A Representation Theorem for Reasoning in First-Order Multi-Agent Knowledge Bases

Christoph Schwering
The University of New South Wales
Sydney, NSW, Australia
c.schwering@unsw.edu.au

Maurice Pagnucco
The University of New South Wales
Sydney, NSW, Australia
m.pagnucco@unsw.edu.au

ABSTRACT

Levesque's notion of *only-knowing* provides a natural formalisation of a knowledge base: it precisely captures the beliefs and non-beliefs that follow from the knowledge base, including introspection and *de dicto* versus *de re* distinctions in a first-order setting. Apart from its attractive properties in terms of specification, a major result about only-knowing is Levesque's representation theorem, which shows how reasoning in (single-agent) knowledge bases can be Turing-reduced to ordinary first-order logic. While numerous proposals have been made to lift the logic of only-knowing to the multi-agent case, generalising the representation theorem has remained an open problem. In this paper, we develop a Turing reduction from reasoning in multi-agent knowledge bases to ordinary, non-epistemic first-order logic and thus obtain a new representation theorem for the multi-agent case.

KEYWORDS

Reasoning in multi-agent systems; only-knowing; first-order logic.

ACM Reference Format:
Christoph Schwering and Maurice Pagnucco. 2019. A Representation Theorem for Reasoning in First-Order Multi-Agent Knowledge Bases. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 9 pages.

1 INTRODUCTION

Levesque's notion of *only-knowing* [12–14] provides a natural formalisation of a knowledge base: the knowledge base is not just known, but *all* that is known. As a consequence, only-knowing precisely captures beliefs and non-beliefs in a natural and semantically perspicuous way. This includes statements involving first-order quantification, introspection, and even *de dicto* versus *de re* distinctions, such as the crucial difference in a card game between "I know that there is a card which my opponent is holding" versus "there is a card which I know my opponent is holding."

Reasoning in such knowledge bases can be cast as a logical entailment problem: given a knowledge base and a query, we want to prove or disprove that only-knowing the knowledge base entails the query. One important result about only-knowing is that such reasoning problems can be reduced to ordinary, non-epistemic first-order reasoning. Apart from establishing a theoretical link between only-knowing and ordinary logic, this result, known as Levesque's *representation theorem*, also offers a viable path towards implementing a reasoning service using off-the-shelf theorem provers [12, 14].

Levesque's original logic only considers the single-agent case. A review of the literature shows that generalising only-knowing to multiple agents turned out to be surprisingly difficult. Earlier proposals exhibit counter-intuitive properties and/or do not extend to the first-order case [6, 7, 9] or their semantics is difficult to work with due to its technical complexity [19]. A more recent approach by Belle and Lakemeyer [3] overcomes these issues by using a simpler semantic model that is more closely related to Levesque's original semantics than Kripke-model approaches [6, 7, 9, 19].

The search for appropriate semantics and proof theories left the question of reducing reasoning in multi-agent knowledge bases to non-epistemic logic with little attention over the years. To our knowledge, only Belle and Lakemeyer [1] discuss such a reduction at all. However, since they adopt Levesque's single-agent representation theorem essentially unaltered, their knowledge bases are subject to severe restrictions that eradicate most of the expressivity of multi-agent only-knowing. In particular, they assume agents have complete knowledge about what the other agents know.

In reality, however, agents rarely have firm knowledge about what other agents know. During a card game, for example, we usually know only our own cards unequivocally. While we do not know our opponents' cards, we do know that they know them. In fact, there are subtle connections between our (lack of) knowledge and what we know them to know: for instance, *we know that if* the other player has the Ace of Spades, *then she knows* that she has it.

These subtle relations are *de dicto* and *de re* distinctions between the different agents. Belle and Lakemeyer's logic of only-knowing [3] provides us with the expressivity to model such scenarios in a natural and concise way.

In this paper, we build on Belle and Lakemeyer's logic to develop a *Turing-reduction from reasoning in first-order multi-agent knowledge bases to ordinary, non-epistemic first-order logic.* Thus, we *generalise Levesque's representation theorem to the multi-agent case.* The class of knowledge bases we consider allows for fine-grained control of what agents know (not) about other agents' knowledge, including *de re* versus *de dicto* distinctions. In particular, it subsumes Belle and Lakemeyer's knowledge bases from [1] as a special case.

On the theoretical side, this reduction establishes a surprising relationship between the logic of multi-agent only-knowing and non-epistemic logic. On the practical side, it opens up ways of implementing multi-agent reasoning by leveraging theorem provers like [8, 15] or extending systems like [17] to multiple agents.

The paper is organised as follows. First, we recapitulate Belle and Lakemeyer's multi-agent logic of only-knowing. Section 3 introduces our class of multi-agent knowledge bases. Section 4 shows how queries can be evaluated in these knowledge bases using only ordinary, non-epistemic first-order reasoning. Then we conclude.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada. © 2019 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

2 THE LOGIC

We consider a first-order logic with equality and modal operators for knowledge. There are two important differences to classical first-order logic: the concept of *standard names*, often simply referred to as *names*, and the modal operators for *knowing* and *only-knowing*.

Standard names are special constant symbols that satisfy the unique-names assumption (that is, distinct names refer to distinct objects) and an infinitary version of domain closure (that is, every object can be referred to by some name). Standard names simplify the technical treatment because quantification can be handled by substituting names for variables; we refer to [14] for further details.

There are two modal operators per agent: one to express that a formula is known, the other to express that a formula is *all* that is known. The latter operator is instrumental in capturing the meaning of a knowledge base. To keep the formal machinery simple, we only consider two agents; it is straightforward to extend the language as well as our results to more agents.

The foundations of this logic are due to Levesque [12–14]. This multi-agent extension follows Belle and Lakemeyer [1, 3].

2.1 The Language

Formally, the *alphabet* of the language consists of countably many predicate symbols of each arity, countably many variable symbols, countably many standard name symbols, modal operators "$\mathbf{K}_A$", "$\mathbf{K}_B$", "$\mathbf{O}_A$", "$\mathbf{O}_B$", and symbols "$=$", "$\neg$", "$\vee$", "$\exists$", "(", ")", ",".

The *terms* of the language are the variables and standard names. For simplicity, we do not consider functions in this paper.

The *formulas* are of the form $P(t_1, \ldots, t_k)$, $t_1 = t_2$, $\neg\alpha$, $(\alpha \vee \beta)$, $\exists x\,\alpha$, $\mathbf{K}_a\,\alpha$, $\mathbf{O}_a\,\alpha$, where P is a k-ary predicate symbol, $t_1, t_2, \ldots, t_k$ are terms, α and β are formulas, x is a variable, and $a \in \{A, B\}$ is an agent. We read $\mathbf{K}_a\,\alpha$ as "agent a knows α" and $\mathbf{O}_a\,\alpha$ as "all that agent a knows is α" or "agent a only-knows α".

We use the following common abbreviations: $t_1 \neq t_2$ for $\neg t_1 = t_2$, $(\alpha \wedge \beta)$ for $\neg(\neg\alpha \vee \neg\beta)$, $\forall x\alpha$ for $\neg\exists x\neg\alpha$, $(\alpha \supset \beta)$ for $(\neg\alpha \vee \beta)$, $(\alpha \equiv \beta)$ for $(\alpha \supset \beta) \wedge (\beta \supset \alpha)$, $\top$ for $\exists x(x = x)$, and $\bot$ for $\neg\top$. We use $\vec{t}$ as a shorthand for $t_1, \ldots, t_k$, and $|\vec{t}|$ for k, and $\exists\vec{x}$ for $\exists x_1 \ldots \exists x_k$. We sometimes omit brackets for ease of readability.

A *perspective vector* is a vector $\vec{a} = \langle a_1, \ldots, a_k\rangle$ of alternating agents $a_i \in \{A, B\}$, that is, $a_i \neq a_{i+1}$ for all $1 \leq i < k$. We abuse notation and identify an agent a with the vector $\langle a\rangle$. For a perspective vector $\vec{a} = \langle a_1, \ldots, a_k\rangle$ and an agent a', we define $\vec{a}\cdot a'$ as follows:

- if $a_k \neq a'$: $\vec{a}\cdot a' = \langle a_1, \ldots, a_k, a'\rangle$,
- if $a_k = a'$: $\vec{a}\cdot a' = \langle a_1, \ldots, a_k\rangle$.

For example, $A\cdot B\cdot B\cdot A\cdot A = A\cdot B\cdot A = \langle A, B, A\rangle$.

The *$\vec{a}$-subformulas* $\Sigma_{\vec{a}}(\alpha)$ of a formula α are defined as follows:

- $\Sigma_{\langle\rangle}(\alpha) = \{\alpha\}$,
- $\Sigma_{\vec{a}\cdot a'}(\alpha)$ is the least set such that if $\mathbf{K}_{a'}\,\beta$ or $\mathbf{O}_{a'}\,\beta$ occurs outside of modal operators in some $\gamma \in \Sigma_{\vec{a}}(\alpha) \cup \Sigma_{\vec{a}\cdot a'}(\alpha)$, then $\beta \in \Sigma_{\vec{a}\cdot a'}(\alpha)$.

For example, if α is the formula $\mathbf{O}_A\,\forall x P(x) \supset \mathbf{K}_A\,\exists y(P(y) \wedge \mathbf{K}_A\,Q(y))$, then $\Sigma_A(\alpha) = \{\forall x P(x),\ \exists y(P(y) \wedge \mathbf{K}_A\,Q(y)),\ Q(y)\}$.

A variable x is *free* in a formula α iff x occurs outside the scope of $\exists x$. We write $\alpha^{t_1}_{t_2}$ to replace t_1 with t_2 in α; when t_1 is a variable, we only replace its free occurrences. We use $(\beta, \vec{y})$ as a shorthand to say that β has free variables $\vec{y}$. In particular, we will often use $(\beta, \vec{y}) \in \Sigma_{\vec{a}}(\alpha)$ as an abbreviation to say that $\beta \in \Sigma_{\vec{a}}(\alpha)$ and β has free variables $\vec{y}$. Thus in the above example, $(Q(y), y) \in \Sigma_A(\alpha)$.

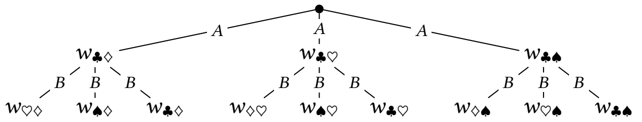
A
A
A
$w_{\clubsuit\diamondsuit}$
$w_{\clubsuit\heartsuit}$
$w_{\clubsuit\spadesuit}$
B
B
B
B
B
B
B
B
B
$w_{\heartsuit\diamondsuit}$
$w_{\spadesuit\diamondsuit}$
$w_{\clubsuit\diamondsuit}$
$w_{\diamondsuit\heartsuit}$
$w_{\spadesuit\heartsuit}$
$w_{\clubsuit\heartsuit}$
$w_{\diamondsuit\spadesuit}$
$w_{\heartsuit\spadesuit}$
$w_{\clubsuit\spadesuit}$

Figure 1: A 2-structure for the card game example, visualised as a tree, where $w_{cc'}$ is a world in which A and B are holding c and c'. A does not know B's card, but A does know that B knows it.

