Armstrong’s Axioms and Navigation Strategies

Kaya Deuser, Pavel Naumov
Vassar College
124 Raymond Avenue
Poughkeepsie, NY 12604
{kdeuser, pnaumov}@vassar.edu

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 database theory. If the assumption of perfect recall is omitted, then Armstrong’s transitivity axiom is not valid, but it can be replaced by a weaker principle. The main technical results are soundness and completeness theorems for the logical systems describing properties of navigability with and without perfect recall.

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 $c$. 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 she 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 a 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 tran-
sitions 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 memoryless 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 a memoryless strategy. The other popular robotic vacuum cleaner, Neato, is scanning the room before cleaning using a memoryless strategy. The other popular robotic vacuum cleaners, Roomba, is only using read-write memory to store information, such as a cleaning schedule, because the same strategy must work to navigate from set \( A \) to set \( B \) or to class \( c \). One can similarly show that there is no memoryless 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 \( f \), there then is a memoryless strategy to do this. Indeed, consider a memoryless 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 sets of classes cannot be reduced to navigability between 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 a memoryless 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 **reflexivity**: \( A \triangleright B \), where \( A \subseteq B \). This property is true for both memoryless and recall strategies because absolutely 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 memoryless and recall strategies is **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 memoryless strategies is **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 \( C \). To see that this property is not universally true for memoryless strategies, note that, in transition system \( T_0 \), memoryless strategy that always uses instruction 0 can be used to navigate from set \( \{a\} \) to set \( \{g\} \) and memoryless 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

<table>
<thead>
<tr>
<th>(a, b)</th>
<th>(c, d)</th>
<th>(e)</th>
<th>(f)</th>
<th>(g)</th>
<th>(h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>m</td>
<td>r</td>
<td>r</td>
<td>m</td>
<td>r'</td>
</tr>
<tr>
<td>-</td>
<td>m</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>r'</td>
</tr>
<tr>
<td>m</td>
<td>m</td>
<td>m</td>
<td>r'</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>m</td>
<td>m</td>
<td>r'</td>
<td>m</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>m</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>m'</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>m</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Navigability between classes in system \( T_0 \).
memoryless 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 database theory as Armstrong’s axioms (Garcia-Molina, Ullman, and Widom 2009, p. 81), where they give a sound and complete axiomatization of functional dependency (Armstrong 1974). We also give a sound and complete axiomatization of universal properties of navigability by memoryless strategies. It consists of the reflexivity and augmentation principles mentioned above as well as the monotonicity principle \((A \cup C) \supset B \rightarrow A \supset B\). The latter principle is true for the recall strategies as well, but it is 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; Ágotnes et al. 2010; Ágotnes, van der Hoek, and Wooldridge 2009; Belardinelli 2014). Alternative logical system were proposed by (More and Naumov 2012), (Wang 2015; 2016), (Alur, Henzinger, and Kupferman 2009; Belardinelli 2014). Alternative logical system were introduced Alternating-Time Temporal Logic (ATL) and (Li and Wang 2017). (Alur, Henzinger, and Kupferman 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 (Ágotnes 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 Ágotnes 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 propose complete logical systems for recall strategies and memoryless 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 memoryless strategies. The rest of the paper is split into two independent sections. The first of them proves the completeness of Armstrong’s axioms for navigability under recall strategies and the second gives an axiomatization for memoryless strategies. The soundness of Armstrong’s axioms with respect to the perfect recall semantics is given in the full version of this paper (Deuser and Naumov 2017).

**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. In the introduction, relation \(\supset\) was viewed as a relation between equivalence classes of a given transition system. Thus, our language depends on these classes and changes from transition system to transition system. In order to have a single language for all transition systems we introduce a fixed finite set of “views” \(V\), whose elements act as names of the equivalence classes in any given transition system.

**Definition 1** \(\Phi\) is the minimal set of formulae such that

1. \(A \supset B \in \Phi\) for all nonempty\(^1\) 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 * of views into equivalence classes of states. Transitions between states under an instruction \(i\) are captured by a transition function \(\Delta_i\).

