Artificial Intelligence 309 (2022) 103727

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Data-informed knowledge and strategies

Junli Jiang[a], Pavel Naumov[b,*]

[a] Institute of Logic and Intelligence, Southwest University, Chongqing, China
[b] University of Southampton, Southampton, United Kingdom

A R T I C L E I N F O

Article history:
Received 19 June 2021
Received in revised form 5 February 2022
Accepted 20 April 2022
Available online 26 April 2022

Keywords:
Knowledge
Strategy
Know-how
Multiagent system
Data
Completeness

A B S T R A C T

The article proposes a new approach to reasoning about knowledge and strategies in multiagent systems. It emphasizes data, not agents, as the source of strategic knowledge. The approach brings together Armstrong's functional dependency from database theory, a data-informed knowledge modality based on a recent work by Baltag and van Benthem, and a newly proposed data-informed strategy modality. The main technical result is a sound and complete logical system that describes the interplay between these three logical operators.

1. Introduction

With technological progress, more information is either stored in databases and remote servers or exchanged directly between autonomous agents. What machines know and can do will rely more and more on the access to such information rather than the individual memory. Personal assistants, like Amazon Alexa, often use external sources of data such as Wikipedia and Answers.com. Self-driving cars can access map updates in the cloud. Vehicular ad hoc networks (vehicular cloud) are being designed to share traffic and other localised information between nearby vehicles [1]. Future medical robots will be treating patients with highly contagious diseases relying on knowledge and skills of doctors and nurses placed in safe remote locations [2].

In this article, we formally define and study the properties of knowledge and abilities in a multiagent setting where data is decoupled from the agents.

1.1. Deep sea rescue example

Suppose that three naval rescue robots, Bluewater (b), Lucky (l), and Extreme (e), are sent on a mission to save the crew of a sunk submarine that has one hour of oxygen left. The area in which the submarine sank could be divided into 9 squares depicted in Fig. 1. The actual location of the submarine (second row, first column) is not known to the robots and it takes one hour for one robot to search through one square. Note that even a single robot in this setting has a strategy to save the crew (search first square in the second row), but the robot does not *know* that this strategy would guarantee the success of the rescue operation.

* Corresponding author.
E-mail addresses: walk08@swu.edu.cn (J. Jiang), p.naumov@soton.ac.uk (P. Naumov).

https://doi.org/10.1016/j.artint.2022.103727

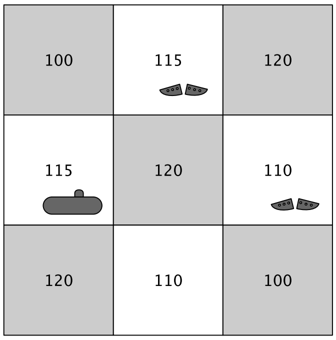

Fig. 1. Rescue example.

Let us now suppose that the rescue robots somehow learned that the sub is located in the second row. Then, they know a joint strategy to save the crew. The strategy consists in three of them searching through different squares in the second row. Moreover, observe that *anyone* who knows in which row (r) the sub is, knows the strategy that Bluewater, Lucky, and Extreme can use to save the crew:

$$\mathsf{S}_r^{b,l,e}(\text{“The crew is safe”}).$$

We write $\mathsf{S}_X^C\varphi$ to state that the knowledge of a strategy that coalition C can use to achieve φ could be gained from the values of variables in set X. In other words, *anyone* who knows the values of the variables in set X knows the strategy that coalition C can use to achieve φ. In this context, we refer to the set of variables X as “dataset”. We read $\mathsf{S}_X^C\varphi$ as “dataset X *informs* a strategy of coalition C to achieve φ”. Note that for $\mathsf{S}_X^C\varphi$ to be true, it is not significant whether the members of the coalition C themselves know values of variables in dataset X. Furthermore, *any* knowledge that the members of the coalition C might have does not affect if $\mathsf{S}_X^C\varphi$ is true or not. For this reason, we refer to the members of the coalition C as *actors* rather than agents.

Recall that Bluewater, just like each of the other robots, has a strategy to save the crew (search the first square in the second row), but it is not true that anyone who knows r would know how Bluewater can do this. Thus,

$$\neg\mathsf{S}_r^{b}(\text{“The crew is safe”}).$$

In other words, Bluewater's strategy to rescue the crew is not informed by the dataset $\{r\}$. The same dataset also does not inform the strategy for Bluewater and Lucky:

$$\neg\mathsf{S}_r^{b,l}(\text{“The crew is safe”}).$$

Let us further assume that the two of the squares contain old shipwrecks. These are the second square in the first row and the third square in the second row, Fig. 1.

Let Boolean variable s is true in the squares that contain shipwrecks and is false in the other squares. Observe that everyone who knows the row and the ship wracks data (s) of the square where the sub has been sunk, would know that the sub is located in one of the first two squares in the second row. Thus, any such person would know how Bluewater and Lucky can achieve the goal:

$$\mathsf{S}_{r,s}^{b,l}(\text{“The crew is safe”}).$$

The validity of the last statement depends on the location of the sub. It would not be true if the sub is located in any of the squares of the third row. In other words, the satisfiability of statement $\mathsf{S}_X^C\varphi$ depends on which of the possible worlds is the current world. In the setting of our example, there are nine possible worlds, corresponding to different locations of the sub. By default, all statements that we consider in this section assume that the current world is the one where the sub is located in the first cell of the second row. Observe that the column (c) and the wreck data do not inform a strategy for Bluewater and Lucky:

$$\neg\mathsf{S}_{c,s}^{b,l}(\text{“The crew is safe”}),$$

because all three squares in the first column have no wrecks.

Let us also assume that the ocean floor in some squares is covered with sand and in the others with rocks. The squares with rock floor are shaded in grey in Fig. 1. Since the sub is lying on the sandy floor and only two of squares with sandy floor have no shipwrecks, everyone who knows the floor type and the ship wreck data, knows the strategy that Bluewater and Lucky can use to save the crew:

$\mathsf{S}^{b,l}_{f,s}$("The crew is safe").

The validity of the last statement also depends on the current world. It would not be true if the sub was lying on rocky floor. One might also observe that row, floor type, and wreck data inform a strategy for Bluewater alone to save the crew:

$\mathsf{S}^{b}_{r,f,s}$("The crew is safe").

The numbers in Fig. 1 show the depth (d), in meters, of the ocean in each square. Observe that the depth of the ocean in white (sandy) squares is equal to either 115 or 110. Thus, anyone who knows the type of the floor on which the sub is lying knows that d is equal to either 115 or 110. We write this as

K_f("The sub is lying at either depth 115 or depth 110").

In general, we write $\mathsf{K}_X\varphi$ if the knowledge of φ about the current world can be gained from the values of variables in dataset X. In other words, anyone who knows the values of the variables in dataset X for the current world knows that statement φ is true in the current world. We read $\mathsf{K}_X\varphi$ as "dataset X *informs* the knowledge of statement φ".

Note that variable f does not inform strategy for Bluewater and Lucky to save the crew:

$\neg\mathsf{S}^{b,l}_{f}$("The crew is safe")

because there are four different squares with sandy floor. However, everyone who knows the value of f knows that the depth at which the sub is lying is either 115 or 110. Thus, everyone who knows the value of f knows that the knowledge of variable d reduces the number of locations where the sub is to just two. Hence, everyone who knows the value of f knows that variable d informs a strategy for Bluewater and Lucky to save the crew:

$\mathsf{K}_f\mathsf{S}^{b,l}_{d}$("The crew is safe").

In this article, in addition to data-informed modalities $\mathsf{S}_X\varphi$ and $\mathsf{K}_X\varphi$, we also consider dependency expression $X \rhd Y$. It means that, *in the current world*, the knowledge of the values of variables in dataset X informs the knowledge of the values of variables in dataset Y. We read $X \rhd Y$ as "dataset X informs dataset Y". For example, note that all squares in the first column have no wrecks. Thus, the knowledge of the column in which the sub is located informs the knowledge of the wreck data: $c \rhd s$. At the same time, $\neg(r \rhd s)$ because no cells in the second row have the same wreck data.

1.2. Literature review

Coalition power modality $\mathsf{S}^C\varphi$ for perfect information games has been originally proposed by Pauly [3,4]. Ågotnes and Alechina have combined this modality with several group and individual knowledge modalities in imperfect information setting [5]. Using their language, one can express statements like "coalition C knows that coalition D has a strategy to achieve φ". Note that *knowing that a strategy exists* is different from *knowing the strategy*. A complete logical system for modality "coalition knows a strategy that it can use to maintain condition φ" has been proposed by Naumov and Tao [6]. Modality "agent knows a strategy she can use to achieve φ in several steps" has been axiomatized by Fervari, Herzig, Li, and Wang [7]. The interplay between distributed knowledge modality, "coalition has a strategy to achieve φ in one step" modality, and "coalition knows a strategy it can use to achieve φ in one step" modality has been described in [8,9]. Several logical systems that combine modality "coalition knows a strategy it can use to achieve φ in one step" with different forms of knowledge is proposed in [5]. Modality "coalition knows a strategy it can use to achieve φ in one step" in a perfect recall setting is axiomatized in [10]. Bisimulations for different logics of "knowing strategy" is proposed in [11]. Neighbourhood semantics for one of these logics is introduced in [12]. Multiple other extensions of Pauly's coalition logic have been proposed, including Logic of Cooperation and Propositional Control [13] and Alternating-time Temporal Logic (ATL) [14]. An epistemic version of ATL is introduced by van der Hoek and Wooldridge [15]. For more discussion of earlier works in this area see [16].

All of the above works that include "knowing strategy" modality assume that the knowing agent (or coalition) is the same as the agent whose strategy she knows. For example, such modality can be used to express statement "Alice knows a strategy that *she herself can use* to achieve φ". However, it cannot be used to say that "Alice knows a strategy that *Bob can use* to achieve φ". A logical system that can be used to express the last statement is proposed by Naumov and Tao [17]. They have introduced modality "coalition C knows a strategy that coalition D can use to achieve φ". They call this modality "second-order know-how" modality and give a complete logical system that describes the interplay between this modality and distributed knowledge modality.

Note that in many real world settings the knowledge is separated from the actors. For example, knowledge can be stored in a database, on a remote server, or deep in the sea inside the flight recorder of a crashed airplane. It can be encrypted or password-protected. In such situations, it is natural to reason about what different agents can do depending on which information they can access. To formalise such reasoning, in this article we propose to completely decouple knowledge from the actors. To do this, we introduce data-informed strategy modality $\mathsf{S}^C_X\varphi$ described in the previous section. This modality has certain resemblance to condition know-how modality $\mathsf{S}_{\psi}\varphi$ that stands for "there is a strategy to achieve φ from any state where ψ is true" [18].

The functional dependency relation $X \rhd Y$ is proposed by Armstrong, who has given its sound and complete axiomatization [19]. His axioms became known in database literature as Armstrong's axioms [20, p. 81]. Beeri, Fagin, and Howard [21] have suggested a variation of Armstrong's axioms that describe properties of multi-valued dependence. Baltag proposed a logical system for expression $X \rhd_a Y$, that stands "agent a knows how to compute dataset Y based on dataset X" [22]. Deuser and Naumov have discussed a connection between Armstrong axioms and strategies in imperfect information setting [23]. It is interesting to point out that dependency $x \rhd y$ between single variables x and y could be expressed in the dynamic epistemic logic of "knowing the value" [24]. Namely, $x \rhd y$ is equivalent to $[x]\mathsf{Kv}(y)$, where $[x]$ is the modality that represents public announcement (inspection) of variable x and $\mathsf{Kv}(y)$ is expression "variable y is publicly known". More generally, if X and Y are finite sets $\{x_1, \dots, x_m\}$ and $\{y_1, \dots, y_n\}$ respectively, then $X \rhd Y$ is equivalent to $[x_1]\dots[x_m]\bigwedge_{i\le n}\mathsf{Kv}(y_i)$. Another approach to dependency is proposed in Dependence Logic [25].

Data-informed knowledge modality $\mathsf{K}_X\varphi$ is introduced by Baltag and van Benthem [26]. They have proposed a complete logical system, A Simple Logic of Functional Dependence, that captures the interplay between functional dependency relation $X \rhd Y$ and data-informed knowledge modality $\mathsf{K}_X\varphi$. Due to the choices Baltag and van Benthem have made in the syntax and the semantics of their system, their logic contains the axiom $\varphi \to \mathsf{K}_X\varphi$ (under certain restrictions on formula φ). Perhaps because of the presence of this axiom, they refer to modality K_X as "dependence" rather than "knowledge" modality. In the current article, we use slightly different syntax and semantics. As a result, the above principle is not universally valid in our setting. The properties of modality K_X in our setting are the standard S5 properties. This is why we refer to K_X as data-informed *knowledge* modality. In Section 3, we further discuss the difference between our setting and the one in [26].

The main technical contribution of the current article is a sound and complete logical system describing the interaction between data-informed strategy modality $\mathsf{S}^C_X\varphi$, data-informed knowledge modality $\mathsf{K}_X\varphi$, and functional dependency expression $X \rhd Y$.

1.3. Outline

The rest of the article is structured as follows. In the next section, we introduce formal models of our logical system. Then, we describe the syntax and the semantics of the system. Section 4 lists its axioms and inference rules. We show soundness of our logical system in Section 5. It turns out that whether our system is strongly complete depends on the finiteness of the set of all data variables. We prove that if this set is finite, then our system is strongly complete with respect to the proposed semantics. We also show that, if the set is infinite, then any strongly sound system in our language is *not* strongly complete. The proof of strong completeness is in Section 6. The incompleteness result is given in Section 7. Finally, in Section 8, we show that even if the set of all variables is infinite, our logical system is complete (but not strongly complete) for the formulae that use only finitely many data variables. In Section 9, we propose and analyse a model checking algorithm for our logical system and in, Section 10, we discuss a possible future work on *multistep* data-informed strategies. Section 11 concludes.

2. Games

Throughout this article, we assume a fixed nonempty set of propositional variables P, a fixed set of data variables V, and a fixed set of actors $\mathcal{A}$. Recall that we use term "actors" rather than "agents" to emphasise that the knowledge in our setting is decoupled from the actions.

By a *dataset* we mean an arbitrary subset of V. In this section, we introduce models of our logical system, called *games*. Informally, a game includes a set of possible states and each of data variables is assigned a value in each of the states. An actor who is informed about a dataset cannot distinguish two states when all variables in the dataset have the same values in both states. Note that it is only important if a given variable has the same value in two given states and it is not important what exactly the value of the variable is. Thus, for the sake of simplicity, our formal definition below does not include values of variables. It only contains an indistinguishability relation $\sim_x$ on possible states associated with each data variable $x \in V$. Intuitively, two states are indistinguishable by a data variable x if this variable has the same value in both states. In this article, by B^A we denote the set of all functions from set A to set B.

Definition 1. A game is a triple $(W, \{\sim_x\}_{x\in V}, \Delta, M, \pi)$, where

1. W is a (possibly empty) set of states,
2. $\sim_x$ is an indistinguishability equivalence relation on set W for each data variable $x \in V$,
3. Δ is a nonempty set of "actions",
4. $M \subseteq W \times \Delta^{\mathcal{A}} \times W$ is a "mechanism" of the game,
5. $\pi(p) \subseteq W$ for each propositional variable $p \in P$.

By a *complete action profile* we mean an arbitrary element of the set $\Delta^{\mathcal{A}}$. By a *coalition* we mean an arbitrary subset $C \subseteq \mathcal{A}$ of actors. By an *action profile* of a coalition C we mean an arbitrary element of the set Δ^C.

Our introductory example has 9 initial states, representing the different locations of the sub, and two final states ("saved" and "not saved"). Note that although in this example the states are naturally divided into initial and final, the same is not true in general. In Definition 1, we assume that, the game might make multiple consecutive transitions between states.

