# Navigability with Intermediate Constraints

Kaya Deuser
Vassar College
Poughkeepsie, New York, USA
kdeuser@vassar.edu

Pavel Naumov
Claremont McKenna College
Claremont, California, USA
Pavel.Naumov@ClaremontMcKenna.edu

July 12, 2018

**Abstract**

The article studies navigability of an autonomous agent in a maze where some rooms may be indistinguishable. In a previous work the authors have shown that the properties of navigability in such a setting depend on whether an agent has perfect recall. Navigability by strategies with perfect recall is a transitive relation and navigability by memoryless strategies is not. Independently, Li and Wang proposed a notion of navigability with intermediate constraints for linear navigation strategies. Linear strategies are different from both perfect recall and memoryless strategies.

This article shows that a certain form of transitivity, expressible in the language with intermediate constraints, holds for memoryless strategies. The main technical result is a sound and complete logical system describing the properties of memoryless strategies in the language with intermediate constraints.

## 1 Introduction

Autonomous agents such as self-navigating missiles, self-driving cars, and robotic vacuum cleaners are often facing the challenge of navigating under conditions of uncertainty about their exact location. A solution to such a problem can be formally described in terms of instructions that transition a system from one state to another, assuming that the agent cannot distinguish some of the states. We refer to such systems as *epistemic transition systems*.

Figure 1 depicts a graph of an epistemic transition system $T_1$. This system has twelve states shown as the nodes of the graph. Dashed lines between states

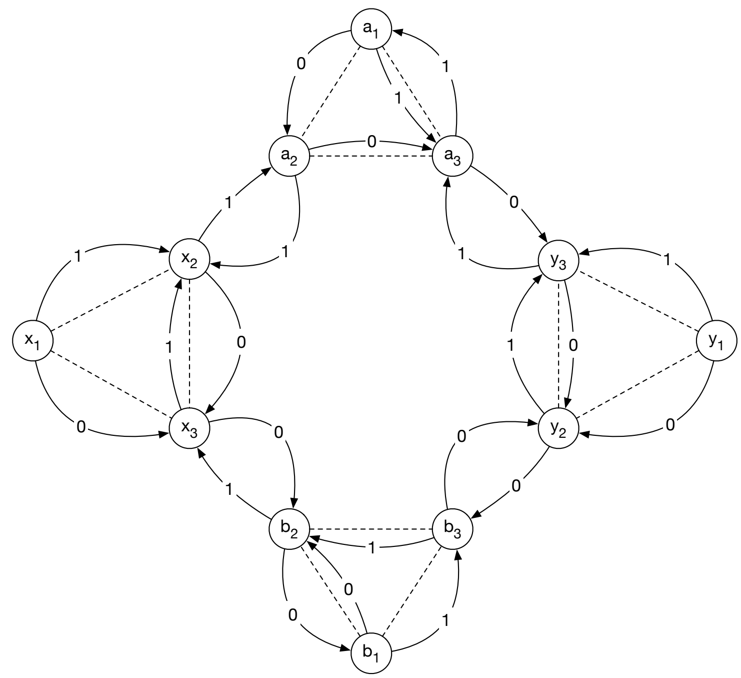


Figure 1: Epistemic transition system $T_1$.

represent the indistinguishability relation. This relation forms four classes of indistinguishable states: $[x_1] = \{x_1, x_2, x_3\}$, $[a_1] = \{a_1, a_2, a_3\}$, $[y_1] = \{y_1, y_2, y_3\}$, and $[b_1] = \{b_1, b_2, b_3\}$. The directed arrows in the figure represent the possible transitions that the system can take. The label on an arrow specifies the instruction that should be invoked to accomplish this transition. For example, instruction 1 can be used to transition the system from state $x_2$ to state $a_2$.

In this article we study navigability between classes of indistinguishable states. For instance, consider a navigation strategy $s$ that invokes instruction 1 in classes $[x_1]$ and $[b_1]$ and instruction 0 in classes $[a_1]$ and $[y_1]$. Note that if the system $T_1$ starts in any state of class $[x_1]$, then this strategy will eventually bring the system to state $y_3$ of class $[y_1]$. Similarly, if the system starts in any state of class $[y_1]$, then the same instruction will eventually bring the system to state $x_3$ of class $[x_1]$. Generally speaking, we say that a strategy can be used to navigate from class $u$ to class $v$ if from *each* state of class $u$ the strategy will eventually transition the system to *some* (not necessarily always the same) state of class $v$.

Note that the same strategy $s$ can also be used to navigate from each state of classes $[x_1]$, $[a_1]$, and $[y_1]$ to state $b_3$ of class $[b_1]$. Generally speaking, we say that a strategy can be used to navigate from set of classes $A$ to set of class $B$ if from *each* state of each class in set $A$ the strategy will eventually transition the system to *some* (not necessarily always the same) state of *some* (not necessarily always the same) class in set $B$.

The main focus of our work is on navigability between classes with intermediate constraints. If $A$, $B$, and $C$ are three sets of indistinguishability classes of states in an epistemic transition system, then by $A \rhd_B C$ we denote the existence of a strategy to navigate from set $A$ to set $C$ under which the system is constrained to intermediate states in set $B$. For the sake of mathematical elegance that will become clear later, we include the initial but not the final state into the intermediate constraint. That is, the intermediate constraint lists all classes in which one would actually need to *use* the strategy to determine which instruction to invoke. For example, $\{[x_1],[a_1]\} \rhd_{\{[x_1],[a_1]\}} \{[y_1]\}$, because the discussed earlier strategy $s$ can be used to navigate from any state in classes $[x_1]$ and $[a_1]$ to state $y_3$ of class $[y_1]$ while being constrained to use only states in classes $[x_1]$ and $[a_1]$ as intermediate states.

The properties of navigability depend on the type of strategies we consider. In this article we will mention three such types: *perfect recall* strategies, *memoryless* strategies, and *linear* strategies. A perfect recall strategy can make a decision about the instruction to be invoked based on the complete history of all classes visited and the instructions used so far. As the name suggests, a memoryless strategy can only take into account the indistinguishability class the system is currently in. Finally, a linear strategy is a list of instructions that should be executed in the given order without taking into account the past and present indistinguishability classes.

The discussed earlier strategy $s$ (that uses instruction 1 in classes $[x_1]$ and $[b_1]$ and instruction 0 in classes $[a_1]$ and $[y_1]$) is memoryless because it chooses an instruction based only on the current class. As we have seen earlier, this strategy can be used to navigate from class $[x_1]$ to class $[y_1]$ while being constrained to classes $[x_1]$ and $[a_1]$. Navigation from class $[y_1]$ to class $[x_1]$ while constrained to classes $[y_1]$ and $[a_1]$ is less trivial. It can be accomplished using a *perfect recall* strategy that uses instruction 1 each time in class $[y_1]$ and on *the first* and *the third* visit to a state in class $[a_1]$ and uses instruction 0 on *the second* visit to a state in class $[a_1]$. However, a *memoryless* strategy to navigate from class $[y_1]$ to class $[x_1]$ while constrained to classes $[y_1]$ and $[a_1]$ does not exist. Indeed, any such strategy will have to transition the system from state $y_3$ to state $a_3$. It also will have to use the same instruction on each visit to each state of class $[a_1]$. If this instruction is 0, then the system will “bounce” back from state $a_3$ to state $y_3$. If the instruction is 1, then the system will “loop” between states $a_1$ and $a_3$. In either of these cases it will never reach a state in class $[x_1]$. There is no single *linear* strategy to navigate from all states of class $[x_1]$ to a state in class $[y_1]$ of the epistemic transition system $T_1$, even without intermediate constraints. However, there are separate such strategies for each state in class $[x_1]$. For example, linear strategy 1100 navigates the system from state $x_1$ to state $y_3$ while constrained to classes $[x_1]$ and $[a_1]$.

We gave a complete axiomatization of the properties of navigability without intermediate constraints for perfect recall and memoryless strategies in [11]. A similar result for linear strategies was given by Wang in [34]. The major property that distinguishes navigability by these three types of strategies is transitivity. While navigability by perfect recall and linear strategies is a transitive relation, navigability by memoryless strategies is not. Indeed, consider transition system

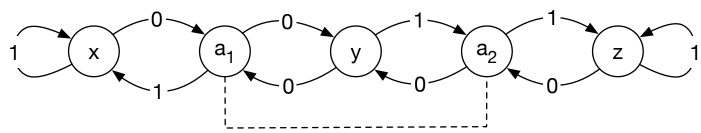


Figure 2: Epistemic transition system $T_2$.

$T_2$ depicted in Figure 2. Note that the memoryless strategy that invokes instruction 0 in each state can be used to navigate from class $[x]$ to class $[y]$ in this transition system. Similarly, the memoryless strategy that invokes instruction 1 in each state can be used to navigate from class $[y]$ to class $[z]$. However, there is no memoryless strategy to navigate from class $[x]$ to class $[z]$ because such a strategy would have to use the same instruction in states $a_1$ and $a_2$.

Complete axiomatization of navigability by linear strategies with intermediate constraints was given by Li and Wang [21]. In this article we propose such axiomatization for memoryless strategies. An important observation about the axiomatization that we propose is that it includes a restricted form of transitivity. Namely, it contains axiom $A \rhd_B C \to (C \rhd_D E \to A \rhd_{B\cup D} E)$, where classes $B$ and $D$ are disjoint. In other words, although a simple language of navigability without constraints does not allow to capture any meaningful form of transitivity for memoryless strategies, the more expressive language with intermediate constraints can do this.

## 2 Other Related Literature

Most of the existing literature on logical systems for reasoning about strategies is focused on modal logics for coalition strategies. Logics of coalition power were proposed by Marc Pauly [28, 29], who also proved the completeness of the basic logic of coalition power. Pauly's approach has been widely studied in literature [14, 33, 9, 30, 3, 4, 7]. An alternative, binary-modality-based, logical system was proposed by More and Naumov [22].

