Armstrong’s Axioms and Navigation Strategies

Kaya Deuser and Pavel Naumov
Vassar College
Poughkeepsie, New York

July 14, 2017

Abstract

The paper investigates navigability with imperfect information. It shows that the properties of navigability with perfect recall are exactly those captured by Armstrong’s axioms from the database theory. If the assumption of perfect recall is omitted, then Armstrong’s transitivity axiom is not valid, but it can be replaced by two new weaker principles. The main technical results are soundness and completeness theorems for the logical systems describing properties of navigability with and without perfect recall.

1 Introduction

Navigation is a commonly encountered task by autonomous agents that need to reach a destination or, more generally, to find a solution to a problem, where the solution is a sequence of instructions that transition a system from one state to another. This task is often performed when the agent does not have precise information about her current location. Examples of such agents are self-navigating missiles, self-driving cars, and robotic vacuum cleaners.

Figure 1 depicts an example $T_0$ of a transition system. This system consists of eight states $a, \ldots, h$ represented by the vertices of the graph. The agent cannot distinguish state $a$ from state $b$ and state $c$ from state $d$, which is denoted by dashed lines connecting the indistinguishable states. The directed edges of the graph represent transitions that the system can make and the labels on these edges represent the instructions that the agent must give to do this. For example, if in state $a$ the agent executes instruction 0, then the system transitions into state $g$, if, instead, the agent executes instruction 1, the system transitions into state $e$. Although in this paper we consider non-deterministic transition systems where the execution of the same instruction can transition the system into one of the several states, for the sake of simplicity the transition system $T_0$ is deterministic.

Note that, in system $T_0$ the agent can navigate from state $a$ to state $c$ by using instruction 1 in state $a$ and the same instruction 1 again in state $e$. However, a different sequence of instructions is required to reach state $c$ from state $b$. As the agent cannot distinguish state $a$ from state $b$, in state $a$ she does not know which instructions to use...
to accomplish her goal. Moreover, if the agent does reach state $c$, she cannot verify that the task is completed, because she cannot distinguish state $c$ from state $d$. For this reason, in this paper instead of navigation between states, we consider navigation between equivalence classes of states with respect to the indistinguishability relation. For example, the agent can navigate from class $[a] = \{a, b\}$ to class $[c] = \{c, d\}$ by using instruction 1 in each state it passes.

**Perfect Recall**

In order to achieve a goal, the agent would need to follow a certain strategy that must be stored in her memory. We assume that the strategy is permanently stored (“hardwired”) in the memory and cannot be changed during the navigation. For example, a robotic vacuum cleaner might be programmed to change direction when it encounters a wall, to make a circle when the dirt sensor is triggered, and to follow straight path otherwise. A crucial question for us is if the vacuum cleaner can remember the walls and the dirty spots it has previously encountered. In other words, we distinguish an agent that can keep track of the classes of states she visited and the instructions she used from an agent who only knows her current state. We say that in the former case the agent has perfect recall and in the later she does not.

A strategy of an agent without perfect recall can only use information available to her about the current state to decide which instruction to use. In other words, a strategy of such an agent is a function that maps classes of indistinguishable states into instructions. A strategy of an agent with perfect recall can use information about the history of previous transitions to decide which instruction to use. In other words, a strategy of such an agent is a function that maps classes of indistinguishable histories into instructions. We call the former amnesic strategies and the later recall strategies.

In theory, a robotic vacuum cleaner without perfect recall is only equipped with read-only memory to store the strategy. A theoretical robotic vacuum with perfect recall in addition to read-only memory that contains the strategy also has an unlimited read-write memory that contains logs of all previous transitions. In practice, the most popular brand of robotic vacuum cleaners, Roomba, is only using read-write memory to store information, such as a cleaning schedule, that is not used for navigation. This means that Roomba is using an amnesic strategy. The other popular robotic vacuum cleaner, Neato, is scanning the room before cleaning to use this information for navigation purposes. Although, of course, Neato only has a read-write memory of a limited
size, the navigation strategy used by Neato is an example of a recall strategy.

**Examples**

Transition system \( T_0 \) has a recall strategy to navigate from class \([a]\) to class \([e]\). This might be surprising, because the same strategy must work to navigate from both state \(a\) and state \(b\) of class \([a]\) to the only state \(e\) of class \([e]\). Yet, one would expect to use instruction 1 in state \(a\) and instruction 0 in state \(b\) to get to \(e\). The recall strategy to navigate from class \([a]\) to class \([e]\) consists in first using instruction 0 once no matter what the starting state is, and then using instruction 1 until state \(e\) is reached. When this strategy is used starting from a state in class \([a]\), the system first transitions into state \(g\), then into state \(a\), and then finally into state \(e\). This is a recall strategy because it uses a different instruction in state \(a\) depending if the system has already visited a state in class \([g]\) or not.

To show that there is no amnesic strategy to navigate from class \([a]\) to class \([e]\) note that any such strategy will have to use the same instruction \(i_0\) at every visit to states \(a\) and \(b\). If \(i_0 = 0\), then when the navigation starts in state \(b\), the system stays “locked” in set of states \([a,g,b]\) and never reaches state \(e\). Similarly, if \(i_0 = 1\), then when the navigation starts in state \(b\), the system stays “locked” in set of states \([b,f,d,h]\) and never reaches state \(e\). Thus, there is no amnesic strategy to navigate from class \([a]\) to class \([e]\).

In some situations, even when there is a path between appropriate states, there might be no amnesic strategy and no recall strategy. For example, transition system \( T_0 \) has no recall strategy to navigate from class \([c]\) to class \([g]\). Indeed, if such a recall strategy exists, it would have to use the same instruction \(i_0\) when the system starts in either state \(c\) or state \(d\). Suppose \(i_0 = 0\). Thus, if the system \(starts\) in state \(c\) it is “locked” in state \(h\). Assume now that \(i_0 = 1\). Hence, if the system \(starts\) in state \(d\) it is “locked” in state \(h\). Therefore, there is no recall strategy from class \([c]\) to class \([g]\).

| \(\{a,b\}\) | \(\{c,d\}\) | \(\{e\}\) | \(\{f\}\) | \(\{g\}\) | \(\{h\}\) |
|---|---|---|---|---|---|
| a | a | r | r | a | r† |
| c | d |  | a | r† |  |
| e |  | a | a | r† | a | a |
| f |  | a | a | r† | a | a |
| g |  | a | a | a | a | a† |
| h |  |  |  |  | a |   |

Table 1: Navigability between classes in system \(T_0\).

Table 1 shows when amnesic strategies and recall strategies between classes of transition system \(T_0\) exist. In this table, letter “a” denotes the existence of an amnesic strategy, letter “r” denotes the existence of a recall strategy, but no amnesic strategy, and dash “-” denotes that neither amnesic strategy nor recall strategy exist. Symbol † marks the cases that we have found to be interesting to think about.