In our example, the set V contains variables r, c, s, f, and d. If w is the current state, in which the sub is located at $(2,1)$, and w' is the state in which the sub is located at $(1,2)$, then $w \sim_f w'$ and $w \sim_d w'$ because squares $(2,1)$ and $(1,2)$ have the same floor type and the same depth.

To keep the notations simple, in Definition 1, we assume that the set of actions Δ is the same for all actors in all states. This assumption is not significant for our results because available actions can always be combined into a single set and the additional actions can be assigned some "default" meaning. At the same time, our assumption that set Δ is nonempty is significant. Without this assumption, the Public Knowledge axiom, introduced in Section 4, is not valid.

Note that the mechanism M is a relation, not a function. Informally, $(w,\delta,w') \in M$ if under complete action profile $\delta \in \Delta^{\mathcal{A}}$ the game can transition from state w to state w'. Defining mechanism as a relation allows us to model nondeterministic games where from a given state under a given complete action profile the game can transition to one of several possible "next" states. Note that we also allow that for some combinations of a state and a complete action profile there might be no next states. We interpret this as a termination of the game.

In our introductory example, the actions consist in searching a square. Since there are nine squares, the set Δ has nine actions corresponding to these squares. The game transitions from the current initial state $w = (2,1)$ to final state "saved" if at least one of the actions is searching in the square where the sub is lying. Otherwise, the game transitions to the final state "not saved". Note that we do not allow the actors to repeat the game. Thus, no further transitions can be made from either of the two final states. We model this by assuming that the mechanism M of this game has no triples whose first element is one of the final states of the game.

As common in modal logics, we interpret propositional variables as properties of states. Informally, $w \in \pi(p)$ if propositional variable p is true in state $w \in W$. This is different from the setting of [26], where Baltag and van Benthem have used atomic predicates instead of propositional variables. The predicates are true or false depending not on the state, but on the values of data variables in the state. In other words, the atomic formulae in their setting capture properties of the variables rather than of the states. We discuss the significance of this difference in the next section.

3. Syntax and semantics

Language Φ of our logical system is defined by the grammar

$$\varphi ::= p \mid X \rhd X \mid \neg\varphi \mid (\varphi \to \varphi) \mid \mathsf{K}_X\varphi \mid \mathsf{S}^C_X\varphi,$$

where $p \in P$ is a propositional variable, $X \subseteq V$ is a dataset, and $C \subseteq \mathcal{A}$ is a coalition. We read $X \rhd Y$ as "dataset X informs dataset Y", $\mathsf{K}_X\varphi$ as "dataset X informs the knowledge of φ", and $\mathsf{S}^C_X\varphi$ as "dataset X informs a strategy of coalition C to achieve φ". By $\overline{\mathsf{K}}_X\varphi$ we mean formula $\neg\mathsf{K}_X\neg\varphi$. We also assume that constant true $\top$, conjunction $\wedge$, and biconditional $\leftrightarrow$ are defined in the standard way. In this article, we omit curly brackets and parenthesis when it does not create confusion. For example, we write x instead of $\{x\}$, $x_1,\dots,x_n$ instead of $\{x_1,\dots,x_n\}$, and $\varphi \to \psi$ instead of $(\varphi \to \psi)$.

For any states $w, w' \in W$ and any dataset $X \subseteq V$, let $w \sim_X w'$ mean that $w \sim_x w'$ for each data variable $x \in X$. In particular, $w \sim_\varnothing w'$ is true for any states $w, w' \in W$. Also, we write $f =_B g$ if $f(b) = g(b)$ for each element b of a set B.

Definition 2. For any state $w \in W$ of a game $(W, \{\sim_x\}_{x\in V}, \Delta, M, \pi)$ and any formula $\varphi \in \Phi$, satisfaction relation $w \Vdash \varphi$ is defined recursively as follows

1. $w \Vdash p$, if $w \in \pi(p)$,
2. $w \Vdash X \rhd Y$, when for each $w' \in W$ if $w \sim_X w'$, then $w \sim_Y w'$,
3. $w \Vdash \neg\varphi$, if $w \nVdash \varphi$,
4. $w \Vdash \varphi \to \psi$, if $w \nVdash \varphi$ or $w \Vdash \psi$,
5. $w \Vdash \mathsf{K}_X\varphi$, if $w' \Vdash \varphi$ for each $w' \in W$ such that $w \sim_X w'$,
6. $w \Vdash \mathsf{S}^C_X\varphi$, when there is an action profile $s \in \Delta^C$ of coalition C such that for all states $w', v \in W$ and each complete action profile $\delta \in \Delta^{\mathcal{A}}$ if $w \sim_X w'$, $s =_C \delta$, and $(w',\delta,v) \in M$, then $v \Vdash \varphi$.

Observe that $\mathsf{K}_\varnothing\varphi$ is the universal modality that says "statement φ is true in each state of the game". If $\mathsf{K}_\varnothing\varphi$ is true in each state, then everyone must know it. For this reason, we read $\mathsf{K}_\varnothing\varphi$ as "statement φ is *public knowledge*".

The sentence "dataset X informs dataset Y" could be interpreted in two ways: locally and globally. Under the first interpretation, the values of variables X *in the current state* determine the values of variables Y. Under the second interpretation, the values of X determine the values of Y *in each state*. For example, suppose that real values of variables x and y are such that $y = x^2$ in each state of the game. Then, the value of x globally determines the value of y, but the value of y does not globally determine the value of x. However, the value of y determines the value of x *locally* in each state where $y = 0$. Item 2 of Definition 2, defines the semantics of expression $X \rhd Y$ as *local* dependency. The global dependency can be captured by the expression $\mathsf{K}_\varnothing(X \rhd Y)$.

The data-informed strategies also can be local and global. For example, supposed that based on the test results X a doctor knows how to adjust a medication. This is a global data-informed strategy because the doctor would know how to adjust medication no matter what the results X are. Of course, for different test results the strategy (adjustment amount) would be different. We can specify such global data-informed strategies as functions that map each X-equivalent class of states into an action. A local strategy might exist only for the values of X in the current state. For example, a doctor might have a strategy to save the life of a cancer patient for the current values of the test results X. For some other value of X she might no longer have such a strategy. A local data-informed strategy is a single action that guarantees result only in the X-equivalence class of the current state. Item 6 of Definition 2 defines modality $\mathsf{S}^C_X\varphi$ as a claim of existence of a local data-informed strategy. The modality for global strategy could be defined as $\mathsf{K}_\varnothing\mathsf{S}^C_X\varphi$.

Recall from the previous section that atomic formulae in [26] capture properties of variables rather than states. The same is true about non-atomic formulae as well. In fact, Baltag and van Benthem define satisfaction $s \Vdash \varphi$ as a relation between an *assignment* s of values to data variables and a formula φ. Because in their setting the validity of any formula is completely determined by the values of variables that occur in the formula, their logical system contains the axiom $\varphi \to \mathsf{K}_X\varphi$ for any formula φ such that all data variables that occur in φ belong to X. This axiom is not valid in our setting.

Technically, the axiom $\varphi \to \mathsf{K}_X\varphi$ is not valid under our approach because we are using a more traditional semantics in which satisfaction $w \Vdash \varphi$ is a relation between a *state* $w \in W$ and a formula φ. Although the approach of [26] is more succinct, it eliminates a possibility to have different states with the same values of all data variables. As a result of this elimination, the data-informed knowledge modality acquires the additional property captured by the above axiom. Besides that axiom, the properties of modality K_X in the current article are the same as in [26].

4. Axioms

In addition to propositional tautologies in language Φ, our logical system contains the following axioms.

1. Truth: $\mathsf{K}_X\varphi \to \varphi$,
2. Negative Introspection: $\neg\mathsf{K}_X\varphi \to \mathsf{K}_X\neg\mathsf{K}_X\varphi$,
3. Distributivity: $\mathsf{K}_X(\varphi \to \psi) \to (\mathsf{K}_X\varphi \to \mathsf{K}_X\psi)$,
4. Reflexivity: $X \rhd Y$, where $Y \subseteq X$,
5. Transitivity: $X \rhd Y \to (Y \rhd Z \to X \rhd Z)$,
6. Augmentation: $X \rhd Y \to (X \cup Z) \rhd (Y \cup Z)$,
7. Introspection of Dependency: $X \rhd Y \to \mathsf{K}_X(X \rhd Y)$,
8. Knowledge Monotonicity: $X \rhd Y \to (\mathsf{K}_Y\varphi \to \mathsf{K}_X\varphi)$,
9. Cooperation: $\mathsf{S}^C_X(\varphi \to \psi) \to (\mathsf{S}^D_X\varphi \to \mathsf{S}^{C\cup D}_X\psi)$, where $C \cap D = \varnothing$,
10. Strategic Monotonicity: $X \rhd Y \to (\mathsf{S}^C_Y\varphi \to \mathsf{S}^C_X\varphi)$,
11. Strategic Introspection: $\mathsf{S}^C_X\varphi \to \mathsf{K}_X\mathsf{S}^C_X\varphi$,
12. Knowledge of Unavoidability: $\mathsf{K}_X\mathsf{S}^\varnothing_Y\varphi \to \mathsf{S}^\varnothing_X\varphi$,
13. Public Knowledge: $\mathsf{K}_\varnothing\varphi \to \mathsf{S}^C_X\varphi$.

The Truth, the Negative Introspection, and the Distributivity axioms are the standard principles from epistemic logic S5. The Reflexivity, the Transitivity, and the Augmentation are well-known Armstrong's axioms for functional dependency [19].

The Introspection of Dependency axiom states that if a dataset X informs a dataset Y, then this is known to anyone with access to X. The Knowledge Monotonicity axioms states that if a dataset X informs a dataset Y and dataset Y informs the knowledge of φ, then dataset X also informs the knowledge of φ. The Cooperation axiom is a variation of Marc Pauly's axiom introduced in the logic of coalition power [3,4]. It states that if a dataset X informs strategies (actions profiles) of disjoint coalitions C and D to achieve $\varphi \to \psi$ and φ, respectively, then the dataset also informs a joint strategy for these coalitions to achieve ψ. The Strategic Monotonicity axiom states that if a dataset X informs a dataset Y, then X informs each strategy informed by Y. The Strategic Introspection axiom states that if a dataset X informs a strategy, then X also informs the knowledge that it informs the strategy.

To understand the meaning of the Knowledge of Unavoidability axiom, note that statement $\mathsf{K}_X\mathsf{S}^C_Y\varphi$ means that "anyone who knows X knows that anyone who knows Y knows a strategy of coalition C to achieve φ". Let us refer to the knowers of X and Y as Xena and Yeily. Note that while Yeily knows the strategy, Xena only knows that the strategy exists and is known to Yeily. Generally speaking, Xena does not know what the strategy is. One important exception, however, is when coalition C is empty. In this case, coalition C has only a single strategy (the unique function from the set Δ^C). In such a situation, knowing that a strategy exists is equivalent to knowing what the strategy is. This is captured by the Knowledge of Unavoidability axiom. The name of the axiom comes from the fact that $\mathsf{S}^\varnothing_Y\varphi$ can also be interpreted as "anyone who knows Y, knows that φ is unavoidable".

By item 3 of Definition 1, each game has at least one action. Such an action can be used by the members of any coalition to achieve any statement which is true in each state of the game. This is captured by the Public Knowledge axiom.

We write $\vdash \varphi$ and say that formula $\varphi \in \Phi$ is a theorem of our system if it is derivable from the above axioms using the Modus Ponens and the Necessitation inference rule:

$$\frac{\varphi, \varphi \to \psi}{\psi} \qquad \frac{\varphi}{\mathsf{K}_X\varphi}.$$

In addition to unary relation $\vdash \varphi$, we also consider a binary relation $F \vdash \varphi$. We write $F \vdash \varphi$ if formula $\varphi \in \Phi$ is provable from the set of formulae $F \subseteq \Phi$ and the theorems of our logical system using only the Modus Ponens inference rule. We say that set F is inconsistent if there is a formula $\varphi \in F$ such that $F \vdash \varphi$ and $F \vdash \neg\varphi$.

We prove the soundness of our logical system in Section 5. The proofs of the next four standard lemmas are omitted.

Lemma 1. $\vdash \mathsf{K}_X\varphi \to \mathsf{K}_X\mathsf{K}_X\varphi$.

Lemma 2 *(Deduction). If $F, \varphi \vdash \psi$, then $F \vdash \varphi \to \psi$.*

Lemma 3. *If $\varphi_1, \dots, \varphi_n \vdash \psi$, then $\mathsf{K}_X\varphi_1, \dots, \mathsf{K}_X\varphi_n \vdash \mathsf{K}_X\psi$.*

Lemma 4 *(Lindenbaum). Any consistent set of formulae can be extended to a maximal consistent set of formulae.*

Lemma 5. *Inference rule $\dfrac{\varphi}{\mathsf{S}^C_X\varphi}$ is derivable.*

Proof. Suppose that $\vdash \varphi$. Thus, $\vdash \mathsf{K}_\varnothing\varphi$ by the Necessitation inference rule. Therefore, $\vdash \mathsf{S}^C_X\varphi$ by the Public Knowledge axiom and the Modus Ponens inference rule. $\square$

Lemma 6. $\vdash \mathsf{S}^C_X\varphi \to \mathsf{S}^D_X\varphi$, *where $C \subseteq D$.*

Proof. Note that formula $\varphi \to \varphi$ is a tautology. Thus, $\vdash \mathsf{S}^{D\setminus C}_X(\varphi \to \varphi)$ by Lemma 5. Hence, $\vdash \mathsf{S}^C_X\varphi \to \mathsf{S}^{(D\setminus C)\cup C}_X\varphi$ by the Cooperation axiom and the Modus Ponens inference rule. Therefore, $\vdash \mathsf{S}^C_X\varphi \to \mathsf{S}^D_X\varphi$ by the assumption $C \subseteq D$ of the lemma. $\square$

5. Soundness

In this section, we prove the soundness of our logical system. The soundness of the Armstrong's axioms, the Truth, the Negative Introspection, and the Distributivity axioms is straightforward. Below, we show the soundness of each of the remaining axioms for an arbitrary game $(W, \{\sim_x\}_{x\in V}, \Delta, M, \pi)$ as a separate lemma. We state the strong soundness for the whole system as Theorem 1 in the end of this section.

Lemma 7 *(Introspection of dependency). If $w \Vdash X \rhd Y$, then $w \Vdash \mathsf{K}_X(X \rhd Y)$.*

Proof. Consider any state $w' \in W$ such that

$$w \sim_X w'. \tag{1}$$

By item 5 of Definition 2, it is enough to prove that $w' \Vdash X \rhd Y$. Towards this proof, consider any state $w'' \in W$ such that $w' \sim_X w''$. By item 2 of Definition 2, it suffices to show that $w' \sim_Y w''$.