**Definition 2** \((S, \sim, *, I, \{\Delta_i\}_{i \in I})\) is a transition system, if

1. \(S\) is a set of states,
2. \(\sim\) is an equivalence (indistinguishability) relation on \(S\),
3. * 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 \(a^*\) instead of \(*a\), where \(a \in V\). An example of a transition system is system \(T_0\) depicted in Figure 1.

**Definition 3** A finite sequence \(w_0, i_1, w_1, \dotsc, i_n, w_n\), 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. \(w_k \in \Delta_{i_k}(w_{k-1})\), 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 \(h = w_0, i_1, w_1, \dotsc, i_n, w_n\) is indistinguishable from history \(h' = w'_0, i_1, w'_1, \dotsc, i_n, w'_n\) if \(w_k \sim w'_k\) for each \(k\) such that \(0 \leq k \leq n\).

\(^1\)If one allows sets \(A\) and \(B\) to be empty, most of the proofs in this paper will remain unchanged, but both logical systems will need an additional axiom \(\neg(A \supset \emptyset)\) for each nonempty set \(A\).
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 \([w]\).

**Lemma 1** If \( w_0, i_1, w_1, \ldots, w_n \approx w'_0, i_1, w'_1, \ldots, w'_n \), then 
\[
[w_k] = [w'_k]
\]
for each \( k \leq n \).

**Definition 5** A memoryless strategy is a function from set \( S/\sim \to \) to set \( I \). A recall strategy maps set \( H/\approx \to \) 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 memoryless 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 memoryless 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 \( w_{k+1} \in \Delta_{i_{k+1}}(w_k) \) 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 memoryless strategy or recall strategy \( s \), let \( Path_s(A) \) 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 \( \text{Visit}_s(B) \) be the set of all paths \( w_0, i_1, w_1, i_2, w_2 \ldots \), under \( s \) such that \( [w_k] \in B^* \) for some \( k \geq 0 \).

We write \( \text{Visit}(B) \) instead of \( \text{Visit}_s(B) \) when value of \( s \) is clear from the context.

**Navigation with Recall Strategies**

In this section we show that Armstrong’s axioms give a 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** \( T \vdash A \triangleright B \) if \( \text{Path}_s(A) \subseteq \text{Visit}_s(B) \) 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, Ullman, and Widom 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 \( T \vdash \varphi \) if \( \varphi \) is provable using a set of additional axioms \( X \).

A relatively straightforward proof of the following soundness theorem for this logical system can be found in the full version of this paper (Deuser and Naumann 2017).

**Theorem 1** If \( \vdash \varphi \), then \( T \vdash \varphi \) for every system \( T \).

**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 \( T(X) = (V \cup \{\emptyset\}, =, *, I, \{\Delta_s\}_{s \in I}) \) 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 \( = \). 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 memoryless 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: \( \{v\} \). We define \( v^* \) to be class \( [v] \).

**Lemma 3** \( u \in A \text{ iff } [u] \in A^* \) 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 12** \( I = \{(A, B) \mid A \triangleright B \in X\} \).

Recall that assumption \( A \triangleright B \in X \) requires sets \( A \) and \( B \) to be nonempty due to Definition 1.

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 black hole state \( \emptyset \).
Lemma 4 For any recall strategy \( s \) and any nonempty set \( G \subseteq V \), let chain of \( G_0^s \subseteq G_1^s \subseteq \cdots \subseteq V \cup \{ \emptyset \} \) be defined as

1. \( G_0^s = G \).
2. \( G_{n+1}^s = G_n^s \cup \{ \text{hd}(h) \mid h \in H \text{ and } \Delta_{s[h]}(\text{hd}(h)) \subseteq G_n^s \} \) 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 \( G \).

Definition 13

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

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 \( \{ G_n^s \}_{n \geq 0} \). Informally, set \( G_0^s \) is the set of all states from which strategy \( s \) “draws” the system into set \( G \) after at most \( n \) transitions. For any history \( h = w_0, i_1, w_1, \ldots, i_n, w_n \), by \( \text{hd}(h) \) we mean the state \( w_n \).

