Expediting Reinforcement Learning by Incorporating Knowledge About Temporal Causality in the Environment

Jan Corazza JAN.CORAZZA@TU-DORTMUND.DE
TU Dortmund University
Research Center Trustworthy Data Science and Security

Hadi Partovi Aria HPARTOVI@ASU.EDU
Arizona State University

Daniel Neider DANIEL.NEIDER@TU-DORTMUND.DE
TU Dortmund University
Research Center Trustworthy Data Science and Security

Zhe Xu XZHE1@ASU.EDU
Arizona State University

Editors: Francesco Locatello and Vanessa Didelez

Abstract

Reinforcement learning (RL) algorithms struggle with learning optimal policies for tasks where reward feedback is sparse and depends on a complex sequence of events in the environment. Probabilistic reward machines (PRMs) are finite-state formalisms that can capture temporal dependencies in the reward signal, along with nondeterministic task outcomes. While special RL algorithms can exploit this finite-state structure to expedite learning, PRMs remain difficult to modify and design by hand. This hinders the already difficult tasks of utilizing high-level causal knowledge about the environment, and transferring the reward formalism into a new domain with a different causal structure. This paper proposes a novel method to incorporate causal information in the form of Temporal Logic-based Causal Diagrams into the reward formalism, thereby expediting policy learning and aiding the transfer of task specifications to new environments. Furthermore, we provide a theoretical result about convergence to optimal policy for our method, and demonstrate its strengths empirically.

Keywords: Temporal Causality, Reinforcement Learning, Probabilistic Reward Machines, Formal Methods

1. Introduction

Reinforcement Learning (RL) has emerged as the forefront method in providing a robust and general framework for intelligent, autonomous decision-making and learning within complex environments. One of the biggest challenges in Reinforcement Learning is integrating high-level, causal knowledge into the learning process. Causal reasoning may come naturally to humans, assisting them in navigating the world by making decisions based on more than just observed outcomes; it involves an understanding of how those outcomes come about. This is in contrast to traditional RL techniques, which often lack the ability to capture temporal cause-effect relationships and hence offer inefficient learning and decision-making. For instance, the knowledge of the likely consequences of actions in terms of future states and rewards can dramatically reduce the amount of exploration that needs to take place to learn effective policies. This problem is most evident in settings with long-term

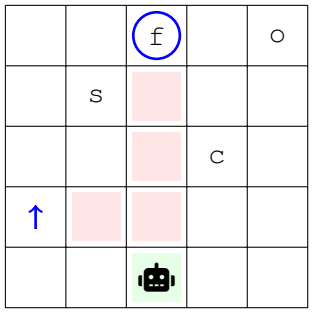
consequences, which calls for RL methods that could incorporate causal knowledge directly into their decision process.

In RL the interaction between the agent and the environment happens step by step. Starting in state s the agent chooses an action a with probability $\pi(a \mid s)$ (the policy), and the environment transitions into a new state s' and gives a reward r. This interaction is formalized in the concept of an MDP, a tuple $M = (S, A, R, p, \gamma)$ where S is the set of states, A the set of actions available to the agent, $R : (S \times A)^\star \times S \to \mathbb{R}$ the reward function mapping trajectories in the MDP to rewards, $p(s' \mid s, a)$ a probabilistic transition function, and $\gamma \in (0, 1)$ the discount factor. The agent's goal is to maximize the expected discounted return, $\max_\pi \mathbb{E}_\pi[\sum_{i=0}^{\infty} \gamma^i r_i]$. A labeling function $L(s, a, s')$ can be provided to attach descriptive propositional variables to transitions in the MDP. An MDP together with a labeling function is called a labeled MDP.

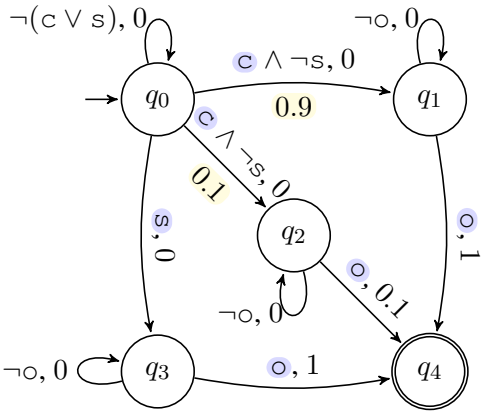
Although MDPs can have a large number of states and a complex transition function, one often has access to high-level causal knowledge of the environment. Figure 1(*a*) illustrates this point on a small example MDP. To complete the task, the agent must choose to bring either coffee or a soda to the office. The high-level knowledge one may supply is that any path from the soda to the office is later blocked by a flower pot, which the agent must avoid. This is due to walls and a one-way door, which constrain the agent's movement. Although special RL algorithms can find the optimal policy for this task, they will not take these temporal-causal constraints into account, and will explore the environment in an inefficient manner. Unfortunately, employing high-level knowledge about causality has shown to be a difficult task, as the current causal RL approaches (e.g., Zhang (2020); Lu et al. (2021); Wang et al. (2021); Bareinboim et al. (2015); Lee and Bareinboim (2018); Mesnard et al. (2021); Forney et al. (2017); Li et al. (2021)) mostly do not take into account the *temporal* aspect of the causal knowledge. This paper aims to address this issue by proposing a novel method that incorporates knowledge about causality directly into the reward function. On the other hand, den Hengst et al. (2022) propose an approach for safe RL that incorporates symbolic reasoning and a temporal domain. However, their primary focus is on ensuring safety rather than expediting the learning process, distinguishing their work from ours, which specifically targets efficient learning by leveraging causality.

1.1. Probabilistic Reward Machines

Common RL algorithms such as Q-learning struggle with tasks where rewards are sparse and depend on a complex sequence of actions that the agent must perform in a specific order. Reward machines, introduced by Icarte et al. (2020) are a finite-state formalism that can capture the reward function in such cases. Q-learning for Reward Machines (QRM), as proposed by Icarte et al. (2020), can exploit this reward structure to expedite learning the optimal policy. The QRM algorithm employs reward machines to significantly enhance the efficiency of problem-solving processes. This algorithm applies an off-policy Q-learning strategy to the reward machine by decomposing it into relevant components at the same time, thereby facilitating the simultaneous learning of each distinct subpolicy. This methodological approach has been empirically validated, demonstrating the algorithm's capability to converge towards an optimal policy in tabular case. Velasquez et al. (2021) introduced a more general variant of reward machines called *probabilistic reward machines* (PRMs). PRMs use a nondeterministic transition function that can capture uncertainty in task outcomes. In the example from Figure 1, uncertainty comes from the fact that the coffee machine may malfunction. Definition 1



(*a*) A labeled 5x5 Gridworld with coffee (c), soda (s), an office (o), and a flower pot (f). The agent can move in the four cardinal directions, and starts in the cell labeled . Other shaded cells are impassable walls. One-way doors are represented by the upwards arrow. The flower pot acts as a sink state.



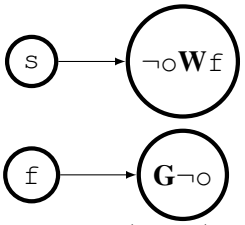
(*b*) A PRM for the task in Figure 1(*a*) (left). Transitions are labeled with propositional formulas and reward outputs. Only transition probabilities different from 1 are shown. State q_4 is a terminal state which ends the task. Formulas on transitions from q_0 to q_1 and q_2 (c∧¬s) contain ¬s in order to disambiguate the transition function in state q_0 on the input c, s. The same could be achieved by using the formula ¬c ∧ s for the transition from q_0 to q_3, and just c for transitions into q_1, q_2.