The assumptions $w \sim_X w'$ and $w' \sim_X w''$ imply $w \sim_X w''$. Hence, by the assumption $w \Vdash X \rhd Y$ of the lemma and item 2 of Definition 2,

$$w \sim_Y w''. \tag{2}$$

At the same time, statement (1), the assumption $w \Vdash X \rhd Y$ of the lemma, and item 2 of Definition 2 imply that $w \sim_Y w'$. Therefore, $w' \sim_Y w''$ by statement (2) and because relation $\sim_Y$ is symmetric and transitive. $\square$

Lemma 8 *(Knowledge monotonicity). If $w \Vdash X \rhd Y$ and $w \Vdash \mathsf{K}_Y\varphi$, then $w \Vdash \mathsf{K}_X\varphi$.*

Proof. Consider any state $w' \in W$ such that $w \sim_X w'$. By item 5 of Definition 2, it is enough to prove that $w' \Vdash \varphi$. Note that the assumption $w \sim_X w'$ implies $w \sim_Y w'$ by the assumption $w \Vdash X \rhd Y$ of the lemma and item 2 of Definition 2. Thus, $w \sim_Y w'$ by the assumption $w \sim_X w'$. Therefore, $w' \Vdash \varphi$ by the assumption $w \Vdash \mathsf{K}_Y\varphi$ of the lemma and item 5 of Definition 2. $\square$

Lemma 9 *(Cooperation). If $w \Vdash \mathsf{S}^C_X(\varphi \to \psi)$, $w \Vdash \mathsf{S}^D_X\varphi$, and sets C and D are disjoint, then $w \Vdash \mathsf{S}^{C\cup D}_X\psi$.*

Proof. By item 6 of Definition 2, the assumptions $w \Vdash \mathsf{S}^C_X(\varphi \to \psi)$ and $w \Vdash \mathsf{S}^D_X\varphi$ imply that there are action profiles $s_1 \in \Delta^C$ and $s_2 \in \Delta^D$ such that

$$\forall w', v \in W\ \forall \delta \in \Delta^{\mathcal{A}}(w \sim_X w', s_1 =_C \delta, (w', \delta, v) \in M \Rightarrow v \Vdash \varphi \to \psi) \tag{3}$$

and

$$\forall w', v \in W\ \forall \delta \in \Delta^{\mathcal{A}}(w \sim_X w', s_2 =_D \delta, (w', \delta, v) \in M \Rightarrow v \Vdash \varphi). \tag{4}$$

Consider action profile

$$s(a) = \begin{cases} s_1(a), & \text{if } a \in C, \\ s_2(a), & \text{if } a \in D, \end{cases} \tag{5}$$

of coalition $C \cup D$. Note that profile s is well-defined because sets C and D are disjoint by the assumption of the lemma. Consider any states $w', v \in W$ and any complete action profile $\delta \in \Delta^{\mathcal{A}}$ such that $w \sim_X w'$, $s =_{C\cup D} \delta$, and $(w', \delta, v) \in M$. By item 6 of Definition 2, it suffices to show that $v \Vdash \psi$.

Assumption $s =_{C\cup D} \delta$ implies that $s_1 =_C \delta$ and $s_2 =_D \delta$ by equation (5). Thus, $v \Vdash \varphi \to \psi$ and $v \Vdash \varphi$ by statements (3) and (4) and assumptions $w \sim_X w'$ and $(w', \delta, v) \in M$. Therefore, $v \Vdash \psi$ by item 4 of Definition 2. □

Lemma 10 *(Strategic monotonicity). If $w \Vdash X \rhd Y$ and $w \Vdash \mathsf{S}^C_Y\varphi$, then $w \Vdash \mathsf{S}^C_X\varphi$.*

Proof. By item 6 of Definition 2, the assumption $w \Vdash \mathsf{S}^C_Y\varphi$ implies that there is an action profile $s \in \Delta^C$ of coalition C such that

$$\forall w', v \in W\ \forall \delta \in \Delta^{\mathcal{A}}(w \sim_Y w', s =_C \delta, (w', \delta, v) \in M \Rightarrow v \Vdash \varphi). \tag{6}$$

Consider any states $w', v \in W$ and any complete action profile $\delta \in \Delta^{\mathcal{A}}$ such that $w \sim_X w'$, $s =_C \delta$, and $(w', \delta, v) \in M$. By item 6 of Definition 2, it suffices to prove that $v \Vdash \varphi$. Indeed, by item 2 of Definition 2, the assumption $w \Vdash X \rhd Y$ of the lemma and statement $w \sim_X w'$ imply that $w \sim_Y w'$. Therefore, $v \Vdash \varphi$ by statement (6). □

Lemma 11 *(Strategic introspection). If $w \Vdash \mathsf{S}^C_X\varphi$, then $w \Vdash \mathsf{K}_X\mathsf{S}^C_X\varphi$.*

Proof. By item 6 of Definition 2, the assumption $w \Vdash \mathsf{S}^C_X\varphi$ implies that there is an action profile $s \in \Delta^C$ of coalition C such that

$$\forall w', v \in W\ \forall \delta \in \Delta^{\mathcal{A}}(w \sim_X w', s =_C \delta, (w', \delta, v) \in M \Rightarrow v \Vdash \varphi). \tag{7}$$

Consider any state $u \in W$ such that $w \sim_X u$. By item 5 of Definition 2, it suffices to show that $u \Vdash \mathsf{S}^C_X\varphi$. Indeed, the assumption $w \sim_X u$ and statement (7) imply that

$$\forall w', v \in W\ \forall \delta \in \Delta^{\mathcal{A}}(u \sim_X w', s =_C \delta, (w', \delta, v) \in M \Rightarrow v \Vdash \varphi).$$

Therefore, $u \Vdash \mathsf{S}^C_X\varphi$ by item 6 of Definition 2. □

Lemma 12 *(Knowledge of unavoidability). If $w \Vdash \mathsf{K}_X\mathsf{S}^{\varnothing}_Y\varphi$, then $w \Vdash \mathsf{S}^{\varnothing}_X\varphi$.*

Proof. Note that there is only one action profile $s \in \Delta^{\varnothing}$ of the empty coalition. As a function, this action profile consists of the empty set of pairs. We denote this profile by s_0.

Consider any states $u \in W$ such that $w \sim_X u$. By item 6 of Definition 2, it suffices to prove that

$$\forall v \in W\ \forall \delta \in \Delta^{\mathcal{A}}(s_0 =_{\varnothing} \delta, (u, \delta, v) \in M \Rightarrow v \Vdash \varphi). \tag{8}$$

Indeed, by item 5 of Definition 2, the assumption $w \sim_X u$ and the assumption $w \Vdash \mathsf{K}_X\mathsf{S}^{\varnothing}_Y\varphi$ of the lemma imply that $u \Vdash \mathsf{S}^{\varnothing}_Y\varphi$. Hence, by item 6 of Definition 2, there is an action profile $s_1 \in \Delta^{\varnothing}$ of the empty coalition such that

$$\forall w', v \in W\ \forall \delta \in \Delta^{\mathcal{A}}(u \sim_Y w', s_1 =_{\varnothing} \delta, (w', \delta, v) \in M \Rightarrow v \Vdash \varphi).$$

Recall that the action profile of the empty coalition is unique. Thus, $s_0 = s_1$. Hence,

$$\forall w', v \in W\ \forall \delta \in \Delta^{\mathcal{A}}(u \sim_Y w', s_0 =_{\varnothing} \delta, (w', \delta, v) \in M \Rightarrow v \Vdash \varphi).$$

The last statement in the case $w' = u$ is equivalent to statement (8). □

Lemma 13 *(Public knowledge). If $w \Vdash \mathsf{K}_\varnothing\varphi$, then $w \Vdash \mathsf{S}^C_X\varphi$.*

Proof. To prove $w \Vdash \mathsf{S}^C_X\varphi$, by item 6 of Definition 2, it suffices to show that $v \Vdash \varphi$ for each state $v \in W$. Indeed, consider any state $v \in W$. Note that, vacuously, $w \sim_\varnothing v$. Therefore, $v \Vdash \varphi$ by item 5 of Definition 2 and the assumption $w \Vdash \mathsf{K}_\varnothing\varphi$ of the lemma. □

The strong soundness theorem below follows from the lemmas above.

Theorem 1 *(Strong soundness). For each state w of an arbitrary game, each set of formulae $F \subseteq \Phi$ and each formula $\varphi \in \Phi$, if $w \Vdash f$ for each formula $f \in F$ and $F \vdash \varphi$, then $w \Vdash \varphi$.* □

6. Strong completeness

In this section, we prove the completeness of our logical system. The proof combines three ideas: harmony, dataset closure, and canonical tree introduced in Section 6.1, Section 6.2, and Section 6.3 respectively. Section 6.3 also describes the canonical game construction. Section 6.4 states and proves basic properties of the canonical games which are used later in Section 6.5 to prove the strong completeness.

6.1. Harmony

Usually, a proof of the completeness for a modal logic such as S5 defines states as the maximal consistent sets of formulae. It also contains a truth lemma that states that a formula is satisfied in a state if and only if the formula belongs to the state as a set. The truth lemma is usually proven by induction on structural complexity of the formula. An important step during the proof by induction is to show that if a state (maximal consistent set) w does *not* contain a formula of the form $\Box\varphi$, then there is a "reachable" state w' such that $\neg\varphi \in w'$. In our proof, the truth lemma is Lemma 31. In the case of modality K, the described above step is formalised as Lemma 25. The latter lemma is used in the induction proof in combination with item 5 of Definition 2.

The situation is very different in the case of the modality S because item 6 of Definition 2 contains quantifiers over two states: w' and v. As a result, in the case of modality S, the mentioned above step in the induction proof must construct *two* states satisfying certain conditions. As it turns out, these two states cannot be constructed consecutively. To construct these states concurrently, Naumov and Tao proposed a "harmony" construction [9]. A modified version of this construction is also used in [17]. The construction consists in defining a certain relation between two sets, called "harmony". First, it is shown that some initial pair of sets is in harmony. Then, it is shown that any pair in harmony can be expanded till the pair is in "complete harmony". Finally, Lindenbaum's lemma is used to complete two sets in complete harmony to two maximal consistent sets.

We further adjust the construction from [17] to handle data-informed modalities. We define our version of "harmony" relation in Definition 3. In Lemma 15, we show that a certain initial pair of sets is in harmony. Lemma 16 shows that any pair in harmony can be extended in a certain way preserving the harmony. Definition 4 introduces the notion of "complete harmony" as the maximal extension of this type. Finally, Lemma 17 states that any pair in harmony can be extended to a pair in a complete harmony. The harmony construction is ultimately used in Lemma 30 to construct the mentioned above states w' and v.

Definition 3. A pair of sets of formulae (F, G) is in *harmony* if $F \nvdash \mathsf{S}^\varnothing_V\neg\bigwedge G'$ for each finite set $G' \subseteq G$.

Recall that V denotes the fixed set of all data variables and that $\mathsf{S}^\varnothing_V\varphi$ stands for "for the given values of V, statement φ is unavoidable after transition". Thus, informally, "$F \nvdash \mathsf{S}^\varnothing_V\neg\bigwedge G'$ for each finite set $G' \subseteq G$" means that "for the given values of V, the assumptions F about the current state are consistent with the set of statements G being satisfied in the next state".

Definition 3 is a slightly modified version of the definition of harmony from [17]. The definition there is for the second-order know-how modality $\mathsf{S}^D_C\varphi$, where both C and D are coalitions of agents. It means "coalition C knows a strategy that coalition D can use to achieve φ". The definition of harmony in [17] requires $F \nvdash \mathsf{S}^\varnothing_C\neg\bigwedge G'$ for each coalition C and each finite set $G' \subseteq G$.

Lemma 14. *If pair of sets of formulae (F, G) is in harmony, then sets F and G are consistent.*

Proof. If set F is inconsistent, then $F \vdash \varphi$ for any formula $\varphi \in \Phi$. In particular, $F \vdash \mathsf{S}^\varnothing_V\neg\bigwedge\varnothing$. Therefore, the pair (F, G) is not in harmony by Definition 3.

Next, let set G be inconsistent. Thus, by Lemma 2 and propositional reasoning, $\vdash \neg\bigwedge G'$ for some finite set $G' \subseteq G$. Hence, $\vdash \mathsf{S}^\varnothing_V\neg\bigwedge G'$ by Lemma 5. Thus, $F \vdash \mathsf{S}^\varnothing_V\neg\bigwedge G'$. Therefore, the pair (F, G) is not in harmony by Definition 3. □

Next, we show that certain initial sets are in harmony.

Lemma 15. *For any consistent set $E \subseteq \Phi$, any formula $\neg \mathsf{S}^C_X \varphi \in E$, any family $\{D_i\}_{i\in I}$ of pairwise disjoint subsets of C, and any set $\{\mathsf{S}^{D_i}_X \psi_i\}_{i\in I}$ of formulae from set E, if*

$$F = \{\chi \mid \mathsf{K}_X \chi \in E\}, \tag{9}$$

$$G = \{\neg\varphi\} \cup \{\psi_i\}_{i\in I} \cup \{\sigma \mid \mathsf{K}_\varnothing \sigma \in E\}, \tag{10}$$

then the pair (F, G) is in harmony.

Proof. Suppose that the pair (F, G) is not in harmony. Thus, by Definition 3, there is a finite set $G' \subseteq G$ such that

$$F \vdash \mathsf{S}^\varnothing_V \neg \bigwedge G'. \tag{11}$$

The assumption that set G' is finite and equation (10) imply that there are formulae

$$\mathsf{K}_\varnothing \sigma_1, \dots \mathsf{K}_\varnothing \sigma_m \in E \tag{12}$$

and indices $i_1, \dots, i_n \in I$ such that

$$\vdash \sigma_1 \to (\sigma_2 \to \dots (\sigma_m \to (\psi_{i_1} \to \dots (\psi_{i_n} \to (\neg\varphi \to \bigwedge G')) \dots)) \dots).$$

Thus, by the laws of propositional reasoning,

$$\vdash \neg \bigwedge G' \to (\sigma_1 \to \dots (\sigma_m \to (\psi_{i_1} \to \dots (\psi_{i_n} \to \varphi) \dots)) \dots).$$

Hence, by Lemma 5,

$$\vdash \mathsf{S}^\varnothing_V \left(\neg \bigwedge G' \to (\sigma_1 \to \dots (\sigma_m \to (\psi_{i_1} \to \dots (\psi_{i_n} \to \varphi) \dots)) \dots)\right).$$

Then, by the Cooperation axiom and the Modus Ponens inference rule,

$$\vdash \mathsf{S}^\varnothing_V \left(\neg \bigwedge G'\right) \to \mathsf{S}^\varnothing_V (\sigma_1 \to \dots (\sigma_m \to (\psi_{i_1} \to \dots (\psi_{i_n} \to \varphi) \dots)) \dots).$$

Thus, by assumption (11) and the Modus Ponens inference rule,

$$F \vdash \mathsf{S}^\varnothing_V (\sigma_1 \to \dots (\sigma_m \to (\psi_{i_1} \to \dots (\psi_{i_n} \to \varphi) \dots)) \dots).$$

Hence, by equation (9), there are formulae

$$\mathsf{K}_X \chi_1, \dots, \mathsf{K}_X \chi_k \in E \tag{13}$$

such that

$$\chi_1, \dots, \chi_k \vdash \mathsf{S}^\varnothing_V (\sigma_1 \to \dots (\sigma_m \to (\psi_{i_1} \to \dots (\psi_{i_n} \to \varphi) \dots)) \dots).$$

Then, by Lemma 3,

$$\mathsf{K}_X \chi_1, \dots, \mathsf{K}_X \chi_k \vdash \mathsf{K}_X \mathsf{S}^\varnothing_V (\sigma_1 \to \dots (\sigma_m \to (\psi_{i_1} \to \dots (\psi_{i_n} \to \varphi) \dots)) \dots).$$

Thus, by the Knowledge of Unavoidability axiom and the Modus Ponens inference rule,

$$\mathsf{K}_X \chi_1, \dots, \mathsf{K}_X \chi_k \vdash \mathsf{S}^\varnothing_X (\sigma_1 \to \dots (\sigma_m \to (\psi_{i_1} \to \dots (\psi_{i_n} \to \varphi) \dots)) \dots).$$

Hence, by assumption (13),

$$E \vdash \mathsf{S}^\varnothing_X (\sigma_1 \to \dots (\sigma_m \to (\psi_{i_1} \to \dots (\psi_{i_n} \to \varphi) \dots)) \dots).$$

Then, by the Cooperation axiom and the Modus Ponens inference rule,

$$E \vdash \mathsf{S}^\varnothing_X \sigma_1 \to \mathsf{S}^\varnothing_X (\sigma_2 \to \dots (\sigma_m \to (\psi_{i_1} \to \dots (\psi_{i_n} \to \varphi) \dots)) \dots).$$