Alur, Henzinger, and Kupferman introduced Alternating-Time Temporal Logic (ATL) that combines temporal and coalition modalities [5]. Completeness of ATL was shown by Goranko and Drimmelen [15]. Van der Hoek and Wooldridge proposed to combine ATL with epistemic modality to form Alternating-Time Temporal Epistemic Logic [32]. However, they did not prove the completeness theorem for the proposed logical system. A completeness theorem for a logical system that combines coalition power and epistemic modalities was proven by Ågotnes and Alechina [1]. An alternative approach to expressing the power to achieve a goal in a temporal setting is the STIT logic [8, 17, 18, 16, 27].

[10] has shown that coalition logic can be embedded into a variation of STIT logic.

The notion of a strategy that we consider in this article is much more restrictive than the the notion of strategy in the works mentioned above. Namely, we assume that the strategy must be based only on the information available to the agent. This is captured in our setting by requiring the strategy to be the same in all indistinguishable states or all indistinguishable histories. This restriction on strategies has been studied before under different names. Jamroga and Ågotnes talk about "knowledge to identify and execute a strategy" [19], Jamroga and van der Hoek discuss "difference between an agent knowing that he has a suitable strategy and knowing the strategy itself" [20]. Van Benthem calls such strategies "uniform" [31]. Naumov and Tao [23] used the term "executable strategy". Ågotnes and Alechina gave a complete axiomatization of an interplay between single-agent knowledge and know-how modalities [2]. Naumov and Tao [23] axiomatized interplay between distributed knowledge modality and know-how coalition strategies for enforcing a condition indefinitely. A similar complete logical system in a *single-agent* setting for know-how strategies to achieve a goal in multiple steps rather than to maintain a goal is developed by Fervari, Herzig, Li, Wang [12]. Naumov and Tao proposed a complete trimodal logical system that describes an interplay between distributed knowledge, coalition power, and know-how coalition power modalities for goals achievable in one step [24] as well as a modal logic for second-order know-how [25]. All of the above works do not consider a perfect recall setting. Modal logic that combines distributed knowledge with coalition power in a perfect recall setting has been recently proposed by Naumov and Tao [26]. Unlike this article, they only consider goals achievable in one step.

Most of the mentioned above works deal with multiagent game-like settings where coalitions of players cooperate to achieve a certain goal. Although navigability explicitly assumes existence of only one agent, it can be thought of as a two-person game between the agent and the environment because of our assumption in this article that transitions in the system might be nondeterministic. Thus, the modal logic approach and the relational approach provide two different languages for describing the same phenomenon.

# 3 Article Outline

The rest of this article is structured as following. In Section 4 we introduce the syntax of our logical system. In Section 5 we define the formal semantics of this system. Section 6 lists the axioms of our system. In Section 7 we give several examples of formal derivations. These results will be used later in the proof of completeness. Section 8 and Section 9 prove the soundness and the completeness of our logical system respectively. Section 10 concludes.

# 4 Syntax

In the introduction we discussed the ternary relation $A \rhd_B C$ as a relation between three sets of indistinguishability classes of a given epistemic transition system. Note that $A$, $B$, and $C$ here are not syntactical objects, but sets of states in a specific system. In the rest of this article we present a formal logical system capturing the universally true properties of this relation in an arbitrary epistemic transition system. Note that sets of classes $A$, $B$ and $C$ are specific to a particular epistemic transition system and thus can not be used to formally state properties common to all epistemic transition systems. In this article we overcome this issue by assuming that there is a fixed *finite* set $V$ of *views*. A transition model specifies an observation function from states to views instead of specifying an indistinguishability relation between states. Informally, two states are indistinguishable if and only if the values of the observation function on these two states are equal. Using views instead of classes the language of our logical system can be defined independently from a particular epistemic transition system as follows.

**Definition 1.** *Let $\Phi$ be the minimal set of formulae such that*

1. $A \rhd_B C \in \Phi$ *for all sets* $A, B, C \subseteq V$,
2. $\neg\varphi, \varphi \to \psi \in \Phi$, *for all* $\varphi, \psi \in \Phi$.

The above assumption that set of views $V$ is finite is used later in the proof of completeness.

# 5 Semantics

In this section we define formal semantics of our logical system.

**Definition 2.** *Epistemic transition system is a tuple* $(S, o, I, \{\to_i\}_{i\in I})$, *where*

1. $S$ *is a set of states,*
2. $o : S \to V$ *is an observation function,*
3. $I$ *is a set of instructions,*
4. $\to_i$ *is a binary relation between states for each* $i \in I$.

For example, for the epistemic transition system $T_2$ depicted in Figure 2, set $S$ is $\{x, a_1, y, a_2, z\}$. Observation function $o$ is such that $o(x) = v_1$, $o(a_1) = o(a_2) = v_2$, $o(y) = v_3$, and $o(z) = v_4$, where $v_1$, $v_2$, $v_3$, and $v_4$ are arbitrary distinctive elements of set $V$. Instruction set $I$ is $\{0, 1\}$. Relation $\to_0$ is $\{(x, a_1), (a_1, y), (y, a_1), (a_2, y), (z, a_2)\}$ and relation $\to_1$ is $\{(x, x), (a_1, x), (y, a_2), (a_2, z), (z, z)\}$.

Note that epistemic transition system $T_2$ is deterministic in the sense that there is a unique state $v$ into which the system transitions from a given state $w$

under a given instruction $i$. Definition 2 specifies a transition system in terms of a transitive relation $\to_i$. We allow several states $v$ such that $w \to_i v$ to model *nondeterministic* transitions. We allow the set of such states $v$ to be empty to model *terminating* transitions. Informally, if in a state $w$ an instruction $i$ is invoked such that there is no $v$ for which $w \to_i v$, then the system terminates and no further instructions are executed.

In the introduction we made a distinction between memoryless and recall strategies. Since the rest of the article deals only with memoryless strategies, we refer to such strategies simply as *strategies*.

**Definition 3.** *A strategy is an arbitrary function from $V$ to $I$.*

A side effect of our choice to use views instead of equivalence classes is that there might be one or more views that are not values of the observation function on any state in the epistemic transition system. Such views define what we call "empty equivalence classes" in the introduction. Per Definition 3, a strategy must still be defined on such views.

The next definition specifies the paths in an epistemic transition system that start in a given set of views and are compatible with a given strategy.

**Definition 4.** *For any set $A \subseteq V$ and any strategy $s$, let $Path_s(A)$ be the set of all (finite or infinite) sequences $w_0, w_1, w_2, \dots \in S$ such that*

1. *$o(w_0) \in A$, and*
2. *$w_k \to_{s(o(w_k))} w_{k+1}$ for each $k \geqslant 0$ for which $w_{k+1}$ exists.*

For epistemic transition system $T_2$, sequence $x, a_1, y$ belongs to set $Path_s(\{o(x)\})$, where $o(x)$ is the view assigned in system in $T_2$ to state $x$ and $s$ is a strategy such that $s(v) = 0$ for each $v \in V$.

A path $\pi'$ is an extension of a path $\pi$ if $\pi$ is a prefix of $\pi'$ or $\pi' = \pi$.

**Definition 5.** *$MaxPath_s(A)$ is the set of all sequences in $Path_s(A)$ that are either infinite or cannot be extended to a longer sequence in $Path_s(A)$.*

Note that we do not require paths to be simple. For epistemic transition system $T_2$, the infinite sequence $y, a_2, z, z, \dots$ belongs to the set $MaxPath_s(\{o(y)\})$, where $o(y)$ is the view assigned in that epistemic transition system to state $y$ and $s$ is a strategy such that $s(v) = 1$ for each $v \in V$.

**Lemma 1.** *Any sequence in $Path_s(A)$ can be extended to a sequence in $MaxPath_s(A)$.*

*Proof.* Any sequence in $Path_s(A)$ which is not in $MaxPath_s(A)$ can be extended to a longer sequence in $Path_s(A)$. Repeating this step multiple times one can get a finite or an infinite sequence in $MaxPath_s(A)$. □

The next definition introduces a technical notation that we use to define the semantics of the restricted navigability relation $A \rhd_B C$.

**Definition 6.** *Let $Until(A, B)$ be the set of all such sequences $w_0, w_1, w_2, \dots$ that there is $k_0 \geqslant 0$ where*

*1.* $o(w_k) \in A$ *for each* $k < k_0$ *and*

*2.* $o(w_{k_0}) \in B$.

**Definition 7.** *For any epistemic transition system* $T$ *and any formula* $\varphi$*, satisfiability relation* $T \vDash \varphi$ *is defined as follows:*

*1.* $T \vDash A \rhd_B C$ *if* $MaxPath_s(A) \subseteq Until(B, C)$ *for some strategy* $s$*,*

*2.* $T \vDash \neg\varphi$ *if* $T \nvDash \varphi$*,*

*3.* $T \vDash \varphi \to \psi$ *if* $T \nvDash \varphi$ *or* $T \vDash \psi$*.*

Note that although we allow non-deterministic transitions, item 1 in the above definition requires all paths from set $MaxPath_s(A)$ to belong to set $Until(B, C)$. This is different from how non-deterministic computation is treated in the automata theory.

# 6 Axioms

In addition to the propositional tautologies in language $\Phi$, our logical system contains the following axioms:

1. Reflexivity: $A \rhd_B C$, where $A \subseteq C$,
2. Augmentation: $A \rhd_B C \to (A \cup D) \rhd_B (C \cup D)$,
3. Transitivity: $A \rhd_B C \to (C \rhd_D E \to A \rhd_{B \cup D} E)$, where $B \cap D = \varnothing$,
4. Early Bird: $A \rhd_B C \to A \rhd_{B \setminus C} C$,
5. Trivial Path: $A \rhd_\varnothing B \to (A \setminus B) \rhd_\varnothing \varnothing$,
6. Path to Nowhere: $A \rhd_B \varnothing \to A \rhd_\varnothing \varnothing$.

If the subscript is omitted from the first three axioms, then the resulting properties are known in the database literature as Armstrong axioms [13, p. 81]. They give a sound and complete axiomatization of the functional dependency relation in database theory [6].

The fourth axiom says that if there is a strategy to navigate from set $A$ to set $C$ passing only through set $B$, then there is a strategy to navigate from set $A$ to set $C$ passing only through set $B \setminus C$. Indeed, once a state of a class of the set $C$ is reached for the first time, there is no need to continue the execution of the strategy. Thus, the navigation path will never have to pass through set $B \cap C$. In essence, the axiom states that the execution of any strategy can be terminated once the desired goal is reached for the first time. For this reason we call this axiom the Early Bird axiom.