Figure 1: An MDP (left) and a PRM (right) that captures the task of bringing either coffee or soda to the office. The coffee machine has a probability of 10% to malfunction and produce bad coffee, leading to a reduced reward of 0.1 instead of 1. Bringing soda to the office results in a reward of 1 deterministically. An example input for the PRM is $\{\texttt{c}, \texttt{s}\}, \emptyset, \{\texttt{o}, \texttt{c}\}$ (a sequence of three labels), which will induce the run $q_0 \mapsto q_3 \mapsto q_3 \mapsto q_4$ with a reward of 1. It is important to note that inputs for PRMs are sets of descriptive propositional variables that are true in a given step, hence why a single label such as $\{\texttt{c}, \texttt{s}\}$ can include multiple (or 0) variables.

formalizes this notion of a finite-state representation of a temporally extended task with probabilistic outcomes.

Definition 1 (Probabilistic Reward Machine (PRM)) *A PRM* $\mathsf{A} = (U, u_I, 2^{AP}, \Gamma, \tau, \sigma, F)$ *is a tuple where* U *is a finite set of states with a distinguished initial state* $u_I \in U$*, AP is a set of atomic propositions and* 2^{AP} *is the set of labels,* $\Gamma \subset \mathbb{R}$ *is a finite set of rewards,* $\tau : (U \times 2^{AP} \times U) \to [0, 1]$ *is a probabilistic transition function,* $\sigma : (U \times 2^{AP} \times U) \to \Gamma$ *is a function mapping each transition to a reward in* Γ*, and* $F \subseteq U$ *is a finite set of terminal states that signal the end of the interaction.*

The agent-environment interaction generates a trajectory $s_0, a_0, s_1, \ldots, a_{n-1}, s_n$ and the corresponding label sequence $\ell_0\ell_1 \cdots \ell_{n-1}$, where $L(s_i, a_i, s_{i+1}) = \ell_i$ for all $i = 0, \ldots, n-1$. The state s_0 may be a unique initial state, or drawn from an initial distribution. After reading a label ℓ in state u, the PRM executes a nondeterministic transition into a new state u' with probability $\tau(u, \ell, u')$, and the agent receives a reward $r = \sigma(u, \ell, u')$. A run of a PRM A on a label sequence $\ell_0\ell_1 \cdots \ell_{n-1}$ is a sequence $u_0, r_0, u_1, \ldots, r_{n-1}, u_n$ where $u_0 = u_I$, and for all $i = 0, \ldots, n-1$, $\tau(u_i, \ell_i, u_{i+1}) > 0$ and $\sigma(u_i, \ell_i, u_{i+1}) = r_i$.



(*a*) $\mathbf{G}(\mathtt{s} \rightarrow \neg\bigcirc\mathbf{W}\mathtt{f}) \wedge \mathbf{G}(\mathtt{f} \rightarrow \mathbf{G}\neg\bigcirc)$

(*b*) $\mathbf{G}(\mathtt{a} \rightarrow \mathbf{G}\neg\mathtt{b})$

Figure 2: Figure 2(*a*) (left) is the TL-CD which captures relevant causal information in the environment from Figure 1(*a*). Figure 2(*b*) (right) is a TL-CD that holds for the case study in Figure 6.

1.2. Temporal Logic-based Causal Diagrams

Linear temporal logic over finite sequences (LTL_f) is a formal reasoning system that can capture causal and temporal properties of label sequences and labeled MDPs. Aside from Boolean operators like $\neg$ and $\vee$, LTL_f introduces temporal operators such as $\mathbf{G}\psi$ (true if and only if ψ holds for every element in the sequence), $\mathbf{X}\psi$ (true iff. ψ holds for the next element of the sequence), and $\psi\mathbf{U}\varphi$ (true iff. ψ holds until φ becomes true, and φ is true in some element of the sequence). We also rely on the weak until operator $\psi\mathbf{W}\varphi$ (true iff. ψ holds until φ becomes true, but φ is not required to become true).

In order to encode knowledge about causality in the underlying MDP, we rely on Temporal Logic-based Causal Diagrams (TL-CDs) introduced in Paliwal et al. (2023). TL-CDs are a special notation that expresses the causal relationship between formulas in LTL_f. The first conjunct induced by the TL-CD in Figure 2(*a*), $\mathbf{G}(\mathtt{s} \rightarrow \neg\bigcirc\mathbf{W}\mathtt{f})$, means that if the agent observes $\mathtt{s}$ (soda) in any step, then it will not observe $\bigcirc$ (the office) before it observes $\mathtt{f}$ (the flower pot). This part of the TL-CD encodes knowledge that soda may only be reached via a one-way door, and the only other exit towards the office will be blocked by the flower pot.

Formally, a TL-CD is a directed graph whose nodes are labeled with LTL_f formulas. For a TL-CD $\mathcal{C}$ one may construct an equivalent LTL_f formula $\varphi^{\mathcal{C}}$ through Equation 1, where $\varphi \blacktriangleright \psi$ iterates over edges that connect formulas φ and ψ in the TL-CD.

$$\varphi^{\mathcal{C}} = \bigwedge_{\varphi \blacktriangleright \psi} \mathbf{G}(\varphi \rightarrow \psi) \tag{1}$$

If $\varphi^{\mathcal{C}}$ is true for a label sequence ℓ, we will write $\ell \models \varphi^{\mathcal{C}}$. A label sequence $\ell = \ell_0\ell_1 \cdots \ell_{n-1}$ is attainable in an MDP $M = (S, A, R, p, \gamma)$ if there exists a trajectory $s_0, a_0, s_1, \ldots, a_{n-1}, s_n$ in M such that $L(s_i, a_i, s_{i+1}) = \ell_i$ and $p(s_i, a_i, s_{i+1}) > 0$ for all $i = 0, 1, \ldots, n-1$. We will say that a TL-CD $\mathcal{C}$ holds for an MDP M if for every label sequence ℓ attainable in M, we have $\ell \models \varphi^{\mathcal{C}}$. In order to simplify working with TL-CDs, we leverage the notion of deterministic finite automata (DFAs). We formalize this notion in Definition 2.

Definition 2 (Deterministic Finite Automaton (DFA)) *A DFA is a tuple* $\mathsf{C} = (Q, q_I, \Sigma, \delta, F)$ *consisting of a finite set of states* Q *with an initial state* q_I*, input alphabet* Σ*, deterministic transition function* $\delta : Q \times \Sigma \rightarrow Q$*, and a finite set of accepting states* $F \subseteq Q$*.*

If the run of the DFA C on an input string ℓ ends in an accepting state $q \in F$, we will write $\ell \in \mathcal{L}(\mathsf{C})$. Every TL-CD $\mathcal{C}$ can be converted into an equivalent DFA C, in the sense that for every ℓ, we have $\ell \in \mathcal{L}(\mathsf{C}) \iff \ell \models \varphi^{\mathcal{C}}$. We will refer to C as the *causal DFA*.

2. Problem statement

One may use QRM to find the optimal policy for the task in Figure 1. However, the PRM in Figure 1(*b*) does not take flower pots and one-way doors into account. Because the agent does not know that knocking over flower pots is forbidden or that choosing soda causes him to enter a room blocked by a flower pot, it will waste time exploring those fruitless trajectories. As PRMs are in essence task specifications, and one may also wish to transfer them into a new environment while preserving the overall goal. In both cases, high-level insights about causality, especially its temporal aspects, could prove helpful by reflecting the dynamics of the MDP in condensed form.