Note that $\mathsf{K}_\varnothing \sigma_1 \in E$ by assumption (12). Thus, $E \vdash \mathsf{S}^\varnothing_X \sigma_1$ by the Public Knowledge axiom and the Modus Ponens inference rule. Hence,

$$E \vdash \mathsf{S}^\varnothing_X (\sigma_2 \to \dots (\sigma_m \to (\psi_1 \to \dots (\psi_n \to \varphi) \dots)) \dots).$$

Then, by repeating the previous step $m-1$ more times,

$$E \vdash \mathsf{S}_X^{\varnothing}(\psi_1 \to (\psi_2 \to \ldots (\psi_n \to \varphi)\ldots)).$$

Hence, by the Cooperation axiom and the Modus Ponens inference rule,

$$E \vdash \mathsf{S}_X^{D_1}\psi_1 \to \mathsf{S}_X^{\varnothing \cup D_1}(\psi_2 \to \ldots (\psi_n \to \varphi)\ldots).$$

Thus, by the assumption $\mathsf{S}_X^{D_1}\psi_1 \in E$ of the lemma and the Modus Ponens inference rule,

$$E \vdash \mathsf{S}_X^{D_1}(\psi_2 \to (\psi_3 \to \ldots (\psi_n \to \varphi)\ldots)).$$

Hence, again by the Cooperation axiom, the Modus Ponens inference rule, and the assumption of the lemma that sets D_1 and D_2 are disjoint,

$$E \vdash \mathsf{S}_X^{D_2}\psi_2 \to \mathsf{S}_X^{D_1 \cup D_2}(\psi_3 \to \ldots (\psi_n \to \varphi)\ldots)).$$

Then, by the assumption $\mathsf{S}_X^{D_2}\psi_2 \in E$ of the lemma and the Modus Ponens inference rule,

$$E \vdash \mathsf{S}_X^{D_1 \cup D_2}(\psi_3 \to \ldots (\psi_n \to \varphi)\ldots)).$$

By repeating the previous step $n-2$ more times,

$$E \vdash \mathsf{S}_X^{D_1 \cup D_2 \cup \cdots \cup D_n}\varphi.$$

Thus, $E \vdash \mathsf{S}_X^C\varphi$ by Lemma 6 and the assumption of the lemma that $\{D_i\}_{i\in I}$ is a family of subsets of C. Therefore, $\neg\mathsf{S}_X^C\varphi \notin E$ because set E is consistent, which contradicts the assumption $\neg\mathsf{S}_X^C\varphi \in E$ of the lemma. $\square$

The next lemma shows that any pair in harmony could be extended.

Lemma 16. *For any pair of sets of formulae (F, G) in harmony, any formula $\mathsf{S}_X^{\varnothing}\varphi \in \Phi$, either the pair $(F \cup \{\neg\mathsf{S}_X^{\varnothing}\varphi\}, G)$ or the pair $(F, G \cup \{\varphi\})$ is in harmony.*

Proof. Suppose that neither the pair $(F \cup \{\neg\mathsf{S}_X^{\varnothing}\varphi\}, G)$ nor the pair $(F, G \cup \{\varphi\})$ is in harmony. Thus, by Definition 3,

$$F, \neg\mathsf{S}_X^{\varnothing}\varphi \vdash \mathsf{S}_V^{\varnothing}\neg\bigwedge G_1, \tag{14}$$

$$F \vdash \mathsf{S}_V^{\varnothing}\neg\bigwedge G_2, \tag{15}$$

for some finite sets $G_1 \subseteq G$ and $G_2 \subseteq G \cup \{\varphi\}$. Then, there must exist a finite set of formulae $G' \subseteq G$ such that,

$$\vdash \bigwedge G' \to \bigwedge G_1,$$

$$\vdash \varphi \to \left(\bigwedge G' \to \bigwedge G_2\right).$$

Hence, by the laws of propositional reasoning,

$$\vdash \neg\bigwedge G_1 \to \neg\bigwedge G',$$

$$\vdash \neg\bigwedge G_2 \to \left(\varphi \to \neg\bigwedge G'\right).$$

Thus, by Lemma 5,

$$\vdash \mathsf{S}_V^{\varnothing}\left(\neg\bigwedge G_1 \to \neg\bigwedge G'\right),$$

$$\vdash \mathsf{S}_V^{\varnothing}\left(\neg\bigwedge G_2 \to \left(\varphi \to \neg\bigwedge G'\right)\right).$$

Then, by the Cooperation axiom and the Modus Ponens inference rule,

$$\vdash \mathsf{S}_V^{\varnothing}\left(\neg\bigwedge G_1\right) \to \mathsf{S}_V^{\varnothing}\left(\neg\bigwedge G'\right),$$

$$\vdash \mathsf{S}_V^{\varnothing}\left(\neg\bigwedge G_2\right) \to \mathsf{S}_V^{\varnothing}\left(\varphi \to \neg\bigwedge G'\right).$$

Hence, by assumptions (14) and (15) and the Modus Ponens inference rule,

$$F, \neg \mathsf{S}^{\varnothing}_{X}\varphi \vdash \mathsf{S}^{\varnothing}_{V}\left(\neg \bigwedge G'\right), \tag{16}$$

$$F \vdash \mathsf{S}^{\varnothing}_{V}\left(\varphi \to \neg \bigwedge G'\right). \tag{17}$$

Note that $\vdash V \rhd X$ by the Reflexivity axiom because $X \subseteq V$. Then, by the Strategic Monotonicity axiom and the Modus Ponens rule, $\vdash \mathsf{S}^{\varnothing}_{X}\varphi \to \mathsf{S}^{\varnothing}_{V}\varphi$. Hence, $\vdash \neg\mathsf{S}^{\varnothing}_{V}\varphi \to \neg\mathsf{S}^{\varnothing}_{X}\varphi$ by the law of contraposition. Thus, by the Modus Ponens inference rule and statements (16) and (17),

$$F, \neg \mathsf{S}^{\varnothing}_{V}\varphi \vdash \mathsf{S}^{\varnothing}_{V}\left(\neg \bigwedge G'\right),$$

$$F \vdash \mathsf{S}^{\varnothing}_{V}\left(\varphi \to \neg \bigwedge G'\right).$$

Then, by the Cooperation axiom and the Modus Ponens inference rule,

$$F, \neg \mathsf{S}^{\varnothing}_{V}\varphi \vdash \mathsf{S}^{\varnothing}_{V}\left(\neg \bigwedge G'\right),$$

$$F \vdash \mathsf{S}^{\varnothing}_{V}\varphi \to \mathsf{S}^{\varnothing}_{V}\left(\neg \bigwedge G'\right).$$

Hence, by Lemma 2,

$$F \vdash \neg\mathsf{S}^{\varnothing}_{V}\varphi \to \mathsf{S}^{\varnothing}_{V}\left(\neg \bigwedge G'\right),$$

$$F \vdash \mathsf{S}^{\varnothing}_{V}\varphi \to \mathsf{S}^{\varnothing}_{V}\left(\neg \bigwedge G'\right).$$

Then, by the laws of propositional reasoning,

$$F \vdash \mathsf{S}^{\varnothing}_{V}\left(\neg \bigwedge G'\right).$$

Therefore, by Definition 3 and the assumption $G' \subseteq G$, the pair (F, G) is not in harmony, which contradicts the assumption of the lemma that it is in harmony. $\square$

Definition 4. A pair of sets of formulae (F, G) is in complete harmony if this pair is in harmony and, for any formula $\mathsf{S}^{\varnothing}_{X}\varphi \in \Phi$, either $\neg\mathsf{S}^{\varnothing}_{X}\varphi \in F$ or $\varphi \in G$.

The next lemma follows from Lemma 16 and Definition 4.

Lemma 17. *For any pair (F, G) in harmony, there is a pair (F', G') in complete harmony such that $F \subseteq F'$, $G \subseteq G'$.* $\square$

6.2. Dataset closure

Another important idea used in our proof of the completeness is "dataset closure". Informally, for each set of formulae F and each dataset X, by closure X^*_F we denote the set of all data variables about which set F can prove that they are informed by set X. This notion goes back to Armstrong's original article on functional dependency. The "saturated" sets of database attributes there [19, Section 6] are essentially our closure sets X^*_F. Closures are used in Definition 7 of the next section to specify the labels of the edges of a tree.

Definition 5. $X^*_F = \{x \in V \mid X \rhd x \in F\}$ for any dataset $X \subseteq V$ and any maximal consistent set of formulae $F \subseteq \Phi$.

In other words, the closure X^*_F is the set of all data variables that, according to set F, are known to each actor who knows the values of all variables in dataset X. Intuitively, such set must include variables from the dataset X itself. We formally prove this in the lemma below.

Lemma 18. $X \subseteq X^*_F$.

Proof. Consider any data variable $x \in X$. Thus, $\vdash X \rhd x$ by the Reflexivity axiom. Hence, $X \rhd x \in F$ because F is a maximal consistent set of formulae. Therefore, $x \in X^*_F$ by Definition 5. $\square$

Note that $X \rhd x \in F$ for each data variable $x \in X^*_F$ by Definition 5. The next lemma shows that if the set of all data variables V is finite, then all such variables x could be brought together on the right-hand-side of $\rhd$ expression. The assumption that set V is finite is crucial. In Section 7, we show that if set V is infinite, then our logical system does not have strongly sound and strongly complete axiomatization. The proof of this incompleteness result, essentially, consists in showing that Lemma 19 does not hold for any infinite set V.

Lemma 19. *If set V is finite, then $F \vdash X \rhd X^*_F$.*

Proof. The set X^*_F is finite by Definition 5 and the assumption of the lemma that set V is finite. Let $X^*_F = \{x_1, \dots, x_n\}$. Note that $F \vdash X \rhd x_i$ for each $i \le n$ by Definition 5. We prove by induction that $F \vdash X \rhd x_1, \dots, x_k$ for each integer k such that $0 \le k \le n$.

Base Case: $F \vdash X \rhd \varnothing$ by the Reflexivity axiom.
Induction Step: Suppose that $F \vdash X \rhd x_1, \dots, x_k$. Then, by the Augmentation axiom and the Modus Ponens inference rule,

$$F \vdash X \cup \{x_{k+1}\} \rhd x_1, \dots, x_k, x_{k+1}. \tag{18}$$

Recall that $F \vdash X \rhd x_{k+1}$. Hence, $F \vdash X \rhd X \cup \{x_{k+1}\}$ by the Augmentation axiom and the Modus Ponens inference rule. Therefore,

$$F \vdash X \rhd x_1, \dots, x_k, x_{k+1}$$

by the Transitivity axiom, statement (18), and the Modus Ponens rule applied twice. □

6.3. Canonical game

In this section, for an arbitrary maximal consistent set of formulae $F_0 \subseteq \Phi$, we define a "canonical" game $G(F_0) = (W, \{\sim_x\}_{x\in V}, \Delta, M, \pi)$ used in the proof of the completeness.

In completeness proofs for modal logics, the states are usually defined as maximal consistent sets of formulae. Two such sets are called indistinguishable by an agent a if they contain the same K_a-formulae. This construction fails in the case of the distributed knowledge. Indeed, two maximal consistent sets containing the same K_a-formulae and the same K_b-formulae do not have to contain the same K_{ab} formulae. As a result, there might be two states indistinguishable to both agents that have different K_{ab} formulae. The same issue exists in our setting if a and b are interpreted as data variables. To solve this issue, we employ the "tree" construction that has been previously used to prove the completeness of modal logics that include distributed knowledge modality [9,17,27,28].

In this article, we modify the tree construction in a significant way by adding "clones" of each non-root node in the tree. For the proof of the completeness to work, the cardinality of the set of all clones of each node should be larger than the cardinality of the set of all actors $\mathcal{A}$. We further explain the intuition behind the clones construction below. To incorporate clones in our tree construction, we fix any set $\mathcal{B}$ of cardinality greater than the cardinality of set $\mathcal{A}$. Note that the proof in [17] is using a different technique that only works if the set $\mathcal{A}$ is finite.

Definition 6. W is a set of all sequences $F_0, X_1, b_1, F_1, \dots, X_n, b_n, F_n$ such that, $n \ge 0$ and

1. F_i is a maximal consistent set of formulae for each $i \ge 1$,
2. $X_i \subseteq V$ is a dataset for each $i \ge 1$,
3. $b_i \in \mathcal{B}$ for each $i \ge 1$,
4. $\{\varphi \mid \mathsf{K}_{X_i}\varphi \in F_{i-1}\} \subseteq F_i$ for each $i \ge 1$.

Item 4 of Definition 6 states that if $\mathsf{K}_{X_i}\varphi \in F_{i-1}$, then $\varphi \in F_i$. Alternatively, the same requirement could be stated in the equivalent form: $\mathsf{K}_{X_i}\varphi \in F_{i-1}$ iff $\mathsf{K}_{X_i}\varphi \in F_i$. We have chosen the form given in item 4 because it results in somewhat simpler proofs. In the next section, we prove a slightly modified alternative form of this requirement as Lemma 23.

For any states $w', w \in W$ such that $w' = F_0, X_1, b_1, \dots, X_{n-1}, b_{n-1}, F_{n-1}$ and $w = F_0, X_1, b_1, \dots, X_{n-1}, b_{n-1}, F_{n-1}, X_n, b_n, F_n$, we say that states w' and w are *adjacent*. The adjacency relation defines an undirected graph structure on set W. It is easy to see that this structure is a tree. Fig. 2 depicts a fragment of such tree. In this fragment, state F_0, X_2, b_2, F_2 is adjacent to state $F_0, X_2, b_2, F_2, X_5, b_5, F_5$. If $w = F_0, X_1, b_1, \dots, X_n, b_n, F_n$, then by $hd(w)$ we mean the maximal consistent set of formulae F_n.

For each node $w = F_0, X_1, b_1, \dots, X_{n-1}, b_{n-1}, F_{n-1}, X_n, b_n, F_n$, the set of nodes $\{F_0, X_1, b_1, \dots, X_{n-1}, b_{n-1}, F_{n-1}, X_n, \beta, F_n \mid \beta \in \mathcal{B}\}$ can be thought of as a set of "clones" of node w in the discussed above tree. For example, node $F_0, X_1, b_1, F_1, X_3, b_6, F_3$ is a clone of node $F_0, X_1, b_1, F_1, X_3, b_3, F_3$, see Fig. 2. The existence of such clones will be used in the proof of Lemma 30.

Definition 7. For any states

$$w' = F_0, X_1, b_1, \dots, X_{n-1}, b_{n-1}, F_{n-1} \in W$$
$$w = F_0, X_1, b_1, \dots, X_{n-1}, b_{n-1}, F_{n-1}, X_n, b_n, F_n \in W,$$

the edge (w', w) is *labeled* with a variable $x \in V$ if $x \in (X_n)^*_{F_{n-1}}$.

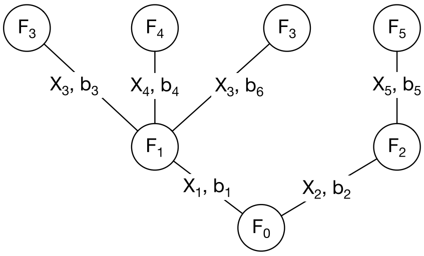

Fig. 2. Fragment of a canonical tree.

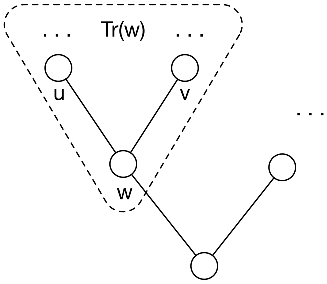

Fig. 3. States w, u, and v belong to $Tr(w)$.

