

Epistemic Skills

Logical Dynamics of Knowing and Forgetting

Xiaolong Liang
School of Philosophy
Shanxi University
Taiyuan, Shanxi, P.R. China
lianghillon@gmail.com

Yì N. Wáng*
Department of Philosophy (Zhuhai)
Sun Yat-sen University
Zhuhai, Guangdong, P.R. China
ynw@xixilogic.org

We present a type of epistemic logics that encapsulates both the dynamics of acquiring knowledge (knowing) and losing information (forgetting), alongside the integration of group knowledge concepts. Our approach is underpinned by a system of weighted models, which introduces an “epistemic skills” metric to effectively represent the epistemic abilities associated with knowledge update. In this framework, the acquisition of knowledge is modeled as a result of upskilling, whereas forgetting is by downskilling. Additionally, our framework allows us to explore the concept of “knowability,” which can be defined as the potential to acquire knowledge through upskilling, and facilitates a nuanced understanding of the distinctions between epistemic *de re* and *de dicto* expressions. We study the computational complexity of model checking problems for these logics, providing insights into both the theoretical underpinnings and practical implications of our approach.

1 Introduction

The study of epistemic logic has become a prolific area within applied modal logic, since its inception as a formal methodology in epistemology [34, 21], and its subsequent application in computer science [16, 27]. A longstanding focus of this field has been to elucidate various forms of group knowledge, with mutual knowledge (what everyone knows), common knowledge, and distributed knowledge being particularly well-known concepts.

On top of this has been an exploration of actions that bring about changes in knowledge, such as the effect of public announcements. This inquiry has given rise to the subfield of dynamic epistemic logic [13], a discipline that incorporates update modalities into its language to depict knowledge updates, with Public Announcement Logic [30] and Action Model Logic [7] being popular approaches (the first can be viewed as a specific instance of the broader framework of the latter). Extensions of Public Announcement Logic that incorporate the concept of knowability have then garnered significant interest [5, 2]. These extensions delve into the nuanced understanding of what it means for something to be knowable in a dynamic informational context.

The literature presents a diverse array of approaches to model the phenomenon of forgetting within the frameworks of both classical and non-classical logics. Among these approaches, two prominent categories emerge: syntactical and semantical strategies for representing knowledge contraction. Syntactical strategies, such as those delineated by the AGM paradigm [3], typically involve the removal of formulas from an agent’s knowledge base, akin to belief contraction. On the other hand, semantical strategies focus on the modification of the interpretation of knowledge. This can include various methods such as erasing the truth values assigned to atomic propositions [26, 22, 12, 36]. Another semantical method involves updating the set of propositions that an agent is aware of [15].

*Corresponding author.

In this study, we propose a unified logical framework designed to model group knowledge, processes that may lead to knowledge update and epistemic necessity and possibility. Our approach is based on weighted modal logic [23, 20]. We extend this foundation by introducing the concept of *epistemic skills*, utilizing weights assigned to the edges in our model to represent the specific skills required to distinguish between pairs of possible worlds. This differentiation introduces a measure of similarity, aligning our work with recent developments in epistemic logic that employ concepts of similarity or distance [28, 14].

Traditionally understood notions such as mutual and common knowledge are preserved in their classical interpretations within our framework. Additionally, we incorporate distributed and field knowledge seamlessly. Our model explicitly defines the skill set each agent possesses, and by leveraging update modalities, we model the acquisition, loss, revision of knowledge as results of upskilling, downskilling and reskilling, respectively. By focusing on operations that modify one's skills, we broaden our analysis to include the concepts of knowability and forgetfulness. In keeping with the perspectives suggested in [5], our guiding principles are: *the knowable is what becomes known after upskilling*, and conversely, *the forgettable is what becomes unknown upon downskilling*. This framework also allows for a more nuanced understanding of the *de re* and *de dicto* distinctions in epistemic sentences.

The structure of the paper is as follows: Section 2 is dedicated to presenting the formal syntax and semantics of our proposed logics. This section also includes a discussion on the use of *epistemic de re* and *de dicto* expressions within our framework. The subsequent section delves into an in-depth analysis of the computational complexity associated with the model checking problems in these logics. The paper concludes with Section 4, where we offer our concluding remarks and reflections on the study.

2 Logics

We extend classical epistemic logic [16, 27] with a mechanism of epistemic skills in the models, allowing us a consistent way of modeling *knowing* and *forgetting*, as well as various notions of group knowledge (such as, *distributed knowledge* and *field knowledge*).

We fix three countably infinite sets before the introduction of formal languages. Namely, \mathbf{P} for the *set of atomic propositions* (*atoms* for short), \mathbf{A} for the *set of agents* and \mathbf{S} for the *set of epistemic skills* (capabilities, professions, or privileges). For simplicity, these sets are unchanged throughout the paper, although it is also possible to treat them as changeable parameters of each of the languages.

2.1 Syntax

The biggest language that we introduce now, named $\mathcal{L}_{CDEF+-\equiv\boxplus\boxminus}$, has its grammar given as follows:

$$\begin{aligned} \varphi ::= & p \mid \neg\varphi \mid (\varphi \rightarrow \varphi) \mid K_a\varphi \mid C_G\varphi \mid D_G\varphi \mid E_G\varphi \mid F_G\varphi \mid \\ & (+S)_a\varphi \mid (-S)_a\varphi \mid (=S)_a\varphi \mid (\equiv_b)_a\varphi \mid \boxplus_a\varphi \mid \boxminus_a\varphi \mid \square_a\varphi \end{aligned}$$

where $p \in \mathbf{P}$, $a, b \in \mathbf{A}$, $G \subseteq \mathbf{A}$ is a finite nonempty group, and $S \subseteq \mathbf{S}$ is a finite nonempty skill set.

As the name shows, we are interested in some of its sublanguages. The basic language \mathcal{L} allows a grammar that builds recursively from atomic propositions with Boolean operators (we choose negation and implication to be primitives) and the modal operator K_a (with $a \in \mathbf{A}$) which is used to characterize *individual knowledge*. Namely, \mathcal{L} is the formal language for classical multi-agent epistemic logic.

Four types of modalities, C_G , D_G , E_G and F_G , are introduced for *common knowledge*, *distributed knowledge*, *mutual knowledge* and *field knowledge*, respectively. In naming a language that extends the basic language, we use combinations of the letters C , D , E and F to indicate the inclusion common,

distributed, mutual or field knowledge operators. For example, \mathcal{L}_{DF} denotes the language that extends the basic language with distributed and field knowledge.

We consider four update modalities, $(+_S)_a$, $(-_S)_a$, $(=_S)_a$ and $(\equiv_b)_a$, where $a, b \in \mathbf{A}$ and S is a finite nonempty subset of \mathbf{S} , which are intended to mean the action of agent a 's expansion with skills S (*upskilling*), subtraction of skills S (*downskilling*), assigning skill set S (*reskilling*) and *learning* from agent b , respectively. These operators are self dual, as one can verify after the semantics is introduced.

Another three operators, \boxplus_a , \boxminus_a and \boxdot_a , are used to mean the action of a 's addition, subtraction and modification of an arbitrary skill set, respectively. Their dual operators are written as \boxplus_a , \boxminus_a and \boxdot_a , respectively, but treated to be non-primitive.

We shall use the symbols $+$, $-$, $=$, \equiv , \boxplus , \boxminus and \boxdot in subscript to signal the introduction of each of the update operators or quantifiers. For example, $\mathcal{L}_{F+\boxplus}$ stands for the language that extends the basic language with field knowledge and the operators $(+_S)_a$ and \boxplus_a (for any $a \in \mathbf{A}$ and $S \subseteq \mathbf{S}$).

As a result, we reach as many as 2^{11} ($= 2048$) languages in total, though many of the combinations may not be of our focus. Other Boolean operators are defined just as in classical logic. When we refer to a *formula*, we are indicating an element of one of these languages, and its specific reference will depend on the context unless otherwise specified.

2.2 Semantics

We introduce a type of models for the interpretation of the languages.

Definition 1. A model is a quadruple (W, E, C, β) where:

- W is a nonempty set of (possible) worlds or states;
- $E : W \times W \rightarrow \wp(\mathbf{S})$, an edge function, assigns each pair of worlds a skill set;
- $C : \mathbf{A} \rightarrow \wp(\mathbf{S})$ is a capability function that assigns a skill set to each agent;
- $\beta : W \rightarrow \wp(\mathbf{P})$ is a valuation.

and satisfies the following two conditions:

- *Positivity:* for all $w, u \in W$, if $E(w, u) = \mathbf{S}$, then $w = u$;
- *Symmetry:* for all $w, u \in W$, $E(w, u) = E(u, w)$.

In the above definition, the function E assigns a skill set to each edge (a pair of worlds), indicating that only individuals with skills outside the set can distinguish between the pair of worlds. The criteria for satisfaction are defined as follows.