**Navigability between Sets**

As we have seen, there is a recall strategy, but no amnesic strategy, to navigate from class \([a]\) to class \([e]\). One can similarly show that there is no amnesic strategy to
navigate from class \([a]\) to class \([f]\). However, if the goal is to navigate from class \([a]\) to either class \([e]\) or to class \([f]\), then there is an amnesic strategy to do this. Indeed, consider an amnesic strategy that uses instruction 1 in every state. This strategy can transition the system from a state of class \([a]\) to a state of a class in set \(\{[e], [f]\}\).

Thus, navigability between \(\textit{sets of classes}\) cannot be reduced to navigability between \(\textit{classes}\). For this reason, in this paper we study properties of navigability between sets of classes. If there is a strategy to navigate from a set of classes \(A\) to a set of classes \(B\), then we write \(A \triangleright B\). It will be clear from the context if we refer to the existence of an amnesic strategy or a recall strategy.

### Universal Properties of Navigability

In the examples above we talked about properties of navigability for the transition system \(T_0\). In the rest of this paper we study universal properties of navigability between sets of classes that are true in all transition systems. An example of such a property is \(\textit{reflexivity}: A \triangleright B\), where \(A \subseteq B\). This property is true for both amnesic and recall strategies because absolutely \(\textit{any}\) strategy can be used to navigate from a subset to the whole set. In fact, in this case the goal is achieved before the navigation even starts.

Another example of a property of navigation which is universally true for both amnesic and recall strategies is \(\textit{augmentation}: A \triangleright B \rightarrow (A \cup C) \triangleright (B \cup C)\). It says that if there is a strategy to navigate from set \(A\) to set \(B\), then there is a strategy to navigate from set \(A \cup C\) to set \(B \cup C\).

An example of a property which is universally true for recall strategies, but is not universally true for amnesic strategies is \(\textit{transitivity}: A \triangleright B \rightarrow (B \triangleright C \rightarrow A \triangleright C)\). It states that if there is a strategy to navigate from set \(A\) to set \(B\) and a strategy to navigate from set \(B\) to set \(C\), then there is a strategy to navigate from set \(A\) to set \(C\).

We prove this property for recall strategies in Lemma \ref{trans}. To see that this property is not universally true for amnesic strategies, note that, in transition system \(T_0\), amnesic strategy that always uses instruction 0 can be used to navigate from set \(\{[a]\}\) to set \(\{[g]\}\) and amnesic strategy that always uses instruction 1 can be used to navigate from set \(\{[g]\}\) to set \(\{[e]\}\). At the same time, as we have shown earlier, there is no amnesic strategy to navigate from set \(\{[a]\}\) to set \(\{[e]\}\).

In this paper we show that reflexivity, augmentation, and transitivity principles form a sound and complete logical system that describes all universal properties of navigability by recall strategies. These are the three principles known in the database theory as Armstrong’s axioms (Garcia-Molina et al., 2009, p. 81), where they give a sound and complete axiomatization of functional dependency. We also give a sound and complete axiomatization of universal properties of navigability by amnesic strategies. It consists of the reflexivity and augmentation principles mentioned above as well as the \(\textit{left monotonicity}\) principle \((A \cup C) \triangleright B \rightarrow A \triangleright B\) and the \(\textit{right monotonicity}\) principle \(A \triangleright B \rightarrow A \triangleright (B \cup C)\). The latter two principles are true for the recall strategies as well, but they are provable from Armstrong’s axioms.
Literature Review

Most of the existing literature on logical systems for reasoning about strategies is focused on modal logics. Logics of coalition power were developed by Pauly (2001, 2002), who also proved the completeness of the basic logic of coalition power. Pauly’s approach has been widely studied in literature Goranko (2001); van der Hoek and Wooldridge (2005); Borgo (2007); Sauro et al. (2006); Ågotnes et al. (2010, 2009); Belardinelli (2014). Alternative logical systems were proposed by More and Naumov (2012), Wang (2015, 2016), and Li and Wang (2017). Alur et al. (2002) introduced Alternating-Time Temporal Logic (ATL) that combines temporal and coalition modalities. van der Hoek and Wooldridge (2003) proposed to combine ATL with epistemic modality to form Alternating-Time Temporal Epistemic Logic. A completeness theorem for a logical system that combines coalition power and epistemic modalities was proven by Ågotnes and Alechina (2012).

The notion of a strategy that we consider in this paper is much more restrictive than 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 histories. This restriction on strategies has been studied before under different names. Jamroga and Ågotnes (2007) talk about “knowledge to identify and execute a strategy”, Jamroga and van der Hoek (2004) discuss “difference between an agent knowing that he has a suitable strategy and knowing the strategy itself”. van Benthem (2001) calls such strategies “uniform”. Naumov and Tao (2017a) use the term “executable strategy”. Naumov and Tao (2017b) proposed a complete trimodal logical system describing an interplay between distributed knowledge, uniform strategic power modality, and standard strategic power modality for achieving a goal by a coalition in one step. Fervari et al. (2017) developed a complete logical system in a single-agent setting for uniform strategies to achieve a goal in multiple steps. Naumov and Tao (2017a) developed a similar system for maintaining a goal in multi-agent setting. Our contribution is different from all of the above papers by being the first to consider recall strategies and to compare their properties to those of amnesic strategies.

Paper Outline

In the next section we define transition systems and the syntax of our logical systems. This section applies equally to recall and amnesic strategies. The rest of the paper is split into two independent sections. The first of them proves the soundness and the completeness of Armstrong’s axioms for navigability under recall strategies and the second gives an axiomatization for amnesic strategies.

2 Syntax and Semantics

In this section we formally define the language of our logical system, the notion of a transition system, and the related terminology. Throughout the rest of the paper we fix a finite set $V$ of “views”.
Definition 1  \( \Phi \) is the minimal set of formulae such that
1. \( A \models B \in \Phi \) for all sets \( A, B \subseteq V \),
2. \( \neg \varphi, \varphi \rightarrow \psi \in \Phi \) for all formulae \( \varphi, \psi \in \Phi \).

Each transition system specifies a mapping \( \ast \) of views into equivalence classes of states. Transitions between states under an instruction \( i \) are captured by a transition function \( \Delta_i \).

Definition 2  \((S, \sim, \ast, I, \{ \Delta_i \}_{i \in I})\) is a transition system, if
1. \( S \) is a set of states,
2. \( \sim \) is an equivalence (indistinguishability) relation on \( S \),
3. \( \ast \) is a function from \( V \) to \( S / \sim \),
4. \( I \) is an arbitrary nonempty set of “instructions”,
5. \( \Delta_i \) maps set \( S \) into nonempty subsets of \( S \) for each \( i \in I \).

We write \( a^\ast \) instead of \( \ast(a) \). An example of a transition system is system \( T_0 \) depicted in Figure 1.