As usual, by a simple path in a graph we mean a path without repeating vertices. We allow zero-length simple paths that start and end in the same vertex. Recall that there is one and only one simple path between any two vertices in a tree.

Definition 8. For any states $w, w' \in W$ and any variable $x \in V$, let $w \sim_x w'$ if every edge along the unique simple path between vertices w and w' is labeled with variable x.

Lemma 20. *Relation $\sim_x$ is an equivalence relation on set W for each variable $x \in V$.* $\square$

Definition 9. For any state $w \in W$, let $Tr(w)$ be the set of all states $u \in W$ such that either $w = u$ or sequence w is a prefix of sequence u.

We call set $Tr(w)$ the *subtree* of a state $w \in W$. Informally, $Tr(w)$ is a subtree of the canonical tree that starts at node w, see Fig. 3. The next two lemmas state an auxiliary property of subtrees used later in this article.

Lemma 21. *$Tr(w) \cap Tr(w') = \varnothing$ for any two states $w, w' \in W$ such that $w = F_0, X_1, b_1, \dots, X_{n-1}, b_{n-1}, F_{n-1}, X_n, b_n, F_n$, $w' = F_0, X_1, b_1, \dots, X_{n-1}, b_{n-1}, F_{n-1}, X'_n, b'_n, F'_n$, and $(X_n, b_n, F_n) \neq (X'_n, b'_n, F'_n)$.*

Lemma 22. *For any states $w, u \in W$ and any data variable $y \in V$, if $w = F_0, X_1, b_1, \dots, b_{n-1}, F_{n-1}, X_n, b_n, F_n \in W$, $u \notin Tr(w)$, and $w \sim_y u$, then $y \in (X_n)^*_{F_{n-1}}$.*

Proof. Let $w_0 = F_0, X_1, b_1, \dots, X_{n-1}, b_{n-1}, F_{n-1}$. Consider the simple path between vertices w and u. Note that the assumption $u \notin Tr(w)$ of the lemma implies that this path must contain edge (w_0, w). Hence, by the assumption $w \sim_y u$ of the lemma and Definition 8, edge (w_0, w) is labeled with variable y. Hence, $y \in (X_n)^*_{F_{n-1}}$ by Definition 7. $\square$

The definition of the canonical mechanism in the proofs of completeness of several extensions of Marc Pauly's Coalition Logic employ "voting" construction [29,6,30,10]. The construction consists in using the set of formulae Φ as the set of possible actions. If a coalition want to achieve outcome φ, then each member of the coalition can simply choose action φ. Of course, not all such requests are granted by the canonical mechanism. Note that in our case a coalition C might have a strategy to achieve φ informed by a dataset X and not have such a strategy informed by a smaller set $Y \subsetneq X$. To be able to model such situations, we require each agent to "sign" the vote with a "key". The key verifies that the agent has access to dataset X. The key potentially could be just the set of values of all variables in dataset X. However, recall that Definition 1 associates not a value, but an equivalence relation with each data variable. For this reason, instead of variable value, we use the corresponding equivalence class of relation $\sim_X$ as the key. To specify an equivalence class, it suffices to specify any element of this class. Thus, a key could be defined simply as a state. In this case, an action of each agent consists in specifying a formula $\varphi \in \Phi$ and a state $w \in W$ of the game. This is captured in the definition below. Similar constructions are used in [9,17].

Definition 10. $\Delta = \Phi \times W$.

If d is a pair (x, y), then by $pr_1(d)$ and $pr_2(d)$ we mean elements x and y respectively.

Definition 11. Mechanism M consists of all triples $(w, \delta, v) \in W \times \Delta^{\mathcal{A}} \times W$ such that for each formula $\mathsf{S}^C_X\varphi \in hd(w)$, if

1. $pr_1(\delta(a)) = \varphi$ for each actor $a \in C$ and
2. $w \sim_X pr_2(\delta(a))$ for each actor $a \in C$,

then $\varphi \in hd(v)$.

Informally, for each formula $\mathsf{S}^C_X\varphi$ that belongs to a set $hd(w)$, our canonical game provides a strategy informed by dataset X for coalition C to achieve φ. The strategy consists in each member of coalition C voting for formula φ and signing the vote with a state from the $\sim_X$-equivalence class of the state w. The fact that such a strategy succeeds is guaranteed by the canonical mechanism M definition above.

Definition 12. $\pi(p) = \{w \in W \mid p \in hd(w)\}$ for each propositional variable $p \in P$.

6.4. Canonical game properties

In Section 6.5, we use the defined above canonical game to finish the proof of the completeness of our logical system. In this section, we establish the properties of the canonical game needed for this proof. We split these properties into three groups.

6.4.1. Properties of modality K

The lemma below shows how formulae of the form $\mathsf{K}_Y\varphi$ can "move" between two adjacent nodes of the canonical tree.

Lemma 23. *If set V is finite, then $\mathsf{K}_Y\varphi \in F_{n-1}$ iff $\mathsf{K}_Y\varphi \in F_n$ for any formula $\varphi \in \Phi$, any $n \geq 1$, any state $F_0, X_1, b_1, F_1, X_2, b_2, \dots, F_{n-1}, X_n, b_n, F_n \in W$, and any dataset $Y \subseteq (X_n)^*_{F_{n-1}}$.*

Proof. ($\Rightarrow$): Suppose that $\mathsf{K}_Y\varphi \in F_{n-1}$. Thus, by Lemma 1 and the Modus Ponens inference rule

$$F_{n-1} \vdash \mathsf{K}_Y\mathsf{K}_Y\varphi. \tag{19}$$

Note that $F_{n-1} \vdash X_n \rhd (X_n)^*_{F_{n-1}}$ by Lemma 19. Also, by the assumption $Y \subseteq (X_n)^*_{F_{n-1}}$ of the lemma and the Reflexivity axiom, $\vdash (X_n)^*_{F_{n-1}} \rhd Y$. Hence, $F_{n-1} \vdash X_n \rhd Y$ by the Transitivity axiom and the Modus Ponens inference rules applied twice. Then, $F_{n-1} \vdash \mathsf{K}_{X_n}\mathsf{K}_Y\varphi$ by the Knowledge Monotonicity axiom and statement (19). Thus, $\mathsf{K}_{X_n}\mathsf{K}_Y\varphi \in F_{n-1}$ because F_{n-1} is a maximal consistent set. Therefore, $\mathsf{K}_Y\varphi \in F_n$ by item 4 of Definition 6.

($\Leftarrow$): Suppose that $\mathsf{K}_Y\varphi \notin F_{n-1}$. Thus, $\neg\mathsf{K}_Y\varphi \in F_{n-1}$ because F_{n-1} is a maximal consistent set of formulae. Hence, $F_{n-1} \vdash \mathsf{K}_Y\neg\mathsf{K}_Y\varphi$ by the Negative Introspection axiom and the Modus Ponens inference rule. Then, again because set F_{n-1} is maximal, $\mathsf{K}_Y\neg\mathsf{K}_Y\varphi \in F_{n-1}$. Thus, $\neg\mathsf{K}_Y\varphi \in F_n$ by item 4 of Definition 6. Therefore, $\mathsf{K}_Y\varphi \notin F_n$ because set F_n is consistent. $\square$

The next lemma shows that formulae of the form $\mathsf{K}_X\varphi$ can "move" between any two $\sim_X$-equivalent nodes of the canonical tree.

Lemma 24. *If set V is finite and $w \sim_X w'$, then $\mathsf{K}_X\varphi \in hd(w)$ iff $\mathsf{K}_X\varphi \in hd(w')$.*

Proof. If $w \sim_X w'$, then, by Definition 8, each edge along the unique simple path between nodes w and w' is labeled with all variables in set X. We prove the lemma by induction on the length of the unique path between nodes w and w'. In the base case, $w = w'$. Thus, $\mathsf{K}_X\varphi \in hd(w)$ iff $\mathsf{K}_X\varphi \in hd(w')$. The induction step follows from Lemma 23. $\square$

As usual in the modal logic, at the core of our proof of completeness is an induction (or "truth") lemma. In our case, this is Lemma 31. The next lemma will be used in the induction step of the proof of Lemma 31 in the case when the formula has the form $\mathsf{K}_Y\varphi$.

Lemma 25. *For any state $w \in W$ and any formula $\neg\mathsf{K}_Y\varphi \in hd(w)$, there is a state $w' \in W$ such that $w \sim_Y w'$ and $\neg\varphi \in hd(w')$.*

Proof. First, we show that the following set of formulae is consistent:

$$G = \{\neg\varphi\} \cup \{\psi \mid \mathsf{K}_Y \psi \in hd(w)\}. \tag{20}$$

Suppose the opposite. Then, there are formulae

$$\mathsf{K}_Y \psi_1, \dots, \mathsf{K}_Y \psi_n \in hd(w) \tag{21}$$

such that $\psi_1, \dots, \psi_n \vdash \varphi$. Thus, $\mathsf{K}_Y \psi_1, \dots, \mathsf{K}_Y \psi_n \vdash \mathsf{K}_Y \varphi$ by Lemma 3. It follows by assumption (21) that $hd(w) \vdash \mathsf{K}_Y \varphi$. Thus, $\neg\mathsf{K}_Y \varphi \notin hd(w)$ because set $hd(w)$ is consistent, which contradicts the assumption $\neg\mathsf{K}_Y \varphi \in hd(w)$ of the lemma. Therefore, set G is consistent.

Second, let w be the sequence $F_0, X_1, b_1, F_1, \dots, X_n, b_n, F_n$ and G' be any extension of set G to a maximal consistent set of formulae. Recall that the cardinality of set $\mathcal{B}$ is greater than the cardinality of set $\mathcal{A}$. Thus, there is at least one element $b \in \mathcal{B}$. Define w' to be the sequence $F_0, X_1, b_1, F_1, \dots, X_n, b_n, F_n, Y, b, G'$. Note that (i) $w' \in W$ by Definition 6 and equation (20) and (ii) $\neg\varphi \in G \subseteq G' = hd(w')$ by equation (20) and the choices of set G' and sequence w'. Finally, to prove $w \sim_Y w'$, note that $Y \subseteq Y^*_{F_n}$ by Lemma 18. Thus, by Definition 7, the edge between vertices w and w' is labeled with each variable in set Y. Therefore, $w \sim_Y w'$ by Definition 8. $\square$

6.4.2. Properties of expression $X \rhd Y$

The two lemmas in this section will be used in the base case of the proof by induction of Lemma 31 when the formula has the form $X \rhd Y$.

Lemma 26. *If set V is finite, then for any state $w \in W$ and any formula $\neg(X \rhd Y) \in hd(w)$, there is a state $w' \in W$ such that $w \sim_X w'$, and $w \nsim_Y w'$.*

Proof. Let state w be sequence $F_0, X_1, b_1, \dots, X_{n-1}, b_{n-1}, F_{n-1}, X_n, b_n, F_n$. Note that the cardinality of set $\mathcal{B}$ is greater than the cardinality of set $\mathcal{A}$. Thus, there is at least one element $b \in \mathcal{B}$. Consider sequence

$$w' = F_0, X_1, b_1, \dots, X_{n-1}, b_{n-1}, F_{n-1}, X_n, b_n, F_n, X, b, F_n.$$

To prove that $w' \in W$, consider any formula $\mathsf{K}_X \varphi \in F_n$. By item 4 of Definition 6, it suffices to show that $\varphi \in F_n$. Indeed, assumption $\mathsf{K}_X \varphi \in F_n$ implies $F_n \vdash \varphi$ by the Truth axiom and the Modus Ponens inference rule. Therefore, $\varphi \in F_n$ because set F_n is maximal.

To prove $w \sim_X w'$, note that $X \subseteq X^*_{F_n}$ by Lemma 18. Thus, by Definition 7, the edge between vertices w and w' is labeled with each data variable in set X. Therefore, $w \sim_X w'$ by Definition 8.

Finally, we show that $w \nsim_Y w'$. By Definition 8, it suffices to prove that the simple path between vertices w and w' is *not* labeled by at least one variable from set Y. Then, by Definition 7, it suffices to show that $Y \nsubseteq X^*_{F_n}$. Suppose the opposite. Thus, $\vdash X^*_{F_n} \rhd Y$ by the Reflexivity axiom. Note that $F_n \vdash X \rhd X^*_{F_n}$ by Lemma 19. Hence, $F_n \vdash X \rhd Y$ by the Transitivity axiom and the Modus Ponens inference rule applied twice. Thus, $\neg(X \rhd Y) \notin F_n = hd(w)$ because set F_n is consistent, which contradicts the assumption $\neg(X \rhd Y) \in hd(w)$ of the lemma. $\square$

Lemma 27. *For any states $w, w' \in W$, if set V is finite, $X \rhd Y \in hd(w)$, and $w \sim_X w'$, then $w \sim_Y w'$.*

Proof. We prove the lemma by induction on the length of the simple path between vertices w and w'. If $w = w'$, then, vacuously, each edge along the simple path between vertices w and w' is labeled with each data variable. Hence, $w \sim_Y w'$ by Definition 8.

Suppose that $w \neq w'$. Consider the unique simple path between vertices w and w'. By the assumption $w \sim_X w'$ of the lemma and Definition 8, each edge along this path is labeled with each data variable in set X. Because $w \neq w'$, there must exist a vertex $u \in W$ on the unique simple path between w and w' such that vertices u and w' are adjacent. Note that the unique simple path between vertices w' and u is a part of the unique simple path between vertices w and w'. Thus, each edge along the simple path between vertices w and u is labeled with each data variable in set X. Hence, by Definition 8,

$$w \sim_X u. \tag{22}$$

Claim 1. *The edge between vertices u and w' is labeled with each data variable in set Y.*

Proof of Claim. We consider the following two cases separately, see Fig. 4:

Case I: $u = F_0, X_1, b_1, F_1, \dots, F_{n-1}$ and $w' = F_0, X_1, b_1, F_1, \dots, X_n, b_n, F_n$. Consider any data variable $y \in Y$. By Definition 7, it suffices to show that $y \in (X_n)^*_{hd(u)}$. Note that $X \rhd Y \in hd(w)$ by the assumption of the lemma. Thus, by the Introspection of Dependency axiom and the Modus Ponens inference rule, $hd(w) \vdash \mathsf{K}_X(X \rhd Y)$. Hence, $\mathsf{K}_X(X \rhd Y) \in hd(w)$

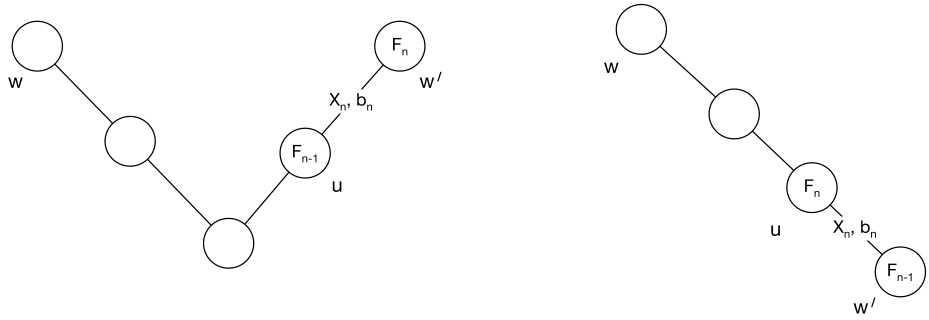

Fig. 4. Case I (left) and Case II (right).

because set $hd(w)$ is maximal. Then, $\mathsf{K}_X(X \rhd Y) \in hd(u)$ by Lemma 24 and statement (22). Thus, by the Truth axiom and the Modus Ponens inference rule,

$$hd(u) \vdash X \rhd Y.$$

Note that $\vdash Y \rhd \{y\}$ by the reflexivity axiom. Hence, by the Transitivity axiom and the Modus Ponens inference rule applied twice,

$$hd(u) \vdash X \rhd y. \tag{23}$$

Recall that u is a vertex on the simple path connecting vertices w and w' and all edges along this path are labeled X. Hence, $X \subseteq (X_n)^*_{hd(u)}$ by Definition 7. Then, $\vdash (X_n)^*_{hd(u)} \rhd X$ by the Reflexivity axiom and the Modus Ponens inference rule. Thus, $\vdash (X_n)^*_{hd(u)} \rhd y$ by the Transitivity axiom and statement (23). Hence, $\vdash X_n \rhd y$ by the Transitivity axiom and Lemma 19. Therefore, $y \in (X_n)^*_{hd(u)}$ by Definition 5.

