{i+1} from M_s^{i+1} to M_s by: for all $t \in S^{i+1}$, for all t' in S , $(t, t') \in \mathfrak{R}^{i+1}$ if and only if either $(t \in S^i \text{ and } (t, t') \in \mathfrak{R}^i)$, or $(t \in S^{N^u}, \text{ and for some } v \in S^i, (t, v) \in \mathfrak{R}^u \text{ and } (v, t') \in \mathfrak{R}^i)$. It is straightforward to check that \mathfrak{R}^{i+1} is a bisimulation.

- $\forall j < i + 1, \forall u \in T^j, M_u^j \models q \wedge t(\varphi[q \setminus x])^q$; by the reasoning presented above, $M_u^j \models t(\varphi[q \setminus x])^q$ is invariant to the successors of the states in T^{j+1} . Therefore, if $M_u^j \models q \wedge t(\varphi[q \setminus x])^q$, then $M_u^{j+1} \models t(\varphi[q \setminus x])^q$.
- $\forall j < i, M^i \uparrow T^j = M^j \uparrow T^j$ follows immediately from the construction.

We now let $M^\omega = (S^\omega, R^\omega, V^\omega)$ where

- $s' \in S^\omega$ iff for some $i, s' \in S^j$ for all $j > i$,
- $uR^\omega v$ iff for some $i, uR^j v$ for all $j > i$,
- $u \in V^\omega(r)$ iff for some $i, u \in V^j(r)$ for all $j > i$,

and let $T^\omega = \emptyset$. It is clear that the limit step will also preserve the induction hypothesis, so we have $M_u^\omega \simeq^{p,q} M_s$ and $M_u^\omega \models q \wedge \blacksquare(q \rightarrow t(\varphi[q \setminus x])^p)$, since by construction $V^\omega(q) = \bigcup_{i < \omega} T^i$. Thus, $M_s \models \exists p \exists q (q \wedge \blacksquare(q \rightarrow t(\varphi[q \setminus x])^p))$ (i.e., $M_s \models \exists \nu x. \varphi$) as required.

The construction is represented in Figure 7.

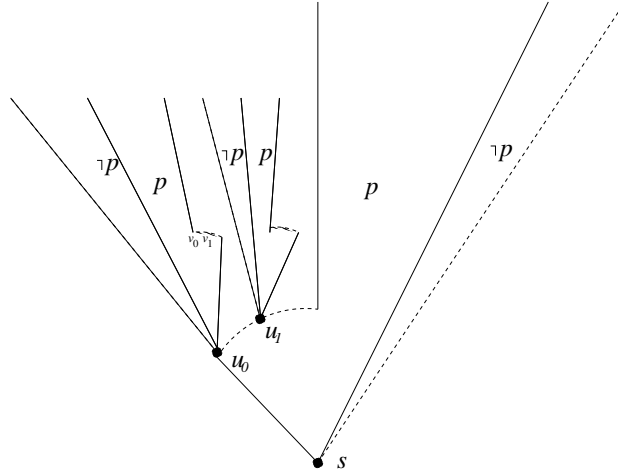


Figure 7: The inductive step for the construction of M^ω . The formula $t(\varphi[q \setminus x])^p$ is independent of any state where p is not true, or any state beyond the frontier defined by u_0, u_1, \dots

Case R^ν

We also use the contrapositive form of the axiom: $\exists \mu x. \varphi \leftrightarrow \mu x. \exists \varphi$. For any $\varphi \in \mathcal{L}_\forall$ we

have that:

$$\begin{aligned}
t(\exists\mu x.\varphi) &\Leftrightarrow \tilde{\exists}p t(\mu x.\varphi)^p \\
&\Leftrightarrow \tilde{\exists}p(\tilde{\forall}q(\blacksquare(t(\varphi[q\backslash x]) \rightarrow q) \rightarrow q))^p \\
&\Leftrightarrow \tilde{\exists}p\tilde{\forall}q(\blacksquare(t(\varphi[q\backslash x])^p \rightarrow q) \rightarrow q) \\
&\Rightarrow \tilde{\forall}q\tilde{\exists}p(\blacksquare(t(\varphi[q\backslash x])^p \rightarrow q) \rightarrow q) & (**) \\
&\Leftrightarrow \tilde{\forall}q\tilde{\exists}p(\blacklozenge(t(\varphi[q\backslash x])^p \wedge \neg q) \vee q) \\
&\Leftrightarrow \tilde{\forall}q(\tilde{\exists}p\blacklozenge(t(\varphi[q\backslash x])^p \wedge \neg q) \vee q) \\
&\Leftrightarrow \tilde{\forall}q(\blacklozenge(\tilde{\exists}p(t(\varphi[q\backslash x])^p \wedge \neg q) \vee q) & (***) \\
&\Leftrightarrow \tilde{\forall}q(\blacklozenge(\tilde{\exists}p t(\varphi[q\backslash x])^p \wedge \neg q) \vee q) \\
&\Leftrightarrow \tilde{\forall}q(\blacksquare(\tilde{\exists}p t(\varphi[q\backslash x])^p \rightarrow q) \rightarrow q) \\
&\Leftrightarrow \tilde{\forall}q(\blacksquare(\exists\varphi[q\backslash x] \rightarrow q) \rightarrow q) \\
&\Leftrightarrow t(\mu x.\exists\varphi)
\end{aligned}$$

The equivalence in (***) is true because \blacklozenge is the existential modality which quantifies over all states 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 $\exists\mu x.\varphi \leftrightarrow \mu x.\exists\varphi$, we now show directly that $\models \mu x.\exists\varphi \rightarrow \exists\mu x.\varphi$ in refinement μ -calculus, for φ a *df* (observe that $\mu x.\varphi$ is then a *df* as well). We use the inductive characterization of $\mu x.\exists\varphi$ of [5] which tells that $M_s \models \mu x.\exists\varphi$ if and only if $s \in \|\exists\varphi\|_\tau$ for some ordinal τ , 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 $M^\tau = M^{[\sigma]}$ with $\sigma = x \mapsto \bigcup_{\tau' < \tau} \|\exists\varphi\|_{\tau'}$.