Lemma 5

\( \emptyset \notin G_n^s \) for each \( n \geq 0 \).

Proof. Suppose \( G_0^s \subseteq G_1^s \subseteq \cdots \subseteq V \cup \{ \emptyset \} \) and set \( V \) is finite, there must exist an integer \( n \) such that \( G_n^s = \bigcup_{k \geq 0} G_k^s \). Therefore, \( G_n^s = G_{n+1}^s \) by Definition 15.

Lemma 6 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 \( X \vdash \{ a_1, \ldots, a_n \} \triangleright B \).

Proof. We prove this statement by induction on \( n \). In the base case, \( X \vdash \{ a_1 \} \triangleright B \) due to the assumption of the lemma.

By the induction hypothesis, \( X \vdash \{ a_1, \ldots, a_{n-1} \} \triangleright B \). Thus, by the Augmentation axiom,

\[
X \vdash \{ a_1, \ldots, a_{n-1}, a_n \} \triangleright B \cup \{ a_n \}.
\]

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

Lemma 7 \( X \vdash G_n^s \triangleright G_{n+1}^s \) for each \( n \geq 0 \).

Proof. The statement of the lemma follows from Lemma 5 and Lemma 6.

Lemma 8 \( X \vdash G_n^s \triangleright G \) for each \( n \geq 0 \).

Proof. We prove this statement by induction on integer \( n \). In the base case, due to Definition 14, it suffices to show that \( \vdash G \triangleright G \), which is an instance of the Reflexivity axiom.

For the induction step, note that \( X \vdash G_n^s \triangleright G_{n+1}^s \) by Lemma 7. At the same time, \( X \vdash G_n^s \triangleright G \) by the induction hypothesis. Hence, \( X \vdash G_n^s \triangleright G \) by the Transitivity axiom.

Lemma 9 There is \( n \geq 0 \), such that \( G_{n+1}^s = G_n^s \).

Proof. Since \( G_0^s \subseteq G_1^s \subseteq \cdots \subseteq V \cup \{ \emptyset \} \) and set \( V \) is finite, there must exist an integer \( n \) such that \( G_n^s = \bigcup_{k \geq 0} G_k^s \). Therefore, \( G_n^s = G_{n+1}^s \) by Definition 15.

Lemma 10 Set \( (\Delta_{s[h]}(\text{hd}(h))) \setminus G_{n+1}^s \) is non-empty for each history \( h \) such that \( \text{hd}(h) \notin G_{n+1}^s \).

Proof. Suppose \( \Delta_{s[h]}(\text{hd}(h)) \subseteq G_n^s \) for some history \( h \). It suffices to show that \( \text{hd}(h) \notin G_{n+1}^s \). Indeed, by Lemma 9 there is \( n \) such that \( G_n^s = G_{n+1}^s \). Hence, \( \text{hd}(h) \in G_{n+1}^s \) by Definition 14. Then, \( \text{hd}(h) \in G_{n+1}^s \) by Definition 15.

Lemma 11 For any positive integer \( k \) and any history \( w_0, i_1, w_1, \ldots, w_{k-1} \) if \( w_{k-1} \notin G_{n-1}^s \), then there is a state \( w_k \notin G_{n-1}^s \), 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 10, set \( (\Delta_{i_k}(w_{k-1})) \setminus G_{n-1}^s \) is not empty. Let \( w_k \) be any state such that \( w_k \in \Delta_{i_k}(w_{k-1}) \) and \( w_k \notin G_{n-1}^s \). Then, \( w_0, i_1, w_1, \ldots, w_{k-1}, i_k, w_k \) is a history by Definition 3.

Lemma 12 For each state \( w_0 \notin G_n^s \) there exists a path \( w_0, i_1, w_1, \ldots \) under recall strategy \( s \) such that \( w_k \notin G_n^s \) for each \( k \geq 0 \).

Proof. Note that single-element sequence \( w_0 \) is a history by Definition 3. Due to Lemma 11, 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. \( w_k \notin G^*_X \).