The Trivial Path axiom states that if there is a strategy to navigate from set $A$ to set $B$ through an empty set of states, then set $A \setminus B$ must be empty.

Finally, the Path to Nowhere axiom states that if there is a strategy to navigate from set $A$ to a state in an empty set, then set $A$ must be empty.

We write $\vdash \varphi$ if formula $\varphi$ is provable in our logical system using the Modus Ponens inference rule. We write $X \vdash \varphi$ if formula $\varphi$ is provable from the axioms of our system and the set of additional axioms $X$.

# 7 Examples of Derivations

The soundness and the completeness of our logical system is proven in Section 8 and Section 9. In this section we give several examples of formal derivations in this system. The results obtained here are used in the proof of the completeness.

**Lemma 2.** $\vdash A \rhd_B C \to A' \rhd_B C$, *where* $A' \subseteq A$.

*Proof.* Assumption $A' \subseteq A$ implies that $\vdash A' \rhd_\varnothing A$ by the Reflexivity axiom. Hence, $\vdash A \rhd_B C \to A' \rhd_{\varnothing \cup B} C$ by the Transitivity axiom. Therefore, $\vdash A \rhd_B C \to A' \rhd_B C$. □

**Lemma 3.** $\vdash A \rhd_B C \to A \rhd_{B'} C$, *where* $B \subseteq B'$.

*Proof.* By the Reflexivity axiom, $\vdash A \rhd_{B' \setminus B} A$. Hence, by the Transitivity axiom, $\vdash A \rhd_B C \to A \rhd_{(B' \setminus B) \cup B} C$. Therefore, $\vdash A \rhd_B C \to A \rhd_{B'} C$, due to the assumption $B \subseteq B'$. □

**Lemma 4.** $\vdash A \rhd_B C \to A \rhd_B C'$, *where* $C \subseteq C'$.

*Proof.* By the Transitivity axiom, $\vdash A \rhd_B C \to (C \rhd_\varnothing C' \to A \rhd_B C')$. Hence, by the laws of logical reasoning,

$$\vdash C \rhd_\varnothing C' \to (A \rhd_B C \to A \rhd_B C'). \tag{1}$$

At the same time, the assumption $C \subseteq C'$ implies $\vdash C \rhd_\varnothing C'$ by the Reflexivity axiom. Hence, $A \rhd_B C \to A \rhd_B C'$ due to statement (1). □

**Lemma 5.** $\vdash B' \rhd_\varnothing \varnothing \to (A \rhd_B C \to A \rhd_{B \setminus B'} C)$.

*Proof.* First, $\vdash A \rhd_B C \to A \rhd_B (C \cup B')$ by Lemma 4. Second, the formula $A \rhd_B (C \cup B') \to A \rhd_{B \setminus (C \cup B')} (C \cup B')$ is an instance of the Early Bird axiom. Third, $\vdash A \rhd_{B \setminus (C \cup B')} (C \cup B') \to A \rhd_{B \setminus B'} (C \cup B')$ by Lemma 3 because $B \setminus (C \cup B') \subseteq B \setminus B'$. The three statements above by the laws of propositional reasoning imply that

$$\vdash A \rhd_B C \to A \rhd_{B \setminus B'} (C \cup B'). \tag{2}$$

At the same time,

$$B' \rhd_\varnothing \varnothing \to (B' \cup C) \rhd_\varnothing C \tag{3}$$

is an instance of the Augmentation axiom. Finally, the formula

$$A \rhd_{B \setminus B'} (C \cup B') \to ((B' \cup C) \rhd_\varnothing C \to A \rhd_{B \setminus B'} C) \tag{4}$$

is an instance of the Transitivity axiom. Taken together, statement (2), statement (3), and statement (4) imply by the laws of propositional reasoning that $\vdash B' \rhd_{\varnothing} \varnothing \to (A \rhd_B C \to A \rhd_{B\setminus B'} C)$. $\square$

**Lemma 6.** $\vdash A \rhd_B C \to (C \rhd_D E \to A \rhd_{B\cup D} E)$ *if* $B \cap D \subseteq C$.

*Proof.* Assumption $B \cap D \subseteq C$ implies that $B\setminus C \subseteq B\setminus D$. Thus, by Lemma 3, $\vdash A \rhd_{B\setminus C} C \to A \rhd_{B\setminus D} C$. At the same time $\vdash A \rhd_B C \to A \rhd_{B\setminus C} C$ by the Early Bird axiom. Hence, by the laws of propositional reasoning,

$$\vdash A \rhd_B C \to A \rhd_{B\setminus D} C.$$

Note that $A \rhd_{B\setminus D} C \to (C \rhd_D E \to A \rhd_{B\cup D} E)$ is an instance of the Transitivity axiom. Therefore, $\vdash A \rhd_B C \to (C \rhd_D E \to A \rhd_{B\cup D} E)$ by the laws of propositional reasoning. $\square$

## 8 Soundness

In this section we prove the soundness of our logical system. The soundness of each of the axioms is stated as a separate lemma. The soundness theorem for the logical system is given in the end of the section. We start with a technical lemma that lists properties of sets *Until* and *MaxPath*. These properties are used in the proofs of the soundness of the axioms.

**Lemma 7.** *For any set* $A, B, C \subseteq V$ *and any strategy* $s$*:*

1. $MaxPath_s(A) \subseteq Until(\varnothing, A)$,
2. $MaxPath_s(A) \subseteq MaxPath_s(A')$, *where* $A \subseteq A'$,
3. $MaxPath_s(A \cup B) = MaxPath_s(A) \cup MaxPath_s(B)$,
4. $Until(A, \varnothing) = \varnothing$,
5. $Until(A, B) \subseteq Until(A, B')$, *where* $B \subseteq B'$,
6. $Until(A, B) \subseteq Until(A', B)$, *where* $A \subseteq A'$,
7. $Until(A, B) \subseteq Until(A\setminus B, B)$,
8. $Until(\varnothing, A) \cap Until(\varnothing, B) \subseteq Until(\varnothing, A \cap B)$.

*Proof.* Statements 1 through 8 follow from Definition 5 and Definition 6. $\square$

Next, we show the soundness of the Reflexivity axiom and the Augmentation axiom.

**Lemma 8.** *If* $A \subseteq C$, *then* $T \vDash A \rhd_B C$.

*Proof.* By Lemma 7 and the assumption $A \subseteq C$,

$$MaxPath_s(A) \subseteq Until(\varnothing, A) \subseteq Until(\varnothing, C) \subseteq Until(B, C).$$

Therefore, $T \vDash A \rhd_B C$ by Definition 7. $\square$

**Lemma 9.** *If $T \vDash A \rhd_B C$, then $T \vDash (A \cup D) \rhd_B (C \cup D)$.*

*Proof.* Let $T \vDash A \rhd_B C$. Thus, $MaxPath_s(A) \subseteq Until(B, C)$ by Definition 7. Hence, by Lemma 7,

$$\begin{aligned} MaxPath_s(A \cup D) &= MaxPath_s(A) \cup MaxPath_s(D) \\ &\subseteq Until(B, C) \cup Until(\varnothing, D) \\ &\subseteq Until(B, C) \cup Until(B, D) \\ &\subseteq Until(B, C \cup D) \cup Until(B, C \cup D) \\ &= Until(B, C \cup D). \end{aligned}$$

Therefore, $T \vDash A, D \rhd_B C, D$ by Definition 7. $\square$

The proof of the soundness of the Transitivity axiom is based on the following auxiliary lemma.

**Lemma 10.** *If $MaxPath_{s_1}(A) \subseteq Until(B, C)$ and $s_1(v) = s_2(v)$ for each $v \in B$, then $MaxPath_{s_2}(A) \subseteq Until(B, C)$.*

*Proof.* Consider any sequence $\pi = w_0, w_1, \dots \in MaxPath_{s_2}(A)$. We prove that $\pi \in Until(B, C)$ by separating the following two cases:
*Case I*: $o(w_k) \in B \backslash C$ for each $k \geqslant 0$. Thus, $s_1(o(w_k)) = s_2(o(w_k))$ due to the assumption of the lemma that $s_1(v) = s_2(v)$ for each view $v \in B$. Hence, the assumption $\pi \in MaxPath_{s_2}(A)$ implies that $\pi \in MaxPath_{s_1}(A)$ by Definition 4. Therefore, $\pi \in Until(B, C)$ due to the assumption $MaxPath_{s_1}(A) \subseteq Until(B, C)$ of the lemma.
*Case II*: $o(w_k) \notin B \backslash C$ for some $k \geqslant 0$. Let $m \geqslant 0$ be the smallest such $k$ that $o(w_k) \notin B \backslash C$. Thus,

1. $o(w_k) \in (B \backslash C)$ for each $k < m$ and
2. $o(w_m) \notin (B \backslash C)$.

Hence, $o(w_k) \in B$ for each $k < m$. We now further split Case II into two different parts:
*Part A*: $o(w_m) \in C$. Therefore, $\pi \in Until(B, C)$ by Definition 6.
*Part B*: $o(w_m) \notin C$. Note that condition $o(w_k) \in B$ for each $k < m$ implies that $s_1(o(w_k)) = s_2(o(w_k))$ for each $k < m$ due to the assumption of the lemma that $s_1(v) = s_2(v)$ for each $v \in B$. Thus, $w_0, w_1, \dots, w_m \in Path_{s_1}(A)$. By Lemma 1, this sequence can be extended to a sequence $\pi' = w_0, w_1, \dots, w_m, \dots \in MaxPath_{s_1}(A)$.

At the same time $o(w_k) \in B$ for each $k \leqslant m$ by the choice of $k$ and $o(w_m) \notin C$ by the assumption of the case. Thus, $\pi' \notin Until(B, C)$. Therefore, $\pi' \in MaxPath_{s_1}(A)$ but $\pi' \notin Until(B, C)$, which contradicts the assumption $MaxPath_{s_1}(A) \subseteq Until(B, C)$ of the lemma. $\square$

We are now ready to finish the proof of the soundness of the remaining axioms of our logical system.