Unfortunately, incorporating knowledge about temporal causality into the reward function remains a difficult and error-prone manual task. In PRMs, this would necessitate adding new states and reasoning about a different, more complicated transition function. Some methods such as JIRP, proposed by Xu et al. (2019) and SRMI, proposed by Corazza et al. (2022) assume that a suitable but unknown representation of the reward function exists, and attempt to recover it from interaction traces. This work proposes an alternative method that leverages TL-CDs in order to automate the process of incorporating knowledge about causality into PRMs. More formally, the problem can be stated as follows. Given a TL-CD $\mathcal{C}$ which holds for an MDP M and a PRM A, produce a PRM B that induces the same optimal policy as A, but utilizes causal information in $\mathcal{C}$ to expedite learning.

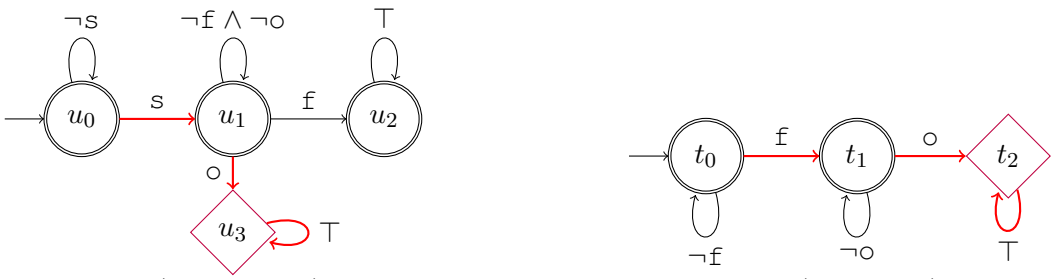
3. Method

This section introduces a novel approach for incorporating causal information from a TL-CD into the reinforcement learning process via a PRM. By creating a product of a TL-CD, represented as a causal DFA, with a PRM, we aim to enhance the RL agent's ability to efficiently learn and make decisions in complex environments.

We first consider the equivalent causal DFA for a given TL-CD. As explained in Section 1.2, the equivalent causal DFA captures the same semantics as the given TL-CD. While TL-CDs are an intuitive notational tool, DFAs are easier to work with computationally. The causal DFA for the TL-CD in Figure 2(*a*) is shown in Figure 3 (in two parts for convenience). State u_3 is a *sink* state, meaning that any run of the DFA which enters u_3 will never leave it. It is also a *rejecting* state. Taken together, this means that any label sequence for which the causal DFA enters u_3 is not the prefix of an attainable sequence in an MDP M if we assume that the TL-CD holds for M.

In other words, when a causal DFA run reaches a rejecting sink state on some prefix of input labels, then the entire input label sequence is unattainable. The reason is that there is no suffix of labels which can cause the run to transition into an accepting state. From now on we will implicitly consider causal DFAs to have at most one rejecting sink state, and that an accepting state is reachable from all other states. This can be achieved by minimization (Sipser (2013)).

We propose to incorporate causal information from a TL-CD $\mathcal{C}$ into a PRM A by computing state values in a new PRM $\mathsf{B}_1 = \mathcal{C} \times \mathsf{A}$, which is a product of the TL-CD (represented by the causal DFA C) and A. The product PRM B_1 synchronizes the runs of the original PRM A and the causal DFA C. B_1 mirrors the output of A, except when C transitions into a rejecting sink state. Then the output of



(*a*) $\mathbf{G}(\mathtt{s} \rightarrow \neg\mathtt{o}\mathbf{W}\mathtt{f})$ (*b*) $\mathbf{G}(\mathtt{f} \rightarrow \mathbf{G}\neg\mathtt{o})$

Figure 3: Two factors of the causal DFA for the TL-CD in Figure 2(*a*). Rejecting sink states are diamond-shaped. Their parallel composition is the true causal DFA, and its states come from the Cartesian product of states in this Figure. For example, the initial state is (u_0, t_0).

B_1 is set to a minimal value m that is lesser than any possible immediate reward and resulting future gain, and will remain there for the rest of the run as C will not leave the sink state. We also compute state values in a "pessimistic" PRM $\mathsf{B}_2 = \mathcal{C} \times (-\mathsf{A})$ in order to uncover temporal-causal information about worst-case reward outcomes. While B_1 outputs the same rewards as A, B_2 negates outputs of A (but also gives minimal outputs m for transitions into rejecting sink states). Because of the minimal reward output m, value iteration in either B_1 or B_2 will disregard transitions that lead C into a rejecting sink state, as explained under Figure 4. Due to negating reward outputs, label sequences that maximize return in B_2, minimize the return in B_1. We combine state value information from B_1 and B_2 into a final PRM B. To obtain B, we start from B_1, and add all states $u \in U^{\mathsf{B}_1}$ that have 0 value in both the machine B_1 and B_2 ($v^\star_{\mathsf{B}_1}(u) = v^\star_{\mathsf{B}_2}(u) = 0$) into the set of terminal states F^{B_1}. Such states have the property that no matter the policy, the future return is constrained with 0 from above and below (and thus, the choice of actions is of no consequence). The product $\mathcal{C} \times \mathsf{A}$ is formalized in Definition 3. We define the value of a PRM state u via the Bellman optimality equation 2, where γ matches the discount factor in the MDP.

$$v^\star(u) = \max_{\ell \in 2^{\mathrm{AP}}} \sum_{u' \in U} \tau\left(u, \ell, u'\right) \cdot \left(\sigma\left(u, \ell, u'\right) + \gamma v^\star\left(u'\right)\right) \tag{2}$$

As Equation 2 is an optimality equation, $v^\star(u)$ is the expected return of a PRM run starting in u and following the most optimistic label sequence (which may or may not be attainable in the MDP). We define the minimal reward output m as $m = -1 - \max_{r \in \Gamma^{\mathsf{A}}} |r| - \max_{u \in U^{\mathsf{A}}} v^\star(u)$. While it may be simpler to use $m = -\infty$, we compute a concrete bound in order to better communicate how our method makes use of state value information. In brief, the formula for m is inspired by the Bellman optimality operator used in value iteration. The terms can be explained in the following way. First, $-\max_{r \in \Gamma^{\mathsf{A}}} |r|$ ensures that the reward is lower than any other immediate reward in the original PRM. Second, $-\max_{u \in U^{\mathsf{A}}} v^\star(u)$ ensures that the reward is lower than any possible future gain starting from a state in the original PRM. Taken together, these two terms ensure that transitions that correspond to rejecting sink states do not contribute to state values.

Definition 3 (PRM & TL-CD product) *Let $M = (S, A, R, p, \gamma)$ be an MDP where the reward function $R : (2^{AP})^\star \to \Gamma$ is given by the PRM $\mathsf{A} = (U^{\mathsf{A}}, u^{\mathsf{A}}_I, 2^{AP}, \Gamma^{\mathsf{A}}, \tau^{\mathsf{A}}, \sigma^{\mathsf{A}}, F^{\mathsf{A}})$, $\mathcal{C}$ a TL-CD that holds for M, and $\mathsf{C} = (Q, q_I, 2^{AP}, \delta, F_{\mathsf{C}})$ its equivalent minimal causal DFA with states Q, initial*

state q_I, a set of accepting states $F_{\mathsf{C}} \subseteq Q$, and transition function δ. Let $Q_{r.s.} \subseteq Q \setminus F_{\mathsf{C}}$ be the set of rejecting sink states of C.