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

\[ \square \]

**Lemma 13** If \( X \vdash A \supset B \), then \( T(X) \models A \supset B \).

**Proof.** Assumption \( X \vdash A \supset B \) implies \( (A, B) \in I \) by Definition 12. Consider recall strategy \( s \) such that \( s[h] = (A, B) \) for each class of histories \( h \). 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 3.

By Definition 11 and Definition 10 it suffices to show that \( [w_1] \in B^* \). Indeed, \( i_1 = (A, B) \) by choice of recall strategy \( s \). Thus, \( \Delta_i(w_0) = \Delta_i(A, B)(w_0) = B \) by Definition 13 and due to the assumption \( w_0 \in A \).

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

\[ \square \]

**Lemma 14** If \( T(X) = E \supset G \), then \( X \vdash E \supset G \).

**Proof.** By Definition 11, assumption \( T(X) = E \supset G \) implies \( Path_s(E) \subseteq Visit(G) \) for some recall strategy \( s \). Case I: \( E \subseteq G^*_X \). By Lemma 9 there exists an integer \( n \geq 0 \), such that \( E \subseteq G^n_\pi \). Thus, \( \vdash E \supset G^n_\pi \) by the Reflexivity axiom. At the same time, \( X \vdash G^*_X \supset G \) by Lemma 8. Therefore, \( X \vdash E \supset G \) by the Transitivity axiom.

Case II: \( E \supset G^*_X \). Then, there is an element \( w_0 \in E \) such that \( w_0 \notin G^*_X \). Hence, \( w_i \notin G \) for all \( k \geq 0 \) because \( G = G^*_0 \subseteq G^*_X \) by Definition 14 and Definition 15. Thus, \( [w_0] \in E^* \) and \( [w_k] \notin G^* \) for all \( k \geq 0 \) by Lemma 3. Then, states in path \( \pi \in Path_s(E) \) by Definition 9 and \( \pi \notin Visit(G) \) by Definition 10. Therefore, \( Path_s(E) \notin Visit(G) \), which contradicts the choice of strategy \( s \).

\[ \square \]

**Lemma 15** \( X \vdash \varphi \) iff \( T(X) = \varphi \) for each \( \varphi \notin \Phi \).

**Proof.** We prove this lemma by induction on the structural complexity of \( \varphi \). The base case follows from Lemmas 13 and 14. The induction case follows from the maximality and the consistency of set \( X \) in the standard way.

We are now ready to state and prove the completeness theorem for the recall strategies.

**Theorem 2** 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 15. Therefore, \( T(X) \not\models \varphi \).

\[ \square \]

**Navigation with Memoryless Strategies**

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

**Definition 16** \( T \models A \supset B \) if \( Path_s(A) \subseteq Visit(B) \) for some memoryless strategy \( s \) of a transition system \( T \).

**Axioms** The logical system for memoryless strategies is the same as for recall strategies with the exception that the Transitivity axiom is replaced by the following principle:

3. Monotonicity: \( A \supset B \rightarrow A \supset B \), where \( A \subseteq A' \).

This principle can be derived from Armstrong’s axioms.

**Theorem 3** If \( \not\vdash \varphi \), 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, see Theorem 1. Soundness of the Monotonicity axiom follows from \( Path_s(A) \subseteq Path_s(A') \), where \( A \subseteq A' \).

**Completeness** In the rest of this section we prove completeness of our logical system with respect to the memoryless 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 \( X \notin \Phi \).

Like in the perfect recall case, the canonical system has one state for each view and an additional “black hole” state \( \diamond \). Unlike the previous construction, the new canonical transition system has more additional states besides state \( \diamond \). 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, \( S = V \cup \{ \diamond \} \cup \{ w(A, B) \mid A \cup B \subseteq V \} \).

The agent can distinguish any two different non-wormhole states, but she cannot distinguish wormholes. In other words, each non-wormhole state \( v \in V \) or \( \diamond \) 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 \( v \in V \), we define \( v^* \) to be class \([v]\).

**Lemma 16** \( u \in A \) iff \( [u] \in A^* \) for each view \( u \in V \) and each set \( A \subseteq V \).

Like in the canonical model for the perfect recall case, for sets \( A, B \subseteq V \) such that \( X \vdash A \supset B \), 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 19.