Suppose $M_s \models \mu x.\exists\varphi$. Since \mathcal{L}_μ^t is bisimulation invariant, without loss of generality we may suppose that M is a countable tree-like model. As M_s satisfies $\mu x.\exists\varphi$, there must be some least ordinal τ whereby $s \in \|\exists\varphi\|_\tau$. We give a proof by induction over τ that $s \in \|\exists\varphi\|_\tau$ implies $M_s \models \exists\mu x.\varphi$. 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 $\mu x.\varphi$ is a *df*, there is a refinement of M^τ 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 φ . Formally, there is a set of states $\{u_0, u_1, \dots\} \in V^\tau(x)$ such that $M'_s \models \exists\varphi$ (i.e., $M'_s \models \tilde{\exists}p t(\varphi)^p$), where $M' = (S', R', V')$ with

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

We note that M'_s is a refinement of M_s^τ . 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 $N_{u_i}^i \preceq M_{u_i}^\tau$ and $N_{u_i}^i \models \mu x.\varphi$. We may append these models to M' , to define $M^* = (S^*, R^*, V^*)$ where $S^* = S' \cup \bigcup_i S^i$, $R^* = R' \cup \bigcup_i R^i \cup \{(t, v_i) \mid (t, u_i) \in R'\}$, and $V^*(y) = V'(y) \cup \bigcup_i V^i(y)$ 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 $M_s^* \models \mu x.\varphi$ as required. \square

The general form of \mathbf{R}^μ 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 \mathbf{R}^ν is not true in the general case, as can be seen by taking $\varphi = p \wedge \square(\diamond \top \rightarrow x)$. 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 \mathbf{RML}^μ proceeds exactly as for Theorem 38, replacing the formulas in cover logic with disjunctive formulas, to get a statement similar to that of Proposition 36.

Proposition 42 *Every formula of \mathcal{L}_\forall^μ is equivalent to a formula of \mathcal{L}^μ .* \dashv

Proof Given a formula ψ , we prove by induction on the number of the occurrences of \exists in ψ that it is equivalent to an \exists -free formula, and therefore to a formula in the modal μ -calculus \mathcal{L}^μ . The base is trivial. Now assume ψ contains $n + 1$ \exists -operators. Choose a subformula of type $\exists\varphi$ of our given formula ψ , where φ is \exists -free (i.e. choose an innermost \exists). As φ is \exists -free, it is semantically equivalent to a formula in disjunctive normal form, and by the completeness of Kozen's axiom system [59] this equivalence is provable in \mathbf{RML}^μ . By **NecR** and **R** it follows that $\exists\varphi$ is provably equivalent to some formula $\exists\psi$ where ψ is a disjunctive formula (analogously to Proposition 32 one can easily show that \mathbf{RML}^μ satisfies substitution of equivalents). Thus without loss of generality, we may assume in the following that φ is in disjunctive normal form. We may now proceed by induction over the complexity of φ , and conclude that $\exists\varphi$ is logically equivalent to a formula χ 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 \mathbf{R}^ν noting that all subformulas of a disjunctive formula are themselves disjunctive); IH.
- $\exists\nu x.\varphi$ iff $\nu x.\exists\varphi$ (by \mathbf{R}^μ); IH.

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

Theorem 43 *The axiom schema \mathbf{RML}^μ is sound and complete for the logic \mathbf{RML}^μ* \dashv

Proof Soundness follows from Theorem 40 and Theorem 28. To see \mathbf{RML}^μ is complete, suppose $\varphi \in \mathcal{L}_\forall^\mu$ is a valid formula. Then by Lemma 42, φ is provably equivalent to some valid formula $\psi \in \mathcal{L}^\mu$. As ψ is valid, it must be provable since **Prop**, **K**, **F1**, **F2**, **NecK**, and **MP** give a sound and complete proof system for the modal μ -calculus [59]. A proof of φ follows by **MP**. \square

7 Complexity

Decidability for both \mathcal{L}_\forall and \mathcal{L}_\forall^μ 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 \mathcal{L} and \mathcal{L}^μ respectively, are recursive and involve transforming formulas into their disjunctive normal forms. Hence they are non-elementary in the size of the original formula. This non-elementary procedure for \mathcal{L}_\forall^μ is optimal as shown in Section 7.1 below.

Unfortunately we were not able to corroborate in this paper the upper complexity claims for RML reported in [56]. But towards some indication of a result in that direction, we further establish a doubly exponential succinctness proof for \mathcal{L}_\forall in Section 7.2. (On complexity, see also ‘Recent results’ in Section 8.)

7.1 RML $^\mu$ is non-elementary

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

Theorem 44 *The satisfiability problem for RML $^\mu$ is non-elementary, even for the single-agent setting.* \dashv

In the rest of this section, we only consider a single-agent setting.

First, we recall a fragment, written CTL^- , of the standard branching-time logic *Computation Tree Logic* (CTL) [14], which in turn is a fragment of \mathcal{L}^μ (see also the example Section 4.2).

$$\text{CTL}^- \ni \varphi ::= \top \mid \perp \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box\varphi \mid \Diamond\varphi \mid \text{EF}\varphi \mid \text{AF}\varphi$$

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

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

Directly translating CTL^- in \mathcal{L}^μ is routine via the following mapping $\tau : \text{CTL}^- \rightarrow \mathcal{L}^\mu$, defined by induction over the formulas: $\tau(\top) = \top$, $\tau(p) = p$, $\tau(\neg\varphi) = \neg\tau(\varphi)$, $\tau(\varphi \wedge \varphi') = \tau(\varphi) \wedge \tau(\varphi')$, $\tau(\Box\varphi) = \Box\tau(\varphi)$, $\tau(\Diamond\varphi) = \Diamond\tau(\varphi)$, $\tau(\text{EF}\varphi) = \mu x. \tau(\varphi) \vee \Diamond x$, $\tau(\text{AF}\varphi) = \mu x. \tau(\varphi) \vee \Box x$.