**Lemma 11.** *If $T \vDash A \rhd_B C$ and $T \vDash C \rhd_D E$, then $T \vDash A \rhd_{B\cup D} E$, where $B \cap D = \varnothing$.*

*Proof.* By Definition 7, the assumption $T \vDash A \rhd_B C$ implies that there is a strategy $s_1$ such that $MaxPath_{s_1}(A) \subseteq Until(B, C)$. Similarly, the assumption $T \vDash C \rhd_D E$ implies that there is a strategy $s_2$ such that $MaxPath_{s_2}(C) \subseteq Until(D, E)$.

Define strategy $s$ as follows

$$s(v) = \begin{cases} s_1(v) & \text{if } v \in B, \\ s_2(v) & \text{otherwise.} \end{cases} \tag{5}$$

By Definition 7, it suffices to show that $MaxPath_s(A) \subseteq Until(B \cup D, E)$. Indeed, consider any sequence $\pi = w_0, w_1, w_2, \dots \in MaxPath_s(A)$. We will show that $\pi \in Until(B \cup D, E)$.

Note that $s_1(v) = s(v)$ for each $v \in B$ by equation (5). Note also that $MaxPath_{s_1}(A) \subseteq Until(B, C)$ by the choice of strategy $s_1$. Thus, $MaxPath_s(A) \subseteq Until(B, C)$ By Lemma 10. Hence, $\pi \in Until(B, C)$ because $\pi \in MaxPath_s(A)$. Thus, by Definition 6, there is an integer $k_0 \geqslant 0$ such that

1. $o(w_k) \in B$ for each integer $k$ such that $k < k_0$,
2. $o(w_{k_0}) \in C$.

Note that $s_2(v) = s(v)$ for each view $v \in D$ by equation (5) and the assumption $B \cap D = \varnothing$. Also, $MaxPath_{s_2}(C) \subseteq Until(D, E)$ by the choice of strategy $s_2$. Thus, $MaxPath_s(C) \subseteq Until(D, E)$ by Lemma 10. Consider now the sequence $\pi' = w_{k_0}, w_{k_0+1}, w_{k_0+2}, \dots \in MaxPath_s(C)$. Then, $\pi' \in Until(D, E)$. Hence, by Definition 6, there is an integer $k'_0 \geqslant k_0$ such that

1. $o(w_k) \in D$ for each integer $k$ such that $k_0 \leqslant k < k'_0$,
2. $o(w_{k'_0}) \in E$.

Therefore, $\pi \in Until(B \cup D, E)$ by Definition 6. ☐

**Lemma 12.** *If $T \vDash A \rhd_B C$, then $T \vDash A \rhd_{B\backslash C} C$.*

*Proof.* Let $T \vDash A \rhd_B C$. Thus, $MaxPath_s(A) \subseteq Until(B, C)$ by Definition 7. Hence, $MaxPath_s(A) \subseteq Until(B, C) \subseteq Until(B\backslash C, C)$, by Lemma 7. Therefore, $T \vDash A \rhd_{B\backslash C} C$ by Definition 7. ☐

**Lemma 13.** *If $T \vDash A \rhd_\varnothing B$, then $T \vDash (A\backslash B) \rhd_\varnothing \varnothing$.*

*Proof.* By Definition 7, the assumption $T \vDash A \rhd_\varnothing B$ implies that there is a strategy $s$ such that $MaxPath_s(A) \subseteq Until(\varnothing, B)$. Thus, by Lemma 7,

$$MaxPath_s(A\backslash B) \subseteq MaxPath_s(A) \subseteq Until(\varnothing, B).$$

At the same time, by Lemma 7,

$$MaxPath_s(A\backslash B) \subseteq MaxPath_s(V\backslash B) \subseteq Until(\varnothing, V\backslash B).$$

Thus, by Lemma 7,

$$\begin{aligned} MaxPath_s(A\backslash B) &\subseteq Until(\varnothing, B) \cap Until(\varnothing, V\backslash B) \\ &= Until(\varnothing, B \cap (V\backslash B)) \\ &= Until(\varnothing, \varnothing). \end{aligned}$$

Therefore, $T \vDash (A\backslash B) \rhd_{\varnothing} \varnothing$ by Definition 7. □

**Lemma 14.** *If $T \vDash A \rhd_B \varnothing$, then $T \vDash A \rhd_{\varnothing} \varnothing$.*

*Proof.* By Definition 7, the assumption $T \vDash A \rhd_B \varnothing$ implies that there is a strategy $s$ such that $MaxPath_s(A) \subseteq Until(B, \varnothing)$. Thus, by Lemma 7,

$$MaxPath_s(A) \subseteq Until(B, \varnothing) = \varnothing \subseteq Until(\varnothing, \varnothing).$$

Therefore, $T \vDash A \rhd_{\varnothing} \varnothing$ by Definition 7. □

We end the section by stating the soundness theorem for our logical system. The theorem follows from the soundness of the individual axioms shown in the lemmas above.

**Theorem 1.** *If $\vdash \varphi$, then $T \vDash \varphi$ for each epistemic transition system $T$.*

# 9 Completeness

Suppose that set $X$ is a fixed maximal consistent set of formulae in the language $\Phi$. In this section we define a canonical epistemic transition system $T(X) = (S, o, I, \{\to_i\}_{i\in I})$ based on set $X$. Sets $S$ and $I$ of this system will be specified in terms of "valid" views.

## 9.1 Valid Views

Note that statement $v \rhd_{\varnothing} \varnothing$ means that there is a path from each state with view $v$ to an empty set. Since no such path exists, statement $v \rhd_{\varnothing} \varnothing$ could be interpreted as saying that there are no states with view $v$. Informally, we think about such views as "invalid" views. All views which are not invalid are referred to as "valid" views. The canonical epistemic system will have at least one state for each valid view.

**Definition 8.** $Valid = \{v \in V \mid X \nvdash v \rhd_{\varnothing} \varnothing\}$.

Below we prove several properties of valid views that are used later in the proof of the completeness.

**Lemma 15.** *$X \vdash \{b_1, \dots, b_n\} \rhd_\varnothing \varnothing$, for each integer $n \geqslant 0$ and all views $b_1, \dots, b_n \in V \backslash Valid$.*

*Proof.* We prove this lemma by induction on $n$. In the case $n = 0$, we need to show that $X \vdash \varnothing \rhd_\varnothing \varnothing$, which is true by the Reflexivity axiom.

Suppose that $X \vdash \{b_1, \dots, b_{n-1}\} \rhd_\varnothing \varnothing$. Thus, by the Augmentation axiom, $X \vdash \{b_1, \dots, b_{n-1}, b_n\} \rhd_\varnothing b_n$. On the other hand, the assumption $b_n \in V \backslash Valid$ by Definition 8 implies $X \vdash b_n \rhd_\varnothing \varnothing$. Therefore, by the Transitivity axiom, $X \vdash \{b_1, \dots, b_{n-1}, b_n\} \rhd_\varnothing \varnothing$. □

**Lemma 16.** *$X \vdash A \rhd_B C \to A \rhd_{B \cap Valid} C$.*

*Proof.* Lemma 15 implies that $X \vdash (B \backslash Valid) \rhd_\varnothing \varnothing$. Hence, by Lemma 5,

$$X \vdash A \rhd_B C \to A \rhd_{B \backslash (B \backslash Valid)} C.$$

In other words, $X \vdash A \rhd_B C \to A \rhd_{B \cap Valid} C$. □

**Lemma 17.** *$X \vdash A \rhd_B C \to A \rhd_B (C \cap Valid)$.*

*Proof.* Lemma 15 implies that $X \vdash (C \backslash Valid) \rhd_\varnothing \varnothing$. Thus, by the Augmentation axiom, $X \vdash C \rhd_\varnothing (C \cap Valid)$. At the same time,

$$A \rhd_B C \to (C \rhd_\varnothing (C \cap Valid) \to A \rhd_B (C \cap Valid))$$

is an instance of the Transitivity axiom. Therefore, by the laws of propositional reasoning, $X \vdash A \rhd_B C \to A \rhd_B (C \cap Valid)$. □

**Lemma 18.** *If $X \vdash A \rhd_B C$, then $(A \backslash C) \cap Valid \subseteq B$.*

*Proof.* Suppose that there is $v \in (A \backslash C) \cap Valid$ such that $v \notin B$. Thus, $v \in A$, $v \notin C$, $v \in Void$, and $v \notin B$.

Recall that $X \vdash A \rhd_B C$ by the assumption of the lemma. It follows that $X \vdash v \rhd_B C$ by Lemma 2 and due to $v \in A$. Thus, $X \vdash v \rhd_B B \cup C$ by Lemma 4. Hence, $X \vdash v \rhd_{B \backslash (B \cup C)} B \cup C$ by the Early Bird axiom. In other words, $X \vdash v \rhd_\varnothing B \cup C$. Thus, $X \vdash v \rhd_\varnothing \varnothing$ by the Trivial Path axiom. Hence, $v \notin Valid$ by Definition 8, which contradicts the choice of view $v$. □

## 9.2 Instructions

For each formula $A \rhd_B C \in X$ our canonical epistemic transition system $T(X)$ will have a strategy to navigate from set $A$ to set $C$ through set $B$. Generally speaking, this strategy will use a dedicated instruction associated with formula $A \rhd_B C$. Formally, the set of all instructions of transition system $T(X)$ is defined as a set of triples $(A, B, C)$ satisfying the three properties listed below:

**Definition 9.** *Let $I$ be the set of all triples $(A, B, C)$ such that*

1. *$A, B, C \subseteq Valid$,*

*2. $X \vdash A \rhd_{A\cup B} C$,*

*3. sets $A$, $B$, and $C$ are pairwise disjoint.*

Note that technically instruction $(A, B, C)$ is associated not with formula $A \rhd_B C$, but rather with formula $A \rhd_{A\cup B} C$. This is done in order to be able to assume that sets $A$, $B$, and $C$ are pairwise disjoint. The next lemma is a general property of sets, which is used later.