The *a_1-depth* of a formula α is the maximum k such that for some perspective vector $\vec{a} = \langle a_1, \ldots, a_k\rangle$, $\Sigma_{\vec{a}}(\alpha) \neq \{\}$. A formula is *$a$-objective* iff its a-depth is 0 and objective if its A- and B-depths are 0. The A- and B-depths of the above example are 1 and 0, respectively.

2.2 The Semantics

Formulas are interpreted with respect to *worlds* and *i-structures*. Intuitively, a world stipulates which facts are true and which are false at an objective level. An i-structure models an agent's knowledge using the possible-worlds approach: an agent, A say, knows the statements that are true in all the worlds she considers possible. To also account for what A knows about B's knowledge, every possible world of A is additionally associated with a set of worlds that A knows B to consider possible. In an i-structure, this arrangement recurses i times.

We denote the set of worlds by $\mathbb{W}$ and the set of i-structures by $\mathbb{E}_i$. A *world* $w \in \mathbb{W}$ is a set of formulas $P(\vec{n})$, where P is a $|\vec{n}|$-ary predicate symbol and $\vec{n}$ are names. An *i-structure* $e \in \mathbb{E}_i$ is

- if $i = 0$: the empty set,
- if $i \geq 1$: a set of tuples (e, w), where $w \in \mathbb{W}$ and $e \in \mathbb{E}_{i-1}$.

For example, consider a card game with only four cards $\diamondsuit, \heartsuit, \spadesuit, \clubsuit$ (for brevity of presentation), and suppose A knows her own card, $\clubsuit$ say, but she does not know B's card. We can model this with an i-structure e_A that contains the worlds where A holds $\clubsuit$ and B holds some other card: $e_A = \{(e_B^{\diamondsuit}, w_{\clubsuit\diamondsuit}), (e_B^{\heartsuit}, w_{\clubsuit\heartsuit}), (e_B^{\spadesuit}, w_{\clubsuit\spadesuit})\}$, where $w_{\clubsuit\diamondsuit}, w_{\clubsuit\heartsuit}, w_{\clubsuit\spadesuit}$ are worlds where A holds $\clubsuit$ and B holds $\diamondsuit, \heartsuit, \spadesuit$, respectively, and $e_B^{\diamondsuit}, e_B^{\heartsuit}, e_B^{\spadesuit}$ are arbitrary $(i-1)$-structures. To also model that A knows that B knows his own but not A's card, we let $e_B^{\diamondsuit} = \{(\{\}, w_{\heartsuit\diamondsuit}), (\{\}, w_{\spadesuit\diamondsuit}), (\{\}, w_{\clubsuit\diamondsuit})\}$, and $e_B^{\heartsuit}, e_B^{\spadesuit}$ analogously. An i-structure can also be viewed as a tree as depicted in Figure 1. For a discussion of i-structures, we refer to [3].

We now define how formulas are interpreted. Let α be a formula without free variables and of A-depth and B-depth at most i and j, respectively. Then *truth* of α with respect to structures $e_A \in \mathbb{E}_i$ and $e_B \in \mathbb{E}_j$ and a world w is defined as follows:

- $e_A, e_B, w \models P(\vec{n})$ iff $P(\vec{n}) \in w$
- $e_A, e_B, w \models n_1 = n_2$ iff $n_1 = n_2$
- $e_A, e_B, w \models \neg\alpha$ iff $e_A, e_B, w \not\models \alpha$
- $e_A, e_B, w \models (\alpha \vee \beta)$ iff $e_A, e_B, w \models \alpha$ or $e_A, e_B, w \models \beta$
- $e_A, e_B, w \models \exists x\alpha$ iff for some name n, $e_A, e_B, w \models \alpha^x_n$
- $e_A, e_B, w \models \mathbf{K}_A\alpha$ iff for all $e'_B \in \mathbb{E}_{i-1}$ and $w' \in \mathbb{W}$, $(e'_B, w') \in e_A$ only if $e_A, e'_B, w' \models \alpha$
- $e_A, e_B, w \models \mathbf{O}_A\alpha$ iff for all $e'_B \in \mathbb{E}_{i-1}$ and $w' \in \mathbb{W}$, $(e'_B, w') \in e_A$ iff $e_A, e'_B, w' \models \alpha$

The semantics of $\mathbf{K}_B\alpha$ and $\mathbf{O}_B\alpha$ is symmetric to the cases for $\mathbf{K}_A\alpha$ and $\mathbf{O}_A\alpha$. Sometimes we prove statements only for agent A with the understanding that the result and proof for B are analogous.

A formula α is *valid*, written $\models \alpha$, iff for all i and j greater than or equal to the A-depth and B-depth of α, for all $e_A \in \mathbb{E}_i$, $e_B \in \mathbb{E}_j$ and $w \in \mathbb{W}$, $e_A, e_B, w \models \alpha$. A formula α is *satisfiable* iff $\not\models \neg\alpha$.

Note that for objective formulas, truth depends solely on w, not on e_A, e_B. Similarly, truth of formulas where no predicate symbols symbol occur outside of $\mathbf{K}_A$ or $\mathbf{O}_A$ depends only on e_A, not on e_B or w. We allow ourselves to omit parameters irrelevant to truth; the naming of the given parameters shall then make it clear which others are omitted. For example, we may write $w \models P \land Q$ for $e_A, e_B, w \models P \land Q$ and $e_A \models \mathbf{K}_A(P \land Q)$ for $e_A, e_B, w \models \mathbf{K}_A(P \land Q)$.

For space reasons, we shall not discuss the properties of the $\mathbf{K}_a$ operator any further apart from mentioning that it has the usual K45-properties [5]. A detailed analysis can be found in [3].

As for $\mathbf{O}_a$, the following lemma tells us that a only-knowing an a-objective formula is always satisfiable and, in fact, uniquely determines the i-structure, modulo i.

Lemma 2.1 *Let α be an A-objective formula without free variables and with A-depth at most i, and let $e_A \in \mathbb{E}_i$.*
Then $e_A \models \mathbf{O}_A\alpha$ iff $e_A = \{(e'_B, w') \in \mathbb{E}_{i-1} \mid e'_B, w' \models \alpha\}$.

Proof. $e_A \models \mathbf{O}_A\alpha$ iff $e_A = \{(e'_B, w') \in \mathbb{E}_{i-1} \mid e_A, e'_B, w' \models \alpha\}$ iff (since α is A-objective) $e_A = \{(e'_B, w') \in \mathbb{E}_{i-1} \mid e'_B, w' \models \alpha\}$. □

A property which will be very helpful later is that agents can only-know at most one formula, modulo logical equivalence.

Property 2.2 *Let α, β be A-objective.*
Then $\models \mathbf{O}_A\alpha \supset \mathbf{O}_A\beta$ iff $\models \alpha \equiv \beta$.

Proof. Let i be greater than or equal to the A-depth of $\mathbf{O}_A\alpha \supset \mathbf{O}_A\beta$.

For the only-if direction, suppose $\models \mathbf{O}_A\alpha \supset \mathbf{O}_A\beta$. Let $e_A = \{(e'_B, w') \in \mathbb{E}_{i-1} \mid e'_B, w' \models \alpha\}$. By Lemma 2.1, $e_A \models \mathbf{O}_A\alpha$. By assumption, $e_A \models \mathbf{O}_A\beta$. By Lemma 2.1, $e_A = \{(e'_B, w') \in \mathbb{E}_{i-1} \mid e'_B, w' \models \beta\}$. Thus $e'_B, w' \models \alpha$ iff $e'_B, w' \models \beta$ for all $e'_B \in \mathbb{E}_{i-1}, w' \in \mathbb{W}$. If j is the B-depth of $\alpha \equiv \beta$, then $\mathbb{E}_{i-1} \supseteq \mathbb{E}_j$. Hence, $\models \alpha \equiv \beta$.

Conversely, suppose $\models \alpha \equiv \beta$ and $e_A \models \mathbf{O}_A\alpha$, $e_A \in \mathbb{E}_i$. By Lemma 2.1, $e_A = \{(e'_B, w') \in \mathbb{E}_{i-1} \mid e'_B, w' \models \alpha\}$. By assumption, $e_A = \{(e'_B, w') \in \mathbb{E}_{i-1} \mid e'_B, w' \models \beta\}$. Thus $e_A \models \mathbf{O}_A\beta$. □

The following property allows us to implicitly assume in the rest of the paper that structures match the relevant formulas' depth.

Property 2.3 (Belle and Lakemeyer [3]) *Let α be a formula without free variables, and i, j be at least the A-depth and B-depth of α.*
Then $\models \alpha$ iff for all $e_A \in \mathbb{E}_i$, $e_B \in \mathbb{E}_j$, $w \in \mathbb{W}$, $e_A, e_B, w \models \alpha$.

3 MULTI-AGENT KNOWLEDGE BASES

The modalities for only-knowing and knowing serve two orthogonal purposes in Levesque's logic: $\mathbf{O}_a$ asserts a knowledge base, $\mathbf{K}_a$ queries the knowledge base. In this section, we formalise the concepts of multi-agent knowledge bases and queries.

Definition 3.1 A formula α with free variables $\vec{x}$ is *a-determinate* iff $\models \forall\vec{x}\big(\alpha \supset \bigvee_{(\beta,\vec{y}) \in \Sigma_a(\alpha)} \exists\vec{y}\mathbf{O}_a\beta\big)$. A formula κ is a *knowledge base* iff it has no free variables, mentions no $\mathbf{K}$, is A- and B-determinate, and for all $\vec{a}$ and $a' \neq b'$, every $\alpha \in \Sigma_{\vec{a}\cdot a'}(\kappa)$ is a'-objective and b'-determinate. A formula λ is a *query* iff it has no free variables and mentions no $\mathbf{O}$.

Before we present examples for a knowledge base and queries, a discussion of A-objectivity and B-determinacy is in order.

Our requirement of α within $\mathbf{O}_A\alpha$ being A-objective is the natural generalisation of Levesque's representation of a single-agent knowledge base as an objective formula [12, 14]. This restriction is motivated by two issues with non-A-objective formulas in $\mathbf{O}_A\alpha$. For one thing, nested $\mathbf{O}_A$ operators are difficult to interpret: for example, what does $\mathbf{O}_A(P \land \mathbf{O}_A Q)$ mean intuitively? For another, they introduce technical challenges because Lemma 2.1 does not extend to non-A-objective α: $\mathbf{O}_A(P \land \mathbf{O}_A Q)$ is unsatisfiable, whereas $\mathbf{O}_A(P \lor \mathbf{O}_A Q)$ is equivalent to $\mathbf{O}_A P$.

The B-determinacy constraint requires α within $\mathbf{O}_A\alpha$ to entail that B only-knows *some*thing. In particular, it allows for many – even infinitely many – alternatives what B might know. For example, in $\mathbf{O}_A\big((P \lor Q) \land (P \supset \mathbf{O}_B P) \land (Q \supset \mathbf{O}_B Q)\big)$ the formula within $\mathbf{O}_A$ is B-determinate: it entails $\mathbf{O}_B P \lor \mathbf{O}_B Q$. Thus our multi-agent knowledge bases are a significantly more expressive generalisations of those considered by Belle and Lakemeyer [1], who assume every agent to have complete knowledge of the other agent's knowledge.