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 CTL^- formula is in *positive form* if negation is applied only to propositional variables. A CTL^- formula φ is *existential* if it is in positive form and there are no occurrences of the universal modalities AF and AG and the modality \Box . The following can be proved by using Proposition 8, enriched for the case of EF formulas (with a transfinite induction argument for this fixed-point formula).

Proposition 45 *Let M_s and N_t be two models with $M_s \succeq N_t$. Then for each existential CTL⁻ formula φ , $N_t \models \varphi$ implies $M_s \models \varphi$. \dashv*

Definition 46 (Refinement CTL⁻) *Refinement CTL⁻ (CTL _{$\bar{\forall}$} ⁻, for short) is the extension of CTL⁻ with the refinement quantifiers \exists and \forall . \dashv*

Definition 47 (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 χ . Formally, $\ell(\epsilon) = 0$, $\ell(Q) = 0$ for every $Q \in \{\exists, \forall\}^*$, and $\ell(QQ'\chi) = \ell(Q'\chi)$ if $Q = Q'$, $\ell(Q'\chi) + 1$ otherwise.*

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

Theorem 48 *Let the class $\mathcal{C}_k = \{\varphi \in \text{CTL}_{\bar{\forall}}^- \mid \delta(\varphi) \leq k\}$. The satisfiability problem for \mathcal{C}_k is k -EXSPACE-hard, for any k . \dashv*

Theorem 48 is proved by a polynomial-time reduction from satisfiability of *Quantified Propositional Temporal Logic* (QPTL) [50]. First, we recall the syntax and the semantics of QPTL. The syntax of QPTL formulas φ over a countable set P of propositional variables is defined as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathbf{X}\varphi \mid \mathbf{F}\varphi \mid \bar{\exists}p.\varphi$$

where $p \in P$, \mathbf{X} is the ‘next’ modality, \mathbf{F} is the ‘eventually’ modality, and $\bar{\exists}$ is the existential quantifier.⁹ We also use standard abbreviation $\mathbf{G}\varphi$ for $\neg\mathbf{F}\neg\varphi$ (‘always’).

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 φ , 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):

$$\begin{aligned} (w, h) \models p &\text{ iff } p \in w(h) \\ (w, h) \models \mathbf{X}\varphi &\text{ iff } (w, h+1) \models \varphi \\ (w, h) \models \mathbf{F}\varphi &\text{ iff there is } h' \geq h \text{ such that } (w, h') \models \varphi \\ (w, h) \models \bar{\exists}p.\varphi &\text{ iff there is } w', w' \equiv_{P \setminus \{p\}} w \text{ and } (w', h) \models \varphi \end{aligned}$$

⁹We distinguish (domain) quantifiers $\bar{\exists}$ and $\bar{\forall}$ in use here, from the refinement quantifiers \exists and \forall , and from the bisimulation quantifiers $\tilde{\exists}$ and $\tilde{\forall}$.

We say that the word w satisfies φ , written $w \models \varphi$, if $(w, 0) \models \varphi$. A QPTL formula φ is in *positive normal form* if it is of the form $Q_1 p_1. Q_2 p_2. \dots. Q_n p_n. \varphi_{n+1}$, where $Q_j \in \{\exists, \forall\}$ for each $1 \leq j \leq n$, and φ_{n+1} is a quantification-free QPTL-formula in which negation is applied only to propositional variables¹⁰. The *quantifier alternation depth* of $Q_1 p_1. Q_2 p_2. \dots. Q_n p_n. \varphi_{n+1}$ is the number of alternations of (existential and universal) quantifiers in the string $Q_1 Q_2 \dots Q_n$. The following is a well-known result.

Theorem 49 [50] *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.* \dashv

Note that Theorem 49 holds even if we assume that formulas in positive normal form like $Q_1 p_1. Q_2 p_2. \dots. Q_n p_n. \varphi_{n+1}$ (with φ_{n+1} quantification-free) are such that p_1, \dots, p_n are pairwise distinct, each proposition occurring in φ_{n+1} is in $\{p_1, \dots, p_n\}$, and $Q_n = \forall$.

Theorem 48 directly follows from Theorem 49 and the following theorem, whose proof is given in the rest of this section.

Theorem 50 *For every $\varphi \in \text{QPTL}$, one can construct in time polynomial in the size of φ a formula $\tilde{\varphi} \in \text{CTL}^-$, such that φ 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 φ .* \dashv

Before proving Theorem 50, we need additional definitions. Let $P = \{p_1, \dots, p_n\}$ and $\tilde{P} = P \cup \{p_0, \bar{p}_1, \dots, \bar{p}_n\}$, where $p_0, \bar{p}_1, \dots, \bar{p}_n$ are fresh propositional variables (intuitively, \bar{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 $1 \leq i \leq j$, either s' has a p_i -state successor or (exclusive) a \bar{p}_i -state successor, and (iii) for all $j+1 \leq i \leq n$, s' has both a p_i -state successor and a \bar{p}_i -state successor.