We define the product $\mathcal{C} \times \mathsf{A}$ as a new PRM $(U, u_I, 2^{AP}, \Gamma, \tau, \sigma, F)$, where

1. *$U = U^{\mathsf{A}} \times Q$, a state of $\mathcal{C} \times \mathsf{A}$ is a pair of states (u, q) with $u \in U^{\mathsf{A}}$ and $q \in Q$;*
2. *$u_I = (u_I^{\mathsf{A}}, q_I)$, the initial state in $\mathcal{C} \times \mathsf{A}$ is the pair of initial states of A and C;*
3. *$\Gamma = \Gamma^{\mathsf{A}} \cup \{m\}$, the output alphabet of $\mathcal{C} \times \mathsf{A}$ is expanded with a possible reward output that is 1 less than any output in the original set of rewards from A;*
4. *$\tau((u, q), \ell, (u', q')) = \tau^{\mathsf{A}}(u, \ell, u') \cdot \mathbb{1}_{\{\delta(q,\ell)=q'\}}$, the probability of $\mathcal{C} \times \mathsf{A}$ transitioning from (u, q) to (u', q') upon reading ℓ is the same as the probability of A transitioning from u to u', given that C transitions from q to q' (otherwise, the probability is 0);*
5. *$\sigma((u, q), \ell, (u', q')) = \begin{cases} \sigma^{\mathsf{A}}(u, \ell, u') & q' \notin Q_{r.s.} \\ m = -1 - \max_{r \in \Gamma^{\mathsf{A}}} |r| - \max_{u \in U^{\mathsf{A}}} v^\star(u) & \text{otherwise} \end{cases}$, the output of the product PRM agrees with A except when C transitions into a rejecting sink state; and*
6. *$F = \{(u, q) : u \in F^{\mathsf{A}}\}$, terminal states in $\mathcal{C} \times \mathsf{A}$ correspond to terminal states in A.*

Performing value iteration acts as a form of look-ahead in the product $\mathcal{C} \times \mathsf{A}$, whose output function is defined so that transitions which lead the causal DFA into a rejecting sink state do not contribute to overall state value. The same is true for $B_2 = \mathcal{C} \times (-\mathsf{A})$, which is defined in the same way, except the output function $-\sigma^{\mathsf{A}}(u, \ell, u')$ provides look-ahead information about the worst-case future outcome. Our method, given in Algorithm 1, improves the convergence speed of QRM by utilizing information about expected rewards that better reflects the temporal causal structure of the environment. In Theorem 1 we show that our method converges to the optimal policy in the limit.

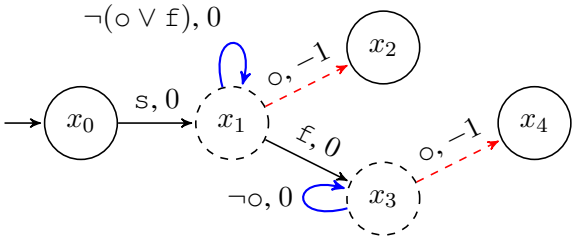


Figure 4: A fragment of the product of the PRM from Figure 1(*b*) and the TL-CD from Figure 2(*a*). Inheriting the q, u, and t names of PRM and causal DFA states from previous figures, $x_0 = (q_0, u_0, t_0)$, $x_1 = (q_3, u_1, t_0)$, $x_2 = (q_4, u_3, t_0)$, $x_3 = (q_3, u_2, t_1)$, and $x_4 = (q_4, u_2, t_2)$. Due to the maximum in Equation 2, dashed transitions do not contribute to state value. Dashed states x_1 and x_3 have 0 value in both B_1 (depicted) and B_2, and will be added to the set of terminal states.

Theorem 1 (Convergence to Optimal Policy) *Let M be an MDP with a non-Markovian reward function captured by PRM A. Let $\mathcal{C}$ be a TL-CD that holds for M, and C the corresponding minimal causal DFA with rejecting sink states $Q_{r.s.}$. Then Algorithm 1 converges to an optimal policy for M with respect to A. In particular, we can easily recover the optimal policy for (M, A) from the optimal policy for (M, B) found in the algorithm.*

Algorithm 1: Reinforcement Learning With Temporal-Causal Information
Require : MDP M, PRM A, minimal causal DFA C with rejecting sink states $Q_{\text{r.s.}}$

$\mathsf{B}_1, \mathsf{B}_2 \leftarrow$ computeProduct(A, C), computeProduct$(-\mathsf{A}, \mathsf{C})$
$v^\star_{\mathsf{B}_1}, v^\star_{\mathsf{B}_2} \leftarrow$ valueIteration(B_1, γ), valueIteration(B_2, γ)
$\mathsf{B} \leftarrow \mathsf{B}_1$
for each $u \in U^{\mathsf{B}}$ **do**
  if $v^\star_{\mathsf{B}_1}(u) = v^\star_{\mathsf{B}_2}(u) = 0$ **then**
    Add u to the set of terminal states of B
  end
end
$Q \leftarrow$ initializeQFunction$()$
while *termination criteria not met* **do**
  $Q \leftarrow$ RunQRMEpisode(Q, B)
end
return Q

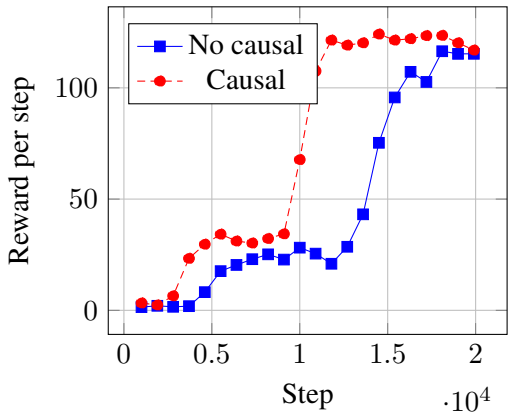
In the full proof of Theorem 1, we introduce 3 transformations on PRMs that realize our method of combining a PRM with a TL-CD. In brief, these transformations allow us to (1) take the parallel composition of a PRM and a DFA, (2) change the outputs of PRMs on transitions into unreachable states, and (3) add states into the set of terminal states of the PRM (under certain conditions); all the while preserving the optimal policy. We then show how Algorithm 1 applies these transformations in order to arrive at the desired PRM B. Since these transformations preserve the optimal policy (or allow for easy recovery of it), we conclude that QRM using the transformed PRM B converges to the optimal policy. In brief, the core contribution of the modifications we propose lies in the ability to exploit causal knowledge contained in the TL-CD. Although we significantly change the structure of PRM by combining it with the TL-CD, we prove that the optimal policy remains the same. However, the changed structure allows for a more nuanced calculation of PRM state values that is not blind to the temporal-causal relations that hold for the MDP. More precisely, we are able to obtain upper and lower bounds on state values, and prove that under certain conditions one does not need to explore the MDP (specifically, when both the upper and lower value bounds are 0). In doing so, we improve the balance between exploration and exploitation and increase the sample efficiency of our algorithm. See the Appendix for further details about the function that computes the PRM and causal DFA intermediate product, and the full proof of Theorem 1.

4. Case Studies

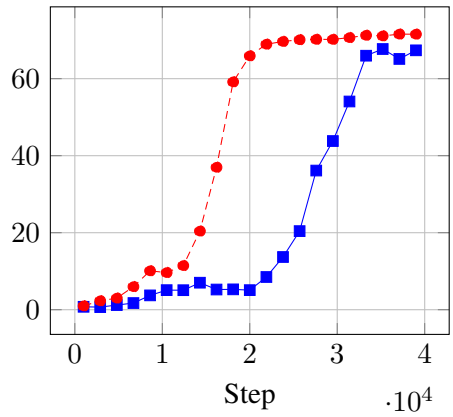
Our method shows promising results across two case studies. The first case study (results in Figure 5(*a*)) is based on the *coffee vs. soda* example from Figure 1. The second case study (results in Figure 5(*b*)) is described in Figure 2.