Definition 2. Given a formula φ , a model $M = (W, E, C, \beta)$ and $w \in W$, we say φ is true or satisfied at w in M , denoted $M, w \models \varphi$, if the following hold inductively:

$$\begin{aligned}
M, w \models p & \Leftrightarrow p \in \beta(w) \\
M, w \models \neg\psi & \Leftrightarrow \text{not } M, w \models \psi \\
M, w \models (\psi \rightarrow \chi) & \Leftrightarrow \text{if } M, w \models \psi \text{ then } M, w \models \chi \\
M, w \models K_a\psi & \Leftrightarrow \text{for all } u \in W, \text{ if } C(a) \subseteq E(w, u) \text{ then } M, u \models \psi \\
M, w \models E_G\psi & \Leftrightarrow M, w \models K_a\psi \text{ for all } a \in G \\
M, w \models C_G\psi & \Leftrightarrow \text{for all positive integers } n, M, w \models E_G^n\psi, \text{ with } E_G^1\psi := E_G\psi \text{ and } E_G^n\psi := E_G^1 E_G^{n-1}\psi \\
M, w \models D_G\psi & \Leftrightarrow \text{for all } u \in W, \text{ if } \bigcup_{a \in G} C(a) \subseteq E(w, u) \text{ then } M, u \models \psi \\
M, w \models F_G\psi & \Leftrightarrow \text{for all } u \in W, \text{ if } \bigcap_{a \in G} C(a) \subseteq E(w, u) \text{ then } M, u \models \psi
\end{aligned}$$

$$\begin{aligned}
M, w \models (+_S)_a \psi &\Leftrightarrow (W, E, C^{a+S}, \beta), w \models \psi, \text{ with } C^{a+S}(a) = C(a) \cup S \text{ and } (\forall x \in \mathbf{A} \setminus \{a\}) C^{a+S}(x) = C(x) \\
M, w \models (-_S)_a \psi &\Leftrightarrow (W, E, C^{a-S}, \beta), w \models \psi, \text{ with } C^{a-S}(a) = C(a) \setminus S \text{ and } (\forall x \in \mathbf{A} \setminus \{a\}) C^{a-S}(x) = C(x) \\
M, w \models (=S)_a \psi &\Leftrightarrow (W, E, C^{a=S}, \beta), w \models \psi, \text{ with } C^{a=S}(a) = S \text{ and } (\forall x \in \mathbf{A} \setminus \{a\}) C^{a=S}(x) = C(x) \\
M, w \models (\equiv_b)_a \psi &\Leftrightarrow (W, E, C^{a \equiv b}, \beta), w \models \psi, \text{ with } C^{a \equiv b}(a) = C(b) \text{ and } (\forall x \in \mathbf{A} \setminus \{a\}) C^{a \equiv b}(x) = C(x) \\
M, w \models \boxplus_a \psi &\Leftrightarrow \text{for all finite nonempty } S \subseteq \mathbf{S}, M, w \models (+_S)_a \psi \\
M, w \models \boxminus_a \psi &\Leftrightarrow \text{for all finite nonempty } S \subseteq \mathbf{S}, M, w \models (-_S)_a \psi \\
M, w \models \square_a \psi &\Leftrightarrow \text{for all finite nonempty } S \subseteq \mathbf{S}, M, w \models (=S)_a \psi.
\end{aligned}$$

Given that G is a finite group, it is clear that the formula $E_G \psi$ is logically equivalent to the $\bigwedge_{a \in G} K_a \psi$. However, this equivalence impacts both the succinctness of the language and the complexity of model checking. Consequently, $E_G \psi$ cannot be treated merely as a straightforward rewriting of $\bigwedge_{a \in G} K_a \psi$.

Note that although $(=S)_a \varphi$ is not a legal formula when S is the empty set \emptyset , we can regard it as a defined formula, i.e., $(=\emptyset)_a \varphi := (=S)_a (-_S)_a \varphi$ (for any qualified set S). In the mean time, it is not hard to verify that both $(+\emptyset)_a \varphi$ and $(-\emptyset)_a \varphi$, if allowed, are equivalent to φ , so there is no need to worry about the cases with empty sets.

The logics (i.e., the sets of valid formulas) that are defined by the above semantics and correspond to our languages will bear the same names, but will be denoted using upright roman typeface, e.g., L , $L_{F+\boxplus}$ and $L_{CDEF+-\equiv\boxplus\boxminus}$.

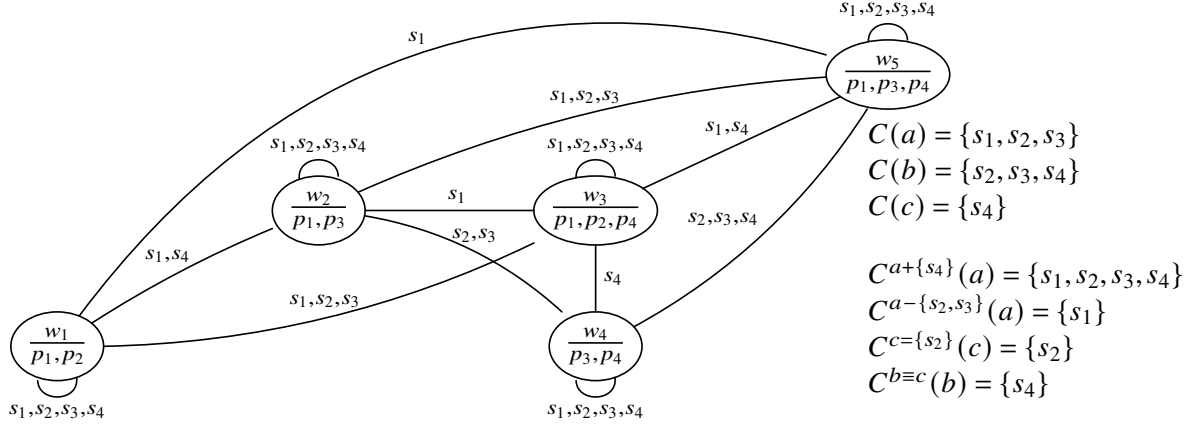
2.3 Representation of a model and truths within it

In this section, we describe an exemplary model and demonstrate several true formulas applicable within this model. Let $s_1, s_2, s_3, s_4, s_5 \in \mathbf{S}$ represent epistemic skills, and $a, b, c \in \mathbf{A}$ denote agents. The model $M = (W, E, C, \beta)$ is defined as follows:

- $W = \{w_1, w_2, w_3, w_4, w_5\}$ is the set of possible worlds.
- $E : W \times W \rightarrow \wp(\mathbf{S})$ is the symmetric closure that satisfies the following:
 - $E(w_1, w_1) = E(w_2, w_2) = E(w_3, w_3) = E(w_4, w_4) = E(w_5, w_5) = \{s_1, s_2, s_3, s_4\}$,
 - $E(w_1, w_2) = E(w_3, w_5) = \{s_1, s_4\}$,
 - $E(w_1, w_3) = E(w_2, w_5) = \{s_1, s_2, s_3\}$,
 - $E(w_1, w_4) = \emptyset$,
 - $E(w_1, w_5) = E(w_2, w_3) = \{s_1\}$,
 - $E(w_2, w_4) = \{s_2, s_3\}$,
 - $E(w_3, w_4) = \{s_4\}$,
 - $E(w_4, w_5) = \{s_2, s_3, s_4\}$.
- C is the capability function that assigns a skill set to each agent, a, b and c :
 - $C(a) = \{s_1, s_2, s_3\}$,
 - $C(b) = \{s_2, s_3, s_4\}$,
 - $C(c) = \{s_4\}$.
- β assigns sets of propositions to each world:
 - $\beta(w_1) = \{p_1, p_2\}$
 - $\beta(w_2) = \{p_1, p_3\}$
 - $\beta(w_3) = \{p_1, p_2, p_4\}$
 - $\beta(w_4) = \{p_3, p_4\}$
 - $\beta(w_5) = \{p_1, p_3, p_4\}$.

The fact that M is a model can be easily verified, and using a diagram to represent M is often helpful (see Figure 1). In the diagram, nodes represent worlds, and undirected edges represent accessibility relations. Each edge is labeled with the skill set that facilitate accessibility between two worlds. If an edge is labeled with an empty set, it indicates no accessibility (as between w_1 and w_4), and such edges are not drawn in the diagram. This helps in visualizing the connections and structure of the model more clearly.

Readers can verify the following logical truths in model $M = (W, E, C, \beta)$ given above:

Figure 1: Illustration of the model M .

1. $M, w_2 \models K_a p_3$, indicating that in world w_2 , agent a knows proposition p_3 .
2. $M, w_4 \models \neg K_b p_1 \wedge \neg K_b \neg p_1$, meaning that in world w_4 , agent b does not know whether proposition p_1 is true or false.
3. $M, w_3 \models K_c (K_a p_3 \vee K_a \neg p_3)$, demonstrating that in world w_3 , agent c knows whether agent a knows proposition p_3 .
4. $M, w_4 \models E_{\{a,b\}}(p_3 \wedge p_4)$, showing that both agents a and b know propositions p_3 and p_4 in w_4 .
5. $M, w_5 \models (\neg C_{\{a,c\}} p_1 \wedge \neg C_{\{a,c\}} \neg p_1) \wedge (\neg C_{\{a,c\}} p_2 \wedge \neg C_{\{a,c\}} \neg p_2)$, indicating that neither the truth nor the falsity of propositions p_1 and p_2 are common knowledge between a and c in world w_5 .
6. $M, w_4 \models D_{\{a,b\}}(\neg p_1 \wedge p_4)$, indicating that in world w_4 , the knowledge that proposition p_1 is false and p_4 is true is distributed between agents a and b .
7. $M, w_4 \models \neg F_{\{a,b\}} \neg p_1 \wedge \neg F_{\{a,b\}} p_4$, showing that in world w_4 , neither $\neg p_1$ nor p_4 are field knowledge for agents a and b .
8. $M, w_5 \models \neg K_a p_4 \wedge (+_{\{s_4\}})_a K_a p_4$. Here, in world w_5 , agent a initially does not know p_4 , but can learn it upon acquiring skill s_4 .
9. $M, w_2 \models K_a p_3 \wedge (-_{\{s_2, s_3\}})_a \neg K_a p_3$, indicating that in world w_2 , agent a knows p_3 but would forget it if she loses skills s_2 and s_3 .
10. $M, w_1 \models E_{\{a,b\}}(\neg K_c p_2 \wedge (=_{\{s_2\}})_c K_c p_2)$. This means that in world w_1 , it is mutual knowledge between agents a and b that, c does not know p_2 , but would know it if her skill set is $\{s_2\}$.
11. $M, w_1 \models (\equiv_c)_b \wedge_{p \in \{p_1, \dots, p_4\}} (F_{\{b,c\}} p \leftrightarrow K_b p)$. This result suggests that in world w_1 , if agent b changes her skill set to match that of agent c , her knowledge will align with the field knowledge shared between them.
12. $M, w_5 \models \diamond_a K_a p_4$, indicating that in world w_5 , there exists a potential skill upgrade under which agent a can come to know p_4 .
13. $M, w_3 \models \diamond_b \wedge_{p \in \{p_1, \dots, p_4\}} (\neg C_{\{a,b\}} p \wedge \neg C_{\{a,b\}} \neg p)$, highlighting that in world w_3 , it is possible through some downskilling for agent b to reach a state where no propositions p_1 through p_4 are common knowledge between agents a and b .
14. $M, w_2 \models K_c p_1 \wedge \neg K_c p_3 \wedge \diamond_c (\neg K_c p_1 \wedge K_c p_3)$, indicating that in world w_2 , it is currently the case that agent c knows p_1 and does not know p_3 , but there is a possible skill update which would make agent c unaware of p_1 while becoming aware of p_3 .