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

¹⁰Every QPTL formula is constructively equivalent to a formula in positive normal form, with linear size.

$$\psi_j := p_0 \wedge \mathbf{AG} \left\{ \left[\bigvee_{p \in \tilde{P}} (p \wedge \bigwedge_{p' \in \tilde{P} \setminus \{p\}} \neg p') \right] \wedge [\neg p_0 \rightarrow \Box \perp] \wedge \right. \\ \left. p_0 \rightarrow \left[\Diamond p_0 \wedge \bigwedge_{j+1 \leq i \leq n} (\Diamond p_i \wedge \Diamond \bar{p}_i) \wedge \bigwedge_{1 \leq i \leq j} (\Diamond (p_i \vee \bar{p}_i) \wedge (\Box \neg p_i \vee \Box \neg \bar{p}_i)) \right] \right\}$$

Intuitively, ψ_j enforces the existence of infinite paths $\pi = s_0 s_1 \dots$ which visit only p_0 -states s_i such that the following holds: the set of successors of s_i ‘encodes’ a specific truth valuation of the variables p_1, \dots, p_j and all the possible truth valuations of the variables p_{j+1}, \dots, p_n .

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 $\mathbf{words}(M_s)$, given by: $w \in \mathbf{words}(M_s)$ iff there is an infinite path $\pi = s_0, s_1, \dots$ of M from s (note that π 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 \bar{p}_j -successor.

Note that if M_s is well-formed w.r.t. 0 , then $\mathbf{words}(M_s) = (2^P)^\omega$. 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 $u_j \in (2^{\{p_1, \dots, p_j\}})^\omega$ such that $\mathbf{words}(M_s) = \{w \in (2^P)^\omega \mid \mathbf{proj}(w, \{p_1, \dots, p_j\}) = u_j\}$. In particular, when $j = n$, $\mathbf{words}(M_s)$ is a singleton.

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

Construction of the $\text{CTL}_{\bar{\vee}}^-$ formula $\tilde{\varphi}$ (in Theorem 50). Pick an QPTL formula $\varphi = Q_1 p_1 \cdot Q_2 p_2 \cdot \dots \cdot Q_n p_n \cdot \varphi_{n+1}$. For each $1 \leq j \leq n$, we let $\varphi_j = Q_j p_j \cdot Q_{j+1} p_{j+1} \cdot \dots \cdot Q_n p_n \cdot \varphi_{n+1}$ (note that φ_1 corresponds to φ).

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

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

- $\Upsilon(p) = \Diamond p$ and $\Upsilon(\neg p) = \Diamond \bar{p}$ for each $p \in P$;
- $\Upsilon(\xi_1 \vee \xi_2) = \Upsilon(\xi_1) \vee \Upsilon(\xi_2)$ and $\Upsilon(\xi_1 \wedge \xi_2) = \Upsilon(\xi_1) \wedge \Upsilon(\xi_2)$;
- $\Upsilon(\mathbf{X}\xi) = \Diamond(p_0 \wedge \Upsilon(\xi))$, $\Upsilon(\mathbf{F}\xi) = \mathbf{EF}(p_0 \wedge \Upsilon(\xi))$, and $\Upsilon(\mathbf{G}\xi) = \mathbf{EG}(p_0 \wedge \Upsilon(\xi))$.

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

Induction case ($1 \leq j \leq n$). Recall $\varphi_j = Q_j p_j \cdot \varphi_{j+1}$.

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

Finally, the $\text{CTL}_{\bar{\vee}}^-$ formula $\tilde{\varphi}$ over \tilde{P} is given by $\tilde{\varphi} := \psi_0 \wedge \widetilde{\varphi}_1$.

Correctness of the construction. Note that the size of $\tilde{\varphi}$ is polynomial in the size of φ . Moreover, the refinement quantifier alternation depth of $\tilde{\varphi}$ is equal to the quantifier alternation depth of φ . Thus, in order to prove Theorem 50, it remains to show that φ is satisfiable iff $\tilde{\varphi}$ is satisfiable. For this, we need three preliminary lemmata.

Lemma 51 *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 ξ in positive normal form, $w \models \xi$ if and only if $M_s \models \Upsilon(\xi)$. \dashv*

Proof Let $\pi = s_0, s_1, \dots$ be the unique infinite path of M from state s (note that π 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 ξ , the following holds: for all $h \geq 0$, $M_{s_h} \models \Upsilon(\xi)$ iff $(w, h) \models \xi$. Hence, the result follows. \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_t of M_s which is well-formed w.r.t. h and minimal. Note that for each $w \in \text{words}(M_s)$ and for each $j \leq h \leq n$, by construction, there exists an h -segment N_t of M_s such that $w \in \text{words}(N_t)$.

Lemma 52 *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, $M_s \models \tilde{\varphi}_j$. \dashv*

Proof The proof is by induction on $n - j = 0, \dots, n - 1$.

Base case ($j = n$). Recall $\varphi_n = \bar{\forall} p_n \cdot \varphi_{n+1}$, where φ_{n+1} is a quantification-free QPTL formula in positive normal form. By construction, $\tilde{\varphi}_n = \forall (\psi_n \rightarrow \Upsilon(\varphi_{n+1}))$. Let N_t be a refinement of M_s which satisfies formula ψ_n (if any). We need to show that $N_t \models \Upsilon(\varphi_{n+1})$. By definition of ψ_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, $w \models \varphi_n = \bar{\forall} p_n \cdot \varphi_{n+1}$, which implies $w \models \varphi_{n+1}$. By Lemma 51, 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 45 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) $\varphi_j = \bar{\exists} p_j \cdot \varphi_{j+1}$ and $\tilde{\varphi}_j = \exists (\psi_j \wedge \tilde{\varphi}_{j+1})$: let $w_0 \in \text{words}(M_s)$. By hypothesis, $w_0 \models \varphi_j$. Hence, there is infinite word w'_0 over 2^P such that $w'_0 \equiv_{P \setminus \{p_j\}} w_0$ and $w'_0 \models \varphi_{j+1}$. Since M_s is well-formed w.r.t. $j - 1$ and $w_0 \in \text{words}(M_s)$, it follows that $w'_0 \in \text{words}(M_s)$ as well. Let N_s be any j -segment of M_s such that $w'_0 \in \text{words}(N_s)$. By definition of ψ_j , $N_s \models \psi_j$. Thus, it suffices to show that $N_s \models \tilde{\varphi}_{j+1}$. Since N_s is well-formed w.r.t. j and minimal, and $w'_0 \in \text{words}(N_s)$, it holds that for each $w' \in \text{words}(N_s)$, $w' \equiv_{\{p_1, \dots, p_j\}} w'_0$. Since every proposition in $\{p_{j+1}, \dots, p_n\}$ does not occur free in φ_{j+1} and $w'_0 \models \varphi_{j+1}$, it follows that for each $w' \in \text{words}(N_s)$, $w' \models \varphi_{j+1}$. Thus, by the induction hypothesis, we obtain that $N_s \models \tilde{\varphi}_{j+1}$, and the result holds.