Definition 3  A finite sequence \( w_0, i_1, w_1, \ldots, i_n, w_n \), where \( n \geq 0 \), is called a history if
1. \( w_k \in S \) for each \( k \) such that \( 0 \leq k \leq n \),
2. \( i_k \in I \), for each \( k \) such that \( 1 \leq k \leq n \),
3. \( w_k \in \Delta_i(w_{k-1}) \), for each \( k \) such that \( 1 \leq k \leq n \).

For example, sequence \( g, 1, a, 1, e, 1, c, 0, h \) is a history for system \( T_0 \). The set of all histories is denoted by \( H \).

Definition 4  History \( h = w_0, i_1, w_1, \ldots, i_n, w_n \) is indistinguishable from history \( h' = w'_0, i_1, w'_1, \ldots, i_n, w'_n \) if \( w_k \sim w'_k \) for each \( k \) such that \( 0 \leq k \leq n \).

For example, histories \( a, 0, g \) and \( b, 0, g \) are indistinguishable in transition system \( T_0 \). Indistinguishability of histories of \( h \) and \( h' \) is denoted by \( h \approx h' \). The equivalence class of history \( h \) with respect to this equivalence relation is denoted by \( [h] \). Equivalence class of a state \( w \) with respect to equivalence relation \( \sim \) is denoted by \( \bar{w} \).

Lemma 1  If \( w_0, i_1, w_1, \ldots, w_n \approx w'_0, i_1, w'_1, \ldots, w'_n \), then \( \bar{w}_k = \bar{w}'_k \) for each \( k \leq n \).

Definition 5  An amnesic strategy is a function from set \( S / \sim \) to set \( I \). A recall strategy maps set \( H / \approx \) to set \( I \).

We write \( s(w) \) and \( s([h]) \) instead of \( s([w]) \) and \( s([h]) \).
Definition 6 An infinite sequence \( w_0, i_1, w_1, i_2, w_2 \ldots \) is called a path under a strategy \( s \) if for each \( k \geq 1 \)
1. \( w_0, i_1, w_1, i_2, w_2, \ldots, w_{k-1} \in H \).
2. \( i_k = s[w_{k-1}] \).

Lemma 2 For any history \( w_0, i_1, w_1, \ldots, i_n, w_n \) and any strategy \( s \), if \( i_k = s[w_{k-1}] \) for each \( k \) such that \( 1 \leq k \leq n \), then there are states \( w_{n+1}, w_{n+2}, \ldots \) and instructions \( i_{n+1}, i_{n+2}, \ldots \) such that sequence \( w_0, i_1, w_1, \ldots, i_n, w_n, i_{n+1}, w_{n+1}, i_{n+2}, w_{n+2}, \ldots \) is a path under strategy \( s \).

Proof. Elements \( i_{n+1}, w_{n+1}, i_{n+2}, w_{n+2}, \ldots \) can be constructed recursively because (a) there is a state \( w_{k+1} \in \Delta_{i_k}(w_k) \) for any state \( w_k \) and any \( i_{k+1} \in I \) by item 5 of Definition 2; (b) \( I \neq \emptyset \) by item 4 of Definition 2.

Definition 7 An infinite sequence \( w_0, i_1, w_1, i_2, w_2 \ldots \) is called a path under a recall strategy \( s \) if for each \( k \geq 1 \)
1. \( w_0, i_1, w_1, i_2, w_2, \ldots, w_{k-1} \in H \).
2. \( i_k = s[w_0, i_1, w_1, i_2, w_2, \ldots, w_{k-1}] \).

Definition 8 \( A^* = \{ a^* \mid a \in A \} \), for all sets \( A \subseteq V \).

Definition 9 For a given amnesic strategy or recall strategy \( s \), let \( \text{Path}_s(A) \) be the set of all paths \( w_0, i_1, w_1, i_2, w_2 \ldots \) under \( s \) such that \( \{w_0\} \in A^* \).

Definition 10 Let set \( \text{Visit}(B) \) be the set of all paths \( w_0, i_1, w_1, i_2, w_2 \ldots \) such that \( \{w_k\} \in B^* \) for some \( k \geq 0 \).

3 Navigation with Recall Strategies

In this section we show that Armstrong’s axioms give a sound and complete axiomatization of navigability between sets of classes with recall strategies. We start with a formal semantics of navigability relation \( \triangleright \) under recall strategies.

Definition 11 \( T = A \triangleright B \) if \( \text{Path}_s(A) \subseteq \text{Visit}(B) \) for some recall strategy \( s \) of transition system \( T \).

Axioms

The axioms of the logical system that we consider in this section are the tautologies in language \( \Phi \) and the following additional principles known as Armstrong’s axioms (Garcia-Molina et al., 2009, p. 81):

1. Reflexivity: \( A \triangleright B \), where \( A \subseteq B \),
2. Augmentation: \( A \triangleright B \rightarrow (A \cup C) \triangleright (B \cup C) \),

3. Transitivity: \( A \triangleright B \rightarrow (B \triangleright C \rightarrow A \triangleright C) \).

We write \( \vdash \varphi \) if formula \( \varphi \) is provable from these axioms using the Modus Ponens inference rule. We write \( X \vdash \varphi \) if \( \varphi \) is provable using a set of additional axioms \( X \).

**Soundness**

The soundness of Armstrong’s axioms with respect to perfect recall semantics is established in Lemma 10, Lemma 11, and Lemma 12 below. We start with technical notions that we need to prove Lemma 12.

**Definition 12** For any history \( w_0, i_1, \ldots, i_n, w_n \) and any set \( B \subseteq V \), let truncation \( (w_0, i_1, \ldots, i_n, w_n)|_B \) be

1. \( w_0, i_1, \ldots, i_n, w_n \) if \( n = 0 \) or \( [w_0] \in B^* \),
2. \( (w_1, i_2, \ldots, i_n, w_n)|_B \), otherwise.

For transition system \( T_0 \), if \( B = \{ b \} \) and \( b^* = [e] \), then truncation \( (g, 1, a, 1, e, 1, c, 0, h)|_B \) is history \( (e, 1, c, 0, h) \).

**Lemma 3** If \( h_1 \approx h_2 \), then \( h_1|_B \approx h_2|_B \).

**Definition 13** For any recall strategies \( s_1 \) and \( s_2 \), any set \( B \subseteq V \), and any history \( h = w_0, i_1, \ldots, w_n \), let

\[
(s_1 \circ_B s_2)[h] = \begin{cases} 
    s_1[h], & \text{if } \forall k \leq n \left( [w_k] \not\in B^* \right), \\
    s_2[h|_B], & \text{otherwise}.
\end{cases}
\]

In other words, “composition” strategy \( s_1 \circ_B s_2 \) follows strategy \( s_1 \) until a state \( w \) is reached such that \( [w] \in B^* \). Once this happens, the memory of all prior states and instructions is erased and strategy \( s_2 \) is executed as if navigation started from state \( w \).