2.4 Epistemic *de re* and *de dicto*

The distinction between epistemic *de re* and *de dicto* modalities was already discussed in [34], with *de re* modalities concerning whether a specific thing possesses or lacks a certain property, and *de dicto* modalities concerning whether a proposition is true or false. Subsequently, as [31] suggests, this distinction becomes more apparent in a formal language when quantifiers over terms are utilized. In the realm of epistemic logic, a *de re* sentence can be expressed as, “There exists a term x such that an entity knows that x possesses or lacks a certain property.” Conversely, a *de dicto* sentence can be formulated as, “An entity knows that there exists a term which possesses or lacks a certain property.”

In the logics introduced in this paper, we are not only able to distinguish between *de re* and *de dicto* modalities, but can also delineate two specific types of *de re* sentences (compare with the case in Group Announcement Logic [2, Section 6]):

- Knowing *de dicto*: “Agent a knows (with her current skills) that there exists a set S of skills such that, with S , she can achieve φ in world w of model (W, E, C, β) .”
Formally, this is expressed as: $(\forall u \in W)[C(a) \subseteq E(w, u) \Rightarrow (\exists S \subseteq \mathbf{S})(W, E, C^{a+S}, \beta), u \models \varphi]$.
- Explicitly knowing *de re*: “There exists a set S of skills such that agent a knows with her current skill set, that with S in addition, she can achieve φ in world w of model (W, E, C, β) .”
Formally, this is represented as: $(\exists S \subseteq \mathbf{S})(\forall u \in W)[C(a) \subseteq E(w, u) \Rightarrow (W, E, C^{a+S}, \beta), u \models \varphi]$.
- Implicitly knowing *de re*: “There exists a set S of skills such that agent a knows, with the addition of S to her skill set, that she can achieve φ in world w of model (W, E, C, β) .”
Formally, this is depicted as: $(\exists S \subseteq \mathbf{S})(\forall u \in W)[C^{a+S}(a) \subseteq E(w, u) \Rightarrow (W, E, C^{a+S}, \beta), u \models \varphi]$.

Although the distinction between *de dicto* and *de re* knowledge remains clear, the nuanced difference between *implicit* and *explicit de re* knowledge hinges on whether the skills from the skill set S are included for the agent to formulate her knowledge.

These distinctions elucidate the complex interplay between knowledge and capabilities in dynamic epistemic scenarios, highlighting subtle differences in how agents process information depending on their skill sets and the nature of their knowledge. All three types of knowledge are expressible using the formal languages we have introduced. Here is how each type is represented:

Proposition 3.

1. Knowledge *de dicto* is expressed by the formula $K_a \diamond_a \varphi$.
2. Explicit knowledge *de re* is expressed by the formula $(\equiv_a)_c \diamond_c K_a (\equiv_c)_a \varphi$ (where c is not in φ).
3. Implicit knowledge *de re* is expressed by the formula $\diamond_a K_a \varphi$.

Proof. Clauses 1 and 3 are straightforwardly validated by the semantics. We focus here on clause 2, where c is any agent not appearing in φ :

$$\begin{aligned}
 & (\exists S \subseteq \mathbf{S})(\forall u \in W) C(a) \subseteq E(w, u) \Rightarrow (W, E, C^{a+S}, \beta), u \models \varphi \\
 \iff & (\exists S \subseteq \mathbf{S})(\forall u \in W) C(a) \subseteq E(w, u) \Rightarrow (W, E, ((C^{c \equiv a})^{c+S})^{a \equiv c}, \beta), u \models \varphi \\
 \iff & (\exists S \subseteq \mathbf{S})(\forall u \in W) C(a) \subseteq E(w, u) \Rightarrow (W, E, (C^{c \equiv a})^{c+S}, \beta), u \models (\equiv_c)_a \varphi \\
 \iff & (\exists S \subseteq \mathbf{S})(W, E, (C^{c \equiv a})^{c+S}, \beta), w \models K_a (\equiv_c)_a \varphi \\
 \iff & (W, E, C^{c \equiv a}, \beta), w \models \diamond_c K_a (\equiv_c)_a \varphi \\
 \iff & (W, E, C, \beta), w \models (\equiv_a)_c \diamond_c K_a (\equiv_c)_a \varphi.
 \end{aligned}$$

Examples of different types of knowledge expressed by formulas The formula $D_G \diamond_a \diamond_b \varphi$ says that, “It is group G ’s distributed knowledge that, with the addition of certain skills by agent a , it becomes achievable that, even with the loss of certain skills by agent b , φ can still be achieved.” This use pertains to be *de dicto*. The formula $(\equiv_a)_c \diamond_c K_a (\equiv_c)_a \varphi$ (where c does not appear in φ) expresses that, “There exists a skill set such that agent a knows that, with exactly this skill set, a can achieve φ .” This use is *explicitly de re*. The formula $\diamond_a K_a \varphi$ says that, “There is an update of agent a ’s skill set through which a knows she can make φ true.” This is *implicitly de re*.

Remark 1. For simplicity, the initial definitions of knowledge *de dicto*, explicit and implicit knowledge *de re* have been presented primarily for individual knowledge using the operator K_a and the actions of knowing represented by the quantifier \boxplus_a . These concepts can be readily extended to include:

- Group knowledge, utilizing operators such as C_G , D_G , E_G and F_G ,
- Quantifiers over actions of downskilling and reskilling, represented by \boxminus_a and \boxplus_a respectively,
- Nested actions and dynamic changes among agents.

For example, the formula $F_G \diamond_{a_1} \diamond_{a_2} \diamond_{a_3} \diamond_{a_4} \varphi$ represents an epistemic *de dicto* statement involving field knowledge and multiple actions (upskilling, downskilling and reskilling) among different agents. The expression $(\equiv_{d_1})_{c_1} \diamond_{c_1} (\equiv_{d_2})_{c_2} \diamond_{c_2} (\equiv_{d_3})_{c_3} \diamond_{c_3} E_I (\equiv_{c_1})_{d_1} (\equiv_{c_2})_{d_2} (\equiv_{c_3})_{d_3} \varphi$ captures explicit knowledge *de re* involving nested contexts and multiple agents, linked to mutual knowledge. Similarly, the formula $\diamond_{b_1} \diamond_{b_2} \diamond_{b_3} D_H \varphi$ illustrates implicit knowledge *de re* involving a sequence of updates and distributed knowledge. In these examples, the agents $a_1, a_2, a_3, a_4, b_1, b_2, b_3, c_1, c_2, c_3, d_1, d_2, d_3$ are not specifically restricted to being within or outside the groups G, H or I . This flexibility allows for a broad application of the concepts across various contexts and group dynamics.

In dynamic epistemic logic, the distinction between *knowing de dicto* and *knowing de re* is enriched through the use of quantifiers for updates, closely aligning with the philosophical inquiries into *knowing that* versus *knowing how*. While previous solutions such as those presented in [5, 2, 6] primarily adopt a syntactical approach, our logic introduces a semantical perspective, providing an alternative to the topological semantics discussed by [33, 9].

3 Complexity of Model Checking

In this section, we study the computational complexity of the model checking problem for the logics introduced in the preceding sections. The *model checking problem* for a logic involves verifying whether a specified formula φ , within a given finite model M and at a particular world w in the model, holds true; formally, whether $M, w \models \varphi$.

3.1 The input

We define the measure of the input. The *length* of a formula φ , denoted $|\varphi|$, is defined to be the number of symbols that occur in φ (including the symbols for brackets), just as in [16, Section 3.1]; or more precisely defined inductively by the structure of φ , i.e., when φ is:

- An atomic formula p : $|p| = 1$;
- Negation $\neg\psi$: $|\neg\psi| = |\psi| + 1$;
- Implication $(\psi \rightarrow \chi)$: $|(\psi \rightarrow \chi)| = |\psi| + |\chi| + 3$;

- Individual knowledge $K_a\psi$: $|K_a\psi| = |\psi| + 2$;
- Group knowledge: $|C_G\psi| = |\psi| + 2|G| + 2$, and similarly for $D_G\psi$, $E_G\psi$ and $F_G\psi$; e.g., $|(p \rightarrow C_{\{a,b,c\}}q)| = 13$;
- An update modality: $|(+_S)_a\psi| = 2|S| + |\psi| + 5$, similarly for $(-_S)_a\psi$ and $(=_S)_a\psi$, and $|(\equiv_b)_a\psi| = |\psi| + 5$;
- A quantifier: $|\boxplus_a\psi| = |\psi| + 2$, and also for $\boxminus_a\psi$ and $\square_a\psi$.

The *size* of a model $M = (W, E, C, \beta)$, denoted $|M|$, is defined as the sum of the following components:

- $|W|$: the size of the domain; ¹
- $|E|$: since E consists of triples (w, u, S) where $w, u \in W$ and $S \subseteq \mathbf{S}$, the size of E is determined by the number of the symbols used to denote this set;
- $|C|$ with respect to a given set A of agents: C is composed pairs (a, S) where $a \in A$ and $S \subseteq \mathbf{S}$; the size of C is the count of all symbols used for its representation; ²
- $|\beta|$: the function β consists of pairs (w, Φ) where $w \in W$ and $\Phi \subseteq \mathbf{P}$; the size of β is the number of the symbols used to represent this set.

Finally, for formula φ and model M (with a designated world w), the *size of the input* is $|\varphi| + |M| + 1$.

3.2 Model checking for logics without quantifiers: in P

We commence by presenting a polynomial-time algorithm designed to ascertain the truth of classical epistemic formulas in a specified world within a given model, addressing the model checking problem for L. Subsequently, we enhance the algorithm to incorporate group knowledge modalities. This extension allows us to establish that the model checking problem for L_{CDEF} fall within the complexity class P. We then proceed to further broaden our results to encompass update modalities, achieving the results for the model checking problems for $L_{CDEF+--\equiv}$ and all of its sublogics.