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

Lemma 53 *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 $M_s \models \tilde{\varphi}_j$. Then, there is a $(j-1)$ -segment N_t of M_s such that $N_t \models \tilde{\varphi}_j$ and for each $w \in \mathbf{words}(N_t)$, $w \models \varphi_j$. \dashv*

Proof The proof is by induction on $n - j = 0, \dots, n - 1$, for which there are two cases. Recall that $\varphi_n = \bar{\forall} p_n \cdot \varphi_{n+1}$.

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

- $j = n$ (*base step*): by construction, $\mathbf{words}(N'_u)$ is a singleton, $\tilde{\varphi}_{n+1} = \Upsilon(\varphi_{n+1})$, and φ_{n+1} is a quantification-free QPTL-formula in positive normal form. Since $w' \in \mathbf{words}(N'_u)$ and $N'_u \models \tilde{\varphi}_{n+1}$, by Lemma 51, it follows that $w' \models \varphi_{n+1}$.
- $j \leq n - 1$ (*induction step*): since $w' \in \mathbf{words}(N'_u)$ and $N'_u \models \tilde{\varphi}_{j+1}$, by the induction hypothesis (note that since N'_u is minimal, for each j -segment N'_v of N'_u , $\mathbf{words}(N'_v) = \mathbf{words}(N'_u)$), it follows that $w' \models \varphi_{j+1}$.

Thus, in both cases $w' \models \varphi_{j+1}$. Since w' is an arbitrary infinite word over 2^P such that $w' \equiv_{P \setminus \{p_j\}} w$, we obtain that $w \models \bar{\forall} p_j \cdot \varphi_{j+1} = \varphi_j$, and the result follows.

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

Now, we can prove the correctness of the construction.

Theorem 54 φ is satisfiable if, and only if, $\tilde{\varphi}$ is satisfiable. ←

Proof First, assume that $\tilde{\varphi} = \psi_0 \wedge \tilde{\varphi}_1$ is satisfiable. Hence, there is a pointed model M_s which satisfies both ψ_0 and $\tilde{\varphi}_1$. By definition of formula ψ_0 , it follows that M_s is well-formed w.r.t. 0. Since $M_s \models \tilde{\varphi}_1$, by Lemma 53, we deduce that there is an infinite word w over 2^P such that $w \models \varphi_1$. Since $\varphi = \varphi_1$, it follows that φ is satisfiable.

Now, assume that φ is satisfiable. Since any proposition in P does not occur free in φ , it follows that for each infinite word w over 2^P , $w \models \varphi$. Let M_s be any pointed model which is well-formed w.r.t. 0. By definition of formula ψ_0 , it holds that $M_s \models \psi_0$. Moreover, since $w \models \varphi$ for each $w \in \mathbf{words}(M_s)$, and $\varphi = \varphi_1$, by Lemma 52 it follows that $M_s \models \tilde{\varphi}_1$. Therefore, $M_s \models \psi_0 \wedge \tilde{\varphi}_1 = \tilde{\varphi}$. Hence, $\tilde{\varphi}$ is satisfiable. □

By using Theorem 48 and the fact that there exists a linear time translation of CTL_{\forall}^- into $\mathcal{L}_{\forall}^{\mu}$ (see page 46) we now obtain the required proof of Theorem 44.

7.2 Succinctness

In this section we establish the following result.

Theorem 55 RML is doubly exponentially more succinct than K, and RML^{μ} is doubly exponentially more succinct than modal μ -calculus. ←

Theorem 55 directly follows from the following result whose proof is given in the rest of this section.

Proposition 56 *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 $n \in \mathbb{N}$, φ_n has size $O(n^2)$ and refining quantifier alternation depth 2, and each equivalent one-agent \mathcal{L}^{μ} formula has size at least $2^{2^{\Omega(n)}}$.* ←

Construction of the \mathcal{L}_{\forall} formulas φ_n in Proposition 56: let $P = \{l, r, \#, 0, 1, a, b\}$. An n -configuration is a string on $\{a, b\}$ of length exactly 2^{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 φ_n in such a way that the following holds: a pointed model $M_s \in \mathcal{C}_n$ satisfies φ_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. An n -block is a pair $bl = (c, i)$ such that $c \in \{a, b\}$ and $1 \leq i \leq 2^{2^n}$. We say that c is the *content* of bl and i is the *position* of bl . Intuitively, bl represents the i th symbol of some n -configuration. First, we define an encoding of (c, i) by a set $\mathbf{code}(c, i)$ of strings over 2^P of length $n + 3$. Since $1 \leq i \leq 2^{2^n}$, 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

¹¹Recall that $f(n) \in \Omega(g(n))$ iff $g(n) \in O(f(n))$.