**Lemma 4** Operation \( s_1 \circ_B s_2 \) is well-defined.

**Proof.** See Definition 12, Lemma 1, and Lemma 3.

**Definition 14** For any path \( w_0, i_1, w_1, i_2, \ldots \) and any set \( B \subseteq V \), let truncation \( (w_0, i_1, w_1, i_2, \ldots)|_B \) be

1. path \( w_0, i_1, w_1, i_2, \ldots \) if \( [w_0] \in B^* \),
2. path \( (w_1, i_2, w_2, i_3 \ldots)|_B \), otherwise.

**Lemma 5** If path \( \pi \in Path_{s_1 \circ_B s_2}(A) \cap Visit(B) \), then \( \pi|_B \in Path_{s_2}(B) \), for any recall strategies \( s_1, s_2 \) and any sets \( A, B \subseteq V \).
Lemma 6 \( \text{Path}_{s_1 \circ B \circ s_2}(A) \subseteq \text{Visit}(B) \) if \( \text{Path}_{s_1}(A) \subseteq \text{Visit}(B) \), for any recall strategies \( s_1, s_2 \) and any sets \( A, B \subseteq V \).

Proof. Assume that there is a path \( w_0, i_1, w_1, \ldots \) in set \( \text{Path}_{s_1 \circ B \circ s_2}(A) \) such that \( w_0, i_1, w_1, \ldots \notin \text{Visit}(B) \). Thus, \( i_k = (s_1 \circ B \circ s_2)[w_0, i_1, w_1, \ldots, w_{k-1}] \) for each \( k \geq 1 \) and \( [w_0] \in A^* \) by Definition 9 and Definition 7. Also, \( [w_k] \notin B^* \) for each \( k \geq 0 \) by Definition 10. Hence, by Definition 13, we have \( i_k = s_1[w_0, i_1, w_1, \ldots, w_{k-1}] \) for each \( k \geq 1 \). Thus, \( w_0, i_1, w_1, \ldots \in \text{Path}_{s_1}(A) \) by Definition 7 and Definition 7. Therefore, \( w_0, i_1, w_1, \ldots \in \text{Visit}(B) \) by the assumption \( \text{Path}_{s_1}(A) \subseteq \text{Visit}(B) \) of the lemma.

Lemma 7 If \( \text{Path}_{s_1 \circ B \circ s_2}(A) \subseteq \text{Visit}(B) \), then \( (\text{Path}_{s_1 \circ B \circ s_2}(A))|_B \subseteq \text{Path}_{s_2}(B) \), for any recall strategies \( s_1, s_2 \) and any sets \( A, B \subseteq V \).

Proof. Consider any path \( \pi \) in set \( \text{Path}_{s_1 \circ B \circ s_2}(A) \). Then, \( \pi \in \text{Visit}(B) \) because \( \text{Path}_{s_1 \circ B \circ s_2}(A) \subseteq \text{Visit}(B) \). Hence, \( \pi|_B \in \text{Path}_{s_2}(B) \) by Lemma 5.

Lemma 8 \( \text{Path}_s(A) \subseteq \text{Visit}(C) \), if \( (\text{Path}_s(A))|_B \subseteq \text{Visit}(C) \), for any recall strategy \( s \) and any sets \( A, B, C \subseteq V \).

Lemma 9 For any recall strategy \( s \) and any sets \( A, B, C \subseteq V \),

1. \( \text{Path}_s(A) \subseteq \text{Path}_s(B) \) if \( A \subseteq B \),
2. \( \text{Path}_s(A) \subseteq \text{Visit}(A) \),
3. \( \text{Path}_s(A \cup C) = \text{Path}_s(A) \cup \text{Path}_s(C) \),
4. \( \text{Visit}(B) \cup \text{Visit}(C) = \text{Visit}(B \cup C) \).

Proof. See Definition 7 and Definition 10.

The next three lemmas prove soundness of Armstrong’s axioms with respect to the perfect recall semantics.

Lemma 10 \( T \vdash A \triangleright B \), where \( A \subseteq B \).

Proof. By Definition 2, set \( I \) is nonempty, and thus it contains at least one instruction \( i_0 \). Let \( s \) be a recall strategy such that \( s[h] = i_0 \) for each history \( h \). By item 2 and item 2 of Lemma 9, \( \text{Path}_s(A) \subseteq \text{Path}_s(B) \subseteq \text{Visit}(B) \). Therefore, \( T \vdash A \triangleright B \) by Definition 11.

Lemma 11 If \( T \vdash A \triangleright B \), then \( T \vdash (A \cup C) \triangleright (B \cup C) \).

Proof. By Definition 11, the assumption \( T \vdash A \triangleright B \) implies that there exists a recall strategy \( s \) such that \( \text{Path}_s(A) \subseteq \text{Visit}(B) \). Thus, by item 3, item 2 and item 4 of Lemma 9

\[
\text{Path}_s(A \cup C) = \text{Path}_s(A) \cup \text{Path}_s(C) \subseteq \text{Visit}(B) \cup \text{Path}_s(C) \\
\subseteq \text{Visit}(B) \cup \text{Visit}(C) \subseteq \text{Visit}(B \cup C).
\]
Therefore, $T \models (A \cup C) \triangleright (B \cup C)$ by Definition 11.

Lemma 12  If $T \models A \triangleright B$ and $T \models B \triangleright C$, then $T \models A \triangleright C$.

Proof. By the assumption $T \models A \triangleright B$ and Definition 11, we have $Path_{s_1}(A) \subseteq Visit(B)$ for some recall strategy $s_1$. Similarly, the assumption $T \models B \triangleright C$ implies that there exists a recall strategy $s_2$ such that $Path_{s_2}(B) \subseteq Visit(C)$.

Statement $Path_{s_1}(A) \subseteq Visit(B)$ implies that $Path_{s_1 \circ s_2}(A) \subseteq Visit(B)$, by Lemma 6. Thus, by Lemma 7, $(Path_{s_1 \circ s_2}(A))|_B \subseteq Path_{s_2}(B)$. Hence, we have $(Path_{s_1 \circ s_2}(A))|_B \subseteq Visit(C)$ because of $Path_{s_2}(B) \subseteq Visit(C)$. Thus, by Lemma 8, $Path_{s_1 \circ s_2}(A) \subseteq Visit(C)$. Therefore, $T \models A \triangleright C$ by Definition 11.

Completeness

In the rest of this section we prove the completeness of Armstrong’s axioms with respect to the perfect recall semantics. We start by defining a canonical transition system $T(X) = (V \cup \{\emptyset\}, =, *, I, \{\Delta_i\}_{i \in I})$ for an arbitrary maximal consistent set of formulae $X \subseteq \Phi$. The set of states of this transition system consists of a single state for each view, plus one additional state that we denote by symbol $\emptyset$. Informally, the additional state is a sink or a “black hole” state from which there is no way out. State $h$ in transition system $T_0$ depicted in Figure 1 is an example of a black hole state. Note that the indistinguishability relation on the states of the canonical transition system is equality relation $\equiv$. That is, the agent has an ability to distinguish any two different states in the system. The fact that equality is suitable as an indistinguishability relation for the canonical transition system with perfect recall is surprising. The indistinguishability relation for the canonical transition system for amnesic strategies, discussed in the next section, is different from equality.