We compared our method against QRM without access to knowledge about causality. In both case studies, our method takes significantly fewer steps to converge to the optimal policy.

For a more thorough comparison and analysis of the method's efficiency, we implemented it across two distinct case studies: a four-door task and a small office world domain. The third case study entails an agent navigating through a scenario where it must open four doors in any arbitrary

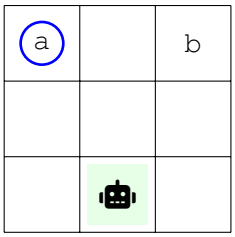


(*a*) Coffee vs. soda task

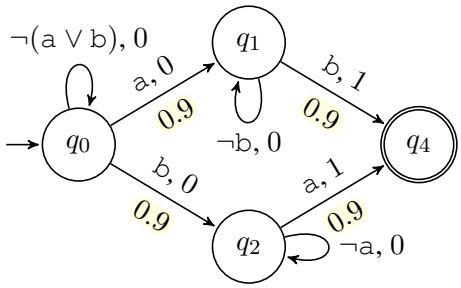


(*b*) Two-doors task

Figure 5: Reward per step averaged over 20 runs. "No causal" refers to using QRM with the original PRM that does not account for additional causal information in the environment. "Causal" are the results for our method. Both graphs showcase QRM convergence to the optimal policy.



(*a*) `3x3` Gridworld environment where the agent must open both door A (`a`) and door B (`b`) in any order. However, the cell with door A traps the agent. The agent can fail at opening the doors with probability 0.1.



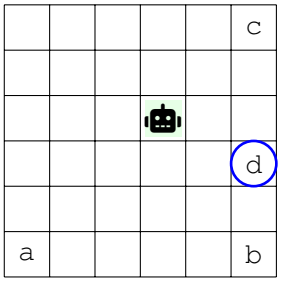
(*b*) The PRM without causal info about the two-door task. Missing transitions are all self-loops with probability 0.1.

Figure 6: The MDP and PRM for the second case study. The TL-CD that adds causal information regarding the sink door A can be found on Figure 2(*b*). It states that after seeing door A, the agent can not later see door B.

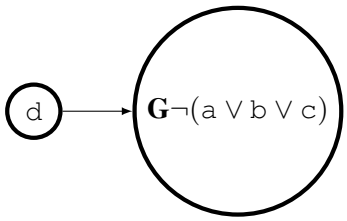
order, as illustrated in Figure 7(*a*). This task involves a significantly more complex PRM owing to the number of possible orders. To evaluate the method's performance and its efficacy in this case study, we use a grid world configuration of 6×6.

The agent here must open door A, door B, door C, and door D in any order. However, door D is a trap, and the agent cannot see doors A, B, or even C after seeing door D. This knowledge, in fact, is encoded in Figure 7(*b*). As this task requires a complex Probabilistic Reward Machine (PRM), we deemed it prudent to relegate its detailed explanation to the Appendix.

Furthermore, Figure 9(*a*) compares our method on the four-doors task to QRM without additional causal information. It can be seen that QRM with causal information results in much higher rewards with faster convergence.



(*a*) 6x6 Gridworld environment where the agent must open door A (a), door B (b), door C (c), and door D (d) in any order. However, the cell with door D traps the agent. The agent may fail to open door B with a probability of 0.1, and it will go to door D instead.



(*b*) TL-CD that holds for the case study of four-doors task.

Figure 7: The MDP and Causal DFA for the third case study.

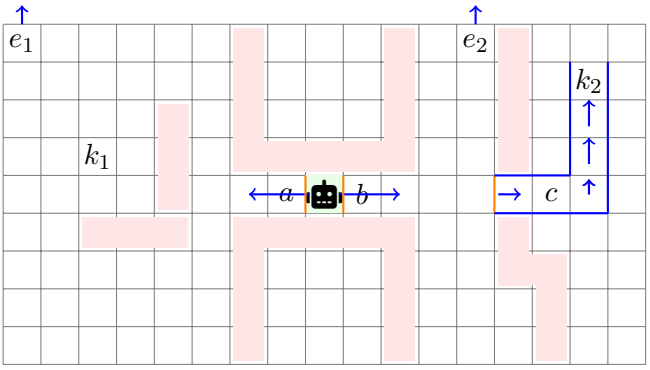
Another case study in which we implemented this method is the small office world domain. For this specific exploration, we considered a small office world with a spatial layout of 17 $\times$ 9, similar to the setup in Paliwal et al. (2023). Within the scope of this case study, the procedure to exit the grid entails a two-step process for the agent: first, it must obtain one of the two available keys, denoted as k_1 or k_2, and then navigate to exit e_1 or e_2, correspondingly aligned with the key acquired. Through one-way doors (indicated by blue arrows), keys, and walls, the agent interacts with the environment. A graphical illustration of this environment, capturing the elements and challenges the agent faces, is provided in Figure 8(*a*), providing a better understanding of the structural and operational complexities of the small office world being explored.

As a result of c being a one-way door, the agent will not be able to pick up key k_2 and exit at e_2, due to the information encoded in figure 8(*c*). In addition, if the agent passes through the door b, it will not be able to exit through the door e_1. Furthermore, Figure 8(*b*) displays the PRM, omitting the causal information regarding the small office world. In order to succeed in exiting the maze and receiving reward 1, the agent must complete both sequences a$-k_1 - e_1$ (open door a, pick up key k_1, and leave at e_1) or b$-k_2 - e_2$ (open door b, pick up key k_2, and exit at e_2). However, a probability of 0.9 suggests a likelihood of the agent exiting through e_1, while a probability of 0.1 indicates a risk of the agent getting stuck.

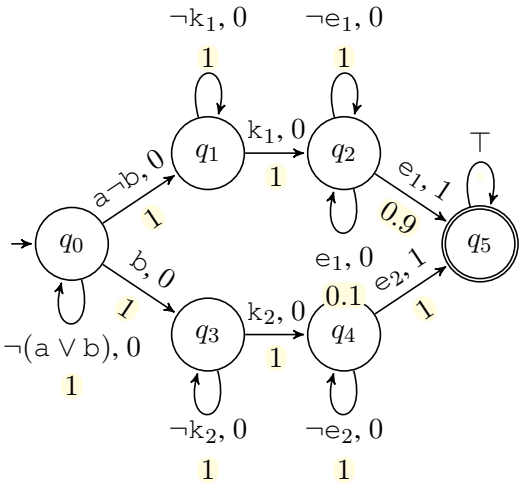
Figure 9(*b*) depicts the performance comparison of our method on the small office world scenario to QRM without additional causal information. In the figure, it can be seen that if the RL agent knows the causal DFA and learns never to open door b, the agent can obtain their optimal reward faster with higher accumulated rewards.

4.1. Performance Impact of Proposed Modifications

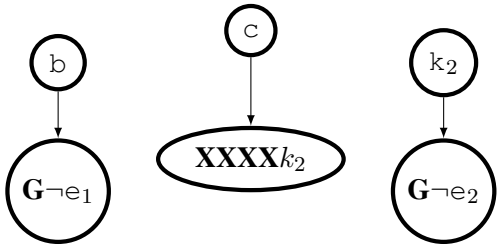
Our method makes significant structural changes to the underlying PRM in order to incorporate temporal-causal information contained in TL-CDs. One of the primary effects of these changes is increasing the size of the state space. Despite this increase in size, experimental results demonstrate that our algorithm improves convergence properties on realistic examples when the provided informa-



(*a*) Map of the small office world. Shaded cells are impassable walls.



(*b*) The PRM without causal info about the small office world.