Case II: $w' = F_0, X_1, b_1, F_1, \dots, F_{n-1}$ and $u = F_0, X_1, b_1, F_1, \dots, X_n, b_n, F_n$. This case is similar to the previous one, except that it uses the set $hd(w')$ instead of the set $hd(u)$ everywhere in the proof. $\square$

To finish the proof of the lemma, note that the simple path between vertices w and u is shorter than the simple path between vertices w and w'. Hence, $w \sim_Y u$, by the induction hypothesis. Also, $u \sim_Y w'$ by Claim 1 and Definition 8. Therefore, $w \sim_Y w'$ because relation $\sim_Y$ is transitive. $\square$

6.4.3. *Properties of modality* S

This section contains the last group of the canonical game properties. Two of them, Lemma 28 and Lemma 30, are used in the induction step of the proof of Lemma 31 when the formula has the form $\mathsf{S}^C_X\varphi$. They are used in the parts of the proof corresponding to different directions of "if and only if" in the statement of Lemma 31. Lemma 29 is an auxiliary statement used in the proof of Lemma 30.

Lemma 28. *For any state $w \in W$ and any formula $\mathsf{S}^C_X\varphi \in hd(w)$ there is an action profile $s \in \Delta^C$ such that for all states $w', v \in W$ and each complete action profile $\delta \in \Delta^{\mathcal{A}}$ if $w \sim_X w'$, $s =_C \delta$, and $(w', \delta, v) \in M$, then $\varphi \in hd(v)$.*

Proof. Let action profile $s \in \Delta^C$ be such that

$$s(a) = (\varphi, w) \tag{24}$$

for each actor $a \in C$. Consider any states $w', v \in W$ and any complete action profile $\delta \in \Delta^{\mathcal{A}}$ such that

$$w \sim_X w', \quad s =_C \delta, \quad \text{and} \quad (w', \delta, v) \in M. \tag{25}$$

It suffices to show that $\varphi \in hd(v)$.

The assumption $\mathsf{S}^C_X\varphi \in hd(w)$ of the lemma implies $hd(w) \vdash \mathsf{K}_X\mathsf{S}^C_X\varphi$ by the Strategic Introspection axiom and the Modus Ponens inference rule. Thus, $\mathsf{K}_X\mathsf{S}^C_X\varphi \in hd(w)$ because set $hd(w)$ is maximal. Hence, $K_X\mathsf{S}^C_X\varphi \in hd(w')$ by Lemma 24 and the part $w \sim_X w'$ of assumption (25). Then, by the Truth axiom and the Modus Ponens inference rule, $hd(w') \vdash \mathsf{S}^C_X\varphi$. Thus, because set $hd(w')$ is maximal, $\mathsf{S}^C_X\varphi \in hd(w')$. Therefore, $\varphi \in hd(v)$ by Definition 11, assumption (24) and the parts $s =_C \delta$ and $(w', \delta, v) \in M$ of assumption (25). $\square$

Before proving the next lemma about the canonical game, we state a property of sets that will be used in the proof of that lemma. Informally, this property can be viewed as an infinite variation of the pigeonhole principle: if the cardinality of the set of pigeons is less than the cardinality of the set of holes, then there is at least one hole that contains no pigeons. In the lemma below, A is the set of pigeons and each set T_b is a hole.

Lemma 29. *For any set A and any pairwise disjoint family of sets $\{T_b\}_{b\in B}$, if the cardinality of set A is less than the cardinality of set B, then there is an index $b \in B$ such that $A \cap T_b = \varnothing$.* $\square$

We are now ready to state the last and the most non-trivial property of the canonical game. The proof of this lemma brings together the harmony construction and the clone nodes construction, both of which we discussed earlier.

Lemma 30. *For any state $w \in W$, any formula $\neg \mathsf{S}^C_X\varphi \in hd(w)$, and any action profile $s \in \Delta^C$, if set V is finite, then there are states $w', v \in W$ and a complete action profile $\delta \in \Delta^{\mathcal{A}}$ such that $w \sim_X w'$, $s =_C \delta$, $(w', \delta, v) \in M$, and $\varphi \notin hd(v)$.*

Proof. Consider any state $w \in W$, any formula $\neg \mathsf{S}^C_X\varphi \in hd(w)$, and any action profile $s \in \Delta^C$. For each formula $\psi \in \Phi$, define coalition

$$D_\psi = \{a \in C \mid pr_1(s(a)) = \psi\}. \tag{26}$$

Claim 2. *$\{D_\psi\}_{\psi\in\Phi}$ is a family of pairwise disjoint subsets of C.*

Proof of Claim. By equation (26), set D_ψ is a subset of C for each formula $\psi \in \Phi$. To show that sets $\{D_\psi\}_{\psi\in\Phi}$ are disjoint, suppose that there is an actor $a \in C$ such that $a \in D_{\psi_1} \cap D_{\psi_2}$ for some formulae $\psi_1, \psi_2 \in \Phi$. Therefore, $\psi_1 = pr_1(s(a)) = \psi_2$ by equation (26). $\square$

Consider the set of formulae

$$\Psi = \{\psi \in \Phi \mid \mathsf{S}^{D_\psi}_X \psi \in hd(w)\}. \tag{27}$$

Next, we are going to apply Lemma 15. Note that $hd(w)$ is a consistent subset of Φ and that $\neg \mathsf{S}^C_X\varphi \in hd(w)$ by the assumption of the lemma. Also, observe that $\{D_\psi\}_{\psi\in\Psi} \subseteq \{D_\psi\}_{\psi\in\Phi}$ is a family of pairwise disjoint subsets of C by Claim 2. Finally, note that $\{\mathsf{S}^{D_\psi}_X \psi\}_{\psi\in\Psi} \subseteq hd(w)$ by equation (27). Therefore, by Lemma 15, the pair (F, G) is in harmony, where

$$F = \{\chi \mid \mathsf{K}_X\chi \in hd(w)\}, \tag{28}$$

$$G = \{\neg\varphi\} \cup \Psi \cup \{\sigma \mid \mathsf{K}_\varnothing \sigma \in hd(w)\}. \tag{29}$$

Then, by Lemma 17, there is a pair (F', G') in complete harmony such that $F \subseteq F'$ and $G \subseteq G'$. Hence, by Definition 4, pair (F', G') in harmony. Thus, by Lemma 14, sets F' and G' are consistent. Then, by Lemma 4, sets F' and G' can be further extended to maximal consistent sets of formulae F'' and G'', respectively, such that

$$F \subseteq F' \subseteq F'' \quad \text{and} \quad G \subseteq G' \subseteq G''. \tag{30}$$

Let state $w \in W$ be the sequence $F_0, X_1, b_1, \dots, b_{n-1}, F_{n-1}, X_n, b_n, F_n$. For each $b \in \mathcal{B}$, define sequence,

$$w_b = F_0, X_1, b_1, \dots, X_{n-1}, b_{n-1}, F_{n-1}, X_n, b_n, F_n, X, b, F''. \tag{31}$$

Recall that we previously referred to such nodes as "clones".

Claim 3. *$w_b \in W$ and $w \sim_{X^*_{hd(w)}} w_b$ for each $b \in \mathcal{B}$.*

Proof of Claim. By Definition 6 and the assumption $w \in W$ of the lemma, to show that $w_b \in W$, it suffices to show that $\{\chi \mid \mathsf{K}_X\chi \in F_n\} \subseteq F''$. The latter is true by equation (28), part $F \subseteq F''$ of statement (30), and because $hd(w) = F_n$. Finally, note that $w \sim_{X^*_{hd(w)}} w_b$ for each $b \in \mathcal{B}$ by Definition 8 and equation (31). $\square$

To continue the proof of the lemma, recall that set $\mathcal{B}$ has been assumed to have larger cardinality than set $\mathcal{A}$. Thus, because $C \subseteq \mathcal{A}$,

$$|\{pr_2(s(a)) \mid a \in C\}| \le |C| \le |\mathcal{A}| < |\mathcal{B}| = |\{Tr(w_b) \mid b \in \mathcal{B}\}|.$$

Note that sets $\{Tr(w_b) \mid b \in \mathcal{B}\}$ are pairwise disjoint by Lemma 21. Hence, by Lemma 29, there must exist at least one element $\beta \in \mathcal{B}$ such that

$$\{pr_2(s(a)) \mid a \in C\} \cap Tr(w_\beta) = \varnothing. \tag{32}$$

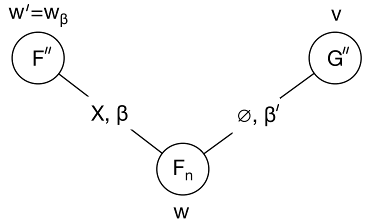

Fig. 5. Towards the proof of Lemma 30.

Let

$$w' = w_\beta. \tag{33}$$

Choose an arbitrary element $\beta' \in \mathcal{B}$ and define state v as follows, see Fig. 5,

$$v = F_0, X_1, b_1, \ldots, X_{n-1}, b_{n-1}, F_{n-1}, X_n, b_n, F_n, \varnothing, \beta', G''. \tag{34}$$

Claim 4. $v \in W$.

Proof of Claim. By Definition 6 and the assumption $w \in W$ of the lemma, it suffices to show that $\{\sigma \mid \mathsf{K}_\varnothing \sigma \in F_n\} \subseteq G''$. The latter is true by equation (29), part $G \subseteq G''$ of statement (30), and because $hd(w) = F_n$. $\square$

Recall that we started the proof of the lemma by fixing a state $w \in W$ and an action profile $s \in \Delta^C$ of coalition C. Define complete action profile $\delta \in \Delta^{\mathcal{A}}$ as follows:

$$\delta(a) = \begin{cases} s(a), & \text{if } a \in C, \\ (\top, w), & \text{otherwise.} \end{cases} \tag{35}$$

Note

$$s =_C \delta. \tag{36}$$

Claim 5. $(w', \delta, v) \in M$.

Proof of Claim. Consider any formula

$$\mathsf{S}^D_Y \psi \in hd(w') \tag{37}$$

such that

$$\forall a \in D\, (pr_1(\delta(a)) = \psi) \tag{38}$$

and

$$\forall a \in D\, (w' \sim_Y pr_2(\delta(a))). \tag{39}$$

By Definition 11, it suffices to show that $\psi \in hd(v)$. We consider the following three cases separately:

Case I: $D = \varnothing$. Assumption (37) implies that $\mathsf{S}^\varnothing_Y \psi \in hd(w_\beta)$ by equation (33). Thus, $\neg\mathsf{S}^\varnothing_Y \psi \notin hd(w_\beta)$ because set $hd(w_\beta)$ is consistent. Hence, $\neg\mathsf{S}^\varnothing_Y \psi \notin F''$ by equation (31). Then, $\neg\mathsf{S}^\varnothing_Y \psi \notin F'$ by part $F' \subseteq F''$ of statement (30). Thus, $\psi \in G'$ by the assumption that the pair (F', G') is in complete harmony and Definition 4. Hence, $\psi \in G''$ by part $G' \subseteq G''$ of statement (30). Therefore, $\psi \in hd(v)$ by equation (34).

Case II: there exists an actor $a \in D \setminus C$. Then, $\delta(a) = (\top, w)$ by equation (35). Hence, $pr_1(\delta(a)) = \top$. Thus, $\psi = \top$ by equation (38). Therefore, $\psi \in hd(v)$ because $hd(v)$ is a maximal consistent set of formulae.

Case III: $\varnothing \neq D \subseteq C$. We further split this case into two subcases:

Subcase IIIa: $Y \subseteq X^*_{hd(w)}$. Note that $pr_1(s(a)) = pr_1(\delta(a))$ for each actor $a \in D$ by equation (35) and the assumption $D \subseteq C$ of Case III. Thus, $pr_1(s(a)) = \psi$ for each actor $a \in D$ by statement (38). Hence, $D \subseteq D_\psi$ by equation (26). Then, by assumption (37), Lemma 6, and the Modus Ponens inference rule,

$$hd(w') \vdash \mathsf{S}^{D_\psi}_Y \psi. \tag{40}$$

At the same time, the assumption $Y \subseteq X^*_{hd(w)}$ of Subcase IIIa implies that $\vdash X^*_{hd(w)} \rhd Y$ by the Reflexivity axiom. Hence, $hd(w') \vdash \mathsf{S}^{D_\psi}_{X^*_{hd(w)}} \psi$ by the Strategic Monotonicity axiom, statement (40) and the Modus Ponens inference rule. Thus, $hd(w') \vdash \mathsf{K}_{X^*_{hd(w)}} \mathsf{S}^{D_\psi}_{X^*_{hd(w)}} \psi$ by the Strategic Introspection axiom and the Modus Ponens inference rule. Hence,

$$\mathsf{K}_{X^*_{hd(w)}} \mathsf{S}^{D_\psi}_{X^*_{hd(w)}} \psi \in hd(w'), \tag{41}$$

because set $hd(w')$ is maximal. Note that $w \sim_{X^*_{hd(w)}} w_\beta$ by Claim 3. Then, $w \sim_{X^*_{hd(w)}} w'$ by equation (33). Thus,

$$\mathsf{K}_{X^*_{hd(w)}} \mathsf{S}^{D_\psi}_{X^*_{hd(w)}} \psi \in hd(w)$$

by statement (41) and Lemma 24. Hence, by the Truth axiom and the Modus Ponens inference rule,

$$hd(w) \vdash \mathsf{S}^{D_\psi}_{X^*_{hd(w)}} \psi.$$

Note that $hd(w) \vdash X \rhd X^*_{hd(w)}$ by Lemma 19. Then, $hd(w) \vdash \mathsf{S}^{D_\psi}_X \psi$ by the Strategic Monotonicity axiom and the Modus Ponens inference rule. Thus, $\mathsf{S}^{D_\psi}_X \psi \in hd(w)$ because set $hd(w)$ is maximal. Hence, $\psi \in \Psi$ by equation (27). Then, $\psi \in G$ by equation (29). Thus, $\psi \in G''$ by equation (30). Therefore, $\psi \in hd(v)$ by equation (34).

Subcase IIIb: there is a data variable $y \in Y \setminus X^*_{hd(w)}$. The assumption $\varnothing \neq D$ of Case III implies that there is an actor $a \in D$. Then, by the assumption $D \subseteq C$ of Case III,

$$a \in C. \tag{42}$$

Also, the assumption $a \in D$ implies $w' \sim_y pr_2(\delta(a))$ by statement (39) and the assumption $y \in Y$ of Subcase IIIb. Thus, $w' \sim_y pr_2(s(a))$ by equation (35) and statement (42). Hence, by equation (33),

$$w_\beta \sim_y pr_2(s(a)). \tag{43}$$

At the same time, $pr_2(s(a)) \notin Tr(w_\beta)$ by equation (32) and statement (42). Therefore, $y \in X^*_{hd(w)}$ by Lemma 22, statement (43), and equation (31), which contradicts the assumption $y \in Y \setminus X^*_{hd(w)}$ of Subcase IIIb.