Each view $v \in V$ is also a state in the canonical transition system. The equivalence class of state $v$ consists of the state itself: $[v] = \{v\}$. We define $v^*$ to be class $[v]$.

Lemma 13 $u \in A$ iff $[u] \in A^*$ for each view $u \in V$ and each set $A \subseteq V$.

Informally, if set $X$ contains formula $A \triangleright B$, then we want the canonical transition system $T(X)$ to have a recall strategy to navigate from set $A^* = \{\{a\} \mid a \in A\} = \{\{a\} \mid a \in A\}$ to set $B^* = \{\{b\} \mid b \in B\} = \{\{b\} \mid b \in B\}$. It turns out that it is sufficient to have just a single instruction that transitions the system from any state in set $A$ to a state in set $B$. We denote this instruction by pair $(A, B)$.

Definition 15 $I = \{(A, B) \mid A \triangleright B \in X\}$.

As discussed above, for any instruction $(A, B)$ we define the nondeterministic transition function $\Delta_{(A, B)}$ to transition the system from a state in $A$ to a state in $B$. If used outside of set of states $A$, instruction $(A, B)$ transitions the system into state $\emptyset$.

Definition 16 $\Delta_{(A, B)}(w) = \begin{cases} B, & \text{if } w \in A, \\ \{\emptyset\}, & \text{otherwise}. \end{cases}$
This concludes the definition of the transition system $T(X)$.

Next, for any recall strategy $s$ and any set of states $G \subseteq V$, we define a family of sets of states $\{G^*_n\}_{n \geq 0}$. Informally, set $G^*_n$ is the set of all states from which strategy $s$ “draws” the system into set $G$ after at most $n$ transitions.

**Definition 17** For any recall strategy $s$ and any set $G \subseteq V$, let chain of $G^*_0 \subseteq G^*_1 \subseteq \cdots \subseteq V \cup \{\emptyset\}$ be defined as

1. $G^*_0 = G$,
2. $G^*_{n+1} = G_n^* \cup \{hd(h) \mid h \in H \text{ and } \Delta_s[h](hd(h)) \subseteq G^*_n\}$ for all $n \geq 0$.

Note that this definition, in essence, has an existential quantifier over history $h$. Thus, informally, strategy $s$ is allowed to “manipulate” the history in order to “draw” the system into set $G$.

**Lemma 14** $\emptyset \notin G^*_n$ for each $n \geq 0$.

**Proof.** We prove this statement by contradiction. Let $n$ be the smallest non-negative integer number such that $\emptyset \in G^*_n$. Note that $n \neq 0$ because $G^*_0 = G \subseteq V$ by Definition 17. Thus, $n > 0$. Hence, by Definition 17 there exists a history $h$ such that $hd(h) = \emptyset$ and $\Delta_s[h](\emptyset) \subseteq G^*_{n-1}$. Note that $\Delta_s[h](\emptyset) = \{\emptyset\}$ by Definition 16. Therefore, $\emptyset \in G^*_{n-1}$, which contradicts the choice of integer $n$. $\Box$

**Definition 18** $G^*_\infty = \bigcup_{k \geq 0} G^*_k$.

We now prove properties of the family of sets $\{G^*_n\}_{n \geq 0}$ that are needed to finish the proof of the completeness.

**Lemma 15** $X \vdash \{g\} \triangleright G^*_n$ for each integer $n \geq 1$ and each state $g \in G^*_n$.

**Proof.** By Definition 17 assumptions $n \geq 1$ and $g \in G^*_n$ imply that there is a history $h$ such that $hd(h) = g$ and

$$\Delta_s[h](g) \subseteq G^*_{n-1}. \quad (1)$$

Furthermore, by Definition 15 there are sets $A, B \subseteq V$ such that $s[h] = (A, B)$.

Case I: $g \notin A$. Then, $\Delta_s[h](g) = \{\emptyset\}$ by Definition 15. Hence, $\{\emptyset\} \subseteq G^*_n$ by equation (1), which contradicts Lemma 14.

Case II: $g \in A$. Then, $\vdash \{g\} \triangleright A$ by the Reflexivity axiom. At the same time, $X \vdash A \triangleright B$ by Definition 15. Thus, by the Transitivity axiom, $X \vdash \{g\} \triangleright B$.

Assumption $g \in A$ implies $\Delta_s[h](g) = B$, by Definition 16. Thus, $B \subseteq G^*_{n-1}$ due to equation (1). Hence, $\vdash B \triangleright G^*_{n-1}$ by the Reflexivity axiom.

Finally, statements $X \vdash \{g\} \triangleright B$ and $\vdash B \triangleright G^*_{n-1}$ by the Transitivity axiom imply $\vdash \{g\} \triangleright G^*_n$. $\Box$

**Lemma 16** For any $n \geq 0$ and any views $a_1, \ldots, a_n \in V$, if $X \vdash \{a_k\} \triangleright B$ for each $k \leq n$, then $X \vdash \{a_1, \ldots, a_n\} \triangleright B$. 

\[\text{11}\]
Proof. We prove this statement by induction on \( n \). For the base case, it suffices to show that \( \emptyset \not\triangleright B \), which is an instance of the Reflexivity axiom.

By the induction hypothesis, \( X \vdash \{a_1, \ldots, a_{n-1}\} \triangleright B \). Thus, by the Augmentation axiom,
\[
X \vdash \{a_1, \ldots, a_{n-1}, a_n\} \triangleright B \cup \{a_n\}. \tag{2}
\]
At the same time, \( X \vdash \{a_n\} \triangleright B \) by the assumption of the lemma. Hence, by the Augmentation axiom \( X \vdash B \cup \{a_n\} \triangleright B \). Thus, \( X \vdash \{a_1, \ldots, a_{n-1}, a_n\} \triangleright B \) by statement (2) and the Transitivity axiom.

Lemma 17 \( X \vdash G^s_n \triangleright G^s_{n-1} \) for each \( n \geq 1 \).

Proof. The statement the lemma follows from Lemma 15 and Lemma 16.

Lemma 18 \( X \vdash G^s_n \triangleright G \) for each \( n \geq 0 \).

Proof. We prove this statement by induction on integer \( n \). In the base case, due to Definition 17, it suffices to show that \( \emptyset \not\triangleright G \), which is an instance of the Reflexivity axiom.

For the induction step, note that \( X \vdash G^s_n \triangleright G^s_{n-1} \) by Lemma 17. At the same time, \( X \vdash G^s_{n-1} \triangleright G \) by the induction hypothesis. Hence, \( X \vdash G^s_n \triangleright G \) by the Transitivity axiom.