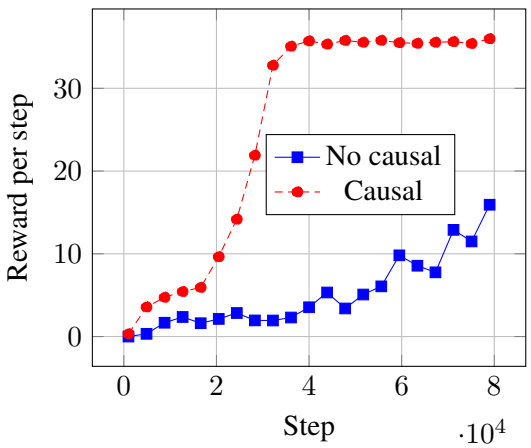


(*c*) TL-CD that holds for the case study of the small office.

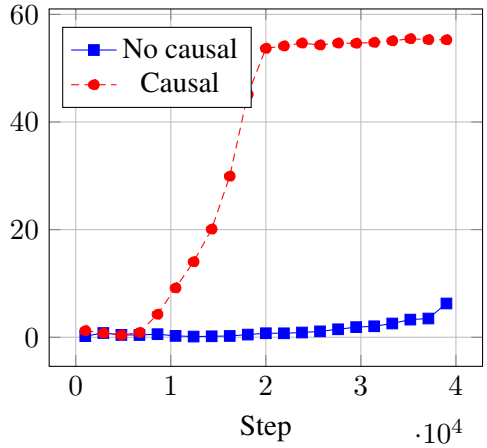
Figure 8: The PRM and Causal DFA for the fourth case study, alongside the office map for reference.

tion is *useful* (allows for pruning redundant paths from the PRM where exploration is not necessary). Strictly speaking, however, we only require that the causal diagrams are *correct* (that they hold for the MDP), not that they contain useful information.

Being robust to mistakes and imprecisions in user-provided causal information (or even malicious inputs) is beneficial. We hypothesize that our algorithm possesses this property, in the sense that its performance does not diminish in the presence of useless or redundant knowledge. In order to check this, we re-ran our experiments in this more difficult setting of perturbed user input. We added useless and redundant causal knowledge, by including an additional factor in the causal DFA product. More precisely, we called `computeProduct` an additional time with a redundant causal DFA with no rejecting sink states. This factor injects no new useful causal information, but increases the state space size 5 times. The results we obtained were fully in line with the ones presented in our case studies, i.e. the improved convergence rate enabled by our method was retained. In conclusion, our algorithm can handle useless or redundant knowledge, and its performance is not diminished by it. We showcase these results in Figure 10.



(*a*) Four-doors task results.



(*b*) Small office world results.

Figure 9: Comparison of task results, using the same reward per step metric averaged over 20 runs.

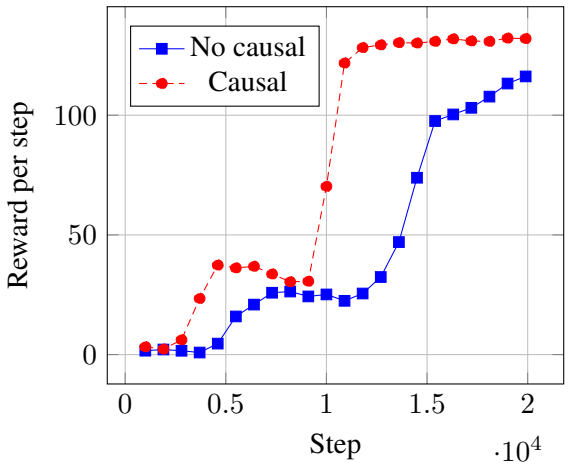


Figure 10: The Coffee vs. soda task. "Causal" are the results for our method, but with additional, redundant causal knowledge added. The resulting PRM has 5 times more states than the one from Figure 5(*a*), but our method achieves the same performance. Like before, "No causal" refers to using QRM with the original PRM (which has neither causal information nor redundancy added).

We make two additional observations relating to state space size.

1. PRMs themselves already provide an immense reduction in state space size, because the policy no longer has to consider the whole history. Instead, the PRM state acts as a form of finite memory in the MDP.

2. Our method is based on removing redundant paths from the PRM (those where exploration is not necessary).

We want to further drive home the point that while using PRMs and incorporating causal information as we propose does contribute to an increased state space, the resulting performance

benefits more than outweigh the costs (and, as our additional analysis shows, some costs are fully avoided).

5. Conclusion and Further Work

The method proposed in this paper addresses the difficult problem of accounting for knowledge about temporal causality in the RL environment. We have shown that an expressive and concise description of temporal and causal relations in the form of a Temporal Logic-based Causal Diagram can be integrated into the reward function formalism. Furthermore, we have shown how the added information about temporal and causal relations can be leveraged to expedite learning without changing the optimal policy.

While our method performs well in case studies, we are convinced that this work can be continued to integrate knowledge about causality even more tightly into the reward function. In particular, look-ahead information contained in state-values of the product PRM may be further utilized by methods like reward shaping. We are also interested in further exploring the interplay between probabilistic outcomes and causal information.

Acknowledgments

This work was supported in part by the National Science Foundation (NSF) under Grants CNS 2304863 and CNS 2339774, and in part by the Office of Naval Research (ONR) under Grant N00014-23-1-2505. Additionally, this work has been financially supported by the Research Center Trustworthy Data Science and Security [1], one of the Research Alliance centers within the UA Ruhr [2].

References

Elias Bareinboim, Andrew Forney, and Judea Pearl. Bandits with unobserved confounders: A causal approach. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/795c7a7a5ec6b460ec00c5841019b9e9-Paper.pdf.

Jan Corazza, Ivan Gavran, and Daniel Neider. Reinforcement learning with stochastic reward machines. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6):6429–6436, Jun. 2022. doi: 10.1609/aaai.v36i6.20594. URL https://ojs.aaai.org/index.php/AAAI/article/view/20594.

F. den Hengst, V. François-Lavet, M. Hoogendoorn, et al. Planning for potential: efficient safe reinforcement learning. *Machine Learning*, 111:2255–2274, 2022. doi: 10.1007/s10994-022-06143-6. URL https://doi.org/10.1007/s10994-022-06143-6.

Andrew Forney, Judea Pearl, and Elias Bareinboim. Counterfactual data-fusion for online reinforcement learners. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*,

1. https://rc-trust.ai
2. https://uaruhr.de

pages 1156–1164. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/forney17a.html.

Rodrigo Toro Icarte, Toryn Q. Klassen, Richard Anthony Valenzano, and Sheila A. McIlraith. Reward machines: Exploiting reward function structure in reinforcement learning. *CoRR*, abs/2010.03950, 2020. URL https://arxiv.org/abs/2010.03950.

Sanghack Lee and Elias Bareinboim. Structural causal bandits: Where to intervene? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/c0a271bc0ecb776a094786474322cb82-Paper.pdf.

Jin Li, Ye Luo, and Xiaowei Zhang. Causal reinforcement learning: An instrumental variable approach. *ERN: Computational Techniques (Topic)*, 2021.

Yangyi Lu, Amirhossein Meisami, and Ambuj Tewari. Causal Markov decision processes: Learning good interventions efficiently. *CoRR*, abs/2102.07663, 2021. URL https://arxiv.org/abs/2102.07663.

Thomas Mesnard, Theophane Weber, Fabio Viola, Shantanu Thakoor, Alaa Saade, Anna Harutyunyan, Will Dabney, Thomas S. Stepleton, Nicolas Heess, Arthur Guez, Eric Moulines, Marcus Hutter, Lars Buesing, and Rémi Munos. Counterfactual credit assignment in model-free reinforcement learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 7654–7664. PMLR, 2021. URL http://proceedings.mlr.press/v139/mesnard21a.html.