This concludes the proof of the claim $(w', \delta, v) \in M$. $\square$

Thus, towards the proof of the lemma, we constructed state $w' \in W$ (equation (33) and Claim 3), state $v \in W$ (Claim 4) and a complete action profile $\delta \in \Delta^{\mathcal{A}}$ such that $w \sim_X w'$ (equations (31) and (33)), $s =_C \delta$ (equation (36)), and $(w', \delta, v) \in M$ (Claim 5). To finish the proof of the lemma, note that $\neg\varphi \in G \subseteq G'' = hd(v)$ by equation (29), equation (30), and equation (34) respectively. $\square$

6.5. Final steps

In this section, we finish the proof of the completeness of our logical system. First, we prove the induction (or "truth") lemma for the canonical game using the properties of the game that we established in Section 6.4. After that, we state and prove the strong completeness of our system.

Lemma 31. *If set V is finite, then $w \Vdash \varphi$ iff $\varphi \in hd(w)$.*

Proof. We prove the statement by induction on the complexity of formula φ.

Suppose that formula φ is a propositional variable p. Note that $w \Vdash p$ iff $w \in \pi(p)$ by item 1 of Definition 2. At the same time, $w \in \pi(p)$ iff $p \in hd(w)$ by Definition 12. Therefore, $w \Vdash p$ iff $p \in hd(w)$.

Suppose that formula φ has the form $X \rhd Y$.

$(\Rightarrow)$: Assume that $X \rhd Y \notin hd(w)$. Thus, $\neg(X \rhd Y) \in hd(w)$ because set $hd(w)$ is maximal. Hence, by Lemma 26, there is a state $w' \in W$ such that $w \sim_X w'$, and $w \nsim_Y w'$. Therefore, $w \nVdash X \rhd Y$ by item 2 of Definition 2.

$(\Leftarrow)$: Assume that $X \rhd Y \in hd(w)$. Recall that set V is finite by the assumption of the lemma. Then, by Lemma 27, for any state $w' \in W$, if $w \sim_X w'$, then $w \sim_Y w'$. Therefore, $w \Vdash X \rhd Y$ by item 2 of Definition 2.

If formula φ is a negation or an implication, then the statement of the lemma follows from the maximality and the consistency of the set $hd(w)$, items 3 and 4 of Definition 2, and the induction hypothesis in the standard way.

Suppose that formula φ has the form $\mathsf{K}_X\psi$.

$(\Rightarrow)$: Assume that $\mathsf{K}_X\psi \notin hd(w)$. Then, $\neg\mathsf{K}_X\psi \in hd(w)$ because $hd(w)$ is a maximal consistent set of formulae. Thus, by Lemma 25, there is a state $w' \in W$ such that $w \sim_X w'$ and $\neg\psi \in hd(w')$. Hence, $\psi \notin hd(w')$ because set $hd(w')$ is consistent. Then, $w' \nVdash \psi$ by the induction hypothesis. Therefore, $w \nVdash \mathsf{K}_X\psi$ by item 5 of Definition 2 and the assumption $w \sim_X w'$.

($\Leftarrow$): Assume that $\mathsf{K}_X\psi \in hd(w)$. Consider any state w' such that $w \sim_X w'$. By item 5 of Definition 2, it suffices to show that $w' \Vdash \psi$. Indeed, the assumption $\mathsf{K}_X\psi \in hd(w)$ implies $\mathsf{K}_X\psi \in hd(w')$ by Lemma 24 and the assumption $w \sim_X w'$. Then, $hd(w') \vdash \psi$ by the Truth axiom and the Modus Ponens inference rule. Thus, $\psi \in hd(w')$ because the set $hd(w')$ is maximal. Then, $w' \Vdash \psi$ by the induction hypothesis.

Finally, suppose that formula φ has the form $\mathsf{S}^C_X\psi$.

($\Rightarrow$): Assume that $\mathsf{S}^C_X\psi \notin hd(w)$. Thus, $\neg\mathsf{S}^C_X\psi \in hd(w)$ because set $hd(w)$ is maximal. Hence, by Lemma 30, for any action profile $s \in \Delta^C$, there are states $w', v \in W$ and a complete action profile $\delta \in \Delta^{\mathcal{A}}$ such that $w \sim_X w'$, $s =_C \delta$, $(w', \delta, v) \in M$, and $\psi \notin hd(v)$. Then, by the induction hypothesis, for any action profile $s \in \Delta^C$, there are states $w', v \in W$ and a complete action profile $\delta \in \Delta^{\mathcal{A}}$ such that $w \sim_X w'$, $s =_C \delta$, $(w', \delta, v) \in M$, and $v \nVdash \psi$. Therefore, $w \nVdash \mathsf{S}^C_X\psi$ by item 6 of Definition 2.

($\Leftarrow$): Assume that $\mathsf{S}^C_X\psi \in hd(w)$. Thus, by Lemma 28, there is an action profile $s \in \Delta^C$ such that for all states $w', v \in W$ and each complete action profile $\delta \in \Delta^{\mathcal{A}}$ if $w \sim_X w'$, $s =_C \delta$, and $(w', \delta, v) \in M$, then $\psi \in hd(v)$. Hence, by the induction hypothesis, there is an action profile $s \in \Delta^C$ such that for all states $w', v \in W$ and each complete action profile $\delta \in \Delta^{\mathcal{A}}$ if $w \sim_X w'$, $s =_C \delta$, and $(w', \delta, v) \in M$, then $v \Vdash \psi$. Therefore, $w \Vdash \mathsf{S}^C_X\psi$ by item 6 of Definition 2. $\square$

Theorem 2 *(Strong completeness). For any set of formulae $F \subseteq \Phi$ and any formula $\varphi \in \Phi$, if the set V of data variables is finite and $F \nvdash \varphi$, then there is a state w of a game such that $w \Vdash f$ for each formula $f \in F$ and $w \nVdash \varphi$.*

Proof. The assumption $F \nvdash \varphi$ implies that set $\{\neg\varphi\} \cup F$ is consistent. Let F_0 be any maximal consistent extension of this set. Consider the canonical game $G(F_0)$. Let w be the single-element sequence whose only element is set F_0. By Definition 6, sequence w is a state of game $G(F_0)$.

Consider any formula $f \in F$. We show that $w \Vdash f$. Indeed, $f \in \{\neg\varphi\} \cup F \subseteq F_0 = hd(w)$ for each formula $f \in F$ by the choice of set F_0 and sequence w. Therefore $w \Vdash f$ by Lemma 31.

Finally, we prove that $w \nVdash \varphi$. Indeed, $\neg\varphi \in \{\neg\varphi\} \cup F \subseteq F_0 = hd(w)$ by the choice of set F_0 and sequence w. Then, $\varphi \notin hd(w)$ because set $hd(w)$ is consistent. Therefore, $w \nVdash \varphi$ by Lemma 31. $\square$

7. Incompleteness

In the previous section, we proved that our logical system is strongly complete under the assumption that the set V of all data variables is finite. In this section, we show that if set V is infinite, then our system is not strongly complete. In fact, we show a more general result that *any* strongly sound logical system in language Φ is not strongly complete. We start the proof with the definitions of the strong soundness and the strong completeness.

Definition 13. Logical system $\mathcal{L}$ is *strongly sound* when for each set of formulae $F \subseteq \Phi$, each formula $\varphi \in \Phi$, and each state w of an arbitrary game, if $w \Vdash f$ for each formula $f \in F$ and $F \vdash_{\mathcal{L}} \varphi$, then $w \Vdash \varphi$.

Definition 14. Logical system $\mathcal{L}$ is *strongly complete*, when for each set of formulae $F \subseteq \Phi$ and each formula $\varphi \in \Phi$, if $w \Vdash \varphi$ for each state w of an arbitrary game such that $w \Vdash f$ for each formula $f \in F$, then $F \vdash_{\mathcal{L}} \varphi$.

The next theorem is the main result of this section.

Theorem 3. *If the set V of data variables is infinite, then any strongly sound logical system $\mathcal{L}$ is not strongly complete.*

Proof. Let $x_1, x_2, \dots$ be an infinite countable set of data variables from the infinite set V. We start the proof by establishing the following claim.

Claim 6. *For each state w of an arbitrary game, if $w \Vdash \varnothing \rhd x_n$ for each integer $n \geq 1$, then $w \Vdash \varnothing \rhd \{x_1, x_2, \dots\}$.*

Proof of Claim. Consider any state u of the game. By item 2 of Definition 2, it suffices to show that $w \sim_{\{x_1,x_2,x_3,\dots\}} u$. Assume the opposite. Thus, there exists $n \geq 1$ such that $w \nsim_{x_n} u$. Then, $w \nVdash \varnothing \rhd x_n$ item 2 of Definition 2. $\square$

Suppose that system $\mathcal{L}$ is strongly complete. Thus, by Definition 14, the above claim implies that

$$\varnothing \rhd x_1, \varnothing \rhd x_2, \dots \vdash_{\mathcal{L}} \varnothing \rhd \{x_1, x_2, \dots\}.$$

Thus, because any proof is using only finitely many assumptions, there must exist an integer $n \geq 0$ such that

$$\varnothing \rhd x_1, \varnothing \rhd x_2, \dots, \varnothing \rhd x_n \vdash_{\mathcal{L}} \varnothing \rhd \{x_1, x_2, \dots\}. \tag{44}$$

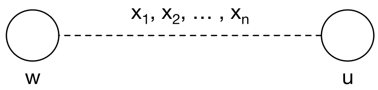

Fig. 6. An epistemic model.

Let us now consider the game depicted in Fig. 6. It has two possible states, w and u, indistinguishable by data variables $x_1,\dots,x_n$ and distinguishable by all other variables, including variable x_{n+1}.

Note that $w \Vdash \varnothing \rhd x_i$ for each $i \le n$ by item 2 of Definition 2 and the assumption $w \sim_{x_i} u$. Also, $w \nVdash \varnothing \rhd \{x_1, x_2, \dots\}$ by the same item 2 of Definition 2 and the assumption $w \nsim_{x_{n+1}} u$. Therefore, logical system $\mathcal{L}$ is not strongly sound by Definition 13 and statement (44). $\square$

It is interesting to point out that if operator $\rhd$ is removed from the language, then we can prove strong completeness even if set V is infinite. On the other hand, because our proof of Theorem 3 is only using this operator (and no modalities K and S), even a pure logical system for $\rhd$ alone is incomplete for infinite set of variables V.

8. Completeness for data-finite formulae

As we have seen in Theorem 2, our logical system is strongly complete if the set of data variables V is finite. We have shown in Theorem 3 that our logical system, just like any other strongly sound system, is not strongly complete if set V is infinite. It is crucial for the proof of Theorem 3 that we allow dataset Y in expression $X \rhd Y$ to be infinite. In this section, we show that if we restrict the language Φ in such a way that datasets in all formulae are finite, then our logical system is complete (but not necessarily strongly complete).

So far, we have been assuming a fixed set of data variables V. In this section, it will be convenient to consider versions of our logical system for different such sets. By $\Phi^{\mathbb{V}}$ we denote the language specified by the grammar given in Section 3 for an arbitrary set of data variables $\mathbb{V}$. By $\vdash^{\mathbb{V}}$ we denote the derivability in our logical system in language $\Phi^{\mathbb{V}}$. Finally, note that Definition 1 of a game assumes the fixed set of data variables V. A game specified by the version of this definition for an arbitrary set of data variables $\mathbb{V}$ will be referred to as "game over set $\mathbb{V}$".

Towards the proof of the completeness, we start with two technical observations. First, note that Theorem 2, originally proven for the fixed set of data variables V, also holds for an arbitrary finite set of data variables $\mathbb{V}$. We restate it in this more general form in the corollary below.

Corollary 1. *For any finite set of data variables $\mathbb{V}$ and any set of formulae $F \subseteq \Phi^{\mathbb{V}}$ and any formula $\varphi \in \Phi^{\mathbb{V}}$, if $F \nvdash \varphi$, then there is a state w of a game over set $\mathbb{V}$ such that $w \Vdash f$ for each formula $f \in F$ and $w \nVdash \varphi$.*

Next, we show a "conservative extension" lemma which states that any game over set $\mathbb{V}$ could be extended to a game over a set $\mathbb{V}' \supseteq \mathbb{V}$ in a way that preserves satisfiability of formulae in language $\Phi^{\mathbb{V}}$.

Lemma 32. *For any game $G = (W, \{\sim_x\}_{x\in\mathbb{V}}, \Delta, M, \pi)$ over a set of data variables $\mathbb{V}$ and any set of data variables $\mathbb{V}' \supseteq \mathbb{V}$ there is a game $G' = (W, \{\sim'_x\}_{x\in\mathbb{V}'}, \Delta, M, \pi)$ over a set $\mathbb{V}'$ such that for any formula $\varphi \in \Phi^{\mathbb{V}}$ and any state $w \in W$,*

$$w \Vdash \varphi \text{ iff } w \Vdash' \varphi,$$

where $\Vdash$ and $\Vdash'$ are satisfaction relations for games G and G' respectively.

Proof. Define relation $\sim'_x$ for any data variable $x \in \mathbb{V}'$ as follows: if $x \in \mathbb{V}$, then $\sim'_x$ is the relation $\sim_x$, else it is the equality relation $=$ on W. It can be shown using Definition 2 and induction on the structural complexity of formula $\varphi \in \Phi^{\mathbb{V}}$ that $w \Vdash \varphi$ iff $w \Vdash' \varphi$ for any state $w \in W$. $\square$

We say that a formula φ is *data-finite* if the set of all data variables that occur in formula φ is finite. In other words, formula φ is data-finite if it can be generated by the grammar in Section 3 using only finite set X. We are now ready to state and prove the completeness theorem for data-finite formulae. Note that this theorem refers to language Φ, derivability relation $\vdash$, and games for the original fixed set of data variables V.

Theorem 4. *If $\varphi \in \Phi$ is a data-finite formula such that $\nvdash \varphi$, then there is a state w of a game such that $w \nVdash \varphi$.*

Proof. Let $\mathbb{V} \subseteq V$ be the set of all data variables that occur in formula φ. Note that set $\mathbb{V}$ is finite because formula φ is data-finite. Assumption $\nvdash \varphi$ of the theorem implies $\nvdash^{\mathbb{V}} \varphi$ because any proof in language $\Phi^{\mathbb{V}}$ is a proof in language Φ. Thus, by Corollary 1, there is a game G over set $\mathbb{V}$ and a state w of that game such that $w \nVdash \varphi$. By Lemma 32, there is a game G' over set $V \supseteq \mathbb{V}$ such that $w \nVdash \varphi$ in game G'. $\square$

```
Ensure: satisfied[u, i] iff u ⊩ ψ_i for each state u ∈ W and each i ≤ n
  for i ≤ n do
    for u ∈ W do
      if ψ_i is a propositional variable then
        if u ∈ π(ψ_i) then
          satisfied[u, i] ← true
        else
          satisfied[u, i] ← false
        end if
      end if
      if ψ_i has the form ¬ψ_j then
        satisfied[u, i] ← ¬(satisfied[u, j])
      end if
      if ψ_i has the form ψ_j → ψ_k then
        satisfied[u, i] ← ¬(satisfied[u, j]) ∨ satisfied[u, k]
      end if
      if ψ_i has the form K_X ψ_j then
        answer ← true
        for u' ∈ W do
          if u ∼_X u' and ¬satisfied[u', ψ_j] then
            answer ← false
            break
          end if
        end for
        satisfied[u, i] ← answer
      end if
      if ψ_i has the form S^C_X ψ_j then
        satisfied[u, i] ← F(C, X, u, j)
      end if
    end for
  end for
```

Fig. 7. Model checking algorithm using function F defined in Fig. 8.

```
Ensure: return = true iff u ⊩ S^C_X ψ_j
  for s ∈ Δ^C do
    answer ← true
    for (u', δ, v) ∈ M do
      if u ∼_X u' and s =_C δ and ¬satisfied[v, j] then
        answer ← false
        break
      end if
    end for
    if answer then
      return true
    end if
  end for
  return false
```

Fig. 8. Boolean function $F(C, x, u, j)$.