Lifting B-determinacy and allowing $\mathbf{K}_B$ operators in the knowledge base would add further expressivity: we could express that A knows B to know a formula but not necessarily only-know it. However, it would also add significantly to the complexity of the reduction because reasoning about B's knowledge then coincides with theorem proving for first-order K45. It is an open question, also in the single-agent case, how this reduction would work.

A technical implication of B-determinacy is that the inner-most $\mathbf{O}_A\alpha$ formulas in a knowledge base must be "closed off" by adding $\mathbf{O}_B\bot$ conjunctively to α. Such $\mathbf{O}_B\bot$ should not be understood to genuinely mean that B only-knows $\bot$ but rather as a marker that indicates the maximal modelled nesting depth has been reached.

An equivalent but technically more cumbersome alternative to Definition 3.1 is to say that queries may only refer to perspectives for which the knowledge base is determinate.

Example

As a running example, we consider a card game situation where each player is holding a single card.[1] We model the cards held by agent a with a unary predicate H_a. The following formula, call it Φ, represents that the agents hold two mutually distinct cards:

$$\begin{gathered}\exists x \forall y \big(H_A(y) \equiv x = y\big) \land \\ \exists x \forall y \big(H_B(y) \equiv x = y\big) \land \\ \neg\exists x \big(H_A(x) \land H_B(x)\big)\end{gathered}$$

Let us further suppose that agent A knows her own card, which we refer to by standard name c. Agent A also knows that B is holding a card but she does not know which one; however, A does know that B knows which card B is holding. This is a typical example of a *de dicto* versus *de re* distinction, or *knowing that* versus *knowing which*: A knows *that* B has some card; B knows *which* one. The former is represented by quantifying a variable inside the modal operator, the latter by quantifying the variable outside of the modal

[1] This paper is only concerned with modelling the agents' knowledge in a static situation. Dynamic extensions of Levesque's logic have been studied elsewhere [1, 11]. As they are orthogonal to the representation theorem, we do not consider them here.

operator. We can thus represent what A knows as follows:

$$\begin{aligned}&\mathbf{O}_A\big(H_A(c) \land \Phi \land\\ &\qquad \forall x\big(H_B(x) \supset \mathbf{O}_B\big(H_B(x) \land \Phi\big)\big)\big)\end{aligned}$$

The first line represents A's objective knowledge about her own and B's card: she is holding c and B is holding another card (implied by Φ). The second line models her knowledge about B's knowledge: A knows that whichever card x B is holding, B knows he has x (and, by Φ, that A is holding some other card).

For this formula to comply with Definition 3.1, we need to ensure A- and B-determinacy. Since in this example, we are interested only in what A knows and what A knows B to know, we can make the formula B-determinate by stipulating $\mathbf{O}_B\bot$ and adding $\mathbf{O}_A\bot$ to the nested $\mathbf{O}_B(H_B(x) \land \Phi)$ and obtain the following knowledge base κ:

$$\begin{aligned}\kappa \;=\; &\mathbf{O}_A\big(H_A(c) \land \Phi \land\\ &\qquad \forall x\big(H_B(x) \supset \mathbf{O}_B\big(H_B(x) \land \Phi \land \mathbf{O}_A\bot\big)\big)\big) \land\\ &\mathbf{O}_B\bot\end{aligned}$$

According to Lemma 2.1 and Property 2.3, there is (essentially) one unique pair of structures e_A, e_B such that $e_A, e_B \models \kappa$: e_B is the empty set, and e_A contains all tuples (e'_B, w') such that

- in w', A is holding (only) c and B is holding (only) one arbitrary other card c', that is, for some name $c' \neq c$, $H_A(n) \in w'$ iff $n = c$, and $H_B(n) \in w'$ iff $n = c'$, and
- e'_B contains all tuples $(\{\}, w'')$ such that in w'', B is holding (only) c' (that same card as in w') and A is holding (only) one arbitrary other card c'', that is, for some name $c'' \neq c'$, $H_B(n) \in w''$ iff $n = c'$, and $H_A(n) \in w''$ iff $n = c''$.

To illustrate how the semantics works, we prove a few queries. The first two are obvious consequences. The third query is more refined and involves an interesting *de re* versus *de dicto* distinctions.

(1) A knows that B is not holding c:

$$\models \kappa \supset \mathbf{K}_A\neg H_B(c)$$

It suffices to show $e_A \models \mathbf{K}_A\neg H_B(c)$ for e_A described above. Indeed, for all $(e'_B, w') \in e_A$, $H_B(c) \notin w$, so the query holds.

(2) For all other cards, A does not know whether B holds them:

$$\models \kappa \supset \forall x\big(x \neq c \supset \neg\big(\mathbf{K}_A H_B(x) \lor \mathbf{K}_A\neg H_B(x)\big)\big)$$

Let $n \neq c$ be a name. We need to show that $e_A \not\models \mathbf{K}_A H_B(n)$ and $e_A \not\models \mathbf{K}_A\neg H_B(n)$. Both are true because there is some $(e'_B, w') \in e_A$ with $H_B(n) \in w$ and some with $H_B(n) \notin w$.

(3) A knows that B is holding a card whose identity is unknown to A but known to B:

$$\models \kappa \supset \mathbf{K}_A\exists x\big(H_B(x) \land \neg\mathbf{K}_A H_B(x) \land \mathbf{K}_B H_B(x)\big)$$

We show $e_A, e'_B, w' \models \exists x\big(H_B(x) \land \neg\mathbf{K}_A H_B(x) \land \mathbf{K}_B H_B(x)\big)$ for all $(e'_B, w') \in e_A$. Let $(e'_B, w') \in e_A$. Let c' be the unique name such that $H_B(c') \in w'$. It remains to be shown that $e_A \not\models \mathbf{K}_A H_B(c')$ and $e'_B \models \mathbf{K}_B H_B(c')$. The former holds since there are $(e''_B, w'') \in e_A$ such that $H_B(c') \notin w''$. The latter holds since for all $(e''_A, w'') \in e'_B$, $H_B(c') \in w''$.

Note the distinction between knowing *that* versus *which* in the last example: $\mathbf{K}_A\exists x H_B(x)$ means that A knows that B is holding some card, and $\neg\mathbf{K}_A H_B(x)$ adds A does not know that B is holding the specific card x (and analogously, $\mathbf{K}_B H_B(x)$ says that B does know).

4 REPRESENTATION THEOREM

In this section we devise a reduction from reasoning in first-order multi-agent knowledge bases to objective first-order reasoning. In terms of computability theory, this reduction is a Turing reduction because it assumes an oracle for objective first-order validity.

Roughly, the reduction works by slicing up the knowledge base and query at the modal operators and transforming these subformulas into objective formulas. The main ideas are:

- As far as the reduction of the knowledge base is concerned, we need to eliminate nested $\mathbf{O}_a$ operators. To this end, we use auxiliary predicates that are subject to additional axioms to mimic part of the behaviour of only-knowing.
- The query is reduced by evaluating each epistemic subformula with respect to the knowledge base reduction. The appropriate auxiliary predicates are then used to encode which only-knowings would prove the query subformula.
- Both reductions call the objective validity oracle. To deal with free variables (which may come from quantifying-in in the knowledge base or query) we use a mechanism to finitely represent the instances for which the formula is valid.

For the rest of the section, let κ be a knowledge base and λ a query. The proofs for this section are provided in Appendix A.

4.1 Known Instances of a Formula

As first component of the reduction, we introduce (a slight variant of) Levesque's $\mathsf{RES}[\![\phi]\!]$ operator, which takes an objective formula ϕ, potentially with free variables and returns a new formula that describes for which, if any, standard names substituted for the free variables the input formula is valid.

While there are infinitely many standard names, by the following lemma it suffices to use a single fresh standard name to represent all the infinitely many names that do not occur in ϕ.

Lemma 4.1 (Levesque [12, 14]) *Let ϕ be an objective formula with one free variable x and let m, n be names that do not occur in ϕ. Then $\models \phi^x_m$ iff $\models \phi^x_n$.*

The proof is straightforward: an induction on $|\phi|$ shows that a world satisfies ϕ^x_m iff the world with m, n swapped satisfies ϕ^x_n.

Levesque's $\mathsf{RES}[\![\phi]\!]$ operator employs this lemma to recursively eliminate free variables and returns a formula that only involves equalities over the variables and standard names that occur ϕ.

Definition 4.2 (Levesque [12, 14]) Let ϕ be an objective formula. Then $\mathsf{RES}[\![\phi]\!]$ is defined as follows:

- if ϕ contains no free variables:

$$\mathsf{RES}[\![\phi]\!] \;=\; \begin{cases}\top & \text{if } \models \phi\\ \bot & \text{otherwise}\end{cases}$$

- if ϕ contains a free variable x, $n_1, \ldots, n_k$ are the names in ϕ, and n' is an arbitrary fresh name:

$$\begin{aligned}\mathsf{RES}[\![\phi]\!] \;=\; &\big(\mathsf{RES}[\![\phi^x_{n_1}]\!] \land x = n_1\big) \lor \ldots \lor\\ &\big(\mathsf{RES}[\![\phi^x_{n_k}]\!] \land x = n_k\big) \lor\\ &\big(\mathsf{RES}[\![\phi^x_{n'}]\!]^{n'}_x \land x \neq n_1 \land \ldots \land x \neq n_k\big)\end{aligned}$$

The following lemma shows that the formula $\mathsf{RES}[\![\phi]\!]$ precisely describes the instances for which ϕ is valid.

Lemma 4.3 (Levesque [12, 14]) *Let ϕ be an objective formula with free variables $\vec{x}$, and let $\vec{m}$ be names. Then:*

$$\models \phi^{\vec{x}}_{\vec{m}} \textit{ iff } \models \mathsf{RES}\llbracket\phi\rrbracket^{\vec{x}}_{\vec{m}}$$

The proof is by induction on the number of free variables in ϕ. The first k disjuncts of $\mathsf{RES}\llbracket\phi\rrbracket$ cover the case where m_i occurs in ϕ. Otherwise, only the last disjunct of $\mathsf{RES}\llbracket\phi\rrbracket$ can become true and the equivalence follows by Lemma 4.1.