**Lemma 19.** *$(A \cup B)\backslash C = (A\backslash C) \cup (B\backslash(A \cup C))$.*

*Proof.*

$$\begin{aligned}(A \cup B)\backslash C &= (A \cup (B\backslash A))\backslash C = (A\backslash C) \cup ((B\backslash A)\backslash C) \\ &= (A\backslash C) \cup (B\backslash(A \cup C))\end{aligned}$$

□

We stated earlier that for each $A \rhd_B C \in X$ there is a dedicated instruction used by the strategy that navigates from set $A$ to set $C$ through set $B$. In some situations this dedicated instruction could be $(A, B, C)$. However in most cases we would need to slightly modify tuple $(A, B, C)$ into tuple $(A', B', C')$ in order for it to satisfy the three conditions from Definition 9. The next lemma specifies $(A', B', C')$ in terms of $(A, B, C)$ and proves that $(A', B', C')$ is an instruction.

**Lemma 20.** *If $X \vdash A \rhd_B C$, then $(A', B', C') \in I$, where $A' = (A\backslash C) \cap Valid$, $B' = (B\backslash(A \cup C)) \cap Valid$, and $C' = C \cap Valid$.*

*Proof.* By Definition 9, it suffices to show that sets $A'$, $B'$, and $C'$ are pairwise disjoint and that $X \vdash A' \rhd_{A'\cup B'} C'$. First, we show that these sets are pairwise disjoint:

$$\begin{aligned}A' \cap B' &= [(A\backslash C) \cap Valid] \cap B' \subseteq A \cap B' \\ &= A \cap [(B\backslash(A \cup C)) \cap Valid] \subseteq A \cap [B\backslash A] = \varnothing, \\ A' \cap C' &= [(A\backslash C) \cap Valid] \cap C' \subseteq [A\backslash C] \cap C' \\ &= [A\backslash C] \cap [C \cap Valid] \subseteq [A\backslash C] \cap C = \varnothing, \\ B' \cap C' &= [(B\backslash(A \cup C)) \cap Valid] \cap C' \subseteq [B\backslash C] \cap C' \\ &= [B\backslash C] \cap [C \cap Valid] \subseteq [B\backslash C] \cap C = \varnothing.\end{aligned}$$

Next, we show that $X \vdash A' \rhd_{A'\cup B'} C'$. Indeed, by Lemma 2, the assumption $X \vdash A \rhd_B C$ implies $X \vdash A\backslash C \rhd_B C$. Hence, $X \vdash A\backslash C \rhd_{A\cup B} C$ by Lemma 3. Thus, $X \vdash A\backslash C \rhd_{(A\cup B)\backslash C} C$ by the Early Bird axiom. Then, by Lemma 19,

$$X \vdash A\backslash C \rhd_{(A\backslash C)\cup(B\backslash(A\cup C))} C.$$

Thus, by Lemma 2,

$$X \vdash (A\backslash C) \cap Valid \rhd_{(A\backslash C)\cup(B\backslash(A\cup C))} C.$$

Hence, by Lemma 16,

$$X \vdash (A \backslash C) \cap Valid \rhd_{((A\backslash C)\cup(B\backslash(A\cup C)))\cap Valid} C.$$

Thus, by Lemma 17,

$$X \vdash (A \backslash C) \cap Valid \rhd_{((A\backslash C)\cup(B\backslash(A\cup C)))\cap Valid} C \cap Valid.$$

Therefore, $X \vdash A' \rhd_{A' \cup B'} C'$ by the choice of sets $A'$, $B'$, and $C'$. $\square$

Informally, the next lemma states that if there is an instruction to navigate from a set $A$ to an empty set, then set $A$ must be empty.

**Lemma 21.** *For any $(A, B, C) \in I$ if $C = \varnothing$, then $A = \varnothing$.*

*Proof.* The assumption $(A, B, C) \in I$ implies that $X \vdash A \rhd_{A\cup B} C$, by Definition 9. Thus, $X \vdash A \rhd_{A\cup B} \varnothing$ due to the assumption $C = \varnothing$. Hence, $X \vdash A \rhd_{\varnothing} \varnothing$ by the Path to Nowhere axiom. Suppose that there is a view $a \in A$. Hence, $X \vdash a \rhd_{\varnothing} \varnothing$ by Lemma 2. Thus, $a \notin Valid$ by Definition 8. At the same time, $A \subseteq Valid$ by Definition 9. Hence, $a \notin A$, which is a contradiction with the choice of view $a$. $\square$

### 9.3 States and Observation Function

There are two types of states in the canonical epistemic transition system $T(X)$. The first type of states comes from our intention for each $v \in Valid$ to have at least one state $w$ such that $o(w) = v$. Thus, we consider each $v \in Valid$ to be a state of the first type and define the observation function on the states of the first type as $o(v) = v$.

In addition to the states of the first type, the canonical epistemic transition system also has states of the second type. Informally, these are intermediate states representing the result of a partial execution of an instruction. If an instruction $i$ might transition the system from a state $w$ of the first type to a state $v$ of the first type, then the same instruction also might transition the system into a partial completion state $(u, i)$ of the second type. If the same instruction $i$ is invoked in state $(u, i)$, then the system will finish the transition into state $v$. If an instruction $j \neq i$ is invoked in state $(u, i)$, then the system abandons the partially completed instruction $i$ and goes into a state prescribed by instruction $j$. The next two definitions formally capture this intuition. Symbol $\sqcup$ represents disjoint union.

**Definition 10.** *$S = Valid \sqcup (Valid \times I)$.*

**Definition 11.**

$$\begin{aligned} o(w) &= \begin{cases} w, & \textit{if } w \in Valid, \\ v, & \textit{if } w = (v, i), \end{cases} \\ o(v, B) &= v. \end{aligned}$$

**Lemma 22.** *$o(w) \in Valid$ for each $w \in S$.*

*Proof.* The statement of the lemma follows from Definition 10 and Definition 11. □

## 9.4 Transitions

Recall that the set of states of a canonical transition model is equal to the disjoint union $Valid \sqcup (Valid \times I)$. We refer to a state as having type one if it belongs to set $Void$ and type two if it belongs to set $Valid \times I$.

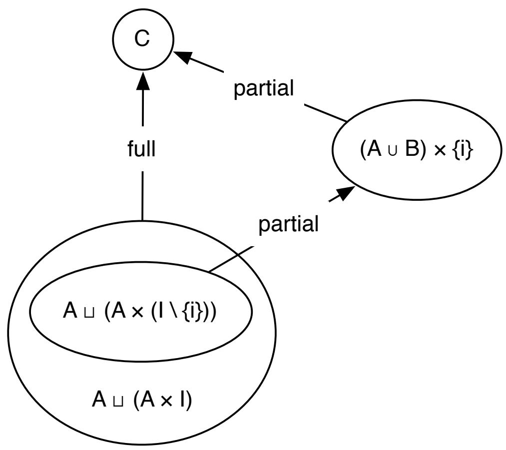


Figure 3: Transitions on instruction $(A, B, C)$

An instruction $(A, B, C)$ could be used to make one of the following transitions, see Figure 3:

1. a "full" transition from any state $w$ such that $o(w) \in A$ to any state $u \in C$ of the first type,
2. a "partial" transition from any state $w$ such that $o(w) \in A$ and state $w$ is not a partial completion for transition $(A, B, C)$, to a state $(v, (A, B, C))$ of the second type such that $v \in A \cup B$,
3. a "partial" transition from any state $w$ such that $o(w) \in A \cup B$ and state $w$ is a partial completion for transition $(A, B, C)$, to any state $u \in C$.

The next definition captures the above informal description.

**Definition 12.** *If $i = (A, B, C) \in I$, then*

$$\begin{aligned}\to_i \quad = \quad & \{(a, c) \mid a \in A \sqcup (A \times I), c \in C\} \\ & \cup \{(a, b) \mid a \in A \sqcup (A \times (I \backslash \{i\})), b \in (A \cup B) \times \{i\}\} \\ & \cup \{(b, c) \mid b \in (A \cup B) \times \{i\}, c \in C\}.\end{aligned}$$

This concludes the definition of the canonical epistemic transition system $T(X) = (S, o, I, \{\to_i\}_{i\in I})$. The next two lemmas prove basic properties of the transition relation $\to_i$. These properties are used later in the proof of the completeness.

**Lemma 23.** *For any strategy $s$, any set $E \subseteq Valid$, any sequence $\pi \in MaxPath_s(E)$, and any element $w$ of $\pi$, if $s(o(w)) = (A, B, C)$ and $o(w) \in A$, then $w$ cannot be the last element of sequence $\pi$.*

*Proof.* By Definition 11, the assumption $o(w) \in A$ implies that $w \in A \sqcup (A \times I)$. By Lemma 21, the same assumption $o(w) \in A$ implies that there is a view $c \in C$. Thus, $w \to_{(A,B,C)} c$ by Definition 12. Hence, $w \to_{s(o(w))} c$ due to the assumption $s(o(w)) = (A, B, C)$. Therefore, element $w$ cannot be the last element of sequence $\pi$ by Definition 5. □

**Lemma 24.** *For any strategy $s$, any set $E \subseteq Valid$, any sequence $\pi \in MaxPath_s(E)$, and any two consecutive elements $w$ and $w'$ of $\pi$ such that*

1. $w \in A \sqcup (A \times I)$,
2. $w' \in (A \cup B) \times \{(A, B, C)\}$,
3. $s(o(w')) = (A, B, C)$,

*the sequence $\pi$ contains an element $w''$ immediately after the element $w'$ such that $o(w'') \in C$.*

*Proof.* By Definition 11, assumption $w \in A \sqcup (A \times I)$ implies that $o(w) \in A$. Hence, by Lemma 21, set $C$ contains at least one element $c$. Note that $w' \to_{(A,B,C)} c$ by Definition 12 due to the assumption $w' \in (A \cup B) \times \{(A, B, C)\}$ of the lemma. Thus, by Definition 5, element $w'$ is not the last element of sequence $\pi$.