9. Model checking

In this section, we propose a model checking algorithm for our logical system and discuss its complexity. By model checking we mean deciding if a statement of the form $w \Vdash \varphi$ holds for a given state w of a given game and a given formula $\varphi \in \Phi$. The proposed algorithm only works when the set of data variables V, the set of agents $\mathcal{A}$, the set of states W of the game, and the set of actions Δ of the game are *finite*. It is a dynamic programming algorithm that pre-computes the Boolean value of the statement $u \Vdash \psi$ for each state $u \in W$ and each proper subformula ψ of formula φ. More specifically, let $\psi_1, \dots, \psi_n$ be the list of all subformulae of formula φ, including formula φ itself, ordered in non-decreasing order of sizes. Note that such ordering guarantees that if formula ψ_j is a proper subformula of formula ψ_i, then $j < i$.

The model checking algorithm, see Fig. 7, computes the Boolean value $satisfied[u, i]$ which is true iff formula ψ_i is satisfied in state u. The most important part of this algorithm, function $F(C, X, u, j)$, is shown separately in Fig. 8. This function returns true iff formula $\mathsf{S}^C_X \psi_j$ is satisfied in state u.

To analyse the execution time of the model checking algorithm, it is important to fix the way indistinguishability relation $\sim_x$ and mechanism relation M are represented by the algorithm. In our analysis, we assume that both of them are represented in the most straightforward way as unordered lists of tuples. In the case of relation $\sim_x$ it is a list of pairs of states and in the case of relation $M \subseteq W \times \Delta^{\mathcal{A}} \times W$ it is a set of triples. In this case, to check if $u \sim_x u'$ is true it takes $O(|\sim_x|)$ steps, where $|\sim_x|$ is the number of pairs in relation $\sim_x$. By $|\sim|$ we denote the maximal value of $|\sim_x|$ for all possible data variables $x \in X$. Then, to check if $u \sim_X u'$ is true for some dataset $X \subseteq V$, it takes $O(|\sim| \times |V|)$ steps. Finally, assuming that

	Local	Global
No Recall	A	C
Perfect Recall	B	D

Fig. 9. Four different multistep data-informed strategy modalities.

an action profile δ and a strategy s are also represented as tuples, it takes at most $O(|\mathcal{A}|)$ steps to check if $\delta =_C s$ for any coalition $C \subseteq \mathcal{A}$. Therefore, an execution of the algorithm depicted in Fig. 8 takes time $O(|\Delta^{\mathcal{A}}| \times |M| \times (|\sim| \times |V| + |\mathcal{A}|))$.

Let us now turn to the analysis of algorithm in Fig. 7. Its part inside the two nested "for" loops has five cases that correspond to formula ψ_i having different forms. The execution time for the case when ψ_i is a propositional variable, in the worse case, is proportional to the length of the list storing the elements of the set $\pi(p) \subseteq W$. Thus, this case takes time $O(|W|)$ in the worst case. In the case when formula ψ_i is either a negation or an implication, the execution time is $O(1)$. In the case when formula ψ_i has the form $\mathsf{K}_X\psi_j$, the execution time is $O(|W| \times |\sim| \times |V|)$. This is true because, as we have observed earlier, it takes $O(|\sim| \times |V|)$ time to check the condition $u \sim_X u'$. For the last case, the execution time is equal to the execution time of the algorithm in Fig. 8, which we have analysed earlier. Thus, the total worst-case execution time of the algorithm in Fig. 7 is

$$O(|n| \times |W| \times (|W| \times |\sim| \times |V| + |\Delta^{\mathcal{A}}| \times |M| \times (|\sim| \times |V| + |\mathcal{A}|))).$$

The same Big-Oh expression also gives the model checking time for any given formula of size n because such a formula has at most n subformulae.

Note that in addition to traditional model checking question if a given formula holds in a given state, the same model checking algorithm can be used to answer "data optimisation" questions. An example of such a question is: what is a minimal dataset X which informs dataset Y in state w? To answer this question, one can run model checking algorithm for statement $w \Vdash X \rhd Y$ with decreasing set X. Similar optimisation questions can also be solved for modalities $\mathsf{K}_X\varphi$ and $\mathsf{S}^C_X\varphi$.

10. Multistep strategies

In the future, we plan to consider multistep data-informed strategies to achieve. Fervari, Herzig, Li, and Wang studied similar strategies for agent-based knowledge [7]. Unfortunately, the Cooperation axiom is not true for such strategies because multistep strategies of coalitions C and D might achieve conditions $\varphi \to \psi$ and φ, respectively, at different times. As a result, there are no interesting properties for coalition strategies of this type that can be captured in our modal language. Thus, such strategies probably should be considered for single actors, as it is done in [7].

There are two important factors to consider when studying multistep data-informed strategies. First, if the actor has *perfect recall* of the information previously available to her and of the actions she has previously taken. This factor is not specific to data-informed strategies. The other factor is specific to data-informed strategies. We refer to it as the distinction between a local and a global access to data variables.[1] To understand what we mean by this, consider formula $w \Vdash \mathsf{S}^a_x\varphi$. Informally, it states that with access to variable x actor a can achieve goal φ in several steps. Note that during these several steps the game will transition through multiple states. If the value of variable x is available to actor a only in state w, then we say that a has a *local* access to variable x. If the value of x is available to a in each state it is passing through, then we say that a has a *global* access to variable x. Note that we can make an assumption that strategy has no recall or perfect recall and, independently from this, make either a local or a global access assumption. This gives us four possible modalities that we denote by A, B, C, and D, see Fig. 9.

We use the two games depicted in Fig. 10 to illustrate modalities A, B, C, and D and to compare their powers. Both of these games have just a single actor a who has just a single action "go forward". We use arrows to show the result of this action in each state. We refer to these two games as the *left* and the *right* games.

First, observe that, in the *left* game, $w_1 \Vdash \neg\mathsf{A}^a_x p$. Indeed, a strategy represented by modality A has only a local (at state w_1) access to variable x and no recall. So, it has enough data to deduce that the starting state is w_1 but it has no memory to be able to do exactly two transitions before it stops at state w_3, see Fig. 10 (left). At the same time, $w_1 \Vdash \mathsf{B}^a_x p$ because a strategy referred to by modality B does have perfect recall, so it would be able to count and stop after two moves.

Second, note that $u_1 \Vdash \neg\mathsf{B}^a_x p$ in the *right* game. This is true because the strategy represented by modality B has only a local (at state u_1) access to variable x. Thus, in spite of a having perfect recall, no matter how many moves it makes, the strategy will never be able to establish if it started at state u_1 or u_2, in both of which variable x has the same value. Without this crucial information, it will not be able to stop at state u_3, see Fig. 10 (right). On the other hand, $u_1 \Vdash \mathsf{C}^a_x p$. Indeed, consider the strategy "stop *the first time* you visit a state where $x = 2$". Note that stopping the first time when $x = 2$ does not require counting and can be done by a strategy with no recall. This strategy does require a global (in each visited state) access to variable x, which is allowed by the definition of modality C. The strategy works starting from either state

[1] In Section 3, we have discussed local and global *dependencies* and local and global *strategies*. These notions are related, but not the same as the notions of local and global *access* that we discuss in the current section.

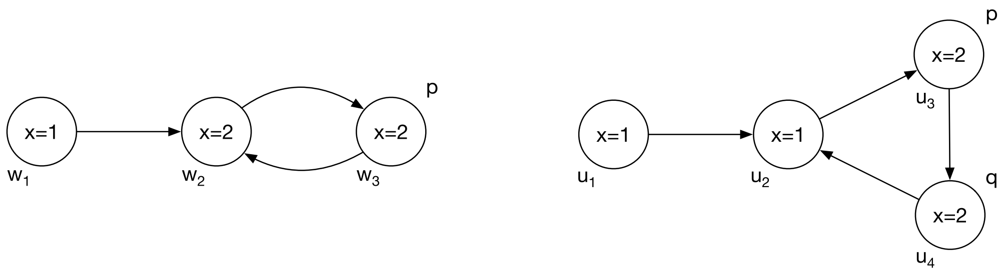

Fig. 10. The left and the right games.

u_1 or state u_2, where values of data variable x are equal. Thus, this strategy is informed by the data variable x in both of these states.

Third, observe that $u_1 \Vdash \neg\mathsf{C}^a_x q$ in the *right* game because a strategy represented by modality C has no recall. Thus, it will not be able to distinguish a visit to state u_3, where q is false, from a visit to state u_4, where q is true, because data variable x has the same value in both of these states, see Fig. 10 (right). At the same time, $u_1 \Vdash \mathsf{D}^a_x q$. Indeed, consider the strategy "stop *the second time* you visit state in which $x = 2$". This strategy requires a global access to variable x and an ability to count, both of which are allowed under the definition of modality D. The strategy works from either state u_1 or state u_2, where values of data variable x are equal. Hence, it is informed by the data variable x in both of these states.

It is clear from the definition of modalities A, B, C, and D that the class of strategies referred to by modality A is the largest and the class referred to by modality D is the smallest. We have already seen above an example of a goal that cannot be achieved by a B-class strategy, but can be achieved by a C-class strategy. We conclude this section with an observation that there are goals that cannot be achieved by a C-class strategy, but can be achieved by a B-class strategy. Indeed, let us go back to the *left* game and observe that $w_1 \Vdash \neg\mathsf{C}^a_x p$. This is again because a strategy represented by modality C cannot count. At the same time, as we have seen earlier in this section, $w_1 \Vdash \mathsf{B}^a_x p$. In the future, we plan to study the interplay between modalities A, B, C, and D.

11. Conclusion

In this article, we introduced a new "data-informed strategy modality" and a complete logical system that described the interplay between the functional dependency expression and the data-informed strategy and knowledge modalities. This system highlights a new approach to multiagent systems that emphasizes the connection between data and knowledge instead of the connection between agents and knowledge traditionally studied in the literature. In addition, we proposed and analysed a model checking algorithm for this system and discussed a possible extension of the system to multistep strategies. Another interesting possible direction for future work is proving decidability of the proposed system. Perhaps tableaux-based approach could be used here similar to how it is done in [5].

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Junli Jiang acknowledges the support of Chongqing Municipal Education Commission Humanities and Social Sciences Research Program, the Key Research Funds for the Key Liberal Science Research Base of Chongqing (NO. 16SKB036), and the Fundamental Research Funds for the Central Universities (SWU1409412).

References

[1] B. Ahmed, A.W. Malik, T. Hafeez, N. Ahmed, Services and simulation frameworks for vehicular cloud computing: a contemporary survey, EURASIP J. Wirel. Commun. Netw. 2019 (1) (2019) 1–21.
[2] Y. Zhu, A. Smith, K. Hauser, Informative path planning for automatic robotic auscultation, in: ICRA 2021 Workshop on Impact of COVID-19 on Medical Robotics and Wearables Research, 2021.
[3] M. Pauly, Logic for social software, Ph.D. thesis, Institute for Logic, Language, and Computation, 2001.
[4] M. Pauly, A modal logic for coalitional power in games, J. Log. Comput. 12 (1) (2002) 149–166, https://doi.org/10.1093/logcom/12.1.149.
[5] T. Ågotnes, N. Alechina, Coalition logic with individual, distributed and common knowledge, J. Log. Comput. 29 (2019) 1041–1069, https://doi.org/10.1093/logcom/exv085.
[6] P. Naumov, J. Tao, Coalition power in epistemic transition systems, in: Proceedings of the 2017 International Conference on Autonomous Agents and Multiagent Systems, AAMAS, 2017, pp. 723–731.
[7] R. Fervari, A. Herzig, Y. Li, Y. Wang, Strategically knowing how, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, 2017, pp. 1031–1038.

[8] P. Naumov, J. Tao, Together we know how to achieve: an epistemic logic of know-how, in: Proceedings of 16th Conference on Theoretical Aspects of Rationality and Knowledge, TARK, July 24–26, 2017, in: EPTCS, vol. 251, 2017, pp. 441–453.
[9] P. Naumov, J. Tao, Together we know how to achieve: an epistemic logic of know-how, Artif. Intell. 262 (2018) 279–300, https://doi.org/10.1016/j.artint.2018.06.007.
[10] P. Naumov, J. Tao, Strategic coalitions with perfect recall, in: Proceedings of Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[11] R. Fervari, F.R. Velázquez-Quesada, Y. Wang, Bisimulations for knowing how logics, Rev. Symb. Log. (2021) 1–37.
[12] Y. Li, Y. Wang, Neighborhood semantics for logic of knowing how, Synthese (2021) 1–29.
[13] W. van der Hoek, M. Wooldridge, On the logic of cooperation and propositional control, Artif. Intell. 164 (1) (2005) 81–119.
[14] R. Alur, T.A. Henzinger, O. Kupferman, Alternating-time temporal logic, J. ACM 49 (5) (2002) 672–713, https://doi.org/10.1145/585265.585270.
[15] W. van der Hoek, M. Wooldridge, Cooperation, knowledge, and time: alternating-time temporal epistemic logic and its applications, Stud. Log. 75 (1) (2003) 125–157, https://doi.org/10.1023/A:1026171312755.
[16] N. Bulling, V. Goranko, W. Jamroga, Logics for reasoning about strategic abilities in multi-player games, in: Models of Strategic Reasoning, Springer, 2015, pp. 93–136.
[17] P. Naumov, J. Tao, Second-order know-how strategies, in: Proceedings of the 2018 International Conference on Autonomous Agents and Multiagent Systems, AAMAS, 2018, pp. 390–398.
[18] Y. Wang, A logic of goal-directed knowing how, Synthese 195 (10) (2018) 4419–4439.
[19] W.W. Armstrong, Dependency structures of data base relationships, in: Information Processing 74, Proc. IFIP Congress, Stockholm, 1974, North-Holland, Amsterdam, 1974, pp. 580–583.
[20] H. Garcia-Molina, J. Ullman, J. Widom, Database Systems: The Complete Book, 2nd edition, Prentice-Hall, 2009.
[21] C. Beeri, R. Fagin, J.H. Howard, A complete axiomatization for functional and multivalued dependencies in database relations, in: SIGMOD '77: Proceedings of the 1977 ACM SIGMOD International Conference on Management of Data, ACM, New York, NY, USA, 1977, pp. 47–61.
[22] A. Baltag, To know is to know the value of a variable, in: L.D. Beklemishev, S. Demri, A. Maté (Eds.), Proceedings of the 11th Conference on "Advances in Modal Logic", held in Budapest, Hungary, August 30–September 2, 2016, in: Advances in Modal Logic, vol. 11, College Publications, 2016, pp. 135–155.
[23] K. Deuser, P. Naumov, Armstrong's axioms and navigation strategies, in: Proceedings of Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[24] J. van Eijck, M. Gattinger, Y. Wang, Knowing values and public inspection, in: Indian Conference on Logic and Its Applications, Springer, 2017, pp. 77–90.
[25] J. Väänänen, Dependence Logic: A New Approach to Independence Friendly Logic, vol. 70, Cambridge University Press, 2007.
[26] A. Baltag, J. van Benthem, A simple logic of functional dependence, J. Philos. Log. (2021) 1–67.
[27] Y.N. Wáng, T. Ågotnes, Simpler completeness proofs for modal logics with intersection, in: International Workshop on Dynamic Logic, Springer, 2020, pp. 259–276.
[28] P. Naumov, J. Tao, An epistemic logic of blameworthiness, Artif. Intell. 283 (June 2020) 103269.
[29] V. Goranko, G. van Drimmelen, Complete axiomatization and decidability of alternating-time temporal logic, Theor. Comput. Sci. 353 (1) (2006) 93–117, https://doi.org/10.1016/j.tcs.2005.07.043.
[30] V. Goranko, S. Enqvist, Socially friendly and group protecting coalition logics, in: Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 372–380.