Lemma 19 There is \( n \geq 0 \), such that \( G^s_\infty = G^s_n \).

Proof. Since \( G^s_0 \subseteq G^s_1 \subseteq G^s_2 \subseteq \cdots \subseteq V \cup \{\emptyset\} \) and set \( V \) is finite, there must exist an integer \( n \) such that \( G^s_n = \bigcup_{k \geq 0} g_k \). Therefore, \( G^s_n = G^s_\infty \) by Definition 18.

Lemma 20 Set \( (\Delta_{s[h]}(hd(h))) \setminus G^s_\infty \) is nonempty for each history \( h \) such that \( hd(h) \notin G^s_\infty \).

Proof. Suppose \( \Delta_{s[h]}(hd(h)) \subseteq G^s_\infty \) for some history \( h \). It suffices to show that \( hd(h) \in G^s_\infty \). Indeed, by Lemma 19 there is an integer \( n \) such that \( G^s_\infty = G^s_n \). Thus, \( \Delta_{s[h]}(hd(h)) \subseteq G^s_n \). Hence, \( hd(h) \in G^s_{n+1} \) by Definition 17. Then, \( hd(h) \in G^s_\infty \) by Definition 18.

Lemma 21 For any positive integer \( k \) and any history \( w_0, i_1, w_1, \ldots, w_{k-1} \), if \( w_{k-1} \notin G^s_\infty \), then there is a state \( w_k \notin G^s_\infty \) such that \( w_0, i_1, w_1, \ldots, w_{k-1}, i_k, w_k \) is a history, where \( i_k = s[w_0, i_1, w_1, \ldots, w_{k-1}] \).

Proof. By Lemma 20 set \( (\Delta_{i_k}(w_{k-1})) \setminus G^s_\infty \) is not empty. Let \( w_k \) be any state such that \( w_k \in \Delta_{i_k}(w_{k-1}) \) and \( w_k \notin G^s_\infty \). Then, \( w_0, i_1, w_1, \ldots, w_{k-1}, i_k, w_k \) is a history by Definition 3.
Lemma 22 For each state \( w_0 \not\in G_f^s \) there exists a path \( w_0, i_1, w_1, \ldots \) under recall strategy \( s \) such that \( w_k \not\in G_f^s \) for each \( k \geq 0 \).

Proof. Note that single-element sequence \( w_0 \) is a history by Definition 3. Due to Lemma 21, there is an infinite sequence \( w_0, i_1, w_1, \ldots \) such that for each integer \( k \geq 1 \),

1. \( w_0, i_1, w_1, \ldots, w_{k-1} \) is a history,
2. \( i_k = s[w_0, i_1, w_1, \ldots, w_{k-1}] \),
3. \( w_k \not\in G_f^s \).

By Definition 7, sequence \( w_0, i_1, w_1, \ldots \) is a path under recall strategy \( s \).

Lemma 23 If \( X \vdash A \triangleright B \), then \( T(X) \models A \triangleright B \).

Proof. Assumption \( X \vdash A \triangleright B \) implies \( (A, B) \in J \) by Definition 15. Consider recall strategy \( s \) such that \( s[\cdot] = (A, B) \) for each class of histories \( [\cdot] \). Consider any path \( w_0, i_1, w_1, \ldots \) under recall strategy \( s \) where \( [w_0] \in A^* \). Then, \( w_0 \in A \) by Lemma 13.

By Definition 11 it suffices to show that \( [w_1] \in B^* \). Indeed, \( i_1 = (A, B) \) by choice of recall strategy \( s \). Thus, \( \Delta_{i_1}(w_0) = \Delta_{(A, B)}(w_0) = B \) by Definition 16 and due to the assumption \( w_0 \in A \).

By Definition 7, sequence \( w_0, i_1, w_1 \) is a history. Hence, \( w_1 \in \Delta_{i_1}(w_0) \) by Definition 3. Thus, \( w_1 \in \Delta_{i_1}(w_0) = B \). Then, \( [w_1] \in B^* \) by Lemma 13.

Lemma 24 If \( T(X) \models E \triangleright G \), then \( X \vdash E \triangleright G \).

Proof. By Definition 11 assumption \( T(X) \models E \triangleright G \) implies \( Path_s(E) \subseteq Visit(G) \) for some recall strategy \( s \).

Case I: \( E \subseteq G_f^s \). By Lemma 19 there exists an integer \( n \geq 0 \), such that \( E \subseteq G_f^n \). Thus, \( \vdash E \triangleright G_f^n \) by the Reflexivity axiom. At the same time, \( X \vdash G_f^n \triangleright G \) by Lemma 18. Therefore, \( X \vdash E \triangleright G \) by the Transitivity axiom.

Case II: \( E \not\subseteq G_f^s \). Then, there is an element \( w_0 \in E \) such that \( w_0 \notin G_f^s \). Thus, by Lemma 22 there is a path \( \pi = w_0, i_1, w_1, \ldots \) under recall strategy \( s \) such that \( w_k \notin G_f^s \) for all \( k \geq 0 \). Hence, \( w_k \notin G \) for all \( k \geq 0 \) because \( G = G_f^0 \subseteq G_f^s \) by Definition 17 and Definition 18. Thus, \( [w_0] \in E^* \) and \( [w_k] \notin G^* \) for all \( k \geq 0 \) by Lemma 13. Then, \( \pi \in Path_s(E) \) by Definition 9 and \( \pi \notin Visit(G) \) by Definition 10. Therefore, \( Path_s(E) \not\subseteq Visit(G) \), which contradicts the choice of strategy \( s \).

Lemma 25 \( X \vdash \varphi \) iff \( T(X) \models \varphi \) for each \( \varphi \in \Phi \).

Proof. We prove this lemma by induction on the structural complexity of \( \varphi \). The base case follows from Lemma 23 and Lemma 24. The induction case follows from the maximality and the consistency of set \( X \) in the standard way.

Now we can state and prove the completeness theorem for the recall strategies.
Theorem 1 If $T \models \varphi$ for every system $T$, then $\vdash \varphi$.

Proof. Suppose $\not\vdash \varphi$. Let $X$ be a maximal consistent set containing formula $\neg \varphi$. Thus, $T(X) \models \neg \varphi$ by Lemma 25. Therefore, $T(X) \not\models \varphi$. $\square$

4 Navigation with Amnesic Strategies

In this section we give a sound and complete axiomatization of navigability under amnesic strategies. We start by modifying Definition 11 to refer to amnesic strategies instead of recall strategies:

Definition 19 $T \models A \triangleright B$ if $\text{Path}_s(A) \subseteq \text{Visit}(B)$ for some amnesic strategy $s$ of a transition system $T$.

Axioms

The logical system for amnesic strategies is the same as for recall strategies with the exception that the Transitivity axiom is replaced by the following two principles:

3. Right Monotonicity: $A \triangleright B \rightarrow A \triangleright B'$, where $B \subseteq B'$,

4. Left Monotonicity: $A' \triangleright B \rightarrow A \triangleright B$, where $A \subseteq A'$.

Both of these two principles can be derived from Armstrong’s axioms.