Let $w''$ be the element of sequence $\pi$ that immediately follows element $w'$. Hence, $w' \to_{s(o(w'))} w''$. Thus, $w' \to_{(A,B,C)} w''$ by the assumption $s(o(w')) = (A, B, C)$ of the lemma. Then, $w'' \in C$ by Definition 12 and due to the assumption $w' \in (A \cup B) \times \{(A, B, C)\}$ of the lemma. Therefore, $o(w'') \in C$ by Definition 11. □

## 9.5 Provability Implies Satisfiability for Atomic Formulae

In this section we show if an atomic proposition is provable from set $X$, then it is satisfied in the canonical epistemic transition system. The converse of this statement is shown later in Lemma 34.

**Lemma 25.** *If $X \vdash A \rhd_B C$, then $T(X) \vDash A \rhd_B C$.*

*Proof.* Let $i_0 = (A', B', C')$, where $A' = (A \backslash C) \cap Valid$, $B' = (B \backslash (A \cup C)) \cap Valid$, and $C' = C \cap Valid$. Thus, $i_0 \in I$ by Lemma 20. Define strategy $s$ to be a constant function such that $s(v) = i_0$ for each view $v \in V$. By Definition 7, it suffices to show that $MaxPath_s(A) \subseteq Until(B, C)$.

Consider any path $\pi = w_0, \dots \in MaxPath_s(A)$. By Definition 4, $o(w_0) \in A$. Note that if $o(w_0) \in C$, then $\pi \in Until(B, C)$ by Definition 6. In the rest of the proof, we assume that $o(w_0) \notin C$. Thus, $o(w_0) \in (A \backslash C) \cap Valid$ by Lemma 22. Hence, $o(w_0) \in A'$ by the choice of set $A'$. Then, by Lemma 23, sequence $\pi$ must contain at least one more element $w_1$ after element $w_0$. Then, $w_0 \to_{s(o(w_0))} w_1$ by Definition 4. Hence, $w_0 \to_{i_0} w_1$ by the choice of strategy $s$. Then, $w_0 \to_{(A',B',C')} w_1$ by the choice of instruction $i_0$. By Definition 12, statement $w_0 \to_{(A',B',C')} w_1$ implies that one of the following three cases takes place:

*Case I:* $w_0 \in A' \sqcup (A' \times I)$ and $w_1 \in C'$. Thus, $o(w_0) \in A'$ and $o(w_1) \in C'$ by Definition 11. Hence, $o(w_0) \in (A \backslash C) \cap Valid$ and $o(w_1) \in C \cap Valid$ due to the choice of sets $A'$ and $C'$. Thus, $o(w_0) \in B$ by Lemma 18 and also $o(w_1) \in C$. Therefore, $\pi \in Until(B, C)$ by Definition 6.

*Case II:* $w_0 \in A' \sqcup (A' \times (I \backslash \{i_0\}))$ and $w_1 \in (A' \cup B') \times \{i_0\}$. Thus, $o(w_0) \in A'$ and $o(w_1) \in A' \cup B'$ by Definition 11. Hence, $o(w_0) \in (A \backslash C) \cap Valid$ and $o(w_1) \in ((A \backslash C) \cap Valid) \cup B'$ by the choice of set $A'$. Hence, $o(w_0) \in B$ and $o(w_1) \in B \cup B'$ by Lemma Lemma 18. Thus, $o(w_0), o(w_1) \in B$ by the choice of set $B'$.

Recall that $w_0 \in A' \sqcup (A' \times (I \backslash \{i_0\}))$, $w_1 \in (A' \cup B') \times \{i_0\}$, and $i_0 = (A', B', C')$. Thus, by Lemma 24, sequence $\pi$ must contain an element $w_2$ immediately after the element $w_1$ such that $o(w_2) \in C'$. Hence $o(w_2) \in C$ by the choice of set $C'$. Thus, we have $o(w_0), o(w_1) \in B$ and $o(w_2) \in C$. Therefore, $\pi \in Until(B, C)$ by Definition 6.

*Case III:* $w_0 \in (A' \cup B') \times \{i_0\}$ and $w_1 \in C'$. Thus, $o(w_0) \in A' \cup B'$ and $o(w_1) \in C'$ by Definition 11. Hence, $o(w_0) \in (((A \backslash C) \cap Valid)) \cup B'$ and $o(w_1) \in C$ by choice of sets $A'$ and $C'$. Thus, $o(w_0) \in B \cup B'$ by Lemma 18 and also $o(w_1) \in C$. Hence, $o(w_0) \in B$ by the choice of set $B'$. Therefore, $\pi \in Until(B, C)$ by Definition 6. □

### 9.6 Satisfiability Implies Provability for Atomic Formulae

The goal of this section is to show the converse of Lemma 25. This result is stated later in the section as Lemma 34. To prove the result, due to Definition 7, it suffices to show that $X \nvdash E \rhd_F G$ implies that $MaxPath_s(E) \nsubseteq Until(F, G)$ for each strategy $s$. In other words, we need to show that for any strategy $s$ there is a path $\pi \in MaxPath_s(E)$ that either never comes to $G$ or leaves $F$ before coming to $G$. To construct this path, we first define $G_*$ as a set of all starting states from which paths under strategy $s$ unavoidably lead to set $G$ never leaving set $F$. According to Definition 14, set $G_*$ is a union of an infinite chain of sets $G = G_0 \subseteq G_1 \subseteq G_2 \dots$. Sets $\{G_i\}_{i \geqslant 0}$ are defined recursively below. The same definition also specifies the auxiliary families of sets $\{H_i\}_{i \geqslant 0}$, $\{A_i\}_{i \geqslant 1}$, $\{B_i\}_{i \geqslant 1}$, $\{C_i\}_{i \geqslant 1}$, $\{A_i^+\}_{i \geqslant 1}$, and $\{B_i^+\}_{i \geqslant 1}$ that will be used to state and prove various properties of family $\{G_i\}_{i \geqslant 0}$.

**Definition 13.** *For any sets $F, G \subseteq V$ and any strategy $s$, let*

1. *$G_0 = G$ and $H_0 = \varnothing$,*

2. *choose any instruction* $(A_n, B_n, C_n) \in I$ *such that*

   (a) $A_n \cup B_n \subseteq F \cup G$,

   (b) $\{a \in A_n \mid s(a) = (A_n, B_n, C_n)\} \backslash G_{n-1}$ *is not empty,*

   (c) $\{a \in A_n \mid s(a) \neq (A_n, B_n, C_n)\} \subseteq G_{n-1}$,

   (d) $\{b \in B_n \mid s(b) \neq (A_n, B_n, C_n)\} \subseteq G_{n-1}$,

   (e) $C_n \subseteq G_{n-1}$

   *and define*

   (i) $A_n^+ = \{a \in A_n \mid s(a) = (A_n, B_n, C_n)\}$,

   (ii) $B_n^+ = \{b \in B_n \mid s(b) = (A_n, B_n, C_n)\}$,

   (iii) $G_n = A_n^+ \cup G_{n-1}$,

   (iv) $H_n = B_n^+ \cup H_{n-1}$,

3. *if no instruction* $(A_n, B_n, C_n) \in I$ *satisfying condition 2a,2b,2c,2d, and 2e exists, then stop.*

Next we state and prove properties of the families of the sets specified in Definition 13.

**Lemma 26.** $A_n \neq A_m$ *for each* $n > m$.

*Proof.* By item (2b) of Definition 13, there must exist a view $a_0 \in \{a \in A_n \mid s(a) = (A_n, B_n, C_n)\}$ such that $a_0 \notin G_{n-1}$. Thus, $a_0 \in A_n^+ \backslash G_{n-1}$ by item (2i) of Definition 13.

Suppose that $A_n = A_m$. Hence, $A_n^+ = A_m^+$ by item (2i) of Definition 13. Notice also that $G_{n-1} \supseteq G_m$ by item (2iii) of Definition 13 and the assumption $n > m$. Then, $a_0 \in A_n^+ \backslash G_{n-1} = A_m^+ \backslash G_{n-1} \subseteq A_m^+ \backslash G_m$. Hence, $A_m^+ \nsubseteq G_m$, which contradicts item (2iii) of Definition 13. □

**Lemma 27.** *Sets* $A_n^+$ *and* $H_{n-1}$ *are disjoint.*

*Proof.* Suppose that there is a view $v$ such that $v \in A_n^+$ and $v \in H_{n-1}$. Hence, by items (1) and (2iv) of Definition 13, there must exist $m < n$ such that $v \in B_m^+$. Thus, $s(v) = (A_n, B_n, C_n)$ and $s(v) = (A_m, B_m, C_m)$ by items (2i) and (2ii) of Definition 13, which contradicts Lemma 26. □

**Lemma 28.** *Sets* $B_n^+$ *and* $H_{n-1}$ *are disjoint.*

*Proof.* Suppose that there is a view $v$ such that $v \in B_n^+$ and $v \in H_{n-1}$. Hence, by items (1) and (2iv) of Definition 13, there must exist $m < n$ such that $v \in B_m^+$. Thus, $s(v) = (A_n, B_n, C_n)$ and $s(v) = (A_m, B_m, C_m)$ by item (2ii) of Definition 13, which contradicts Lemma 26. □

**Lemma 29.** $(A_n^+ \cup B_n^+) \cap (G_{n-1} \cup H_{n-1}) \subseteq G_{n-1}$, *for each* $n \geqslant 1$.

By Lemma 27 and Lemma 28,

$$
\begin{aligned}
&(A_n^+ \cup B_n^+) \cap (G_{n-1} \cup H_{n-1}) \\
&= ((A_n^+ \cup B_n^+) \cap G_{n-1}) \cup ((A_n^+ \cup B_n^+) \cap H_{n-1}) \\
&= ((A_n^+ \cup B_n^+) \cap G_{n-1}) \cup (A_n^+ \cap H_{n-1}) \cup (B_n^+ \cap H_{n-1}) \\
&\subseteq G_{n-1} \cup \varnothing \cup \varnothing = G_{n-1}.
\end{aligned}
$$