string over $\{0, 1\}$ of length n)¹² of the position j of the j th bit in the binary encoding of i . This leads to the following definition. An n -sub-block is a string over 2^P of length $n + 2$ of the form $sbl = \{\#\}, \{b_1\}, \dots, \{b_n\}, \{B\}$, where $b_1, \dots, b_n, B \in \{0, 1\}$. 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, \dots, b_n . Intuitively, sbl encodes the position and the content B of a bit along the binary encoding of an integer $1 \leq i \leq 2^{2^n}$. 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 an 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 $(V^M)^{-1}(s_0), (V^M)^{-1}(s_1), \dots$ such that s_0, s_1, \dots is a maximal path of M starting from s . A pointed model M_s *encodes an n -block* (c, i) if

$$\text{Traces}(M_s) = \text{code}(c, i) \text{ and } M_s \models \bigwedge_{d=0}^{n-1} \square^d (\diamond 1 \wedge \diamond 0) \in \mathcal{L}$$

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 $(V^M)^{-1}(s_r) = \{r\}$;
- for each $dir \in \{l, r\}$, s_{dir} has 2^{2^n} successors $s_{1,dir}, \dots, s_{2^{2^n},dir}$. Moreover, for each $1 \leq i \leq 2^{2^n}$, $M_{s_{i,dir}}$ encodes the n -block $(c_{i,dir}, i)$, where $c_{i,dir}$ is the i th symbol of the n -configuration w_{dir} .

If additionally $w_l = w_r$, then we say that M_s is *balanced*. 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}_\forall formula φ_n (for each $n \geq 0$), we first show Lemma 57. This lemma asserts that there is an \mathcal{L}_\forall formula ψ_n of size $O(n^2)$ which allows one 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.

Lemma 57 *For each $n \geq 0$, one can construct a one-agent \mathcal{L}_\forall formula ψ_n of size $O(n^2)$ and refinement quantifier alternation 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 ψ_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 an 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 . ⊣

¹²Here, it is not relevant to specify the form of the binary encoding which is used.

Proof The \mathcal{L}_\forall formula ψ_n is defined as follows:

$$\psi_n := \xi_n \wedge \forall(\theta_n \rightarrow \bigvee_{b \in \{0,1\}} \square^{n+3}b)$$

where ξ_n and θ_n are \mathcal{L} formulas defined as follows:

$$\begin{aligned} \xi_n &:= \diamond\top \wedge \square\diamond\top \wedge \bigwedge_{d=0}^{n-1} \square^{d+2}(\diamond 1 \wedge \diamond 0) \wedge \square^{n+2}\diamond\top \wedge \square^{n+3}\diamond\top \\ \theta_n &:= \diamond\top \wedge \square\diamond\top \wedge \square^2\diamond\top \wedge \bigwedge_{d=1}^n \bigvee_{b \in \{0,1\}} \square^{d+2}(b \wedge \diamond\top) \wedge \square^{n+3}\diamond\top \end{aligned}$$

Note that ψ_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 ξ_n , we obtain the following.

Fact 1: $M'_{s'}$ satisfies ξ_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 $\forall(\theta_n \rightarrow \bigvee_{b \in \{0,1\}} \square^{n+3}b)$ of the definition of ψ_n , the formula θ_n intuitively enforces one to select the refinements $M'_{s'}$ of M_s encoding only n -blocks having the same position. Formally, by definition of θ_n , we obtain the following.

Fact 2: Let $M''_{s''}$ be a refinement of $M'_{s'}$. Then, $M''_{s''}$ satisfies θ_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 $\forall(\theta_n \rightarrow \bigvee_{b \in \{0,1\}} \square^{n+3}b)$ of definition of ψ_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. \square

For each $n \geq 0$, let ψ_n be the \mathcal{L}_\forall formula satisfying the statement of Lemma 57. Then, the one-agent \mathcal{L}_\forall formula φ_n is defined as follows:

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

By construction and Lemma 57, we easily obtain the following result.

Lemma 58 *For each $n \geq 0$, the \mathcal{L}_\forall formula φ_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 φ_n iff M_s is balanced. \dashv*

Proof of Proposition 56: by Lemma 58, in order to complete the proof of Proposition 56, we need to show that for each $n \geq 0$, each one-agent \mathcal{L}^μ formula equivalent to φ_n has size at least $2^{2^{\Omega(n)}}$. For this, we use a well-known automata-characterization of (one-agent) \mathcal{L}^μ in terms of *parity symmetric alternating (finite-state) automata* (PSAA) which operate on pointed models [61]. 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 ε 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 $\pi = x_0 x_1 \dots$ of T -nodes such that $x_0 = \varepsilon$ and for any i , x_{i+1} is a child of x_i . For an alphabet Σ , a Σ -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 \vee and \wedge (we also allow the formulas **true** and **false**). A subset Y of X *satisfies* $\theta \in \mathcal{B}_+(X)$ iff the truth assignment that assigns **true** to the elements in Y and **false** to the elements of $X \setminus Y$ satisfies θ .

A *parity symmetric alternating automaton* (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, $\delta : Q \times 2^P \rightarrow \mathcal{B}_+(\{\square, \diamond\} \times Q)$ 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 $\{\square, \diamond\} \times Q$. An atom (\diamond, q) means that from the current state s (of the pointed input model) \mathcal{A} moves to some successor of s and the location is updated to q . On the other hand, an atom (\square, q) means that from the current state s the automaton splits in multiple copies and, for each successor s' of s , one of such copies moves to s' and the location is updated to q .

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 $\langle T, r \rangle$.¹³ Moreover, we require that $r(\varepsilon) = (q_0, s_0)$ (initially, \mathcal{A} is in the initial 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 $H \subseteq \{\square, \diamond\} \times Q$ 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 $at = (\square, q')$, then for each successor s' of s in M , $(q', s') \in L(y)$.

For an infinite path $\pi = y_0 y_1 \dots$ of T , let $inf(\pi)$ be the set of locations in Q that appear in $r(y_0)r(y_1) \dots$ infinitely often. The run $\langle T, r \rangle$ is *accepting* if for each infinite path π 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 59 [61] *Given a one-agent \mathcal{L}^μ formula φ over P , one can construct a PSAA \mathcal{A}_φ with $O(|\varphi|)$ locations whose language is the set of pointed models over P satisfying φ .¹*

¹³Intuitively, 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 .