3.2.1 Model checking in L

Given a model $M = (W, E, C, \beta)$, a world $w \in W$ and a formula φ , we decide whether $M, w \models \varphi$. In order to do so, we present an algorithm (Algorithm 1) for calculating $Val(M, \varphi)$, the *truth set* of φ in M , i.e., $\{x \in W \mid M, x \models \varphi\}$. The question about whether $M, w \models \varphi$ holds is thus reduced to the membership testing in $Val(M, \varphi)$, which takes at most $|W|$ steps in addition to the time costs on computing $Val(M, \varphi)$.

It is not hard to verify that $Val(M, \varphi)$ is indeed the set of worlds of M at which φ is true. In particular, in the case for the K_a operator,

$$\begin{aligned}
M, w \models K_a\psi &\iff \forall y \in W : C(a) \subseteq E(w, y) \Rightarrow M, y \models \psi \\
&\iff \forall y \in W : C(a) \subseteq E(w, y) \Rightarrow y \in Val(M, \psi) & \text{(IH)} \\
&\iff w \in \{x \in W \mid \forall y \in W : C(a) \subseteq E(x, y) \Rightarrow y \in Val(M, \psi)\}
\end{aligned}$$

The cost for computing $Val(M, \varphi)$ is in polynomial time. In the case for $K_a\psi$ —the most time-consuming case here—there are two while-loops over W , and checking $C(a) \subseteq E(x, y)$ costs at most

¹Model checking is typically impractical for infinite sets due to computational limitations; therefore, we restrict our analysis to finite sets. This is consistently applied in the following discussions as well.

²Theoretically, the function C maps each agent (from an infinite set) to a specific skill set. This mapping is not feasible with finite input, but in practical scenarios, we limit the number of agents. It is essential to ensure that the set A includes all agents relevant to the formula being checked.

Algorithm 1 Function $Val(M, \varphi)$: computing the truth set for basic formulas

Input: model $M = (W, E, C, \beta)$ and formula φ Output: $\{x \mid M, x \models \varphi\}$ 1: Initialize: $tmpVal \leftarrow \emptyset$ 2: if $\varphi = p$ then return $\{x \in W \mid p \in \beta(x)\}$ 3: else if $\varphi = \neg\psi$ then return $W \setminus Val(M, \psi)$ 4: else if $\varphi = \psi \rightarrow \chi$ then 5: return $(W \setminus Val(M, \psi)) \cup Val(M, \chi)$ 6: else if $\varphi = K_a\psi$ then	7: for all $x \in W$ do 8: Initialize: $n \leftarrow \text{true}$ 9: for all $y \in W$ do 10: if $C(a) \subseteq E(x, y)$ and $y \notin Val(M, \psi)$ then 11: $n \leftarrow \text{false}$ 12: if $n = \text{true}$ then $tmpVal \leftarrow tmpVal \cup \{x\}$ 12: return $tmpVal$ \triangleright This returns $\{x \in W \mid \forall y \in W : C(a) \subseteq E(x, y) \Rightarrow y \in Val(M, \psi)\}$
---	--

$|C| \cdot |E|$ steps, and the membership checking $y \notin Val(M, \psi)$ (when $Val(M, \psi)$ is at hand) takes at most $|W|$ steps; so this case costs at most $|W|^2 \cdot (|C| \cdot |E| + |W|)$. Moreover, the algorithm for computing $Val(M, \varphi)$ calls itself recursively, but only for a subformula of φ , and the maximum number of recursion is bounded by $|\varphi|$, i.e., the length of φ . So the total time cost for computing $Val(M, \varphi)$ is $|W|^2 \cdot (|C| \cdot |E| + |W|) \cdot |\varphi|$. Considering the input size, we find that the total time cost is within $O(n^5)$. So the following lemma holds.

Lemma 4. *The model checking problem for L is in P.*

3.2.2 Model checking group knowledge

Building on the previous result, we now aim to encompass scenarios that include group knowledge. To facilitate this extension, we will first introduce a definition and a couple of lemmas that underpin it.

Definition 5. For a formula φ , let $A_\varphi = \{G \mid \text{“}E_G\text{” or “}C_G\text{” appears in } \varphi\}$. For a model $M = (W, E, C, \beta)$,

- For all worlds $w, u \in W$, $E_\varphi(w, u) = E(w, u) \cup \{G \in A_\varphi \mid (\exists a \in G) C(a) \subseteq E(w, u)\}$,
- For all worlds $w, u \in W$, $E_\varphi^+(w, u) = E_\varphi(w, u) \cup \{G \in A_\varphi \mid (\exists n \geq 1)(\exists w_0, \dots, w_n \in W) w_0 = w \text{ and } w_n = u \text{ and } G \in \bigcap_{0 \leq i < n} E_\varphi(w_i, w_{i+1})\}$,

where without loss of generality we assume that $A_\varphi \cap A = \emptyset$. For short, we write M_φ^+ for $(W, E_\varphi^+, C, \beta)$.

It should be noted that the above definition involves an abuse of notation by treating groups of agents as skills. To ensure formal correctness, a one-to-one mapping can be defined from each group to a new skill in S .

Proposition 6. For any model M and any formula φ , M_φ^+ is a model. □

Lemma 7. Given formulas φ and χ , a group G , a model M and a world w of M :

1. $M, w \models \varphi$ iff $M_\chi^+, w \models \varphi$;
2. If “ C_G ” appears in χ , then $M, w \models C_G\varphi$ iff $M, u \models \varphi$ for any world u such that $G \in E_\chi^+(w, u)$.

Proof. 1. For any agent a , formula χ and world w, u , we have $C(a) \subseteq E(w, u)$ iff $C(a) \subseteq E_\chi(w, u)$ iff $C(a) \subseteq E_\chi^+(w, u)$. Thus it is easy to verify that (M, w) and (M_χ^+, w) satisfy exactly the same formulas.

2. We first verify the base case for $C_G\varphi$:

$$\begin{aligned}
 M, w \models E_G\varphi &\iff \text{for any } a \in G, M, w \models K_a\varphi \\
 &\iff \text{for any } a \in G \text{ and } u \in W, C(a) \subseteq E(w, u) \text{ implies } M, u \models \varphi \\
 &\iff \text{for any } u \in W \text{ and } a \in G, C(a) \subseteq E(w, u) \text{ implies } M, u \models \varphi \\
 &\iff \text{for any } u \in W, M, u \models \varphi \text{ if } C(a) \subseteq E(w, u) \text{ for some } a \in G \\
 &\iff \text{for any } u \in W, G \in E_\chi(w, u) \text{ implies } M, u \models \varphi \\
 &\iff M, u \models \varphi \text{ for any world } u \text{ such that } G \in E_\chi(w, u)
 \end{aligned}$$

$$\begin{aligned}
 \text{and so } M, w \models C_G\varphi &\iff M, w \models E_G^k\varphi \text{ for all } k \in \mathbb{N}^+ \\
 &\iff M, u \models \varphi \text{ for any world } u \text{ such that } G \in E_\chi^+(w, u) \quad (*)
 \end{aligned}$$

where (*) can be shown as follows: Suppose $M, w \not\models E_G^n \varphi$ for some $n \in \mathbb{N}^+$, then by induction on n , we have $w_1, \dots, w_n \in W$ such that $M, w_n \not\models \varphi$ and $G \in E_\chi(w, w_1) \cap \bigcap_{1 \leq i < n} E_\chi(w_i, w_{i+1})$. Hence $M, w_n \not\models \varphi$ and $G \in E_\chi^+(w, w_n)$. Suppose $M, u \not\models \varphi$ for a world u such that $G \in E_\chi^+(w, u)$, w.l.o.g, assume that there exist $w_0, \dots, w_n \in W$ such that $w_0 = w, w_n = u, G \in \bigcap_{0 \leq i < n} E_\chi(w_i, w_{i+1})$ and $M, w_n \not\models \varphi$. Thus using the above result n times we have $M, w \not\models E_G^n \varphi$. \square

Lemma 8. *The model checking problem for \mathcal{L}_{CDEF} (hence for all of its sublogics) is in P.*

Proof. It suffices to provide a polynomial algorithm for the types of formulas $C_G\psi, D_G\psi, E_G\psi$ and $F_G\psi$. The details are given in Algorithm 2. As in the proof of Lemma 4, checking $C(a) \subseteq E(t, u)$ costs at most

Algorithm 2 Function $Val(M, \varphi)$ extended: cases with group knowledge operators

<pre> 1: Initialize: $tmpVal \leftarrow \emptyset$ 2: if ... then ... ▷ Same as in Algorithm 1 3: else if $\varphi = C_G\psi$ then 4: for all $x \in W$ do 5: Initialize: $n \leftarrow \text{true}$ 6: for all $y \in W$ do 7: if $G \in E_\varphi^+(x, y)$ and $y \notin Val(M, \psi)$ then 8: $n \leftarrow \text{false}$ 9: if $n = \text{true}$ then 10: $tmpVal \leftarrow tmpVal \cup \{x\}$ 11: return $tmpVal$ ▷ Returns $\{x \in W \mid \forall y \in W : G \in E_\varphi^+(x, y) \Rightarrow y \in Val(M, \psi)\}$ 12: else if $\varphi = D_G\psi$ then 13: for all $x \in W$ do 14: Initialize: $n \leftarrow \text{true}$ 15: for all $y \in W$ do 16: if $\bigcup_{a \in G} C(a) \subseteq E(x, y)$ and 17: $y \notin Val(M, \psi)$ then 18: $n \leftarrow \text{false}$ 19: if $n = \text{true}$ then 20: $tmpVal \leftarrow tmpVal \cup \{x\}$ </pre>	<pre> 20: return $tmpVal$ ▷ Returns $\{x \in W \mid \forall y \in W : \bigcup_{a \in G} C(a) \subseteq E(x, y) \Rightarrow y \in Val(M, \psi)\}$ 21: else if $\varphi = E_G\psi$ then 22: for all $x \in W$ do 23: initialize $n \leftarrow \text{true}$ 24: for all $y \in W$ do 25: if $G \in E_\varphi(x, y)$ and $y \notin Val(M, \psi)$ then 26: $n \leftarrow \text{false}$ 27: if $n = \text{true}$ then $tmpVal \leftarrow tmpVal \cup \{x\}$ 28: return $tmpVal$ ▷ Returns $\{t \in W \mid \forall u \in W : G \in E_\varphi(t, u) \Rightarrow u \in Val(M, \psi)\}$ 29: else if $\varphi = F_G\psi$ then 30: for all $x \in W$ do 31: Initialize: $n \leftarrow \text{true}$ 32: for all $y \in W$ do 33: if $\bigcap_{a \in G} C(a) \subseteq E(x, y)$ and 34: $y \notin Val(M, \psi)$ then 35: $n \leftarrow \text{false}$ 36: if $n = \text{true}$ then $tmpVal \leftarrow tmpVal \cup \{x\}$ 37: return $tmpVal$ ▷ Returns $\{x \in W \mid \forall y \in W : \bigcap_{a \in G} C(a) \subseteq E(x, y) \Rightarrow y \in Val(M, \psi)\}$ </pre>
---	--

$|C| \cdot |E|$ steps, here we furthermore need to calculate the cost caused by group knowledge operators.