Theorem 2 If $\vdash \varphi$, then $T \models \varphi$ for every system $T$.

Proof. Soundness of the Reflexivity axiom and the Augmentation axiom is similar to the case of perfect recall. Soundness of the Right Monotonicity axiom follows from $\text{Visit}(B) \subseteq \text{Visit}(B')$, where $B \subseteq B'$. Soundness of the Left Monotonicity axiom follows from $\text{Path}_s(A) \subseteq \text{Path}_s(A')$, where $A \subseteq A'$. $\square$

Completeness

In the rest of this section we prove completeness of our logical system with respect to the amnesic semantics. First, we define a canonical transition system $T(X) = (S, \sim$, $\ast, I, \{\Delta_i\}_{i \in I})$ for an arbitrary maximal consistent set of formulae $X \subseteq \Phi$.

Like in the perfect recall case, the canonical system has one state for each view and an additional “black hole” state $\emptyset$. Unlike the previous construction, the new canonical transition system has more additional states besides state $\emptyset$. Drawing on our original intuition of a transition system as a maze, we think about these new states as “wormholes”. For any sets of states $A$ and $B$ in the maze there is a wormhole state $w(A, B)$ that can be used to travel one-way from set $A$ to set $B$. Then, $S = V \cup \{\emptyset \} \cup \{w(A, B) \mid A, B \subseteq V\}$.

The agent can distinguish any two different non-wormhole states, but she cannot distinguish wormholes. In other words, each non-wormhole state $v \in V \cup \{\emptyset\}$ forms
its own indistinguishability class \([v] = \{v\}\), while all wormholes belong to the same single indistinguishability class of wormholes. Like in the perfect recall case, for each \(v ∈ V\), we define \(v^*\) to be class \([v]\).

**Lemma 26** \(u ∈ A\) iff \([u] ∈ A^*\) for each view \(u ∈ V\) and each set \(A ⊆ V\).

Like in the canonical model for the perfect recall case, for sets \(A, B \subseteq V\) such that \(X ⊨ A △ B\), we introduce an instruction \((A, B)\) that can be used to navigate from set \(A\) to set \(B\). Unlike the perfect recall case, we introduce such an instruction only if sets \(A\) and \(B\) are disjoint. This is an insignificant technical restriction that we use to simplify the proof of Lemma 29.

**Definition 20** \(I = \{(A, B) \mid X ⊨ A △ B \text{ and } A ∩ B = ∅\}\).

In the perfect recall case, instruction \((A, B)\) can be used to transition the system directly from a state in set \(A\) to a state in set \(B\). In our case, this transition happens via the wormhole state \(w(A, B)\). In other words, when instruction \((A, B)\) is invoked in a state from set \(A\), the system transitions into state \(w(A, B)\). When the same instruction is invoked in \(w(A, B)\), the system transitions into a state in \(B\).

**Definition 21**

\[
\Delta_{(A,B)}(u) = \begin{cases} 
\{w(A,B)\}, & \text{if } u ∈ A, \\
B, & \text{if } u = w(A,B), \\
\emptyset, & \text{otherwise}. 
\end{cases}
\]

**Lemma 27** If \(X ⊨ A △ B\) and sets \(A\) and \(B\) are disjoint, then \(T(X) ⊨ A △ B\).

**Proof.** Assumptions \(X ⊨ A △ B\) and \(A ∩ B = ∅\) imply that \((A, B) ∈ I\), by Definition 20. Consider strategy \(s\) such that \(s(x) = (A, B)\) for each class \(x\). By Definition 19, it suffices to show that \(Path_s(A) ⊆ Visit(B)\).

Consider any \(w_0, i_1, w_1, \cdots ∈ Path_s(A)\). By Definition 9, sequence \(w_0, i_1, w_1, \ldots\) is a path under strategy \(s\) such that \([w_0] ∈ A^*\). Hence, \(w_0 ∈ A\) by Lemma 26. Thus,

\[
\Delta_{(A,B)}(w_0) = \{w(A,B)\}
\]

by Definition 21.

At the same time, \(w_1 ∈ Δ_{i_1}(w_0)\) by Definition 5. Hence, \(w_1 ∈ Δ_{s[w_0]}(w_0)\) by Definition 6. Thus, \(w_1 ∈ Δ_{(A,B)}(w_0)\) by the choice of strategy \(s\). Then, \(w_1 = w(A,B)\) by equation (3). Hence,

\[
\Delta_{(A,B)}(w_1) = B
\]

by Definition 21.

Similarly, \(w_2 ∈ Δ_{i_2}(w_1)\) by Definition 5. Hence, \(w_2 ∈ Δ_{s[w_1]}(w_1)\) by Definition 6. Thus, \(w_2 ∈ Δ_{(A,B)}(w_1)\) by the choice of strategy \(s\). Then, \(w_2 ∈ B\) by equation (4). Hence, \([w_2] ∈ B^*\) by Lemma 26. Therefore, \(w_0, i_1, w_1, i_2, w_2, \cdots ∈ Visit(B)\) by Definition 16. 

\(\Box\)
Lemma 28 If $X \vdash A \triangleright B$, then $T(X) \models A \triangleright B$.

Proof. Suppose that $X \vdash A \triangleright B$. Thus, $X \vdash A \setminus B \triangleright B$ by the Left Monotonicity Axiom. Hence, $T(X) \models A \setminus B \triangleright B$ by Lemma 27. Therefore, $T(X) \models A \triangleright B$ due to the soundness of the Augmentation axiom, see Theorem 2. $\square$

Recall that all wormhole states belong to a single indistinguishability class of wormholes. For any amnesic strategy $s$, let $(A_s, B_s)$ be the instruction assigned by strategy $s$ to the class of wormholes. Once strategy $s$ is fixed, the states of the canonical transition system can be partitioned into five groups: set $A_s$, set $V \setminus A_s$, the single element set $\{\emptyset\}$ containing the black hole state, the single element set $\{w(A_s, B_s)\}$ containing the wormhole state $w(A_s, B_s)$, and the set $\{w(C, D) | (C, D) \neq (A_s, B_s)\}$ of all other wormholes. Definition 21 restricts transitions under strategy $s$ that are possible between these five groups of states. For example, from set $V \setminus A_s$ one can transition either into set $\{\emptyset\}$ or into set $\{w(C, D) | (C, D) \neq (A_s, B_s)\}$. The arrows in Figure 2 show all possible transitions between these five groups of states allowed under Definition 21. These five groups of states can be further classified into “above the line” and “below the line” states, as shown. Notice that once the system transitions into one of the “below the line” states, it is trapped there and it will never be able to transition under the amnesic strategy $s$ into an “above the line” state.

![Figure 2: Classification of states.]