Example

As an example, consider $H_A(x) \equiv H_A(z)$, which contains no names itself but two free variables. Under the $\mathsf{RES}\llbracket\cdot\rrbracket$ operator this formula evaluates to the following formula:[2]

$$\begin{aligned}
& \mathsf{RES}\llbracket H_A(x) \equiv H_A(z)\rrbracket \\
= \;& \mathsf{RES}\llbracket H_A(x) \equiv H_A(n')\rrbracket^{n'}_{z} \\
= \;& \big(\mathsf{RES}\llbracket H_A(n') \equiv H_A(n')\rrbracket \wedge x = n'\big)^{n'}_{z} \vee \\
& \big(\mathsf{RES}\llbracket H_A(n'') \equiv H_A(n')\rrbracket^{n''}_{x} \wedge x \neq n'\big)^{n'}_{z} \\
= \;& \big(\top \wedge x = n'\big)^{n'}_{z} \vee \\
& \big(\bot^{n''}_{x} \wedge x \neq n'\big)^{n'}_{z} \\
= \;& \big(\top \wedge x = z\big) \vee \\
& \big(\bot \wedge x \neq z\big) \\
\overset{\text{eq}}{=} \;& x = z
\end{aligned}$$

Here, the $\mathsf{RES}\llbracket\cdot\rrbracket$ operator first picks variable z and substitutes the fresh name n' for it (the order in which variables and which fresh names are selected is not specified in Definition 4.2 and can be chosen arbitrarily). The recursive call to $\mathsf{RES}\llbracket\cdot\rrbracket$ then eliminates x by trying both n' as well as another fresh name n''.

Observe that these freshly introduced names n', n'' are substituted again with the original variables z, x, so that the resulting formula does not contain any new names.

4.2 Knowledge Base Reduction

One step of our reduction is to slice up the knowledge base and translate the only-known subformulas to objective formulas. In order to ensure that the assertional force of the original formula α is identical to the new objective formula, we will substitute the subjective subformulas $\mathbf{O}_{a'}\beta$ of α with auxiliary predicates.

Additional axioms will impose on these auxiliary predicates the same constraints that are inherent to only-knowing. In particular, we need to ensure that Property 2.2 carries over to the auxiliary predicates: an agent can only-know at most one formula, modulo logical equivalence.

For the rest of this section, we introduce for every agent a' and every $(\alpha, \vec{x}) \in \bigcup_{\vec{a}} \Sigma_{\vec{a}\cdot a'}(\kappa)$ a fresh $|\vec{x}|$-ary auxiliary predicate $P_{a',\alpha}$. Moreover, let $*$ be a function that maps the variables in κ, λ to fresh variables. We write α^* for the formula α with all with all variables x replaced with x^*.

Definition 4.4 The *knowledge base reduction* $\langle\!\langle \vec{a}, \alpha \rangle\!\rangle$ of an $\alpha \in \Sigma_{\vec{a}}(\kappa)$ is the conjunction $(\alpha^\sharp \wedge \Omega_{\vec{a}})$, where $\alpha^\sharp$ is the formula obtained by replacing every objective occurrence of $\mathbf{O}_{a'}\beta$ with free variables $\vec{y}$ in α with the auxiliary predicate $P_{a',\beta}(\vec{y})$, and $\Omega_{\vec{a}}$ is the conjunction of formulas

$$\forall \vec{y} \forall \vec{z}^* \left(P_{a',\beta}(\vec{y}) \supset \left(P_{a',\gamma}(\vec{z}^*) \equiv \mathsf{RES}\llbracket \langle\!\langle \vec{a}\cdot a', \beta \rangle\!\rangle \equiv \langle\!\langle \vec{a}\cdot a', \gamma \rangle\!\rangle^* \rrbracket\right)\right)$$

over all agents a' and $(\beta, \vec{y}), (\gamma, \vec{z}) \in \Sigma_{\vec{a}\cdot a'}(\kappa)$.

Note that by construction, $\langle\!\langle \vec{a}, \alpha \rangle\!\rangle$ is an objective formula, and that $\mathsf{RES}\llbracket\cdot\rrbracket$ is only applied to recursively reduced formulas. That way, the oracle is called for objective formulas only. The base case for the recursion is when $\bigcup_{a'} \Sigma_{\vec{a}\cdot a'}(\kappa) = \{\}$ and thus $\Omega_{\vec{a}} = \top$.

Also observe how $\Omega_{\vec{a}}$ expresses Property 2.2 for the auxiliary predicates $P_{a',\beta}$: when a' only-knows β ($P_{a',\beta}$ is true), then a' only-knows γ ($P_{a',\gamma}$ holds) iff (the reductions of) β and γ are equivalent.

The following lemma states that the original and the reduced formula are equivalent as far as objective information is concerned. We let $w \approx_{\vec{a}} w'$ iff w and w' agree on all but the auxiliary predicates, that is, for all $P \notin \bigcup_{a'}\{P_{a',\alpha} \mid \alpha \in \Sigma_{\vec{a}\cdot a'}(\kappa)\}$, $P(\vec{n}) \in w$ iff $P(\vec{n}) \in w'$. Moreover, we let $e_A, e_B, w \approx_{\vec{a}} w'$ iff $w \approx_{\vec{a}} w'$ and the auxiliary predicates reflect what is only-known, that is, for all agents a', for all $(\alpha, \vec{x}) \in \Sigma_{\vec{a}\cdot a'}(\kappa)$, $e_A, e_B \models \mathbf{O}_{a'}\alpha^{\vec{x}}_{\vec{n}}$ iff $w' \models P_{a',\alpha}(\vec{n})$.

Lemma 4.5 *Let $(\alpha, \vec{x}), (\beta, \vec{y}) \in \Sigma_{\vec{a}}(\kappa)$.*

(i) For all e_A, e_B, w with $e_A, e_B, w \models \alpha^{\vec{x}}_{\vec{m}}$, there is some w' such that $w' \models \langle\!\langle \vec{a}, \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}}$ and $e_A, e_B, w \approx_{\vec{a}} w'$.

(ii) For all w with $w \models \langle\!\langle \vec{a}, \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}}$, there is some e_A, e_B such that $e_A, e_B, w \models \alpha^{\vec{x}}_{\vec{m}}$ and $e_A, e_B, w \approx_{\vec{a}} w$.

(iii) $\models \alpha^{\vec{x}}_{\vec{m}} \equiv \beta^{\vec{y}}_{\vec{n}}$ iff $\models \langle\!\langle \vec{a}, \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}} \equiv \langle\!\langle \vec{a}, \beta \rangle\!\rangle^{\vec{y}}_{\vec{n}}$.

The proof is by backward induction on $|\vec{a}|$ and can be found in Appendix A. An easy consequence is that testing A-determinacy can be decided using non-modal reasoning.

Corollary 4.6 *Let $(\alpha, \vec{x}) \in \Sigma_{\vec{a}}(\kappa)$. Then:*

$$\alpha \textit{ is } a'\textit{-determinate iff } \models \forall \vec{x} \left(\langle\!\langle \vec{a}, \alpha \rangle\!\rangle \supset \bigvee\nolimits_{(\beta,\vec{y}) \in \Sigma_{a'}(\alpha)} \exists \vec{y} P_{a',\beta}(\vec{y})\right)$$

Example

Before we proceed with the query reduction, we illustrate the knowledge base reduction with our running example knowledge base κ:

$$\mathbf{O}_A \left(H_A(c) \wedge \Phi \wedge \forall x \left(H_B(x) \supset \mathbf{O}_B \left(H_B(x) \wedge \Phi \wedge \mathbf{O}_A \bot\right)\right)\right) \wedge \mathbf{O}_B \bot$$

We begin with the reduction of κ itself. Let μ denote the formula within the outer $\mathbf{O}_A$, that is, $H_A(c) \wedge \Phi \wedge \forall x \left(H_B(x) \supset \mathbf{O}_B \left(H_B(x) \wedge \Phi \wedge \mathbf{O}_A \bot\right)\right)$. Applying Definition 4.4 to κ replaces $\mathbf{O}_A \mu$ with $P_{A,\mu}$ and $\mathbf{O}_B \bot$ with $P_{B,\bot}$ and adds the axioms $\Omega_{\langle\rangle}$:

$$\begin{aligned}
\langle\!\langle \langle\rangle, \kappa \rangle\!\rangle \;=\; & \big(P_{A,\mu} \wedge P_{B,\bot} \wedge \\
& \quad \big(P_{A,\mu} \supset \big(P_{A,\mu} \equiv \mathsf{RES}\llbracket \langle\!\langle A, \mu \rangle\!\rangle \equiv \langle\!\langle A, \mu \rangle\!\rangle \rrbracket\big)\big) \wedge \\
& \quad \big(P_{B,\bot} \supset \big(P_{B,\bot} \equiv \mathsf{RES}\llbracket \langle\!\langle B, \bot \rangle\!\rangle \equiv \langle\!\langle B, \bot \rangle\!\rangle \rrbracket\big)\big)\big) \\
\overset{\text{eq}}{=} \;& P_{A,\mu} \wedge P_{B,\bot} \qquad \text{(since both } \mathsf{RES}\llbracket\cdot\rrbracket \text{ evaluate to } \top\text{)}
\end{aligned}$$

Next, we reduce what A only-knows, that is, μ. Let η denote the formula within $\mathbf{O}_B$ in μ, that is, $H_B(x) \wedge \Phi \wedge \mathbf{O}_A \bot$. Notice the free variable x in η. $\mathbf{O}_B \eta$ will be replaced with $P_{B,\eta}(x)$ and Ω_A will be non-trivial. Let z be x^*. Then applying Definition 4.4 obtains:

$$\begin{aligned}
\langle\!\langle A, \mu \rangle\!\rangle \;=\; & \big(H_A(c) \wedge \Phi \wedge \forall x \big(H_B(x) \supset P_{B,\eta}(x)\big) \wedge \\
& \quad \forall x \forall z \big(P_{B,\eta}(x) \supset \\
& \qquad \big(P_{B,\eta}(z) \equiv \mathsf{RES}\llbracket \langle\!\langle A\cdot B, \eta \rangle\!\rangle \equiv \langle\!\langle A\cdot B, \eta \rangle\!\rangle^{x}_{z} \rrbracket\big)\big)\big)
\end{aligned}$$

[2] To improve readability, we will sometimes simplify formulas (in an equivalence-preserving way) in the examples; we mark these steps with the $\overset{\text{eq}}{=}$.

To evaluate the $\mathsf{RES}[\![\cdot]\!]$ expression, we first need to reduce η:

$$\begin{aligned} \langle\!\langle A{\cdot}B, \eta\rangle\!\rangle &= \big(H_B(x) \land \Phi \land P_{A,\bot} \land \\ &\qquad \big(P_{A,\bot} \supset \big(P_{A,\bot} \equiv \mathsf{RES}[\![\langle\!\langle A{\cdot}B{\cdot}B,\bot\rangle\!\rangle \equiv \langle\!\langle A{\cdot}B{\cdot}B,\bot\rangle\!\rangle]\!]\big)\big)\big) \\ &\overset{\text{eq}}{=} H_B(x)\land\Phi\land P_{A,\bot} \qquad \text{(since } \mathsf{RES}[\![\cdot]\!] \text{ evaluates to } \top) \end{aligned}$$

The $\mathsf{RES}[\![\cdot]\!]$ expression in $\langle\!\langle A,\mu\rangle\!\rangle$ then evaluates similarly to the example from Section 4.1:

$$\begin{aligned} &\mathsf{RES}[\![\langle\!\langle A{\cdot}B,\eta\rangle\!\rangle \equiv \langle\!\langle A{\cdot}B,\eta\rangle\!\rangle^{x}_{z}]\!] \\ =\ &\mathsf{RES}[\![H_B(x)\land\Phi\land P_{A,\bot} \equiv H_B(z)\land\Phi\land P_{A,\bot}]\!] \\ =\ &(x = z \land \top) \lor (x \neq z \land \bot) \\ \overset{\text{eq}}{=}\ &x = z \end{aligned}$$

Now we can finally fully evaluate the reduction of μ:

$$\begin{aligned} \langle\!\langle A,\mu\rangle\!\rangle &= \big(H_A(c)\land\Phi\land\forall x\big(H_B(x)\supset P_{B,\eta}(x)\big)\land \\ &\qquad \forall x\forall z\big(P_{B,\eta}(x)\supset\big(P_{B,\eta}(z)\equiv x = z\big)\big)\big) \end{aligned}$$

4.3 Query Reduction

The other step of our reduction is to unwind the modal operators in the query and recursively replace the nested subformulas with their known instances. The known instances are determined relative to the reductions of the relevant subformulas $\mathbf{O}_{a'}\gamma$ in the knowledge base and conditioned on $P_{a',\gamma}(\vec{z})$, that is, on the fact that a' only-knows γ.