For D_G and F_G , notice that the number of agents in any group G that appears in φ is less than $|\varphi|$, so checking $\bigcup_{a \in G} C(a) \subseteq E(t, u)$ and $\bigcap_{a \in G} C(a) \subseteq E(t, u)$ costs at most $|C| \cdot |E| \cdot |\varphi|$ steps. Thus for the logics extended with these operators, the complexity for model checking would not go beyond P.

For E_G and C_G , we need to ensure that there is a polynomial-time algorithm for computing $E_\varphi(w, u)$ and $E_\varphi^+(w, u)$ and checking whether G is an element of them. By Definition 5 and Lemma 7, computing the set A_φ costs at most $|\varphi|$ steps, since there are at most $|\varphi|$ modalities appearing in φ ; moreover, the size of G is at most $|\varphi|$. To compute $E_\varphi(w, u)$ for any given w and u , it costs at most $|E|$ steps to compute $E(w, u)$ and at most $|\varphi|^2 \cdot |C| \cdot |E|$ steps to check for every $G \in A_\varphi$ whether there exists $a \in G$ such that $C(a) \subseteq E(w, u)$. So the cost of computing the whole function E_φ can be finished in at most $|W|^2 \cdot (|E| + |\varphi|^2 \cdot |C| \cdot |E|)$ steps. Now we consider the computation of E_φ^+ . Assume that we have a string that describes E_φ , then we check for all pairs $(x, y), (y, z) \in W^2$ whether there exists a “G” appearing in φ such that $G \in E_\varphi(x, y) \cap E_\varphi(y, z)$; if it is, we add G as a member of $E_\varphi(x, z)$. Keep doing this until E_φ does not change any more. Every round of checking takes at most $2|\varphi|^2 \cdot |W|^3$ steps, and it will be stable

in at most $|\varphi| \cdot |W|^2$ rounds. Then we obtain the function E_φ^+ as we want. Every membership checking for $G \in E_\varphi^+(w, v)$ is finished in polynomial steps. So the whole process is still in P. \square

3.2.3 Model checking formulas with update modalities

As we address the case involving update modalities, let us consider a model $M = (W, E, C, \beta)$ and a world $w \in W$, and examine the formulas $(+_S)_a\psi$, $(-_S)_a\psi$, $(=)_S)_a\psi$ and $(\equiv)_b)_a\psi$. According to the semantics provided in Definition 2,

$$M, w \models (+_S)_a\psi \iff M^{a+S}, w \models \psi$$

where $M^{a+S} = (W, E, C^{a+S}, \beta)$ is defined such that

$$C^{a+S}(x) = \begin{cases} C(x), & \text{if } x \neq a \\ C(a) \cup S, & \text{if } x = a \end{cases}$$

From this, we deduce that verifying whether $M, w \models (+_S)_a\psi$ is reducible to checking if $M^{a+S}, w \models \psi$, effectively eliminating the leftmost update modality from consideration. An algorithm that invokes the model checking algorithm on the latter can be executed in linear time since it involves generating the updated model M^{a+S} directly from the original model M and considering the new formula ψ , which is a substring of the original formula. Hence, the complete algorithm, including the invocation of the model checking, will conclude within polynomial time.

The cases with $(-_S)_a\psi$, $(=)_S)_a\psi$ and $(\equiv)_b)_a\psi$ follow a similar process, with the distinction that each involves a different modification to the model. Nonetheless, the computational cost remains within P for both scenarios. This leads us to the following theorem:

Theorem 9. *The model checking problems for $L_{CDEF+-\equiv}$ and all of its sublogics are in P.*

3.3 Model checking quantified formulas: PSPACE complete

We demonstrate the PSPACE hardness by reducing, in polynomial time, the problem of undirected edge geography (UEG) – a variant of the generalized geography [32, 25] – to the model checking problem for any of L_{\boxplus} , L_{\boxminus} or L_{\square} , since UEG is a game for which determining a winning strategy is known to be PSPACE complete [17]. The PSPACE upper bound is established using a polynomial space algorithm that builds upon the algorithms introduced earlier.

Let $G = (D, R)$ be an undirected graph; i.e., D is a finite nonempty set, and R is a symmetric and irreflexive relation on D . Given a node $d \in D$, the pair (G, d) is referred to as a *rooted undirected graph*. The undirected edge geography (UEG) game on (G, d) involves two players, and unfolds as follows.

1. **Player I's Move:** Player I starts by selecting edge $\{d, d_1\} \in R$. If no such edge exists, the game ends and Player II wins as Player I cannot make a valid move.
2. **Player II's Move:** After Player I's move selecting an edge $\{d_i, d_{i+1}\}$, Player II must choose an edge $\{d_{i+1}, d_{i+2}\}$ that has not been chosen in previous moves. If Player II cannot make such a move, the game ends and Player I wins.
3. **Alternating Turns:** After Player II's move selecting an edge $\{d_j, d_{j+1}\}$, it is Player I's turn again to choose an edge $\{d_{j+1}, d_{j+2}\}$ not previously chosen. If Player I cannot make such a move, the game ends and Player II wins.
4. **Repeat Step 2:** The game continues by alternating turns following the process described in step 2.

Alternatively, UEG game on (G, d) can be recursively defined by modifying the graph after each move:

- The current player chooses an edge $\{d, d'\} \in R$; if this is impossible, he loses, and the game ends.
- The game then proceeds with the opposing player starting a new game on (G', d') where $G' = (D, R \setminus \{\{d, d'\}\})$.

The *UEG problem*, based on a rooted undirected graph, aims to determine whether Player I possesses a winning strategy.

Definition 10 (induced model). *Let $G = (D, R)$ represent an undirected graph. For each edge $\{x, y\} \in R$, assign a unique epistemic skill $s_{\{x, y\}} \in \mathbf{S}$ (ensuring that $s_{\{x', y'\}} \neq s_{\{x'', y''\}}$ for any distinct unordered pairs $\{x', y'\}$ and $\{x'', y''\}$), and for each node $x \in D$, assign a unique atomic proposition $p_x \in \mathbf{P}$ (ensuring that $p_{x'} \neq p_{x''}$ for any distinct nodes x' and x'').*

Define the induced model M_G as the tuple (D, E, C, β) where:

- E : For every $x, y \in D$, if $\{x, y\} \in R$, then $E(x, y) = \{s_{\{x, y\}}\}$; otherwise, $E(x, y) = \emptyset$;
- C : For all agents a , $C(a) = \emptyset$;
- β : For each node $x \in D$, $\beta(x) = \{p_x\}$.

This model M_G is well-defined and compactly represents the relationships and properties within the graph G . The size of E is $O(|D|^2)$ due to the pairwise relationship between nodes, while the size of β is $O(|D|)$, reflecting the unique property assignment per node. The size of C remains $O(|D|)$, given that only a limited number of agents are actually be utilized, as confirmed by the following definition and the definition of the size of the input.

Definition 11 (induced formula). *Let $G = (D, R)$ be an undirected graph. Consider n agents $a_1, \dots, a_n \in \mathbf{A}$, where n is the smallest positive even number greater than or equal to $|R|$. For each i where $1 \leq i \leq n$ (for φ_i , only consider even numbers i), define:*

$$\begin{aligned} \psi_i &:= \neg K_{a_i} \perp \wedge \bigvee_{x \in D} K_{a_i} p_x \\ \chi_i &:= \bigvee_{x, y \in D \text{ with } x \neq y, 1 \leq j < i} (p_x \wedge \hat{K}_{a_j} p_y \wedge K_{a_i} p_y) \\ \varphi_i &:= \diamond_{a_1} (\psi_1 \wedge \neg \chi_1 \wedge K_{a_1} \boxplus_{a_2} (\neg \psi_2 \vee \chi_2 \vee \\ &\quad \hat{K}_{a_2} \boxplus_{a_3} (\psi_3 \wedge \neg \chi_3 \wedge K_{a_3} \boxplus_{a_4} (\neg \psi_4 \vee \chi_4 \vee \\ &\quad \hat{K}_{a_4} \boxplus_{a_5} (\psi_5 \wedge \neg \chi_5 \wedge K_{a_5} \boxplus_{a_6} (\neg \psi_6 \vee \chi_6 \vee \\ &\quad \dots \\ &\quad \hat{K}_{a_{i-2}} \boxplus_{a_{i-1}} (\psi_{i-1} \wedge \neg \chi_{i-1} \wedge K_{a_{i-1}} \boxplus_{a_i} (\neg \psi_i \vee \chi_i) \dots))))). \end{aligned}$$

In the above, \hat{K}_a is the dual of K_a . The induced formula φ_G for the graph G is defined as φ_n .

Let us try to understand the induced formula. In a game, each agent a_i plays in the i -th move. The formulas ψ_i represents the condition where the player a_i at the i -th move chooses exactly one edge from the current node. The formula χ_i captures the scenario where the edge chosen by player a_i at the i -th move has been selected in a previous move, thus representing an invalid game move under the new edge rule. The conjunction $\psi_i \wedge \neg \chi_i$ ensures that each move in the game involves selecting a new, unvisited edge. As for complexity, the length of ψ_i is in $O(|D|)$, as it involves a disjunction over each node in D . The length of χ_i is in $O(|D|^2 \cdot |R|)$. The overall formula φ_G thus has its length in $O(|D|^2 \cdot |R|^2)$.