**Definition 17** \( I = \{(A, B) \mid X \vdash A \supset B \text{ and } A \cap B = \emptyset \} \).

Recall that assumption \( X \vdash A \supset B \) implies that sets \( A \) and \( B \) are nonempty due to Definition 1.

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 18

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

Lemma 17 If \( X \vdash A \triangleright B \) and sets \( A \) and \( B \) are disjoint, then \( T(X) \vdash A \triangleright B \).

Proof. Assumptions \( X \vdash A \triangleright B \) and \( A \cap B = \emptyset \) imply that \( (A, B) \in I \), by Definition 17. Consider a memoryless strategy \( s \) such that \( s(x) = (A, B) \) for each class \( x \). By Definition 16, it suffices to show that \( \text{Path}_s(A) \subseteq \text{Visit}(B) \).

Consider any \( w_0, i_1, w_1, \ldots \in \text{Path}_s(A) \). Then, by Definition 9, sequence \( w_0, i_1, w_1, \ldots \) is a path under strategy \( s \) such that \( \{w_0\} \in A^* \). Hence, \( w_0 \in A \) by Lemma 16. Thus,

\[ \Delta_{(A,B)}(w_0) = \{w(A,B)\} \tag{3} \]

by Definition 18.

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

\[ \Delta_{(A,B)}(w_1) = B \tag{4} \]

by Definition 18.

Similarly, \( w_2 \in \Delta_{i_2}(w_1) \) by Definition 3. Hence, \( w_2 \in \Delta_{[w_2]}(w_1) \) by Definition 6. Thus, \( w_2 \in \Delta_{(A,B)}(w_1) \) by the choice of strategy \( s \). Then, \( w_2 \in B \) by equation (4). Hence, \( \{w_2\} \in B^* \) by Lemma 16. Therefore, \( w_0, i_1, w_1, i_2, w_2, \ldots \in \text{Visit}(B) \) by Definition 10.

\[ \Box \]

Lemma 18 If \( X \vdash A \triangleright B \), then \( T(X) \vdash A \triangleright B \).

Proof. Suppose that \( X \vdash A \triangleright B \).

If \( A \triangleright B \neq \emptyset \), then \( X \vdash A \triangleright B \) by the Monotonicity Axiom. Hence, \( T(X) \vdash A \triangleright B \) by Lemma 17. Therefore, \( T(X) \vdash A \triangleright B \) due to the soundness of the Augmentation axiom, see Theorem 3.

If \( A \triangleright B = \emptyset \), then \( A \subseteq B \). Therefore, \( T(X) \vdash A \triangleright B \) due to the soundness of the Reflexivity axiom.

Recall that all wormhole states belong to a single indistinguishability class of wormholes. For any memoryless 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 \( \{C\} \) 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 \( \{w(C, D) \mid (C, D) \neq (A_s, B_s)\} \) of all other wormholes. Definition 18 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 \( \{C\} \) or into set \( \{w(C, D) \mid (C, D) \neq (A_s, B_s)\} \). The arrows in Figure 2 show all possible transitions between these five groups of states allowed under Definition 18. 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 memoryless strategy \( s \) into an “above the line” state.

![Figure 2: Transitions under memoryless strategy \( s \).](image-url)
\( E \setminus G \subseteq A \), implies that \( E \subseteq A \cup G \). Therefore, \( X \vdash E \triangleright G \) by the Monotonicity axiom.

**Lemma 20** \( X \vdash \varphi \iff T(X) \models \varphi \) for each \( \varphi \in \Phi \).

**Proof.** We prove this lemma by induction on the structural complexity of \( \varphi \). The base case follows from Lemma 18 and Lemma 19. The induction case follows from the maximality and the consistency of set \( X \) in the standard way.  

We are now ready to state and prove the completeness theorem for memoryless strategies.

**Theorem 4** 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 20. Therefore, \( T(X) \not\models \varphi \).

**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 database theory. In the absence of perfect recall, the transitivity axiom is no longer valid, but it could be replaced by the Monotonicity axiom.

**References**