Definition 4.7 The *query reduction* $\|\vec{a},\alpha\|$ of an $\alpha\in\Sigma_{\vec{a}}(\lambda)$ is the formula obtained by replacing every objective occurrence of $\mathbf{K}_{a'}\beta$ in α with the disjunction of formulas

$$\exists\vec{z}^*\,\big(P_{a',\gamma}(\vec{z}^*)\land\mathsf{RES}[\![(P_{a',\gamma}(\vec{z}^*)\land\langle\!\langle\vec{a}{\cdot}a',\gamma\rangle\!\rangle^*)\supset\|\vec{a}{\cdot}a',\beta\|]\!]\big)$$

over all $(\gamma,\vec{z})\in\Sigma_{\vec{a}\cdot a'}(\kappa)$.

It is immediate that since all $\mathbf{K}_{a'}\beta$ in α are replaced with objective formulas, the end result $\|\vec{a},\alpha\|$ is an objective formula, too. Notice that $\|\vec{a},\alpha\|$ recursively evaluates $\|\vec{a},\beta\|$ before applying $\mathsf{RES}[\![\cdot]\!]$ to ensure that our oracle is only applied to objective formulas. The base case of this recursion is when α does not contain any formula.

The next lemma establishes an equivalence between modal and non-modal reasoning by way of our reductions.

Lemma 4.8 *Let* $(\alpha,\vec{x})\in\Sigma_{\vec{a}\cdot a'}(\kappa)$ *and* $(\beta,\vec{y})\in\Sigma_{\vec{a}\cdot a'}(\lambda)$. *Then:*

$$\models \mathbf{O}_{a'}\alpha^{\vec{x}}_{\vec{m}}\supset\mathbf{K}_{a'}\beta^{\vec{y}}_{\vec{n}} \textit{ iff } \models P_{a',\alpha}(\vec{m})\land\langle\!\langle\vec{a}{\cdot}a',\alpha\rangle\!\rangle^{\vec{x}}_{\vec{m}}\supset\|\vec{a}{\cdot}a',\beta\|^{\vec{y}}_{\vec{n}}$$

The proof is by induction on the depth of β and can be found in Appendix A.

With Lemma 4.8 in hand, it is not surprising that the reductions can also be applied to the entire knowledge base κ and query λ to eliminate the modal aspect in deciding the validity $\kappa\supset\lambda$. Our representation theorem states exactly this.

Theorem 4.9 (Representation theorem)

$$\models\kappa\supset\lambda \textit{ iff } \models\langle\!\langle\langle\rangle,\kappa\rangle\!\rangle\supset\|\langle\rangle,\lambda\|$$

The proof is essentially a simpler variant of the proof of Lemma 4.8 and can be found in Appendix A.

Example

To illustrate how this reduction works, let us consider the third sample query from Section 3:

$$\mathbf{K}_A\exists x\big(H_B(x)\land\neg\mathbf{K}_AH_B(x)\land\mathbf{K}_BH_B(x)\big)$$

Let us call the query λ, and let δ denote the formula within the outer $\mathbf{K}_A$, that is, $\exists x\big(H_B(x)\land\neg\mathbf{K}_AH_B(x)\land\mathbf{K}_BH_B(x)\big)$, and let μ and η be as in the example above. Applying Definition 4.7 yields:

$$\|\langle\rangle,\lambda\| = \big(P_{A,\mu}\land\mathsf{RES}[\![P_{A,\mu}\land\langle\!\langle A,\mu\rangle\!\rangle\supset\|A,\delta\|]\!]\big)$$

To evaluate this further, we first determine the reduction of δ and then evaluate the $\mathsf{RES}[\![\cdot]\!]$ expression. By Definition 4.7 we obtain:

$$\begin{aligned} \|A,\delta\| &= \exists x\big(H_B(x)\land \\ &\qquad \neg\big(P_{A,\mu}\land\mathsf{RES}[\![P_{A,\mu}\land\langle\!\langle A,\mu\rangle\!\rangle\supset\|A,H_B(x)\|]\!]\big)\land \\ &\qquad \exists z\big(P_{B,\eta}(z)\land \\ &\qquad\qquad \mathsf{RES}[\![P_{B,\eta}(z)\land\langle\!\langle A{\cdot}B,\eta\rangle\!\rangle^{x}_{z}\supset\|A{\cdot}B,H_B(x)\|]\!]\big)\big) \end{aligned}$$

The reductions $\|A,H_B(x)\|$ and $\|A{\cdot}B,H_B(x)\|$ are $H_B(x)$. The first $\mathsf{RES}[\![\cdot]\!]$ expression in $\|A,\delta\|$ evaluates to $\bot$ since there is no n such that $P_{A,\mu}\land\langle\!\langle A,\mu\rangle\!\rangle\supset H_B(n)$ is valid. The second one evaluates to:

$$\begin{aligned} &\mathsf{RES}[\![P_{B,\eta}(z)\land\langle\!\langle A{\cdot}B,\eta\rangle\!\rangle^{x}_{z}\supset\|A{\cdot}B,H_B(x)\|]\!] \\ =\ &\mathsf{RES}[\![P_{B,\eta}(z)\land H_B(z)\land\Phi\land P_{A,\bot}\supset H_B(x)]\!] \\ =\ &(x = z\land\top)\lor(x\neq z\land\bot) \\ \overset{\text{eq}}{=}\ &x = z \end{aligned}$$

Now we can continue evaluating $\|A,\delta\|$ as follows:

$$\begin{aligned} \|A,\delta\| &= \exists x\big(H_B(x)\land\neg(P_{A,\mu}\land\bot)\land\exists z\big(P_{B,\eta}(z)\land x = z\big)\big) \\ &\overset{\text{eq}}{=} \exists x\big(H_B(x)\land P_{B,\eta}(x)\big) \end{aligned}$$

Now that we have $\langle\!\langle A,\mu\rangle\!\rangle$ and $\|A,\delta\|$, we are ready to evaluate the $\mathsf{RES}[\![\cdot]\!]$ expression in $\|\langle\rangle,\lambda\|$:

$$\begin{aligned} &\mathsf{RES}[\![P_{A,\mu}\land\langle\!\langle A,\mu\rangle\!\rangle\supset\|A,\delta\|]\!] \\ =\ &\mathsf{RES}[\![\big(P_{A,\mu}\land H_A(c)\land\Phi\land\forall x\big(H_B(x)\supset P_{B,\eta}(x)\big)\land \\ &\qquad \forall x\forall z\big(P_{B,\eta}(x)\supset\big(P_{B,\eta}(z)\equiv x = z\big)\big)\supset \\ &\qquad \exists x\big(H_B(x)\land P_{B,\eta}(x)\big)]\!] \\ =\ &\top \qquad \text{(since } \Phi \text{ implies } \exists xH_B(x)) \end{aligned}$$

Thus we have finally fully determined $\langle\!\langle\langle\rangle,\kappa\rangle\!\rangle$ and $\|\langle\rangle,\lambda\|$. The formula $\kappa\supset\lambda$ therefore eventually reduced to:

$$\big(\langle\!\langle\langle\rangle,\kappa\rangle\!\rangle\supset\|\langle\rangle,\lambda\|\big) = \big((P_{A,\mu}\land P_{B,\bot})\supset(P_{A,\mu}\land\top)\big)$$

This formula is obviously valid. Therefore, by Theorem 4.9, $\models\kappa\supset\lambda$.

5 CONCLUSION AND DISCUSSION

We showed within Levesque's framework of only-knowing how reasoning in first-order multi-agent knowledge bases can be Turing-reduced to non-epistemic reasoning with an oracle for objective first-order validity.

The reduction proceeds by slicing up the knowledge base at the modal operators and introducing auxiliary predicates to objectively represent the nested modalities. In the query, epistemic subformulas are replaced with the appropriate auxiliary predicates to express which fragments of the knowledge base can entail the query. It is remarkable that by leveraging Levesque's $\mathsf{RES}[\![\cdot]\!]$ operator as an interface to the objective first-order validity oracle, the auxiliary-predicates approach suffices to handle variables quantified into a modal context both in the knowledge base and in the query.

An important feature of our knowledge bases is that agents may have incomplete knowledge about what the other agents only-know. We only require that an agent can enumerate all – perhaps infinitely many – alternatives about what the other agent's may know. In particular, this allows for powerful *de dicto* versus *de re* distinctions.

On the other hand, our knowledge bases do not allow us to combine only-knowing with plain knowing to say, for example, that A only-knows that B knows but not necessarily only-knows a formula. While this restriction is not particular to the multi-agent case and applies to Levesque's representation theorem too, we plan to study if and how this restriction can be lifted. This would also mean the representation theorem would extend to non-monotonic reasoning [10]. Another interesting open question is how common knowledge [2] affects the representation theorem and whether the multi-agent representation theorem also combines well with other variants of the original representation theorem [16–18].

In this paper we only considered two agents. Interestingly, it is easy to generalise the logic as well as our representation theorem to (up to) infinitely many agents, represented as first-order objects that can be quantified over. This will be useful in practical use-cases.

Also on the practical side of future work, we plan to study how the representation carries over to limited-belief logic and integrate it into the Limbo reasoning system [17], which trades completeness for computational tractability [4]. The main challenge here is to first find an appropriate limited-belief version of multi-agent only-knowing which may or may not be compatible with this paper's results. Another path towards implementation is via off-the-shelf provers like [8] where built-in integer sorts could be leveraged to simulate standard names. This option seems attractive for off-line reasoning as in verification tasks.

A PROOF OF THEOREM 4.9

Here we prove the results from Section 4. Let κ be a knowledge base and λ be a query.

Lemma A.1 *Let α be a-objective.*
Then $\mathbf{O}_a\alpha \land \mathbf{K}_a\beta$ is satisfiable iff $\models \mathbf{O}_a\alpha \supset \mathbf{K}_a\beta$.