The formula φ_G constructs a logical framework that mirrors the gameplay in an undirected graph:

- \diamond_{a_1} : Indicates the potential for player a_1 to make a valid move by upskilling.
- $\psi_1 \wedge \neg \chi_1$: Ensures that a_1 's choice is a new edge (valid move).

- $K_{a_1} \boxplus_{a_2}$: Player a_1 ensures that no matter how player a_2 responds (upskills), the game's next state must be described by the formula that follows. And that formula describes that either a_2 does not find a new edge to choose (leading to the end the game), or, if a_2 chooses a new edge, then the formula starting with $\hat{K}_{a_3} \boxplus_{a_3}$ must hold, indicating a situation similar to the first clause above (but for a_3).

This recursive and intertwined structure of φ_G effectively captures the strategic progression of the game, with each player's move affecting the possible moves of the next player, all within the framework of an undirected graph where each node represents a game state or choice.

We now introduce a lemma that establishes a connection between the undirected edge geography problem and the logics we have developed.

Lemma 12. *For any rooted undirected graph (G, d) , Player I has a winning strategy in the undirected edge geography game on (G, d) , if and only if $M_G, d \models \varphi_G$.*

Proof. We show the lemma by induction on $|R|$. Base case $|R| = 0$, $n = 2$. For any $x, y \in D$, $\{x, y\} \notin R$. Player I loses in this case. Let $M_G = (D, E, C, \beta)$ be the induced model. Then $E(x, y) = \emptyset$ for any $x, y \in D$. We need to show $M_G, d \not\models \varphi_G$. For any finite non-empty $S \subseteq \mathbb{S}$, consider the model $M' = (D, E, C^{a_1+S}, \beta)$. Since $\varphi_G = \varphi_2 = \boxplus_{a_1}(\psi_1 \wedge \neg\chi_1 \wedge K_{a_1} \boxplus_{a_2}(\neg\psi_2 \vee \chi_2))$, where $\psi_1 = \neg K_{a_1} \perp \wedge \bigvee_{x \in D} K_{a_1} p_x$, $\chi_1 = \perp$, $\psi_2 = \neg K_{a_2} \perp \wedge \bigvee_{x \in D} K_{a_2} p_x$, and $\chi_2 = \bigvee_{x \neq y \in D} (p_x \wedge \hat{K}_{a_1} p_y \wedge K_{a_2} p_y)$. It is clear that $M', d \not\models \psi_1$, since $M', d \models K_{a_1} \perp$. It follows that $M', d \not\models \psi_1 \wedge \neg\chi_1 \wedge K_{a_1} \boxplus_{a_2}(\neg\psi_2 \vee \chi_2)$. Since S is arbitrary, we have $M_G, d \not\models \varphi_G$.

Base case $|R| = 1$, and so $n = 2$. Let $\{d, d'\}$ be the unique edge in R . Let $M_G = (D, E, C, \beta)$ be the induced model. $E(d, d') = E(d', d) = \{s_{\{d, d'\}}\}$, and $E(x, y) = \emptyset$ otherwise. Player I has a winning strategy in this case, and we show that $M_G, d \models \varphi_G$. Consider $S = \{s_{\{d, d'\}}\}$. Let $M' = (D, E, C^{a_1+S}, \beta)$, with $\varphi_G = \varphi_2 = \boxplus_{a_1}(\psi_1 \wedge \neg\chi_1 \wedge K_{a_1} \boxplus_{a_2}(\neg\psi_2 \vee \chi_2))$, where:

- $\psi_1 = \neg K_{a_1} \perp \wedge \bigvee_{x \in D} K_{a_1} p_x$ ($M', d \models \psi_1$, for $M', d \models \neg K_{a_1} \perp \wedge K_{a_1} p_{d'}$)
- $\chi_1 = \perp$ ($M', d \models \neg\chi_1$)
- $\psi_2 = \neg K_{a_2} \perp \wedge \bigvee_{x \in D} K_{a_2} p_x$
- $\chi_2 = (p_d \wedge \hat{K}_{a_1} p_{d'} \wedge K_{a_2} p_{d'}) \vee (p_{d'} \wedge \hat{K}_{a_1} p_d \wedge K_{a_2} p_d) \vee \bigvee_{x \neq y \in D \setminus \{d, d'\}} (p_x \wedge \hat{K}_{a_1} p_y \wedge K_{a_2} p_y)$.

For any finite nonempty $S' \subseteq \mathbb{S}$, let $M'' = (D, E, (C^{a_1+S})^{a_2+S'}, \beta)$, we have one of the following cases:

- (1) $S' \not\subseteq S$, then $\forall x \in D$, $(C^{a_1+S})^{a_2+S'}(a_2) \not\subseteq E(d, x)$, hence $M'', d' \models \neg\psi_2$, for $M'', d' \models K_{a_2} \perp$.
- (2) $S' \subseteq S$, then $M'', d' \models p_{d'} \wedge \hat{K}_{a_1} p_d \wedge K_{a_2} p_d$. Thus, $M'', d' \models \chi_2$ for its right disjunct is satisfied.

In both case $M'', d' \models \neg\psi_2 \vee \chi_2$, and so $M', d' \models \boxplus_{a_2}(\neg\psi_2 \vee \chi_2)$, and $M', d \models K_{a_1} \boxplus_{a_2}(\neg\psi_2 \vee \chi_2)$. Together with the verifications above, we have $M_G, d \models \varphi_G$.

The case $|R| = k$. The direction from left to right. Suppose that Player I has a winning strategy, by which she chooses in the first move $\{d, d'\}$. Let $M_G = (D, E, C, \beta)$ be the induced model. We need to show that $M_G, d \models \varphi_G$, where $\varphi_G = \boxplus_{a_1}(\psi_1 \wedge \neg\chi_1 \wedge K_{a_1} \varphi_{G, \boxplus_{a_2}})$, in which $\varphi_{G, \boxplus_{a_2}}$ is the subformula of φ_G beginning with \boxplus_{a_2} (see Def. 11). Consider $S = \{s_{\{d, d'\}}\}$, and let $M' = (D, E, C^{a_1+S}, \beta)$:

- $\psi_1 = \neg K_{a_1} \perp \wedge \bigvee_{x \in D} K_{a_1} p_x$ ($M', d \models \psi_1$, for $M', d \models \neg K_{a_1} \perp \wedge K_{a_1} p_{d'}$)
- $\chi_1 = \perp$ ($M', d \models \neg\chi_1$)

Now we show $M', d \models K_{a_1} \varphi_{G, \boxplus_{a_2}}$; namely, $M', d' \models \varphi_{G, \boxplus_{a_2}}$, where $\varphi_{G, \boxplus_{a_2}} = \boxplus_{a_2} (\neg \psi_2 \vee \chi_2 \vee \hat{K}_{a_2} \varphi_{G, \boxplus_{a_3}})$ in which $\varphi_{G, \boxplus_{a_3}}$ is the subformula of φ_G beginning with \boxplus_{a_3} . For any finite nonempty $S' \subseteq \mathbf{S}$, let $M'' = (D, E, (C^{a_1+S})^{a_2+S'}, \beta)$, and it suffices to show that

$$M'', d' \models \neg \psi_2 \vee \chi_2 \vee \hat{K}_{a_2} \varphi_{G, \boxplus_{a_3}}, \quad (\dagger)$$

where $\psi_2 = \neg K_{a_2} \perp \wedge \bigvee_{x \in D} K_{a_2} p_x$ and $\chi_2 = \bigvee_{x \neq y \in D} (p_x \wedge \hat{K}_{a_1} p_y \wedge K_{a_2} p_y)$. Consider the possible cases:

- (1) There does not exist $x \in D$ such that $S' \subseteq E(d', x)$, or
- (2) There exists $d'' \in D$ such that $S' \subseteq E(d', d'')$ (note that S' must be singleton).

In case (1), $M'', d' \models K_{a_2} \perp$, so $M'', d' \models \neg \psi_2$, hence (\dagger) holds. In case (2), Player I has a winning strategy in the continued game on (G_2, d'') with $G_2 = (D, R \setminus \{\{d, d'\}, \{d', d''\}\})$ (note that d'' cannot be d or d'). It suffices to show the following result:

$$M'', d'' \models \varphi_{G, \boxplus_{a_3}} \iff M_{G_2}, d'' \models \varphi_{G_2}. \quad (\ddagger)$$

Since $M_{G_2}, d'' \models \varphi_{G_2}$ holds by the induction hypothesis, by (\ddagger) , we have $M'', d'' \models \varphi_{G, \boxplus_{a_3}}$. This makes the rightmost disjunct of (\dagger) true in M'', d' , and completes the whole proof.

Let $M_{G_2} = (D, E_2, C, \beta)$. To see (\ddagger) , $M'', d'' \models \varphi_{G, \boxplus_{a_3}}$, i.e., $(D, E, (C^{a_1+S})^{a_2+S'}, \beta), d'' \models \varphi_{G, \boxplus_{a_3}}$
 $\iff (D, E_2, (C^{a_1+S})^{a_2+S'}, \beta), d'' \models \varphi'_{G, \boxplus_{a_3}}$, where $\varphi'_{G, \boxplus_{a_3}}$ is adapted from $\varphi_{G, \boxplus_{a_3}}$ by the following:

- Delete all occurrences of $\bigvee_{x \neq y \in D} (p_x \wedge \hat{K}_{a_1} p_y \wedge K_{a_1} p_y)$ from $\varphi_{G, \boxplus_{a_3}}$
- Delete all occurrences of $\bigvee_{x \neq y \in D} (p_x \wedge \hat{K}_{a_2} p_y \wedge K_{a_1} p_y)$ from $\varphi_{G, \boxplus_{a_3}}$

(This equivalence holds since $E(d, d') = E(d', d) = \emptyset$, which implies that any formulas $\hat{K}_{a_1} \varphi$ and $\hat{K}_{a_2} \varphi$ are false in any world x of model (D, E_2, C', β) , where C' is any capability function updated from $(C^{a_1+S})^{a_2+S'}$ without changing the capabilities of a_1 and a_2 .)

$\iff (D, E_2, C, \beta), d'' \models \varphi''_{G, \boxplus_{a_3}}$, where $\varphi''_{G, \boxplus_{a_3}}$ a variant of $\varphi'_{G, \boxplus_{a_3}}$ by replacing any a_{i+2} with a_i ,

(This holds since $(C^{a_1+S})^{a_2+S_2}(a_{i+2}) = C(a_i) = \emptyset$; note that a_1 and a_2 does not exist in $\varphi'_{G, \boxplus_{a_3}}$.)