Lemma 29 If $T(X) \models E \triangleright G$, then $X \vdash E \triangleright G$.

Proof. If $E \subseteq G$, then $\vdash E \triangleright G$ by the Reflexivity axiom. In the rest of the proof we suppose that there is $e_0 \in E \setminus G$. By Definition 19 assumption $T(X) \models E \triangleright G$ implies that there exists a strategy $s$ such that $\text{Path}_s(E) \subseteq \text{Visit}(G)$.

First, we show that $E \setminus G \subseteq A_s$. Suppose there is a view $e_1 \in E$ such that $e_1 \notin G$ and $e_1 \notin A_s$. By Definition 5 single-element sequence $e_1$ is a history. Thus, by Lemma 2 there is a path $\pi \in \text{Path}_s(E)$ that starts with state $e_1$. Since path $\pi$ starts in a state from set $V \setminus A_s$, all non-initial states of this path are “below the line”, see Figure 2. Hence, $\pi \notin G$, because $e_1 \notin G$ and none of the states “below the line” are in set $G$ either. Thus, $\pi \in \text{Path}_s(E)$ and $\pi \notin \text{Visit}(G)$, which contradicts the choice of strategy $s$. Therefore, $E \setminus G \subseteq A_s$. In particular $e_0 \in A_s$. 16
Second, we show that \( s[e_0] = (A_s, B_s) \). Suppose that \( s[e_0] = (C, D) \), where \( (C, D) \neq (A_s, B_s) \). If \( e_0 \notin C \), then \( \pi = e_0, (C, D), \{ \} \), \( s[\{], \{ \}, s[\{], \ldots \) is a path under strategy \( s \) by Definition \( 6 \) and Definition \( 21 \). Note that \( \pi \notin \text{Visit}(G) \) because \( e_0 \notin G \) and \( \pi \in \text{Path}_s(E) \). This contradicts \( \text{Path}_s(E) \subseteq \text{Visit}(G) \). Similarly, if \( e_0 \in C \), then sequence \( \pi = e_0, (C, D), w(C, D), (A_s, B_s), \{ \}, s[\{], \ldots \) is a path such that \( \pi \notin \text{Visit}(G) \) and \( \pi \in \text{Path}_s(E) \), which again contradicts \( \text{Path}_s(E) \subseteq \text{Visit}(G) \).

Third, we prove that \( B_s \subseteq G \). Suppose that there is a state \( b_0 \in B_s \setminus G \). Sequence \( e_0, (A_s, B_s), w(A_s, B_s), (A_s, B_s), b_0 \) is a history by Definition \( 3 \) Definition \( 21 \) and the choice of instruction \( (A_s, B_s) \). Thus, by Lemma \( 2 \) there is a path \( \pi \in \text{Path}_s(E) \) that starts as \( e_0, (A_s, B_s), w(A_s, B_s), (A_s, B_s), b_0 \). Assumption \( b_0 \in B_s \setminus \text{G} \) implies that \( b_0 \in V \setminus A_s \), because sets \( A_s \) and \( B_s \) are disjoint by Definition \( 20 \). Thus, \( \pi \) contains only states "below the line", see Figure \( 2 \), none of which are in set \( G \subseteq V \). Recall also that \( e_0, b_0 \notin \text{G} \) by the choice of states \( e_0 \) and \( b_0 \). Thus, \( \pi \in \text{Path}_s(E) \) and \( \pi \notin \text{Visit}(G) \), which again contradicts \( \text{Path}_s(E) \subseteq \text{Visit}(G) \).

Note that \( X \vdash A_s \triangleright B_s \) by Definition \( 20 \). Hence, \( X \vdash E \setminus G \triangleright B_s \) by the Left Monotonicity axiom and because \( E \setminus G \subseteq A_s \). Thus, \( X \vdash E \setminus G \triangleright G \) by Right Monotonicity axiom and because \( B_s \subseteq G \). Therefore, \( X \vdash E \triangleright G \) by the Augmentation axiom.

**Lemma 30** \( X \vdash \varphi \) iff \( T(X) \models \varphi \) for each \( \varphi \in \Phi \).

**Proof.** We prove this lemma by induction on the structural complexity of \( \varphi \). The base case follows from Lemma \( 28 \) and Lemma \( 29 \). The induction case follows from the maximality and the consistency of set \( X \) in the standard way.

Now we can state and prove the completeness theorem for amnesic strategies.

**Theorem 3** If \( T \models \varphi \) for every system \( T \), then \( \models \varphi \).

**Proof.** Suppose \( \not\models \varphi \). Let \( X \) be a maximal consistent set containing formula \( \neg \varphi \). Thus, \( T(X) \models \neg \varphi \) by Lemma \( 30 \). Therefore, \( T(X) \not\models \varphi \).

### 5 Conclusion

In this paper we have shown that the properties of navigability under perfect recall strategies are exactly those described by Armstrong’s axioms for functional dependency in the database theory. In the absence of perfect recall, the Transitivity axiom is no longer valid, but it could be replaced by the Left and the Right Monotonicity axioms.

### References

Thomas Ågotnes and Natasha Alechina. Epistemic coalition logic: completeness and complexity. In *Proceedings of the 11th International Conference on Autonomous*
Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge. Reasoning about coalitional games. *Artificial Intelligence*, 173(1):45 – 79, 2009. ISSN 0004-3702. doi: 10.1016/j.artint.2008.08.004.

Thomas Ågotnes, Philippe Balbiani, Hans van Ditmarsch, and Pablo Seban. Group announcement logic. *Journal of Applied Logic*, 8(1):62 – 81, 2010. ISSN 1570-8683. doi: 10.1016/j.jal.2008.12.002.

Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002. ISSN 0004-5411. doi: 10.1145/585265.585270.

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.

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.

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

Raul Fervari, Andreas Herzig, Yanjun Li, and Yanjing Wang. Strategically knowing how. In *26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017. (to appear).

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

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.

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. doi: 10.3166/jancl.17.423-475.

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

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.
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. ISSN 1529-3785. doi: 10.1145/2287718.2287722.

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, 2017a.

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*, 2017b. (to appear).

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

Marc Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002. doi: 10.1093/logcom/12.1.149.

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. ISBN 1-59593-303-4. doi: 10.1145/1160633.1160663.

Johan van Benthem. Games in dynamic-epistemic logic. *Bulletin of Economic Research*, 53(4):219–248, 2001. doi: 10.1111/1467-8586.00133.

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. doi: 10.1023/A:1026171312755.

Wiebe van der Hoek and Michael Wooldridge. On the logic of cooperation and propositional control. *Artificial Intelligence*, 164(1):81 – 119, 2005. ISSN 0004-3702.

Yanjing Wang. A logic of knowing how. In *Logic, Rationality, and Interaction*, pages 392–405. Springer, 2015. doi: 10.1007/978-3-662-48561-3_32.

Yanjing Wang. A logic of goal-directed knowing how. *Synthese*, 2016. (to appear).