Proof. For the only-if direction, suppose $\mathbf{O}_a\alpha \land \mathbf{K}_a\beta$ is satisfiable and $e_a \models \mathbf{O}_a\alpha$. By Lemma 2.1 and assumption, $e_a \models \mathbf{K}_a\beta$. Conversely, suppose $\models \mathbf{O}_a\alpha \supset \mathbf{K}_a\beta$ and let e_a be as in Lemma 2.1. By Lemma 2.1, $e_a \models \mathbf{O}_a\alpha$ and by assumption, $e_a \models \mathbf{K}_a\beta$. □

Lemma 4.5 *Let $(\alpha, \vec{x}), (\beta, \vec{y}) \in \Sigma_{\vec{a}}(\kappa)$.*

(i) For all e_A, e_B, w with $e_A, e_B, w \models \alpha^{\vec{x}}_{\vec{m}}$, there is some w' such that $w' \models \langle\!\langle \vec{a}, \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}}$ and $e_A, e_B, w \approx_{\vec{a}} w'$.

(ii) For all w with $w \models \langle\!\langle \vec{a}, \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}}$, there is some e_A, e_B such that $e_A, e_B, w \models \alpha^{\vec{x}}_{\vec{m}}$ and $e_A, e_B, w \approx_{\vec{a}} w$.

(iii) $\models \alpha^{\vec{x}}_{\vec{m}} \equiv \beta^{\vec{y}}_{\vec{n}}$ iff $\models \langle\!\langle \vec{a}, \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}} \equiv \langle\!\langle \vec{a}, \beta \rangle\!\rangle^{\vec{y}}_{\vec{n}}$.

Proof. The proof is by backward induction on $|\vec{a}|$. For the base case, suppose that $\Sigma_{\vec{a}\cdot a'}(\kappa) = \{\}$ for all a'. Thus α, β are objective and $\langle\!\langle \vec{a}, \alpha \rangle\!\rangle = \alpha$ and $\langle\!\langle \vec{a}, \beta \rangle\!\rangle = \beta$. Thus (i), (ii), (iii) hold immediately.

For the induction step, suppose the lemma holds $\vec{a}\cdot a'$ for all $a' \in \{A, B\}$ and consider $\vec{a}$.

For (i), let $e_A, e_B, w \models \alpha^{\vec{x}}_{\vec{m}}$. Let w' be such that $w \approx_{\vec{a}} w'$ and for every $(\gamma, \vec{z}) \in \Sigma_{\vec{a}\cdot a'}(\kappa)$, $P_{a',\gamma}(\vec{o}) \in w'$ iff $e_A, e_B \models \mathbf{O}_{a'}\gamma^{\vec{z}}_{\vec{o}}$. A simple subinduction on $|\alpha|$ shows that $e_A, e_B, w \models \alpha^{\vec{x}}_{\vec{m}}$ iff $w' \models \alpha^{\sharp\vec{x}}_{\vec{m}}$, where $\alpha^{\sharp}$ is as in Definition 4.4. Moreover, for all $(\gamma, \vec{z}), (\gamma', \vec{z}') \in \Sigma_{\vec{a}\cdot a'}(\kappa)$, if $e_A, e_B \models \mathbf{O}_{a'}\gamma^{\vec{z}}_{\vec{o}}$, then $e_A, e_B \models \mathbf{O}_{a'}\gamma'^{\vec{z}'}_{\vec{o}'}$ iff $\models \mathbf{O}_{a'}\gamma^{\vec{z}}_{\vec{o}} \equiv \mathbf{O}_{a'}\gamma'^{\vec{z}'}_{\vec{o}'}$ iff (by Property 2.2) $\models \gamma^{\vec{z}}_{\vec{o}} \equiv \gamma'^{\vec{z}'}_{\vec{o}'}$ iff (by induction for (iii)) $\models \langle\!\langle \vec{a}\cdot a', \gamma \rangle\!\rangle^{\vec{z}}_{\vec{o}} \equiv \langle\!\langle \vec{a}\cdot a', \gamma' \rangle\!\rangle^{\vec{z}'}_{\vec{o}'}$ iff $\models \mathsf{RES}[\![\langle\!\langle \vec{a}\cdot a', \gamma \rangle\!\rangle \equiv \langle\!\langle \vec{a}\cdot a', \gamma' \rangle\!\rangle^{*}]\!]^{\vec{z}\vec{z}'^{*}}_{\vec{o}\vec{o}'}$. Thus and by construction of $\langle\!\langle \vec{a}, \alpha \rangle\!\rangle$, $w' \models \langle\!\langle \vec{a}, \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}}$ and $e, w \approx_{\vec{a}} w'$.

For (ii), let $w \models \langle\!\langle \vec{a}, \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}}$. If $P_{A,\gamma}(\vec{o}) \in w$ for some $(\gamma, \vec{z}) \in \Sigma_{\vec{a}\cdot A}(\kappa)$ and some $\vec{o}$, let $e_A = \{(e'_B, w') \mid (\gamma, \vec{z}) \in \Sigma_{\vec{a}\cdot A}(\kappa), P_{A,\gamma}(\vec{o}) \in w, e'_B, w' \models \gamma^{\vec{z}}_{\vec{o}}\}$, and otherwise, let e_A be such that $e_A \neq \{(e'_B, w') \mid e'_B, w' \models \gamma^{\vec{z}}_{\vec{o}}\}$ for all $(\gamma, \vec{z}) \in \Sigma_{\vec{a}\cdot A}(\kappa)$ and $\vec{o}$, which exists since there are unrepresentable states [14] (choose e_B analogously).

First we show for all $(\gamma, \vec{z}) \in \Sigma_{\vec{a}\cdot A}(\kappa)$ and all names $\vec{o}$ that $P_{A,\gamma}(\vec{o}) \in w$ iff $e_A = \{(e'_B, w') \mid e'_B, w' \models \gamma^{\vec{z}}_{\vec{o}}\}$ (the argument for B is symmetric) (*).

For the only-if direction of (*), suppose $P_{A,\gamma}(\vec{o}) \in w$. The $\supseteq$ direction holds by construction. For the $\subseteq$ direction, suppose $(e'_B, w') \in e_A$. Then $e'_B, w' \models \gamma'^{\vec{z}'}_{\vec{o}'}$ for some $(\gamma', \vec{z}') \in \Sigma_{\vec{a}\cdot A}(\kappa)$ with $P_{A,\gamma'}(\vec{o}') \in w$. By construction of $\langle\!\langle \vec{a}, \alpha \rangle\!\rangle$, $\models \mathsf{RES}[\![\langle\!\langle \vec{a}\cdot A, \gamma \rangle\!\rangle \equiv \langle\!\langle \vec{a}\cdot A, \gamma' \rangle\!\rangle^{*}]\!]^{\vec{z}\vec{z}'^{*}}_{\vec{o}\vec{o}'}$. By Lemma 4.3, $\models \langle\!\langle \vec{a}\cdot A, \gamma \rangle\!\rangle^{\vec{z}}_{\vec{o}} \equiv \langle\!\langle \vec{a}\cdot A, \gamma' \rangle\!\rangle^{\vec{z}'}_{\vec{o}'}$. By induction for (iii), $\models \gamma^{\vec{z}}_{\vec{o}} \equiv \gamma'^{\vec{z}'}_{\vec{o}'}$. Thus $e'_B, w' \models \gamma^{\vec{z}}_{\vec{o}}$.

For the if direction of (*), suppose $e_A = \{(e'_B, w') \mid e'_B, w' \models \gamma^{\vec{z}}_{\vec{o}}\}$. By construction of e_A, $P_{A,\gamma'}(\vec{o}') \in w$ for some $(\gamma', \vec{z}') \in \Sigma_{\vec{a}\cdot A}(\kappa)$ and some $\vec{o}'$. By the only-if direction of (*), $e_A = \{(e'_B, w') \mid e'_B, w' \models \gamma'^{\vec{z}'}_{\vec{o}'}\}$. Thus $\models \gamma'^{\vec{z}'}_{\vec{o}'} \equiv \gamma^{\vec{z}}_{\vec{o}}$. By induction for (iii), $\models \langle\!\langle \vec{a}\cdot A, \gamma' \rangle\!\rangle^{\vec{z}}_{\vec{o}} \equiv \langle\!\langle \vec{a}\cdot A, \gamma \rangle\!\rangle^{\vec{z}'}_{\vec{o}'}$. By Lemma 4.3, $\models \mathsf{RES}[\![\langle\!\langle \vec{a}\cdot A, \gamma' \rangle\!\rangle \equiv \langle\!\langle \vec{a}\cdot A, \gamma \rangle\!\rangle^{*}]\!]^{\vec{z}\vec{z}'^{*}}_{\vec{o}\vec{o}'}$. By construction of $\langle\!\langle \vec{a}, \alpha \rangle\!\rangle$, $P_{A,\gamma}(\vec{o}) \in w$.

Finally, we now prove by subinduction on $|\alpha|$ that $w \models \langle\!\langle \vec{a}, \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}}$ iff $e_A, e_B, w \models \alpha^{\vec{x}}_{\vec{m}}$. The only non-trivial case is for the case $\mathbf{O}_{a'}\gamma^{\vec{z}}_{\vec{o}}$, which holds by (*) and by Lemma 2.1. Thus $e_A, e_B, w \models \alpha^{\vec{x}}_{\vec{m}}$.

For the only-if direction of (iii), suppose $\models \alpha^{\vec{x}}_{\vec{m}} \equiv \beta^{\vec{y}}_{\vec{n}}$ and $w \models \langle\!\langle \vec{a}, \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}}$ (the case for $\langle\!\langle \vec{a}, \beta \rangle\!\rangle^{\vec{y}}_{\vec{n}}$ is analogous). By (ii), there is an e_A, e_B, w with $e_A, e_B, w \models \alpha^{\vec{x}}_{\vec{m}}$ and $e_A, e_B, w \approx_{\vec{a}} w$. By assumption, $e_A, e_B, w \models \beta^{\vec{y}}_{\vec{n}}$. A simple subinduction on $|\beta|$ shows $w \models \beta^{\sharp\vec{y}}_{\vec{n}}$. Thus and by assumption, $w \models \langle\!\langle \vec{a}, \beta \rangle\!\rangle^{\vec{y}}_{\vec{n}}$.

For the if direction of (iii), suppose $\models \langle\!\langle \vec{a}, \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}} \equiv \langle\!\langle \vec{a}, \beta \rangle\!\rangle^{\vec{y}}_{\vec{n}}$ and $e_A, e_B, w \models \alpha^{\vec{x}}_{\vec{m}}$ (the case for $\beta^{\vec{y}}_{\vec{n}}$ is analogous). By (i), there is a w' such that $w' \models \langle\!\langle \vec{a}, \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}}$ and $e_A, e_B, w \approx_{\vec{a}} w'$. By assumption, $w' \models \langle\!\langle \vec{a}, \beta \rangle\!\rangle^{\vec{y}}_{\vec{n}}$. A simple subinduction on $|\beta|$ shows $e_A, e_B, w \models \beta^{\vec{y}}_{\vec{n}}$. □

Corollary 4.6 *Let $(\alpha, \vec{x}) \in \Sigma_{\vec{a}}(\kappa)$. Then:*

α is a'-determinate iff $\models \forall\vec{x}\,\big(\langle\!\langle \vec{a}, \alpha \rangle\!\rangle \supset \bigvee_{(\beta,\vec{y})\in\Sigma_{a'}(\alpha)} \exists\vec{y}\,P_{a',\beta}(\vec{y})\big)$

Proof. For the only-if direction, suppose the left-hand side holds and $w \models \langle\!\langle \vec{a}, \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}}$. By Lemma 4.5, $e_A, e_B, w \models \alpha^{\vec{x}}_{\vec{m}}$ and $e_A, e_B, w \approx_{\vec{a}} w$ for some e_A, e_B. By assumption, for some $(\beta, \vec{y}) \in \Sigma_{a'}(\alpha) \subseteq \Sigma_{\vec{a}\cdot a'}(\kappa)$ and names $\vec{n}$, $e_{a'} \models \mathbf{O}_{a'}\beta^{\vec{y}}_{\vec{n}}$. Thus $w \models P_{a',\beta}(\vec{n})$.