$\iff M_{G_2}, d'' \models \varphi_{G_2}$, i.e., $(D, E_2, C, \beta), d'' \models \varphi_{G_2}$ (since $\varphi_{G_2} = \varphi''_{G, \boxplus_{a_3}}$)

From right to left. If Player I does not have a winning strategy, we must show that $M_G, d \not\models \varphi_G$. Let the induced model M_G be (D, E, C, β) . Since Player I does not have a winning strategy, then:

- (a) There is no $x \in D$ such that $\{d, x\} \in R$, and Player I loses in this case; or
- (b) Player I does not have a winning strategy by choosing in the first move any $x \in D \setminus \{d\}$ such that $\{d, x\} \in R$.

For case (a): Since $E(d, x) = \emptyset$ for any x , we get $M_G, d \not\models \varphi_G$ similarly to the case when $|R| = 0$.

For case (b): Consider an arbitrary finite nonempty $S \subseteq \mathbf{S}$. Then:

- (1) For all $x \in D$, $S \not\subseteq E(d, x)$; or
- (2) There exists $d' \in D$ such that $S \subseteq E(d, d')$ (note that d' cannot be d).

We need to show $M_G, d \not\models \varphi_G$ where φ_G is given in Def. 11. Let $M' = (D, E, C^{a_1+S}, \beta)$. In case (1), since $M', d \models K_{a_1} \perp$, $M', d \not\models \psi_1$ (with $\psi_1 = \neg K_{a_1} \perp \wedge \bigvee_{x \in D} K_{a_1} p_x$), and so $M, d \not\models \varphi_G$.

In case (2) (under the case (b)), there must exist $d'' \in D \setminus \{d, d'\}$ such that Player I does not have a winning strategy in the game on (G_2, d'') where $G_2 = (D, R \setminus \{\{d, d'\}, \{d', d''\}\})$; for otherwise Player I

has a winning strategy (this is also the case when there is no such a d''), leading to a contradiction. Let $S' = \{s_{\{d', d''\}}\}$, then $S' \subseteq E(d', d'')$. Let $M'' = (D, E, (C^{a_1+S})^{a_2+S'}, \beta)$. It suffices to show that

$$M'', d' \not\models \neg\psi_2 \vee \chi_2 \vee \hat{K}_{a_2} \varphi_{G, \diamond_{a_3}}, \quad (*)$$

Consider $\psi_2 = \neg K_{a_2} \perp \wedge \bigvee_{x \in D} K_{a_2} p_x$. Since $M'', d' \models \neg K_{a_2} \perp \wedge K_{a_2} p_{d''}$, we have $M'', d' \not\models \neg\psi_2$. As for $\chi_2 = \bigvee_{x \neq y \in D} (p_x \wedge \hat{K}_{a_1} p_y \wedge K_{a_2} p_y)$, since $M'', d' \models \hat{K}_{a_1} p_y \wedge K_{a_2} p_y$ implies $y = d \neq d'' = y$, we have $M'', d' \not\models \chi_2$. Finally we show that $M'', d' \not\models \hat{K}_{a_2} \varphi_{G, \diamond_{a_3}}$. Since there is exact one $x \in D$ (which must be d'' by the definition of S') such that $S' \subseteq E(d', x)$, it suffices to prove $M'', d'' \not\models \varphi_{G, \diamond_{a_3}}$. Note that (\ddagger) from the proof of the converse direction can also be shown here, it suffices to show that $M_{G_2}, d'' \not\models \varphi_{G_2}$, and this holds by the induction hypothesis. \square

Corollary 13. *Undirected edge geography is polynomial time reducible to the model checking problem for L_{\boxplus} .*

Remark 2. *It is important to note that the reduction discussed previously utilizes only the modalities \boxplus and \diamond . However, we can also perform a reduction using exclusively the modalities \square and \diamond . This alternative reduction is structurally similar to the original, with the primary modification being the replacement of \boxplus with \square . Additionally, a reduction that employs only the modalities \boxplus and \diamond is also feasible. In this case, we replace \boxplus with \boxminus . Furthermore, there is a requirement to modify the skill set $C(a_i)$ to $\{s_{\{w,v\}} \mid w, v \in D\}$. The model checking problems for any logics that include at least one of the modalities $\boxplus, \boxminus, \square, \diamond, \diamond$ or \diamond – remain PSPACE hard. This complexity assertion holds even in the absence of additional modalities such as $C_G, D_G, E_G, F_G, (+S)_a, (-S)_a, (=S)_a$, and $(\equiv_S)_a$.*

Lemma 14. *The model checking problem for $L_{CDEF+-\equiv\boxplus\boxminus}$ is in PSPACE.*

Proof. With the presence of Algorithm 2, it suffices to provide a polynomial space algorithm for the types of formulas $\boxplus_a \varphi, \square_a \varphi$ and $\boxminus_a \varphi$. The details are given in Algorithm 3.

Algorithm 3 Function $Val((W, E, C, \beta), \varphi)$ extended: cases with quantifiers

1: Initialize: $tmpVal \leftarrow \emptyset$	14: Initialize: $n \leftarrow \text{true}$
2: Initialize: $S_1 \leftarrow (\bigcup_{w,v \in W} E(w, v)) \cup (\bigcup_{a \text{ appears in } \varphi} C(a))$	15: for all $S \subseteq S_2$ do
3: Initialize: $S_2 \leftarrow S_1 \cup \{s\} \triangleright$ Here $s \in S$ is new for S_1	16: \quad if $S \neq \emptyset$ and $t \notin Val((W, E, C^{a=S}, \beta), \psi)$
4: if ... then ... \triangleright Same as in Algorithm 2	17: $\quad \quad$ then $n \leftarrow \text{false}$
5: else if $\varphi = \boxplus_a \psi$ then	18: \quad if $n = \text{true}$ then $tmpVal \leftarrow tmpVal \cup \{t\}$
6: \quad for all $t \in W$ do	19: return $tmpVal \triangleright$ Returns $\{t \in W \mid \forall S \subseteq S_1 : t \in Val((W, E, C^{a=S}, \beta), \psi)\}$
7: $\quad \quad$ Initialize: $n \leftarrow \text{true}$	20: else if $\varphi = \square_a \psi$ then
8: $\quad \quad$ for all $S \subseteq S_2$ do	21: \quad for all $t \in W$ do
9: $\quad \quad \quad$ if $S \neq \emptyset$ and $t \notin Val((W, E, C^{a=S}, \beta), \psi)$	22: $\quad \quad$ Initialize: $n \leftarrow \text{true}$
10: $\quad \quad \quad$ then $n \leftarrow \text{false}$	23: $\quad \quad$ for all $S \subseteq S_2$ do
11: $\quad \quad$ if $n = \text{true}$ then $tmpVal \leftarrow tmpVal \cup \{t\}$	24: $\quad \quad \quad$ if $S \neq \emptyset$ and $t \notin Val((W, E, C^{a=S}, \beta), \psi)$
12: return $tmpVal \triangleright$ Returns $\{t \in W \mid \forall S \subseteq S_1 : t \in Val((W, E, C^{a=S}, \beta), \psi)\}$	25: $\quad \quad \quad$ then $n \leftarrow \text{false}$
13: else if $\varphi = \boxminus_a \psi$ then	26: $\quad \quad$ if $n = \text{true}$ then $tmpVal \leftarrow tmpVal \cup \{t\}$
14: \quad for all $t \in W$ do	27: return $tmpVal \triangleright$ Returns $\{t \in W \mid \forall S \subseteq S_1 : t \in Val((W, E, C^{a=S}, \beta), \psi)\}$

Here we furthermore need to check the space cost caused by the new modalities. But notice that all the space cost of checking $Val((W, E, C, \beta), \varphi)$ is in $O(|M|)$. So the space cost of the algorithm is immediately linear. So the model checking problem for $L_{CDEF+-\equiv\boxplus\boxminus}$ is in PSPACE. \square

We reach the following result from Corollary 13 (considering that UEG is PSPACE complete) and Lemma 14.

Theorem 15. *The model checking problems for all logics with quantifiers (i.e., at least one of \boxplus , \boxminus and \boxdot) that extends the base logic L is PSPACE complete.*

4 Discussion

We have developed a variety of logics that incorporate individual and group knowledge, actions such as knowing, forgetting, revising, and learning, as well as the necessity and possibility of these actions. These logics are highly expressive, yet the computational cost for model checking remains manageable. Specifically:

- For logics devoid of quantifiers, the complexity of model checking falls within the class P, aligning with many traditional epistemic logics.
- For logics that include quantifiers, the complexity is PSPACE complete. This matches the complexity found in similar types of logics, such as Group Announcement Logic [2], Coalition Announcement Logic [29, 19, 4], and Subset Space Arbitrary Announcement Logic [6].³

Logicians are deeply interested in the decidability of validity/satisfiability problems for logics that include quantifiers over updates. Known complexities, such as the undecidability of Arbitrary Public Announcement Logic (APAL) and Group Announcement Logic [18, 1], have spurred ongoing research into decidable alternatives [18, 11, 10]. Even the development of variants that are recursively axiomatizable represents an advancement [35, 8], especially given that APAL is not likely to have this feature.

Our research also aims to explore the decidability and computational complexity of satisfiability and validity problems within our logics. Although our ongoing efforts have yielded PSPACE completeness and EXPTIME completeness results for many of our less complex logics (for example, the satisfiability problems for logics devoid of common knowledge, update modalities, and quantifiers are PSPACE complete, whereas those lacking update modalities and quantifiers but incorporating common knowledge are EXPTIME complete), a definitive result for the full logic $L_{CDEF+-==\boxplus\boxminus\boxdot}$ remains elusive. Additionally, while we have successfully axiomatized some of our logics in previous studies [24], an axiomatic system for the full logic is not yet developed. These areas are designated for future exploration and development.