**Lemma 30.** *$X \vdash G_n \rhd_{G_n \cup B_n^+} G_{n-1}$, for each $n \geqslant 1$.*

*Proof.* Note that $C_n \subseteq G_{n-1}$ by item (2e) of Definition 13. Thus, $\vdash C_n \rhd_{\varnothing} G_{n-1}$ by the Reflexivity axiom. Also, $X \vdash A_n \rhd_{A_n \cup B_n} C_n$ by Definition 9. Thus, $X \vdash A_n \rhd_{A_n \cup B_n} G_{n-1}$ by the Transitivity axiom. Hence,

$$X \vdash A_n \rhd_{(A_n \setminus A_n^+) \cup A_n^+ \cup (B_n \setminus B_n^+) \cup B_n^+} G_{n-1}$$

because $A_n^+ \subseteq A_n$ and $B_n^+ \subseteq B_n$ by item (2i) and item (2ii) of Definition 13. Note that $A_n \setminus A_n^+ \subseteq G_{n-1}$ and $B_n \setminus B_n^+ \subseteq G_{n-1}$ by items (2c), (2d), (2i) and (2ii) of Definition 13. Thus, by Lemma 3,

$$X \vdash A_n \rhd_{A_n^+ \cup G_{n-1} \cup B_n^+} G_{n-1}.$$

Hence, by Lemma 2 and due to item (2i) of Definition 13,

$$X \vdash A_n^+ \rhd_{A_n^+ \cup G_{n-1} \cup B_n^+} G_{n-1}.$$

Thus, $X \vdash A_n^+ \cup G_{n-1} \rhd_{A_n^+ \cup G_{n-1} \cup B_n^+} G_{n-1}$ by the Augmentation axiom. Then, $X \vdash G_n \rhd_{G_n \cup B_n^+} G_{n-1}$ by by item (2iii) of Definition 13. □

**Lemma 31.** *$X \vdash G_n \rhd_{G_n \cup H_n} G$*

*Proof.* We prove this statement by induction on $n$. If $n = 0$, then $G_n = G$ by item (1) of Definition 13. Therefore, $\vdash G_n \rhd_{G_n \cup H_n} G$ by the Reflexivity axiom.

Suppose that $n > 0$. Thus, $X \vdash G_n \rhd_{G_n \cup B_n^+} G_{n-1}$ by Lemma 30. Thus, $X \vdash G_n \rhd_{(G_n \setminus G_{n-1}) \cup (B_n^+ \setminus G_{n-1})} G_{n-1}$ by the Early Bird axiom. Thus, $X \vdash G_n \rhd_{A_n^+ \cup (B_n^+ \setminus G_{n-1})} G_{n-1}$, by by item (2iii) of Definition 13. Hence, $X \vdash G_n \rhd_{A_n^+ \cup B_n^+} G_{n-1}$ by Lemma 3. At the same time, $X \vdash G_{n-1} \rhd_{G_{n-1} \cup H_{n-1}} G$ by the induction hypothesis. Thus, $X \vdash G_n \rhd_{A_n^+ \cup B_n^+ \cup G_{n-1} \cup H_{n-1}} G$ by Lemma 6 taking into account Lemma 29. Therefore, $X \vdash G_n \rhd_{G_n \cup H_n} G$ by items (2iii) and (2iv) of Definition 13. □

**Definition 14.** $G_* = \bigcup_n G_n$.

**Lemma 32.** *There is $n \geqslant 0$ such that $G_* = G_n$.*

*Proof.* By Definition 13 and Definition 14, we have $G_0 \subseteq G_1 \subseteq G_2 \cdots \subseteq G_* \subseteq V$. Thus, the statement of the lemma follows from the assumption in Section 4 that set $V$ is finite. □

**Lemma 33.** *$G_n \cup H_n \subseteq F \cup G$.*

*Proof.* Consider any $n \geqslant 0$. Note that $A_n \cup B_n \subseteq F \cup G$ by line (2a) of Definition 13. Thus, $A_n^+ \cup B_n^+ \subseteq F \cup G$ by line (2i) and line (2ii) of Definition 13. Hence, $A_n^+ \cup B_n^+ \subseteq F \cup G$ for all $n \geqslant 0$. Then, $G_n \cup H_n \subseteq F \cup G$ for all $n \geqslant 0$ by by line (2a), line (2iii), and line (2iv) of Definition 13. □

We are now ready to state and prove the main lemma of this section. The statement of this lemma is the contrapositive of Lemma 25.

**Lemma 34.** *If $T(X) \vDash E \rhd_F G$, then $X \vdash E \rhd_F G$.*

*Proof.* Suppose that $T(X) \vDash E \rhd_F G$. Thus, by Definition 7, there is a strategy $s$ such that $MaxPath_s(E) \subseteq Until(F,G)$.

Consider chain of sets $G_0 \subseteq G_1 \subseteq G_2 \subseteq \dots$ and set $G_*$, as specified in Definition 13 and Definition 14, constructed based on sets $F$ and $G$ as well as strategy $s$. We consider the following two cases separately:
*Case I:* $E \cap Valid \subseteq G_*$. Thus, by the Reflexivity axiom

$$\vdash (E \cap Valid) \rhd_\varnothing G_*. \tag{6}$$

At the same time, $X \vdash (E \backslash Valid) \rhd_\varnothing \varnothing$ by Lemma 15. Hence, $X \vdash (E\backslash Valid) \cup (E \cap Valid) \rhd_\varnothing (E \cap Valid)$ by the Augmentation axiom. In other words, $X \vdash E \rhd_\varnothing (E \cap Valid)$. This, together with statement (6) by the Transitivity axiom implies that $X \vdash E \rhd_\varnothing G_*$. Thus, by Lemma 32, there is $n \geqslant 0$ such that $X \vdash E \rhd_\varnothing G_n$. Hence, $X \vdash E \rhd_{G_n \cup H_n} G$ by Lemma 31 and the Transitivity axiom. Hence, $X \vdash E \rhd_{F \cup G} G$ by Lemma 33 and Lemma 3. Thus, $X \vdash E \rhd_{(F\cup G)\backslash G} G$ by the Early Bird axiom. Note that $(F \cup G)\backslash G \subseteq F$. Therefore, $X \vdash E \rhd_F G$ by Lemma 3.
*Case II:* there is $e \in (E \cap Valid)\backslash G_*$. Let

$$W = (V\backslash G_*) \cup \{(w,i) \in (V\backslash G_*) \times I \mid s(o(w)) \neq i\}.$$

Let $\pi$ be a maximal (either finite or infinite) sequence $w_0, w_1, \dots$ of elements from set $W$ such that

1. $w_0 = e$,
2. $w_i \to_{s(o(w_i))} w_{i+1}$ for all $i \geqslant 0$.

**Claim 1.** *Sequence $\pi$ is finite.*

*Proof of Claim.* If sequence $\pi$ is infinite then $\pi \in MaxPath_s(E)$ by Definition 5, $o(w_0) = o(e) \in (E \cap Valid)\backslash G_* \subseteq E$, and item (2) above. At the same time $\pi \notin Until(F,G)$ by Definition 6 because $o(w_i) \in o(W) \subseteq V\backslash G_* \subseteq V\backslash G_0 = V\backslash G$ for each $i \geqslant 0$. Thus, $MaxPath_s(E) \nsubseteq Until(F,G)$, which is a contradiction with the choice of strategy $s$. □

Let $w_k$ be the last element of sequence $\pi$ and $(A,B,C) = s(o(w_k))$. By $pr_1$ and $pr_2$ we mean the first and the second projection of a pair.

**Claim 2.** *If $w_k \in V \times I$, then $pr_2(w_k) \neq (A,B,C)$.*

*Proof of Claim.* Suppose that $pr_2(w) = (A, B, C)$. Then, by the choice of instruction $(A, B, C)$, we have $pr_2(w_k) = s(o(w_k))$. Thus, $w_k \notin W$ by the choice of set $W$, which is a contradiction with the choice of sequence $\pi$. □

**Claim 3.** $o(w_k) \in A$.

*Proof of Claim.* Suppose $o(w_k) \notin A$. First we show $\pi \in MaxPath_s(E)$. Assume $\pi \notin MaxPath_s(E)$. Thus, by Definition 5 and because $o(w_0) \in E$, there must exist state $w_{k+1} \in S$ such that $w_k \rightarrow_{s(o(w_k))} w_{k+1}$. Hence, $w_k \rightarrow_{(A,B,C)} w_{k+1}$ by the choice of the instruction $(A, B, C)$. Thus, by Definition 12, the assumption $o(w_k) \notin A$ implies that $w_k \in (A \cup B) \times \{(A, B, C)\}$, which is a contradiction with Claim 2. Therefore, $\pi \in MaxPath_s(E)$.

Recall that $MaxPath_s(E) \subseteq Until(F, G)$ by the choice of strategy $s$. Hence, $\pi \in Until(F, G)$. Thus, by Definition 6, there is $m \geqslant 0$ such that $o(w_m) \in G$. Hence, $o(w_m) \in G_0$ by Definition 13. Thus, $o(w_m) \in G_*$ by Definition 14. Therefore, $w_m \notin W$ by the choice of $W$ and Definition 11, which is a contradiction with the choice of sequence $\pi$. □

**Claim 4.** $C \subseteq G_*$.

*Proof of Claim.* Suppose that there is $c \in C$ such that $c \notin G_*$. Note that $o(w_k) \in A$ by Claim 3. Thus, $w_k \rightarrow_{(A,B,C)} c$ by Definition 12 and the assumption $c \in C$. At the same time, the assumption $c \notin G_*$ implies $c \in V\backslash G_*$. Which implies that $c \in W$ by the choice of set $W$. Hence, sequence $\pi$ can be extended by at least one more element, namely by state $c$, which is a contradiction with the choice of sequence $\pi$. □

**Claim 5.** $A \cup B \subseteq F \cup G$.

*Proof of Claim.* Suppose that there is $x \in (A \cup B)\backslash(F \cup G)$. Recall that $o(w_k) \in A$. Thus, $w_k \in A \sqcup (A \times (I\backslash\{(A, B, C)\}))$ by Definition 11 and Claim 2. Thus, $w_k \rightarrow_{(A,B,C)} (x, (A, B, C))$ by Definition 12 and because $x \in A \cup B$. Let $\pi' = \pi, (x, (A, B, C))$. In other words, $\pi'$ is the extension of sequence $\pi$ by an additional element $(x, (A, B, C))$. Note that $\pi' \in Path_s(E)$ by the choice of sequence $\pi$ and because $w_k \rightarrow_{(A,B,C)} (x, (A, B, C))$. By Lemma 1, sequence $\pi'$ can be extended to a sequence $\pi'' \in MaxPath_s(E)$. Thus, $\pi'' \in Until(F, G)$ by the choice of strategy $s$.