Conversely, suppose the right-hand side holds and $e_A, e_B, w \models \alpha^{\vec{x}}_{\vec{m}}$. By Lemma 4.5, there is some w' such that $w' \models \langle\!\langle \vec{a}, \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}}$ and

$e_A, e_B, w \approx_{\vec{a}} w'$. By assumption, $w' \models P_{a',\beta}(\vec{n})$ for some $(\beta, \vec{y}) \in \Sigma_{a'}(\alpha) \subseteq \Sigma_{\vec{a}\cdot a'}(\kappa)$ and some names $\vec{n}$. Thus $e_{a'} \models \mathbf{O}_{a'}\beta^{\vec{y}}_{\vec{n}}$. □

Lemma A.2 *Let $(\alpha, \vec{x}) \in \Sigma_{\vec{a}\cdot a'}(\kappa)$ and $(\beta, \vec{y}) \in \Sigma_{\vec{a}\cdot a'}(\lambda)$. Then:*

$$\models P_{a',\alpha}(\vec{m}) \wedge \langle\!\langle \vec{a}\cdot a', \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}} \supset \|\vec{a}\cdot a', \beta\|^{\vec{y}}_{\vec{n}} \textit{ iff } \models P_{a',\alpha}(\vec{m}) \supset \|\vec{a}, \mathbf{K}_{a'}\beta\|^{\vec{y}}_{\vec{n}}$$

Proof. $\models P_{a',\alpha}(\vec{m}) \wedge \langle\!\langle \vec{a}\cdot a', \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}} \supset \|\vec{a}\cdot a', \beta\|^{\vec{y}}_{\vec{n}}$ iff (by Lemma 4.3) $\models \mathsf{RES}[\![P_{a',\alpha}(\vec{x}^*) \wedge \langle\!\langle \vec{a}\cdot a', \alpha \rangle\!\rangle \supset \|\vec{a}\cdot a', \beta\|]\!]^{\vec{x}^*\vec{y}}_{\vec{m}\,\vec{n}}$ iff (since $\mathsf{RES}[\![\cdot]\!]$ does not mention $P_{a',\alpha}$) $\models P_{a',\alpha}(\vec{m}) \supset \exists\vec{x}^*(P_{a',\alpha}(\vec{x}^*) \wedge \mathsf{RES}[\![P_{a',\alpha}(\vec{x}^*) \wedge \langle\!\langle \vec{a}\cdot a', \alpha \rangle\!\rangle \supset \|\vec{a}\cdot a', \beta\|]\!]^{\vec{y}}_{\vec{n}})$ iff $\models P_{a',\alpha}(\vec{m}) \supset \|\vec{a}, \mathbf{K}_{a'}\beta\|^{\vec{y}}_{\vec{n}}$. □

Lemma 4.8 *Let $(\alpha, \vec{x}) \in \Sigma_{\vec{a}\cdot a'}(\kappa)$ and $(\beta, \vec{y}) \in \Sigma_{\vec{a}\cdot a'}(\lambda)$. Then:*

$$\models \mathbf{O}_{a'}\alpha^{\vec{x}}_{\vec{m}} \supset \mathbf{K}_{a'}\beta^{\vec{y}}_{\vec{n}} \textit{ iff } \models P_{a',\alpha}(\vec{m}) \wedge \langle\!\langle \vec{a}\cdot a', \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}} \supset \|\vec{a}\cdot a', \beta\|^{\vec{y}}_{\vec{n}}$$

Proof. The proof is by induction on the nesting depth of modal operators in of β. For the base case, suppose there are no modal operators β is 0.

For the only-if direction, suppose the left-hand side holds and $w \models P_{a',\alpha}(\vec{m}) \wedge \langle\!\langle \vec{a}\cdot a', \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}}$. By Lemma 4.5, there is an e_A, e_B such that $e_A, e_B, w \models \alpha^{\vec{x}}_{\vec{m}}$ and $e_A, e_B, w \approx_{\vec{a}\cdot a'} w$. By Lemma 2.1 and assumption, $e_A, e_B, w \models \beta^{\vec{y}}_{\vec{n}}$. Since β is objective, $w \models \|\vec{a}\cdot a', \beta\|^{\vec{y}}_{\vec{n}}$.

For the if direction, suppose the right-hand side holds and $e_{a'} \models \mathbf{O}_{a'}\alpha^{\vec{x}}_{\vec{m}}$. We need to show $e_{a'} \models \mathbf{K}_{a'}\beta^{\vec{y}}_{\vec{n}}$, which is equivalent to $e_A, e_B, w \models \beta^{\vec{y}}_{\vec{n}}$ for all $(e'_{b'}, w) \in e_{a'}$. Let $(e_{b'}, w) \in e_{a'}$. By assumption, $e_A, e_B, w \models \alpha^{\vec{x}}_{\vec{m}}$. By Lemma 4.5, there is a w' with $w' \models \langle\!\langle \vec{a}\cdot a', \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}}$ and $e_A, e_B, w \approx_{\vec{a}\cdot a'} w'$. Since α is a'-objective, $\langle\!\langle \vec{a}\cdot a', \alpha \rangle\!\rangle$ does not mention $P_{a',\alpha}$, so without loss of generality, we can assume that $P_{a',\alpha}(\vec{m}) \in w'$. By assumption, $w' \models \|\vec{a}\cdot a', \beta\|^{\vec{y}}_{\vec{n}}$. Since β is objective, $e_A, e_B, w \models \beta^{\vec{y}}_{\vec{n}}$.

For the induction step, suppose the the lemma holds for all formulas with nesting depth of modal operators less than that of β.

For the only-if direction, suppose the left-hand side holds and suppose $w \models P_{a',\alpha}(\vec{m}) \wedge \langle\!\langle \vec{a}\cdot a', \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}}$. We need to show that $w \models \|\vec{a}\cdot a', \beta\|^{\vec{y}}_{\vec{n}}$. Since α is a'-objective, $\langle\!\langle \vec{a}\cdot a', \alpha \rangle\!\rangle$ does not mention $P_{a',\gamma}$ for any $\gamma \in \Sigma_{\vec{a}\cdot a'}(\kappa)$, so we can assume without loss of generality for all $(\gamma, \vec{z}) \in \Sigma_{\vec{a}\cdot a'}(\kappa)$ and names $\vec{o}$, that $P_{a',\gamma}(\vec{o}) \in w$ iff $\models \mathbf{O}_{a'}\alpha^{\vec{x}}_{\vec{m}} \equiv \mathbf{O}_{a'}\gamma^{\vec{z}}_{\vec{o}}$. By Lemmas 2.1 and 4.5, there is an e_A, e_B such that $e_A, e_B \models \mathbf{O}_{a'}\alpha^{\vec{x}}_{\vec{m}}$ and $e_A, e_B, w \models \alpha^{\vec{x}}_{\vec{m}}$ and $e_A, e_B, w \approx_{\vec{a}\cdot a'} w$ and, by construction of w, also $e_A, e_B, w \approx_{\vec{a}} w$. By assumption, $e_A, e_B, w \models \beta^{\vec{y}}_{\vec{n}}$. We show by subinduction on $|\beta|$ that $e_A, e_B, w \models \beta^{\vec{y}}_{\vec{n}}$ iff $w \models \|\vec{a}\cdot a', \beta\|^{\vec{y}}_{\vec{n}}$.

The only non-trivial case is to show for subformulas $(\beta', \vec{y}') \in \Sigma_{\vec{a}\cdot a'\cdot a''}(\kappa)$ that $e_A, e_B \models \mathbf{K}_{a''}\beta'^{\vec{y}'}_{\vec{n}'}$ iff $w \models \|\vec{a}\cdot a', \mathbf{K}_{a''}\beta'\|^{\vec{y}'}_{\vec{n}'}$. Note that since either $a' = a''$ or α is a''-determinate, there is a $(\alpha', \vec{x}') \in \Sigma_{\vec{a}\cdot a'\cdot a''}(\kappa)$ and names $\vec{m}'$ with $e_A, e_B \models \mathbf{O}_{a''}\alpha'^{\vec{x}'}_{\vec{m}'}$ (*). Moreover, note that if $w \models \|\vec{a}\cdot a', \mathbf{K}_{a''}\beta'\|^{\vec{y}'}_{\vec{n}'}$, then $w \models P_{a'',\alpha''}(\vec{m}'')$ for some $(\alpha'', \vec{x}'') \in \Sigma_{\vec{a}\cdot a'\cdot a''}(\kappa)$, which by $e_A, e_B, w \approx_{\vec{a}} w$ and $e_A, e_B, w \approx_{\vec{a}\cdot a'} w$ implies that $e_A, e_B \models \mathbf{O}_{a''}\alpha''^{\vec{x}''}_{\vec{m}''}$, so by Property 2.2, $e_A, e_B \models \mathbf{O}_{a''}\alpha''^{\vec{x}''}_{\vec{m}''}$, and again by $e_A, e_B, w \approx_{\vec{a}} w$ and $e_A, e_B, w \approx_{\vec{a}\cdot a'} w$, $w \models P_{a'',\alpha'}(\vec{m})$ (**). Then $e_A, e_B \models \mathbf{K}_{a''}\beta'^{\vec{y}'}_{\vec{n}'}$ iff (by Lemma A.1 and (*)) $\models \mathbf{O}_{a''}\alpha'^{\vec{x}'}_{\vec{m}'} \supset \mathbf{K}_{a''}\beta'^{\vec{y}'}_{\vec{n}'}$ iff (by main induction) $\models P_{a'',\alpha'}(\vec{m}') \wedge \langle\!\langle \vec{a}\cdot a'\cdot a'', \alpha' \rangle\!\rangle^{\vec{x}'}_{\vec{m}'} \supset \|\vec{a}\cdot a'\cdot a'', \beta'\|^{\vec{y}'}_{\vec{n}'}$ iff (by Lemma A.2) $\models P_{a'',\alpha'}(\vec{m}') \supset \|\vec{a}\cdot a', \mathbf{K}_{a''}\beta'\|^{\vec{y}'}_{\vec{n}'}$ iff (by (**)) $w \models \|\vec{a}\cdot a', \mathbf{K}_{a''}\beta'\|^{\vec{y}'}_{\vec{n}'}$.

