Navigability with Bounded Recall

Kaya Deuser
Department of Computer Science
Vassar College
Poughkeepsie, New York 12604
kdeuser@vassar.edu

Pavel Naumov
Department of Mathematical Sciences
Claremont McKenna College
Claremont, California 91711
pnaumov@cmc.edu

Maze Navigation

In this paper we study properties of navigation by machines with finite memories, which we refer to as having *bounded recall*. As an example, consider the maze depicted in Figure 1. This maze has seven rooms: a, b, c, d, e, f, and g. Some rooms of the maze are indistinguishable, which is shown in the diagram using dashed lines. Additionally, there are one-way passages between rooms, shown by the arrows connecting them. In this example there are exactly two passages leading out of each room, labeled 0 and 1. One can think about the labels as signs at the entrances to the passages, but in this paper we refer to labels as *actions* that an agent must execute to navigate through the passage. For example, to navigate from room a to room g, an agent could invoke the sequence of actions $1, 1, 0, 0, 1$.

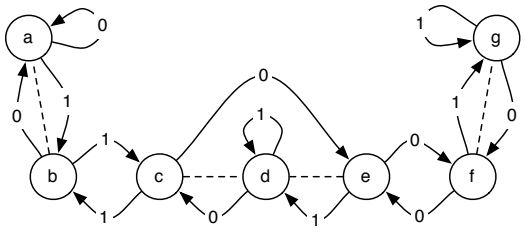

Figure 1: Maze.

The focus of this paper is on navigability by agents with bounded recall. We model such agents by Mealy machines (Mealy 1955). These machines determine output (action) based on the current state and the current input (observation). Mealy machines have been previously used in circuit design (Benini and De Micheli 1995), in machine learning (Shahbaz and Groz 2009; Aarts, Schmaltz, and Vaandrager 2010; Aarts et al. 2014), and for software specification (Bloem et al. 2009; Herrmannsdoerfer, Benz, and Juergens 2009).

Figure 2 (left) depicts a single-state Mealy machine m_1 that can navigate from room a to room g of the maze depicted in Figure 1. Note that rooms a and b of the maze are indistinguishable. Thus, *single-state* Mealy machine m_1 has to output the same action in both of these rooms. In case of machine m_1 this action is 1. In other words, machine m_1 specifies an action for each class of indistinguishable rooms rather than for each room. We show this by a transition from state q back to state q of the machine m_1 labeled with $[a]/1$.

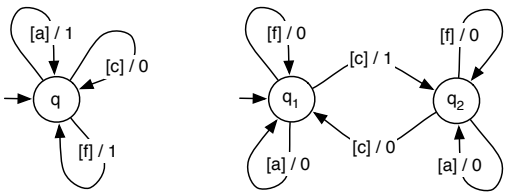

Figure 2: Mealy machines m_1 (left) and m_2 (right).

It is easy to see that there is no single-state Mealy machine that can navigate from state g to state a. Indeed, to navigate from g to a, the hypothetical machine would need to use different actions (1 and 0) in indistinguishable states c and d, which is not possible if the machine has only one state. However, navigation from room g to room a is possible by two-state Mealy machine m_2 depicted in Figure 2 (right), when starting from state q_1. First, machine m_2 loops from state q_1 back into q_1 twice, while using action 0 to transition from room g to room f and then to room e. Next, it transitions from state q_1 into state q_2 and then back to state q_1, as action 1 is used to move from room e to room d and then action 0 is used to move into room c. Finally, it transitions from state q_1 into state q_2 and loops in state q_2, while moving from room c to room b using action 1 and then transitioning to room a using action 0.

Navigability between Classes

From the point of view of logically capturing properties of navigability, we find it more convenient to talk about navigability from one indistinguishability class to another rather than between individual rooms. We say that a machine can navigate from a class of indistinguishable rooms $[a]$ to a class $[g]$ if the machine can navigate from *each* room in class $[a]$ to *at least one* room in class $[g]$. For example, machine m_1 can navigate from class $[a] = \{a, b\}$ to class $[g] = \{f, g\}$ because no matter if it starts in room a or room b it reaches

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

room $f \in [g]$. Similarly, machine m_2 can navigate from class $[g]$ to class $[a]$. Furthermore, we say that a Mealy machine can navigate from a *set* of indistinguishability classes X to a *set* of indistinguishability classes Y if it can navigate from each room in each class of set X to a room in a class of set Y. We write $X \rhd_n Y$ if there is a Mealy machine with at most n states that can navigate from a set of classes X to a set of classes Y. The language of navigability between sets of classes rather than individual classes is a more powerful one. Indeed, statement $(\{[a]\} \rhd_n \{[c]\}) \wedge (\{[b]\} \rhd_n \{[c]\})$ asserts that there are *two* n-state machines, one of which can navigate from class $[a]$ to class $[c]$, while the other machine can navigate from class $[b]$ to class $[c]$. At the same time, statement $\{[a], [b]\} \rhd_n \{[c]\}$ claims that there is *one* n-state machine that can navigate from anywhere in the union $[a] \cup [b]$ to class $[c]$. The main contribution of this paper is a sound, complete, and decidable axiomatization of the relation $X \rhd_n Y$.