Proposition 56 directly follows from Proposition 59 and the following result.

Lemma 60 *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}_\forall formula φ_n . Then, the number of locations of \mathcal{A}_n is at least 2^{2^n} . \dashv*

Proof Let $n \geq 0$ and \mathcal{A}_n as in the statement of the lemma (note that \mathcal{A}_n exists by Proposition 59 together with Proposition 42), and Q be the set of \mathcal{A}_n -locations. For each n -configuration w , let $M_{s_w}^w$ be some balanced 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 $M_{s_w}^w$ so that:

- Q_l is the set of locations associated with the replicas 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., $Q_l = \{q \in Q \mid \text{for some } x \in T, r(x) = (q, s_l)\}$. (Note that $Q_l = \emptyset$ if $\langle T, r \rangle$ does not visit the left successor s_l of s_w .)

First, we show that $H(w) \neq \emptyset$. By hypothesis and Lemma 58, there must exist some accepting run of \mathcal{A}_n over the input $M_{s_w}^w$. Now, by construction, $H(w)$ is a set of subsets of Q , and $H(w)$ is non-empty if and only if there is some accepting run of \mathcal{A}_n over $M_{s_w}^w$. (If no accepting run of \mathcal{A}_n visits the left successor s_l of s_w in M^w , then $H(w)$ is a singleton containing just the empty set.) Hence, non-emptiness of $H(w)$ follows. Next, we prove the following.

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 $s \in S^M$, let $[M_s]$ denote the restriction of M to the set of states reachable from s in M . For all n -configurations w and $dir \in \{l, r\}$, let $s_{w,dir}$ be the *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 $M_{s_w}^{w,w'}$ 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 $[M_{s_{w',l}}^{w,w'}] = [M_{s_{w',l}}^{w'}]$ and $[M_{s_{w,r}}^{w,w'}] = [M_{s_{w,r}}^w]$. Evidently, $M_{s_w}^{w,w'}$ is a pointed model encoding the pair (w', w) . Since $w \neq w'$, by hypothesis and Lemma 58, \mathcal{A}_n does not accept $M_{s_w}^{w,w'}$. 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 $M_{s_w}^{w,w'}$, 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 $2^{2^{2^n}}$ and the number of subsets of Q is $2^{|Q|}$, we obtain that $|Q| \geq 2^{2^n}$, and the result holds. \square

8 Conclusions and perspectives

Conclusions 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 μ -calculus are only the beginning. One could think of refinement CTL, refinement PDL, (yet other) 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 logic without refinement is certainly not to be expected; e.g., we estimate that refinement modal logic is more expressive than the base modal logic on the \mathcal{KT} model class.

Recent results Following the initial submission of the paper, some further results have been obtained in this area, typically involving one of the authors. In [12] it was established that the complexity of refinement modal logic for a single agent is AEXP_{pol} -complete, which means that the satisfiability of an RML formula can be decided by an exponential-time bounded Alternating Turing Machine with a linearly-bounded number of alternations. In [28] an axiomatization of the multi-agent refinement modal logic of knowledge is given, among other results. As a generalization of quantifying over announcements (arbitrary announcements), in [8] semantics were also given for quantifying over action models and the question was posed how to axiomatize this logic: in [26] it is shown that quantifying over action models is equally expressive as the refinement quantifier, i.e., ‘there is a refinement after which φ is true’ means the same as ‘there is an action model such that after its execution φ is true’. This answers one of the open questions on logics with quantification over information change, posed in the recent survey [54]. That survey also puts various other proposals on propositional quantification in perspective, such as [52], [18], [7] (going back to [6]), and [60]—for details, see [54]. It should not be forgotten to mention that many of these, including our own proposal, go back to the original publication [21].

Further research We wish to determine the complexity of model checking in the various refinement modal logics. 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 to other forms of propositional quantification over information change also needs further investigation.

Acknowledgements

We acknowledge a very insightful and detailed review from a journal referee. Hans van Ditmarsch is also affiliated to IMSc (Institute of Mathematical Sciences), Chennai, India. We acknowledge support from ERC project EPS 313360, and from EU 7th Framework Programme under grant agreement no. 295261 (MEALS).

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] Rajeev Alur, Thomas A. Henzinger, Orna Kupferman, and Moshe Y. Vardi. Alternating refinement relations. In *International Conference on Concurrency Theory*, pages 163–178, 1998.
- [4] 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.
- [5] A. Arnold and D. Niwinski. *Rudiments of μ -calculus*. North Holland, 2001.
- [6] G. Aucher. Characterizing updates in dynamic epistemic logic. In *Proceedings of Twelfth KR*. AAAI Press, 2010.
- [7] G. Aucher. DEL-sequents for regression and epistemic planning. *Journal of Applied Non-Classical Logics*, 22(4):337–367, 2012.
- [8] 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.
- [9] A. Baltag, L.S. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. In *Proc. of 7th TARK*, pages 43–56. Morgan Kaufmann, 1998.
- [10] 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.
- [11] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, Cambridge, 2001. Cambridge Tracts in Theoretical Computer Science 53.
- [12] Laura Bozzelli, Hans P. van Ditmarsch, and Sophie Pinchinat. The complexity of one-agent refinement modal logic. In Luis Fariñas del Cerro, Andreas Herzig, and Jérôme Mengin, editors, *JELIA*, volume 7519 of *Lecture Notes in Computer Science*, pages 120–133. Springer, 2012.
- [13] 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*, LNCS 249, pages 256–270. Springer, 1987.