At the same time, $w_1, \ldots, w_k \in W$ by the choice of sequence $\pi$. Thus, we have $o(w_1), \ldots, o(w_k) \notin G_*$ by the choice of set $W$. Then, by Definition 14, $o(w_1), \ldots, o(w_k) \notin G_0$. Hence, $o(w_1), \ldots, o(w_k) \notin G$ by Definition 13. Recall that $x \in (A \cup B)\backslash(F \cup G)$. Thus, $o(x, (A, B, C)) \notin F \cup G$. Then, $o(w_1), \ldots, o(w_k), o(x, (A, B, C)) \notin G$ and $o(x, (A, B, C)) \notin F$. Therefore, $\pi'' \notin Until(F, G)$ by Definition 6, which is a contradiction with the above observation $\pi'' \in Until(F, G)$. □

**Claim 6.** $\{x \in A \cup B \mid s(x) \neq (A, B, C)\} \subseteq G_*$.

*Proof of Claim.* Suppose that there is $x \in A \cup B$ such that $s(x) \neq (A, B, C)$ and $x \notin G_*$. Recall that $o(w_k) \in A$. Thus, $w_k \in A \sqcup (A \times (I\backslash\{(A, B, C)\}))$ by

Definition 11 and Claim 2. Thus, $w_k \to_{(A,B,C)} (x, (A, B, C))$ by Definition 12 and because $x \in A \cup B$.

At the same time, $o(x, (A, B, C)) = x \notin G_*$ by Definition 11 and the assumption $x \notin G_*$. Hence, $(x, (A, B, C)) \in W$ by the choice of $W$ and the assumption $s(x) \neq (A, B, C)$.

Therefore, sequence $\pi$ can be extended by at least one more element, namely by state $(x, (A, B, C))$, which is a contradiction with the choice of sequence $\pi$. □

We are now ready to finish the proof of the lemma. Note that set $G_n \backslash G_{n-1}$ is not empty for each $n \geqslant 0$ by item (2b) of Definition 13. Thus, the recursive construction of chain $G_0 \subseteq G_1 \subseteq G_2 \ldots$, as given in Definition 13, must terminate due to set $V$ being finite. Suppose that the last element of the chain $G_0 \subseteq G_1 \subseteq G_2 \ldots$ is set $G_{k-1}$. To come to a contradiction, it suffices to show that at least one more set can be added to the chain $G_0 \subseteq G_1 \subseteq G_2 \ldots$ by choosing instruction $(A_n, B_n, C_n)$ to be $(A, B, C)$. To prove the latter, we need to show that instruction $(A, B, C)$ satisfies conditions (2a) through (2e) of Definition 13. Indeed, condition (2a) is satisfied by Claim 5. Condition (2b) is satisfied because $s(o(w_k)) \in A$ by Claim 3 and $o(w_k) \notin G_{k-1} = G_*$ because $w_k \in W$ by Claim 2. Conditions (2c) and (2d) are satisfied by Claim 6. Finally, condition (2e) is satisfied by Claim 4. This concludes the proof of the lemma. □

## 9.7 Completeness: The Final Steps

In this section we use Lemma 25 and Lemma 34 to finish the proof of the completeness theorem. The completeness theorem itself is stated below as Theorem 2.

**Lemma 35.** *$T(X) \vDash \varphi$ if and only if $\varphi \in X$.*

*Proof.* Induction on the structural complexity of formula $\varphi$. In the base case the statement of the lemma follows from Lemma 25 and Lemma 34. The induction step follows from the maximality and the consistency of set $X$ in the standard way. □

**Theorem 2.** *If $T \vDash \varphi$ for every epistemic transition system $T$, then $\vdash \varphi$.*

*Proof.* Suppose $\nvdash \varphi$. Let $X$ be a maximal consistent set containing formula $\neg\varphi$. Thus, $T(X) \vDash \neg\varphi$ by Lemma 35. Therefore, $T(X) \nvDash \varphi$. □

# 10 Conclusion

In our previous work [11], we have shown that navigability by perfect recall strategies is transitive and navigability by memoryless strategies is not. In this article we observe that the situation is different if one considers a more general notion of navigability with intermediate constraints. Namely, in this new setting certain form of transitivity holds even for memoryless strategies. The main technical contribution of the article is a sound and complete axiomatization of all properties of navigability by memoryless strategies with intermediate constraints.

In this article and in [11], we compare the properties of the navigability by perfect recall and memoryless strategies. An interesting possible direction for future work could be to consider navigability by agents with restricted memory. Such as, for example, navigability by finite state machines of a given size.

# References


[1] Thomas Ågotnes and Natasha Alechina. Epistemic coalition logic: completeness and complexity. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2 (AAMAS)*, pages 1099–1106, 2012.

[2] Thomas Ågotnes and Natasha Alechina. Coalition logic with individual, distributed and common knowledge. *Journal of Logic and Computation*, 2016. exv085.

[3] Thomas Ågotnes, Philippe Balbiani, Hans van Ditmarsch, and Pablo Seban. Group announcement logic. *Journal of Applied Logic*, 8(1):62 – 81, 2010.

[4] Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge. Reasoning about coalitional games. *Artificial Intelligence*, 173(1):45 – 79, 2009.

[5] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.

[6] W. W. Armstrong. Dependency structures of data base relationships. In *Information Processing 74 (Proc. IFIP Congress, Stockholm, 1974)*, pages 580–583. North-Holland, Amsterdam, 1974.

[7] Francesco Belardinelli. Reasoning about knowledge and strategies: Epistemic strategy logic. In *Proceedings 2nd International Workshop on Strategic Reasoning, SR 2014, Grenoble, France, April 5-6, 2014*, volume 146 of *EPTCS*, pages 27–33, 2014.

[8] Nuel Belnap and Michael Perloff. Seeing to it that: A canonical form for agentives. In *Knowledge representation and defeasible reasoning*, pages 167–190. Springer, 1990.

[9] Stefano Borgo. Coalitions in action logic. In *20th International Joint Conference on Artificial Intelligence*, pages 1822–1827, 2007.

[10] Jan Broersen, Andreas Herzig, and Nicolas Troquard. A normal simulation of coalition logic and an epistemic extension. In *Proceedings of the 11th conference on Theoretical aspects of rationality and knowledge*, pages 92–101. ACM, 2007.

[11] Kaya Deuser and Pavel Naumov. Armstrong's axioms and navigation strategies. In *Proceedings of Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[12] Raul Fervari, Andreas Herzig, Yanjun Li, and Yanjing Wang. Strategically knowing how. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1031–1038, 2017.

[13] Hector Garcia-Molina, Jeffrey Ullman, and Jennifer Widom. *Database Systems: The Complete Book*. Prentice-Hall, second edition, 2009.

[14] Valentin Goranko. Coalition games and alternating temporal logics. In *Proceedings of the 8th conference on Theoretical aspects of rationality and knowledge*, pages 259–272. Morgan Kaufmann Publishers Inc., 2001.

[15] Valentin Goranko and Govert van Drimmelen. Complete axiomatization and decidability of alternating-time temporal logic. *Theoretical Computer Science*, 353(1):93 – 117, 2006.

[16] John Horty and Eric Pacuit. Action types in stit semantics. *The Review of Symbolic Logic*, pages 1–21, 2017.

[17] John F Horty. *Agency and deontic logic*. Oxford University Press, 2001.

[18] John F Horty and Nuel Belnap. The deliberative stit: A study of action, omission, ability, and obligation. *Journal of philosophical logic*, 24(6):583–644, 1995.

[19] Wojciech Jamroga and Thomas Ågotnes. Constructive knowledge: what agents can achieve under imperfect information. *Journal of Applied Non-Classical Logics*, 17(4):423–475, 2007.

[20] Wojciech Jamroga and Wiebe van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 63(2-3):185–219, 2004.

[21] Yanjun Li and Yanjing Wang. Achieving while maintaining: A logic of knowing how with intermediate constraints. In *Logic and Its Applications: Proceedings of 7th Indian Conference, ICLA 2017, Kanpur, India, January 5-7, 2017*, pages 154–167. Springer, 2017.

[22] Sara Miner More and Pavel Naumov. Calculus of cooperation and game-based reasoning about protocol privacy. *ACM Trans. Comput. Logic*, 13(3):22:1–22:21, August 2012.

[23] Pavel Naumov and Jia Tao. Coalition power in epistemic transition systems. In *Proceedings of the 2017 International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 723–731, 2017.

[24] Pavel Naumov and Jia Tao. Together we know how to achieve: An epistemic logic of know-how. In *16th conference on Theoretical Aspects of Rationality and Knowledge (TARK), July 24-26, 2017, EPTCS 251*, pages 441–453, 2017.

[25] Pavel Naumov and Jia Tao. Second-order know-how strategies. In *Proceedings of the 2018 International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2018. (to appear).

[26] Pavel Naumov and Jia Tao. Strategic coalitions with perfect recall. In *Proceedings of Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[27] Grigory K Olkhovikov and Heinrich Wansing. Inference as doxastic agency. part i: The basics of justification stit logic. *Studia Logica*, pages 1–28, 2016.

[28] Marc Pauly. *Logic for Social Software*. PhD thesis, Institute for Logic, Language, and Computation, 2001.

[29] Marc Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.

[30] Luigi Sauro, Jelle Gerbrandy, Wiebe van der Hoek, and Michael Wooldridge. Reasoning about action and cooperation. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '06, pages 185–192, New York, NY, USA, 2006. ACM.

[31] Johan van Benthem. Games in dynamic-epistemic logic. *Bulletin of Economic Research*, 53(4):219–248, 2001.

[32] Wiebe van der Hoek and Michael Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75(1):125–157, 2003.

[33] Wiebe van der Hoek and Michael Wooldridge. On the logic of cooperation and propositional control. *Artificial Intelligence*, 164(1):81 – 119, 2005.

[34] Yanjing Wang. A logic of goal-directed knowing how. *Synthese*, Nov 2016.