For the if direction, suppose the right-hand side holds and $e_{a'} \models \mathbf{O}_{a'}\alpha^{\vec{x}}_{\vec{m}}$. We need to show $e_{a'} \models \mathbf{K}_{a'}\beta^{\vec{y}}_{\vec{n}}$, which is equivalent to $e_A, e_B, w \models \beta^{\vec{y}}_{\vec{n}}$ for all $(e'_{b'}, w) \in e_{a'}$. Let $(e_{b'}, w) \in e_{a'}$. By assumption, $e_A, e_B, w \models \alpha^{\vec{x}}_{\vec{m}} \wedge \mathbf{O}_{a'}\alpha^{\vec{x}}_{\vec{m}}$. By Lemma 4.5, there is a w' such that $w' \models \langle\!\langle \vec{a}\cdot a', \alpha \rangle\!\rangle^{\vec{x}}_{\vec{m}}$ and $e_A, e_B, w \approx_{\vec{a}\cdot a'} w'$. Since α is a'-objective, $\langle\!\langle \vec{a}\cdot a', \alpha \rangle\!\rangle$ does not mention $P_{a',\gamma}$ for any $\gamma \in \Sigma_{\vec{a}\cdot a'}(\kappa)$, so we can assume without loss of generality for all $(\gamma, \vec{z}) \in \Sigma_{\vec{a}\cdot a'}(\kappa)$ and names $\vec{o}$, that $P_{a',\gamma}(\vec{o}) \in w'$ iff $e_A, e_B \models \mathbf{O}_{a'}\gamma^{\vec{z}}_{\vec{o}}$. In particular, this implies $P_{a',\alpha}(\vec{m}) \in w'$ and $e_A, e_B, w \approx_{\vec{a}} w'$. Then by assumption, $w' \models \|\vec{a}\cdot a', \beta\|^{\vec{y}}_{\vec{n}}$. We show by subinduction on $|\beta|$ that $w' \models \|\vec{a}\cdot a', \beta\|^{\vec{y}}_{\vec{n}}$ iff $e_A, e_B \models \mathbf{K}_{a'}\beta^{\vec{y}}_{\vec{n}}$.

The only non-trivial case is to show for subformulas $(\beta', \vec{y}') \in \Sigma_{\vec{a}\cdot a'\cdot a''}(\kappa)$ that $w' \models \|\vec{a}\cdot a', \mathbf{K}_{a''}\beta'\|^{\vec{y}'}_{\vec{n}'}$ iff $e_A, e_B \models \mathbf{K}_{a''}\beta'^{\vec{y}'}_{\vec{n}'}$. Note that since either $a' = a''$ or α is a''-determinate, there is a $(\alpha', \vec{x}') \in \Sigma_{\vec{a}\cdot a'\cdot a''}(\kappa)$ and names $\vec{m}'$ such that $e_A, e_B \models \mathbf{O}_{a''}\alpha'^{\vec{x}'}_{\vec{m}'}$ (*). Then $w' \models \|\vec{a}\cdot a', \mathbf{K}_{a''}\beta'\|^{\vec{y}'}_{\vec{n}'}$ iff (by Lemma 4.3) for some $(\alpha', \vec{x}') \in \Sigma_{\vec{a}\cdot a'\cdot a''}(\kappa)$ and names $\vec{m}'$, $P_{a'',\alpha'}(\vec{m}') \in w'$ and $\models P_{a'',\alpha'}(\vec{m}') \wedge \langle\!\langle \vec{a}\cdot a'\cdot a'', \alpha' \rangle\!\rangle^{\vec{x}'}_{\vec{m}'} \supset \|\vec{a}\cdot a'\cdot a'', \beta'\|^{\vec{y}'}_{\vec{n}'}$ iff (since $e_A, e_B, w \approx_{\vec{a}\cdot a'} w'$ and $e_A, e_B, w \approx_{\vec{a}} w'$ and by main induction) for some $(\alpha', \vec{x}') \in \Sigma_{\vec{a}\cdot a'\cdot a''}(\kappa)$ and names $\vec{m}'$, $e_A, e_B \models \mathbf{O}_{a''}\alpha'^{\vec{x}'}_{\vec{m}'}$ and $\models \mathbf{O}_{a''}\alpha'^{\vec{x}'}_{\vec{m}'} \supset \mathbf{K}_{a''}\beta^{\vec{y}'}_{\vec{n}'}$ iff (by Lemma A.1 and (*)) $e_A, e_B \models \mathbf{K}_{a''}\beta'^{\vec{y}'}_{\vec{n}'}$. □

Theorem 4.9 (Representation theorem)

$$\models \kappa \supset \lambda \textit{ iff } \models \langle\!\langle \langle\rangle, \kappa \rangle\!\rangle \supset \|\langle\rangle, \lambda\|$$

Proof. For the only-if direction, suppose the left-hand side holds and $w \models \langle\!\langle \langle\rangle, \kappa \rangle\!\rangle$. By Lemma 4.5, there is an e_A, e_B such that $e_A, e_B, w \models \kappa$ and $e_A, e_B, w \approx_{\langle\rangle} w$. We now show by induction on $|\lambda|$ that $e_A, e_B, w \models \lambda$ iff $w \models \|\langle\rangle, \lambda\|$. The base case for atomic formulas follows from $e_A, e_B, w \approx_{\langle\rangle} w$. The only non-trivial induction step modal subformulas $\mathbf{K}_{a'}\beta$. Let $(\beta, \vec{y}) \in \Sigma_{a'}(\lambda)$. Then $e_{a'} \models \mathbf{K}_{a'}\beta^{\vec{y}}_{\vec{n}}$ iff (since κ is a'-determinate) for some $(\alpha, \vec{x}) \in \Sigma_{a'}(\kappa)$, $e_{a'} \models \mathbf{O}_{a'}\alpha^{\vec{x}}_{\vec{m}}$ and $e_{a'} \models \mathbf{K}_{a'}\beta^{\vec{y}}_{\vec{n}}$ iff (by Lemma A.1) for some $(\alpha, \vec{x}) \in \Sigma_{a'}(\kappa)$, $\models \mathbf{O}_{a'}\alpha^{\vec{x}}_{\vec{m}} \supset \mathbf{K}_{a'}\beta^{\vec{y}}_{\vec{n}}$ iff (by Lemmas A.2 and 4.8) for some $(\alpha, \vec{x}) \in \Sigma_{a'}(\kappa)$, $\models P_{a',\alpha}(\vec{m}) \supset \|\langle\rangle, \mathbf{K}_{a'}\beta\|^{\vec{y}}_{\vec{n}}$ iff (by construction of $\|\langle\rangle, \mathbf{K}_{a'}\beta\|$) for some $(\alpha, \vec{x}) \in \Sigma_{a'}(\kappa)$, $w \models P_{a',\alpha}(\vec{m}) \wedge \|\langle\rangle, \mathbf{K}_{a'}\beta\|^{\vec{y}}_{\vec{n}}$ iff for some $(\alpha, \vec{x}) \in \Sigma_{a'}(\kappa)$, $w \models P_{a',\alpha}(\vec{m}) \wedge \|\langle\rangle, \mathbf{K}_{a'}\beta\|^{\vec{y}}_{\vec{n}}$ iff (since κ is a'-determinate and $e_A, e_B, w \approx_{\langle\rangle} w$) $w \models \|\langle\rangle, \mathbf{K}_{a'}\beta\|^{\vec{y}}_{\vec{n}}$.

Conversely, suppose the right-hand side holds and $e_A, e_B, w \models \kappa$. By Lemma 4.5, there is an w' such that $w' \models \langle\!\langle \langle\rangle, \kappa \rangle\!\rangle$ and $e_A, e_B, w \approx_{\langle\rangle} w'$. We now show by induction on $|\lambda|$ that $w' \models \|\langle\rangle, \lambda\|$ iff $e_A, e_B, w \models \lambda$. The base case for atomic formulas follows from $e_A, e_B, w \approx_{\langle\rangle} w'$. The only non-trivial induction step modal subformulas $\mathbf{K}_{a'}\beta$, which works by the same argument as for the only-if direction. □

ACKNOWLEDGMENTS

We would like to thank Gerhard Lakemeyer for his comments on a draft of this paper.

REFERENCES

[1] Vaishak Belle and Gerhard Lakemeyer. 2014. Multiagent Only Knowing in Dynamic Systems. *Journal of Artificial Intelligence Research* 49 (2014), 363–402.

[2] Vaishak Belle and Gerhard Lakemeyer. 2015. Only Knowing Meets Common Knowledge. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*. 2755–2761.

[3] Vaishak Belle and Gerhard Lakemeyer. 2015. Semantical considerations on multiagent only knowing. *Artificial Intelligence* 223 (2015), 1–26.

[4] Yijia Chen, Abdallah Saffidine, and Christoph Schwering. 2018. The Complexity of Limited Belief Reasoning – The Quantifier-Free Case. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*. Stockholm, Sweden, 1774–1780.

[5] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. 1995. *Reasoning about knowledge.*

[6] Joseph Y. Halpern. 1993. Reasoning about only knowing with many agents. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI)*. 655–661.

[7] Joseph Y. Halpern and Gerhard Lakemeyer. 2001. Multi-agent Only Knowing. *Journal of Logic and Computation* 11, 1 (2001), 41–70.

[8] Laura Kovács and Andrei Voronkov. 2013. First-Order Theorem Proving and Vampire. In *Computer Aided Verification*. Springer Berlin Heidelberg, 1–35.

[9] Gerhard Lakemeyer. 1993. All they know: a study in multi-agent autoepistemic reasoning. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Vol. 93. 376–381.

[10] Gerhard Lakemeyer and Hector J. Levesque. 2005. Only-knowing: Taking it beyond autoepistemic reasoning. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*. 633–638.

[11] Gerhard Lakemeyer and Hector J. Levesque. 2011. A semantic characterization of a useful fragment of the situation calculus with knowledge. *Artificial Intelligence* 175, 1 (2011), 142–164.

[12] Hector J. Levesque. 1984. Foundations of a functional approach to knowledge representation. *Artificial Intelligence* 23, 2 (1984).

[13] Hector J. Levesque. 1990. All I know: a study in autoepistemic logic. *Artificial Intelligence* 42, 2 (1990).

[14] Hector J. Levesque and Gerhard Lakemeyer. 2001. *The Logic of Knowledge Bases*. MIT Press.

[15] Stephan Schulz. 2013. System Description: E 1.8. In *Proceedings of the 19th Logic for Programming Artificial Intelligence and Reasoning, Stellenbosch*. 735–743.

[16] Christoph Schwering. 2016. *Conditional Beliefs in Action*. Ph.D. Dissertation. RWTH Aachen University.

[17] Christoph Schwering. 2017. A Reasoning System for a First-Order Logic of Limited Belief. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*. 1247–1253.

[18] Christoph Schwering and Gerhard Lakemeyer. 2015. Projection in the Epistemic Situation Calculus with Belief Conditionals. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*. Austin, Texas, 1583–1589.

[19] Arild Waaler and Bjørnar Solhaug. 2005. Semantics for multi-agent only knowing. In *Proceedings of the Tenth conference on Theoretical Aspects of Rationality and Knowledge (TARK)*. 109–125.