- [14] 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.
- [15] G. d’Agostino and M. Hollenberg. Logical questions concerning the μ -calculus: Interpolation, Lyndon and Los-Tarski. *Journal of Symbolic Logic*, 65(1):310–332, 2000.
- [16] G. d’Agostino and G. Lenzi. An axiomatization of bisimulation quantifiers via the μ -calculus. *Theor. Comput. Sci.*, 338(1-3):64–95, 2005.
- [17] 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.
- [18] P. Economou. *Extensions and Applications of Dynamic Epistemic Logic*. PhD thesis, Oxford University, 2010.
- [19] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge MA, 1995.
- [20] Guillaume Feuillade and Sophie Pinchinat. Modal specifications for the control theory of discrete-event systems. *Discrete Event Dynamic Systems*, 17(2):181–205, 2007.
- [21] K. Fine. Propositional quantifiers in modal logic. *Theoria*, 36(3):336–346, 1970.
- [22] T. French. *Bisimulation quantifiers for modal logic*. PhD thesis, University of Western Australia, 2006.
- [23] 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. Proc. of the seventh conference “Advances in Modal Logic”.
- [24] J.D. Gerbrandy and W. Groeneveld. Reasoning about information change. *Journal of Logic, Language, and Information*, 6:147–169, 1997.
- [25] J. Hales. Refinement quantifiers for logics of belief and knowledge. Honours Thesis, University of Western Australia, 2011.
- [26] J. Hales. Arbitrary action model logic and action model synthesis. In *Proc. of 28th LICS*, pages 253–262. IEEE, 2013.
- [27] J. Hales, T. French, and R. Davies. Refinement quantified logics of knowledge. *Electr. Notes Theor. Comput. Sci.*, 278:85–98, 2011.
- [28] J. Hales, T. French, and R. Davies. Refinement quantified logics of knowledge and belief for multiple agents. In *Advances in Modal Logic 9*, pages 317–338. College Publications, 2012.

- [29] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, Cambridge MA, 2000. Foundations of Computing Series.
- [30] D. Janin and I. Walukiewicz. Automata for the modal mu-calculus and related results. In *Proc. of 20th MFCS*, LNCS 969, pages 552–562. Springer, 1995.
- [31] D. Janin and I. Walukiewicz. 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.
- [32] B. Kooi. Expressivity and completeness for public update logics via reduction axioms. *Journal of Applied Non-Classical Logics*, 17(2):231–254, 2007.
- [33] O. Kupferman, M. Vardi, and P. Wolper. Module checking. *Information and Computation*, 164(2):322–344, 2001.
- [34] 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.
- [35] Ralf Küsters. Memoryless determinacy of parity games. In *Automata logics, and infinite games*, pages 95–106. Springer, 2002.
- [36] 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.
- [37] 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.
- [38] René Mazala. Infinite games. *Automata logics, and infinite games*, pages 197–204, 2002.
- [39] J.S. Miller and L.S. Moss. The undecidability of iterated modal relativization. *Studia Logica*, 79(3):373–407, 2005.
- [40] C. Morgan. *Programming from Specifications: Second Edition*. Prentice Hall International, Hemstead, UK, 1994.
- [41] 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.
- [42] M. Pauly. *Logic for social software*. PhD thesis, University of Amsterdam, 2001. ILLC Dissertation Series DS-2001-10.

- [43] J.A. Plaza. Logics of public communications. In *Proc. of the 4th ISMIS*, pages 201–216. Oak Ridge National Laboratory, 1989.
- [44] Jean-Baptiste Raclet. *Quotient de spécifications pour la réutilisation de composants*. PhD thesis, Université de Rennes I, December 2007. (In French).
- [45] 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.
- [46] 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 199–127. IEEE Computer Society Press, 2009.
- [47] P. Ramadge and W. Wonham. On the supervisory control of discrete event systems. In *Proc. of the IEEE*, pages 81–98, 1989.
- [48] Stéphane Riedweg and Sophie Pinchinat. Quantified mu-calculus for control synthesis. In Branislav Rován and Peter Vojtás, editors, *MFCs*, volume 2747 of *Lecture Notes in Computer Science*, pages 642–651. Springer, 2003.
- [49] 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.
- [50] 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.
- [51] J. Tsitsoklis. On the control of discrete event dynamical systems. *Mathematics of Control Signals and Systems*, 2(2):95–107, 1989.
- [52] J. van Benthem. An essay on sabotage and obstruction. In *Mechanizing Mathematical Reasoning*, volume 2605 of *LNCS 2605*, pages 268–276. Springer, 2005.
- [53] J. van Benthem. One is a lonely number: on the logic of communication. In *Logic colloquium 2002. Lecture Notes in Logic, Vol. 27*, pages 96–129. A.K. Peters, 2006.
- [54] H. van Ditmarsch. Quantifying notes. In *Proc. of 19th WoLLIC*, LNCS 7456, pages 89–109. Springer, 2012.
- [55] 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*, LNAI 5605. Presented at LOFT 2008 and KRAMAS 2008, pages 51–65. Springer, 2009.

- [56] 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.
- [57] H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*, volume 337 of *Synthese Library*. Springer, 2007.
- [58] Y. Venema. Lecture notes on the modal μ -calculus. (Draft), 2012.
- [59] I. Walukiewicz. Completeness of Kozen’s axiomatisation of the propositional μ -calculus. *INFCTRL: Information and Computation (formerly Information and Control)*, 157, 2000.
- [60] X. Wen, H. Liu, and F. Huang. An alternative logic for knowability. In *Logic, Rationality, and Interaction (Proceedings of LORI-3)*, LNCS 6953, pages 342–355. Springer, 2011.
- [61] T. Wilke. CTL⁺ is exponentially more succinct than CTL. In *Proc. 19th FSTTCS*, LNCS 1738, pages 110–121. Springer, 1999.
- [62] J. Woodcock and J. Davies. *Using Z — Specification, Refinement and Proof*. Prentice Hall, 1996.