Yash Paliwal, Rajarshi Roy, Jean-Raphaël Gaglione, Nasim Baharisangari, Daniel Neider, Xiaoming Duan, Ufuk Topcu, and Zhe Xu. Reinforcement learning with temporal-logic-based causal diagrams. In Andreas Holzinger, Peter Kieseberg, Federico Cabitza, Andrea Campagner, A. Min Tjoa, and Edgar Weippl, editors, *Machine Learning and Knowledge Extraction*, pages 123–140, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-40837-3.

Michael Sipser. *Introduction to the Theory of Computation*. Course Technology, Boston, MA, third edition, 2013. ISBN 113318779X.

Alvaro Velasquez, Andre Beckus, Taylor Dohmen, Ashutosh Trivedi, Noah Topper, and George K. Atia. Learning probabilistic reward machines from non-markovian stochastic reward processes. *CoRR*, abs/2107.04633, 2021. URL https://arxiv.org/abs/2107.04633.

Lingxiao Wang, Zhuoran Yang, and Zhaoran Wang. Provably efficient causal reinforcement learning with confounded observational data. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 21164–21175, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/b0b79da57b95837f14be95aaa4d54cf8-Abstract.html.

Zhe Xu, Ivan Gavran, Yousef Ahmad, Rupak Majumdar, Daniel Neider, Ufuk Topcu, and Bo Wu. Joint inference of reward machines and policies for reinforcement learning. *CoRR*, abs/1909.05912, 2019. URL http://arxiv.org/abs/1909.05912.

Junzhe Zhang. Designing optimal dynamic treatment regimes: A causal reinforcement learning approach. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11012–11022. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/zhang20a.html.

Section A contains the algorithm for the `computeProduct` function used in Algorithm 1. Section B contains the proof of Theorem 1. In Section C, the PRM related to the four-door task is depicted.

Appendix A. Computing the PRM and causal DFA product

Algorithm 2 computes the (intermediate) product B_1 (B_2) of a PRM A and causal DFA C. This function is called in Line 1 of Algorithm 1 in Section 3. Note that this function does not perform value iteration, this is done in Line 2 of Algorithm 1. The set of terminal states in the intermediate product B_1 is a subset of the set of terminal states in the final product B.

The `transitions` dictionary used in Algorithm 2 represents the transition and output functions of B_1. It maps triplets $((u,q),\ell,(u',q'))$ to to pairs (p,r), where $p = \tau^{\mathsf{B}_1}((u,q),\ell,(u',q'))$, and $r = \sigma^{\mathsf{B}_1}((u,q),\ell,(u',q'))$.

Appendix B. Formal Statements and Proofs

Throughout this Section, let $M = (S, A, R, p, L, \gamma)$ be a labeled MDP, and let $\mathsf{A} = (U, u_I, 2^{\mathrm{AP}}, \Gamma, \tau, \sigma, F)$ be a PRM that encodes R in M. A state $s \in S$ is reachable if there exists an attainable trajectory $s_0, a_0, s_1, \ldots, a_{n-1}, s_n$ in M such that $s = s_i$ for some $i = 0, \ldots, n$.

Lemma 1 (Transformation 1) *Let $\mathsf{A} = (U, u_I, 2^{AP}, \Gamma, \tau, \sigma, F)$ be a PRM, C a DFA with transition function δ, and $\mathsf{A} \times \mathsf{C}$ their parallel composition with output function $\sigma^{\mathsf{A}\times\mathsf{C}}((u,q),\ell,(u',q')) = \sigma(u,\ell,u')$. Let $\pi^\star(s,(u,q))$ be an optimal policy in the product MDP $M \times (\mathsf{A} \times \mathsf{C})$. Then $\pi(s,u) = \pi^\star(s,(u,F(u)))$ is an optimal policy in the product MDP $M \times \mathsf{A}$, where $F_s : U \to Q$ is a tiebreaking reachability function that maps u to a fixed but arbitrary q such that (s,u,q) is reachable in $M \times (\mathsf{A} \times \mathsf{C})$.*

Proof Since Q-learning with an ϵ-greedy exploration strategy visits every reachable state infinitely often in the limit, we can learn a mapping F_s along with the optimal policy for $M \times (\mathsf{A} \times \mathsf{C})$, by setting $F_s(u) = q$ once we reach (s,u,q) for the first time, and we have not yet observed (s,u,q') for any $q' \in Q$. Then, from the fact that

1. Q-learning learns the optimal Q-function in all reachable states, and
2. the optimal Q-function $q^\star(s,u,q,a)$ for $M \times (\mathsf{A} \times \mathsf{C})$ is the same as $q^\star(s,u,a)$ for $M \times \mathsf{A}$ once we project out the DFA component $q \in Q$,

it will follow that we can use F_s to construct the optimal policy in $M \times \mathsf{A}$.

Let $q^\star(s,u,a)$ be the optimal Q-function for $M \times \mathsf{A}$, and $q^\star(s,u,q,a)$ the optimal Q-function for $M \times (\mathsf{A} \times \mathsf{C})$. We will proceed by showing that $q^\star(s,u,q_1,a) = q^\star(s,u,q_2,a)$ for all $q_1, q_2 \in Q$.

Let $\tilde{q}(s,u,q,a) = q^\star(s,u,a)$. We will show that $\tilde{q}$ is a solution to the state value Bellman optimality equation for $M \times (\mathsf{A} \times \mathsf{C})$, given in Equation 3. In the following formulas, let $\ell = L(s,a,s')$.

Algorithm 2: computeProduct(A, C)

Input : PRM A, minimal causal DFA C with rejecting sink states $Q_{\text{r.s.}}$

```
appears ← A.appears ∪ C.appears ; // M.appears is the set of relevant atomic propositions in M.
pairToSelfStateMap ← {} ; // Dictionary mapping pairs of states in A and C to a single state in B′
selfToPairStateMap ← {}
nonTerminalStates ← ∅ ; // Contains non-terminal states of the intermediate product PRM.
terminalStates ← ∅ ; // Contains terminal states of the intermediate product PRM.
transitions ← {} ; // Dictionary representation of τ and σ functions of B′.
stateCounter ← 0
for u in A.states do
    for q in C.states do
        pairToSelfStateMap[(u, q)] ← stateCounter
        selfToPairStateMap[stateCounter] ← (u, q)
        if u in A.terminalStates then
            terminalStates ← {stateCounter} ∪ terminalStates
        end
        else
            nonTerminalStates ← {stateCounter} ∪ nonTerminalStates
        end
        stateCounter ← stateCounter + 1
    end
end
for u in A.nonTerminalStates do
    for q in C.states do
        state ← pairToSelfStateMap[(u, q)]
        transitions[state] ← {}
        for inputSymbol in GENERATEINPUTS(appears) do
            inputSymbolPrm ← A.appears ∩ inputSymbol
            inputSymbolDfa ← C.appears ∩ inputSymbol
            transitions[state][inputSymbol] ← {}
            nextDfaState ← C.transitions[q][inputSymbolDfa]
            for nextPrmState in A.transitions[u][inputSymbolPrm] do
                nextState ← pairToSelfStateMap[(nextPrmState, nextDfaState)]
                probability, reward ← A.transitions[u][inputSymbolPrm][nextPrmState]
                if nextDfaState in Q_r.s. then
                    reward ← m
                end
                transitions[state][inputSymbol][nextState] ← (probability, reward)
            end
        end
    end
```

$$q^\star(s,u,q,a) = \sum_{\substack{s'\in S\\ u'\in U\\ q'\in Q}} p(s',u',q' \mid s,u,q,a)\left(\sigma^{\mathsf{A}\times\mathsf{C}}((u,q),\ell,(u',q')) + \gamma \max_{a'\in A} q^\star(s',u',q',a')\right)$$
$$= \sum_{\substack{s'\in S\\ u'\in U}} p(s',u' \mid s,u,a)\left(\sigma^{\mathsf{A}}(u,\ell,u') + \gamma \max_{a'\in A} q^\star(s',u',\delta(q,\ell),a')\right) \tag{3}$$