We have introduced a new update modality for learning, $(\equiv_b)_a$, which denotes the action where agent a learns the skills of agent b . This operator essentially replaces a 's skill set with that of b . However, we can also devise variants that facilitate skill set modification through incremental learning (e.g., $(\cup_b)_a$) or decremental learning (e.g., $(\cap_b)_a$ “retaining only beneficial skills from b ”, or $(\setminus_b)_a$ “eliminating undesirable skills of b ”). Additionally, the concept of “deskilling,” derived from the richness of natural language, refers to a reduction in the skills required to perform a task. This could be modeled as an update action that modifies the edge function, thereby requiring fewer skills to distinguish between worlds, potentially leading to knowledge acquisition. Incorporating these diverse learning modalities does not increase the complexity of the model checking problem, though it may add complexity to the validity problem. The exploration of quantifiers over learning operators presents another intriguing area of study.

Acknowledgements We express our gratitude to the anonymous reviewers for their invaluable comments and suggestions. We acknowledge the financial support by the MOE Project of Humanities and Social Sciences (No. 24YJA72040002) and the National Social Science Fund of China (Grant No. 20&ZD047).

³It is worth noting that model checking in Arbitrary Public Announcement Logic is also believed to be PSPACE complete [5]. However, a detailed validation of this claim has not yet been found by us.

References

- [1] T. Ågotnes, H. van Ditmarsch & T. French (2016): *The Undecidability of Quantified Announcements*. *Studia Logica* 104(4), pp. 597–640, doi:10.1007/s11225-016-9657-0.
- [2] Thomas Ågotnes, Philippe Balbiani, Hans van Ditmarsch & Pablo Seban (2010): *Group Announcement Logic*. *Journal of Applied Logic* 8(1), pp. 62–81, doi:10.1016/j.jal.2008.12.002.
- [3] Carlos E. Alchourrón, Peter Gärdenfors & David Makinson (1985): *On the Logic of Theory Change: Partial Meet Contraction and Revision Functions*. *The Journal of Symbolic Logic* 50, pp. 510–530, doi:10.2307/2274239.
- [4] Natasha Alechina, Hans van Ditmarsch, Rustam Galimullin & Tuo Wang (2021): *Verification and Strategy Synthesis for Coalition Announcement Logic*. *Journal of Logic, Language and Information* 30(4), pp. 671–700, doi:10.1007/s10849-021-09339-6.
- [5] Philippe Balbiani, Alexandru Baltag, Hans van Ditmarsch, Andreas Herzig, Tomohiro Hoshi & Tiago de Lima (2008): ‘Knowable’ as ‘Known after an Announcement’. *The Review of Symbolic Logic* 1(3), pp. 305–334, doi:10.1017/S1755020308080210.
- [6] Philippe Balbiani, Hans van Ditmarsch & Andrey Kudinov (2013): *Subset Space Logic with Arbitrary Announcements*. In: *Proceedings of ICLA 2013*, pp. 233–244, doi:10.1007/978-3-642-36039-8_21.
- [7] Alexandru Baltag, Lawrence S. Moss & Sławomir Solecki (1998): *The Logic of Public Announcements, Common Knowledge, and Private Suspicions*. In I. Gilboa, editor: *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 98)*, pp. 43–56, doi:10.5555/645876.671885.
- [8] Alexandru Baltag, Aybüke Özgün & Ana Lucia Vargas Sandoval (2023): *Arbitrary Public Announcement Logic with Memory*. *Journal of Philosophical Logic* 52(1), pp. 53–110, doi:10.1007/s10992-022-09664-6.
- [9] Alexandru Baltag, Aybüke Özgün & Ana Lucia Vargas Sandoval (2017): *Topo-Logic as a Dynamic-Epistemic Logic*. In Alexandru Baltag, Jeremy Seligman & Tomoyuki Yamada, editors: *Logic, Rationality, and Interaction*, Springer Berlin Heidelberg, pp. 330–346, doi:10.1007/978-3-662-55665-8_23.
- [10] Hans van Ditmarsch & Tim French (2022): *Quantifying over Boolean announcements*. *Logical Methods in Computer Science* 18(1), doi:10.46298/lmcs-18(1:20)2022.
- [11] Hans van Ditmarsch, Tim French & Sophie Pinchinat (2010): *Future Event Logic – Axioms and Complexity*. In Lev D. Beklemishev, Valentin Goranko & Valentin B. Shehtman, editors: *Advances in Modal Logic 8, papers from the eighth conference on "Advances in Modal Logic," held in Moscow, Russia, 24-27 August 2010*, College Publications, pp. 77–99.
- [12] Hans van Ditmarsch, Andreas Herzig, Jérôme Lang & Pierre Marquis (2009): *Introspective Forgetting*. *Synthese* 169(2), pp. 405–423, doi:10.1007/s11229-009-9554-4.
- [13] Hans van Ditmarsch, Wiebe van der Hoek & Barteld Kooi (2008): *Dynamic Epistemic Logic*. *Synthese Library* 337, Springer Netherlands, doi:10.1007/978-1-4020-5839-4.
- [14] Huimin Dong, Xu Li & Yi N. Wang (2021): *Weighted Modal Logic in Epistemic and Deontic Contexts*. In Sujata Ghosh & Thomas Icard, editors: *Proceedings of the Eighth International Conference on Logic, Rationality and Interaction (LORI 2021)*, *Lecture Notes of Theoretical Computer Science* 13039, Springer, pp. 73–87, doi:10.1007/978-3-030-88708-7_6.
- [15] Ronald Fagin & Joseph Y. Halpern (1988): *Belief, Awareness, and Limited Reasoning*. *Artificial Intelligence* 34(1), pp. 39–76, doi:10.1016/0004-3702(87)90003-8.
- [16] Ronald Fagin, Joseph Y. Halpern, Yoram Moses & Moshe Y. Vardi (1995): *Reasoning about Knowledge*. The MIT Press, doi:10.7551/mitpress/5803.001.0001.
- [17] Aviezer S Fraenkel, Edward R Scheinerman & Daniel Ullman (1993): *Undirected Edge Geography*. *Theoretical Computer Science* 112(2), pp. 371–381, doi:10.1016/0304-3975(93)90026-P.
- [18] Tim French & Hans van Ditmarsch (2008): *Undecidability for Arbitrary Public Announcement Logic*. In Carlos Areces & Robert Goldblatt, editors: *Advances in Modal Logic*, 7, College Publications, pp. 23–42.

- [19] Rustam Galimullin, Natasha Alechina & Hans van Ditmarsch (2018): *Model Checking for Coalition Announcement Logic*. In Frank Trollmann & Anni-Yasmin Turhan, editors: *KI 2018: Advances in Artificial Intelligence*, Springer International Publishing, Cham, pp. 11–23, doi:10.1007/978-3-030-00111-7_2.
- [20] Mikkel Hansen, Kim Guldstrand Larsen, Radu Mardare & Mathias Ruggaard Pedersen (2018): *Reasoning about Bounds in Weighted Transition Systems*. *Logical Methods in Computer Science* 14(4), pp. 1–32, doi:10.23638/LMCS-14(4:19)2018.
- [21] Jaakko Hintikka (1962): *Knowledge and Belief: An Introduction to the Logic of Two Notions*. Cornell University Press, Ithaca, New York.
- [22] Jôme Lang, Paolo Liberatore & Pierre Marquis (2003): *Propositional Independence: Formula-Variable Independence and Forgetting*. *Journal of Artificial Intelligence Research* 18(1), pp. 391–443, doi:10.5555/1622420.1622431.
- [23] Kim G. Larsen & Radu Mardare (2014): *Complete Proof Systems for Weighted Modal Logic*. *Theoretical Computer Science* 546(12), pp. 164–175, doi:10.1016/j.tcs.2014.03.007.
- [24] Xiaolong Liang & Yi N. Wáng (2022): *Epistemic Logic over Weighted Graphs*. In: *Proceedings of the Second International Workshop on Logics for New-Generation AI*, College Publications, pp. 43–58.
- [25] David Lichtenstein & Michael Sipser (1980): *GO Is Polynomial-Space Hard*. *Journal of the ACM* 27(2), pp. 393–401, doi:10.1145/322186.322201.
- [26] Fangzhen Lin & Ray Reiter (1994): *Forget It!* In: *Working Notes of AAAI Fall Symposium on Relevance*, pp. 154–159.
- [27] John-Jules Ch. Meyer & Wiebe van der Hoek (1995): *Epistemic Logic for AI and Computer Science*. Cambridge University Press, doi:10.1017/CBO9780511569852.
- [28] Pavel Naumov & Jia Tao (2015): *Logic of Confidence*. *Synthese* 192, pp. 1821–1838, doi:10.1007/s11229-014-0655-3.
- [29] Marc Pauly (2002): *A Modal Logic for Coalition Power in Games*. *Journal of Logic Computation* 12(1), pp. 149–166, doi:10.1093/logcom/12.1.149.
- [30] Jan A. Plaza (1989): *Logics of Public Communications*. In M. L. Emrich, M. S. Pfeifer, M. Hadzikadic & Z. W. Ras, editors: *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems (ISMIS '89)*, Oak Ridge National Laboratory, pp. 201–216.
- [31] W. V. Quine (1956): *Quantifiers and Propositional Attitudes*. *The Journal of Philosophy* 53(5), pp. 177–187, doi:10.2307/2022451.
- [32] Thomas J. Schaefer (1978): *On the Complexity of Some Two-Person Perfect-Information Games*. *Journal of Computer and System Sciences* 16(2), pp. 185–225, doi:10.1016/0022-0000(78)90045-4.
- [33] Yi N. Wáng & Thomas Ågotnes (2013): *Subset Space Public Announcement Logic*. In Kamal Lodaya, editor: *Proceedings of ICLA, Lecture Notes in Computer Science 7750*, Springer, pp. 245–257, doi:10.1007/978-3-642-36039-8_22.
- [34] Georg H. von Wright (1951): *An Essay in Modal Logic*. Studies in Logic and the Foundations of Mathematics, North-Holland Publishing Company.
- [35] Kang Xu & Yi N. Wáng (2018): *Group Simple Announcement Logic*. *Studies in Logic* 11(1), pp. 1–22.
- [36] Yan Zhang & Yi Zhou (2009): *Knowledge forgetting: Properties and applications*. *Artificial Intelligence* 173(16), pp. 1525–1537, doi:10.1016/j.artint.2009.07.005.