Axioms

This paper builds on our previous paper (Deuser and Naumov 2018), where we gave sound and complete axiomatizations of navigation by strategies with perfect recall and memoryless strategies. The former can be viewed as infinite-state Mealy machines and the latter as single-state Mealy machines. If the navigability by perfect recall strategies between sets of indistinguishability classes X and Y is denoted by $X \rhd_\infty Y$, then our axioms from (Deuser and Naumov 2018) for such strategies are

1. Reflexivity: $X \rhd_\infty Y$, where $X \subseteq Y$,
2. Augmentation: $X \rhd_\infty Y \to X \cup Z \rhd_\infty Y \cup Z$,
3. Transitivity: $X \rhd_\infty Y \to (Y \rhd_\infty Z \to X \rhd_\infty Z)$.

The above three axioms are 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).

If navigability by memoryless strategies is denoted by $X \rhd_1 Y$, then our axioms from (Deuser and Naumov 2018) for memoryless strategies are

1. Reflexivity: $X \rhd_1 Y$, where $X \subseteq Y$,
2. Augmentation: $X \rhd_1 Y \to X \cup Z \rhd_1 Y \cup Z$,
3. Monotonicity: $X' \rhd_1 Y \to X \rhd_1 Y$, if $X \subseteq X'$.

Note that the second list does not contain the Transitivity axiom because, as we show in (Deuser and Naumov 2018), this axiom is not valid for memoryless strategies. Instead of the Transitivity axiom, the last list of axioms includes the Monotonicity axiom. The Monotonicity axiom is also valid for the perfect recall navigability, but it is provable from the rest of the axioms for the relation $\rhd_\infty$ and therefore is not included on the first list of axioms.

In this paper we give a sound and complete axiomatization of the relation $X \rhd_n Y$. It consists of the following axioms:

1. Reflexivity: $X \rhd_n X$,
2. Augmentation: $X \rhd_n Y \to X \cup Z \rhd_n Y \cup Z$,
3. Transitivity: $X \rhd_n Y \to (Y \rhd_k Z \to X \rhd_{n+k} Z)$,
4. Monotonicity: $X' \rhd_n Y \to X \rhd_n Y$, if $X \subseteq X'$.

The proof of completeness combines the techniques from the two completeness proofs in (Deuser and Naumov 2018) and further extends them by introducing finite-length wormholes. It also uses the pigeonhole principle to show that Mealy machines of a small size cannot navigate through sufficiently many wormholes.

References

Aarts, F.; Kuppens, H.; Tretmans, J.; Vaandrager, F.; and Verwer, S. 2014. Improving active mealy machine learning for protocol conformance testing. *Machine learning* 96(1-2):189–224.

Aarts, F.; Schmaltz, J.; and Vaandrager, F. 2010. Inference and abstraction of the biometric passport. In Margaria, T., and Steffen, B., eds., *Leveraging Applications of Formal Methods, Verification, and Validation*, 673–686. Berlin, Heidelberg: Springer Berlin Heidelberg.

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

Benini, L., and De Micheli, G. 1995. Transformation and synthesis of fsms for low-power gated-clock implementation. In *Proceedings of the 1995 International Symposium on Low Power Design*, ISLPED '95, 21–26. New York, NY, USA: ACM.

Bloem, R.; Chatterjee, K.; Henzinger, T. A.; and Jobstmann, B. 2009. Better quality in synthesis through quantitative objectives. In Bouajjani, A., and Maler, O., eds., *Computer Aided Verification*, 140–156. Berlin, Heidelberg: Springer Berlin Heidelberg.

Deuser, K., and Naumov, P. 2018. Armstrong's axioms and navigation strategies. In *Proceedings of Thirty-Second AAAI Conference on Artificial Intelligence*.

Garcia-Molina, H.; Ullman, J.; and Widom, J. 2009. *Database Systems: The Complete Book*. Prentice-Hall, second edition.

Herrmannsdoerfer, M.; Benz, S.; and Juergens, E. 2009. Cope - automating coupled evolution of metamodels and models. In Drossopoulou, S., ed., *ECOOP 2009 – Object-Oriented Programming*, 52–76. Berlin, Heidelberg: Springer Berlin Heidelberg.

Mealy, G. H. 1955. A method for synthesizing sequential circuits. *The Bell System Technical Journal* 34(5):1045–1079.

Shahbaz, M., and Groz, R. 2009. Inferring mealy machines. In Cavalcanti, A., and Dams, D. R., eds., *FM 2009: Formal Methods*, 207–222. Berlin, Heidelberg: Springer Berlin Heidelberg.