The second equality holds because

1. $p(s',u',q' \mid s,u,q,a) = 0$ for $\delta(q,\ell) \neq q'$,
2. $p(s',u',q' \mid s,u,q,a) = p(s',u' \mid s,u,a)$ for $\delta(q,\ell) = q'$, and
3. $\sigma^{\mathsf{A}\times\mathsf{C}}((u,q),\ell,(u',q')) = \sigma^{\mathsf{A}}(u,\ell,u')$.

by definition of $\mathsf{A}\times\mathsf{C}$. Equation 3 shows that the Bellman optimality equation for $q^\star(s,u,q,a)$ reduces to the Bellman optimality equation for $q^\star(s,u,a)$. More precisely, the parameters for the system of nonlinear equations given in Equation 3 are the same as those in the system for $q^\star(s,u,a)$, except that each individual equation is repeated $|Q|$ times (once for every DFA state). Therefore, we have $\tilde{q}(s,u,q,a) = q^\star(s,u,a)$ as a solution.

Now all that is left is the fact that Q-learning in $M\times(\mathsf{A}\times\mathsf{C})$ will converge to the optimal Q-function that is independent of the $q\in Q$ component, and that Q-learning will converge to the same Q-function in $M\times\mathsf{A}$. Values in unreachable states will remain unaffected by learning updates, and will not affect the return from the optimal policy.

■

Definition 4 (Unreachable PRM state) *Let $\mathsf{A} = (U, u_I, 2^{AP}, \Gamma, \tau, \sigma, F)$ be a PRM. A state $u\in U$ is M-unreachable if for every input sequence λ s.t. $\mathsf{A} \xrightarrow{\lambda} u$ (A transitions into u upon reading λ) we have that every trajectory $s_0,a_0,s_1,\ldots,a_{n-1},s_n$ such that $L(s_0,a_0,s_1,\ldots,a_{n-1},s_n) = \lambda$ is unattainable in M (has probability 0 according to the transition function p of M).*

Lemma 2 (Transformation 2) *Let $\alpha\in\mathbb{R}$ be an arbitrary real. Let $\mathsf{A} = (U,u_I,2^{AP},\Gamma,\tau,\sigma,F)$ be a PRM, and let $V\subset U$ be a set of M-unreachable PRM states of A. Let $\mathsf{A}' = \mathsf{A}/V \to \alpha = (U,u_I,2^{AP},\Gamma\cup\alpha,\tau,\sigma^{\mathsf{A}'},F)$ be a PRM obtained by setting the output of every transition into an unreachable state $u\in V$ to α. In other words, $\sigma^{\mathsf{A}'}(u,u') = \alpha$ for all $u\in U$ and $u\in V$. Let $\pi^\star(s,u)$ be an optimal policy in the product MDP $M\times\mathsf{A}'$. Then $\pi^\star(s,u)$ is also an optimal policy in the product MDP $M\times\mathsf{A}$.*

Proof MDPs $M\times\mathsf{A}$ and $M\times\mathsf{A}'$ share the same state space $S\times U$, probabilistic transition function p, and initial state distribution. They may differ only in their (Markovian) reward function, specifically, on transitions into unreachable states $(s,u)\in S\times U$ for all $s\in S$ and $u\in V\subset U$. By definition 4, trajectories that induce a PRM transition into an M-unreachable state in A are unattainable.

This proof proceeds similarly to proof of Lemma 1, except it is even easier because we can work with a system of linear (not optimality) equations. The statement of the lemma concerning optimality

will follow from the general reduction, i.e. the lemma holds for an arbitrary policy π not just the optimal one. In Equation 4 we set out the system of Bellman equations in $M \times \mathsf{A}'$, and show that it reduces to the one for $M \times \mathsf{A}$.

$$
\begin{aligned}
q_\pi^{M\times\mathsf{A}'}(s,u,a) &= \sum_{\substack{s'\in S\\ u'\in U}} p(s',u' \mid s,u,a)\left(\sigma^{\mathsf{A}'}(u,\ell,u') + \gamma \sum_{a'\in A} \pi(a' \mid s') q_\pi^{M\times\mathsf{A}'}(s',u',a')\right) \\
&= \sum_{\substack{s'\in S\\ u'\in U\setminus V}} p(s',u' \mid s,u,a)\left(\sigma^{\mathsf{A}}(u,\ell,u') + \gamma \sum_{a'\in A} \pi(a' \mid s') q_\pi^{M\times\mathsf{A}'}(s',u',a')\right) \\
&+ \sum_{\substack{s'\in S\\ u'\in V}} p(s',u' \mid s,u,a)\left(\sigma^{\mathsf{A}'}(u,\ell,u') + \gamma \sum_{a'\in A} \pi(a' \mid s') q_\pi^{M\times\mathsf{A}'}(s',u',a')\right)
\end{aligned}
\tag{4}
$$

If u is an M-reachable state, the second sum vanishes because $p^{M\times\mathsf{A}}(s',u' \mid s,u,a) = p^M(s', \mid s,a) \cdot \tau^{\mathsf{A}}(u, L(s,a,s'), u') = 0$ for $u' \in V$ (otherwise, u' would be M-reachable if u was M reachable). Therefore, solutions in rows corresponding to M-reachable states are independent of rows corresponding to M-unreachable states, and the equations are the same as in the system for $q^{M\times\mathsf{A}}$, where the second sum also vanishes. Therefore, as Q-learning explores all reachable states infinitely often, and the reachable states in both product MDPs are the same and share the same Q-function, Q-learning will find the same optimal policy in both MDPs. The values of Q-functions corresponding to unreachable states are of no consequence.

■

Definition 5 captures the structure of the set of M-unreachable states in a PRM A induced by rejecting sinks states of a causal DFA. This property of a set of unreachable states V models the deterministic transition function of a causal DFA.

Definition 5 (Dependent Set of Unreachable PRM States) *Let $A = (U, u_I, 2^{AP}, \Gamma, \tau, \sigma, F)$ be a PRM, and V a subset of M-unreachable states in A. We say that V is a dependent set of M-unreachable states in A if it is a set of M-unreachable states and the following property holds for all labels $\ell \in 2^{AP}$ and states $u \in U^{\mathsf{A}}$: $(\exists u' \in V)\ \tau^{\mathsf{A}}(u,\ell,u') > 0 \implies (\forall u'' \notin V)\ \tau^{\mathsf{A}}(u,\ell,u'') = 0$.*

Intuitively, a set of M-unreachable states V is dependent if it is not possible to transition into both V and $U \setminus V$ from any $u \in U$.

Lemma 3 (Transformation 3) *Let $A = (U, u_I, 2^{AP}, \Gamma, \tau, \sigma, F)$ be a PRM, V a dependent set of M-unreachable states in A, and $m = -1 - \max_{r\in\Gamma^{\mathsf{A}}} |r| - \max_{u\in U^{\mathsf{A}}} v^\star(u)$. Let $\mathsf{B}_1 = A/V \to m$ be a PRM that mirrors the output of A, (except on transitions into unreachable states in V where the output is m), and $\mathsf{B}_2 = (-A)/V \to m$ a PRM that negates the output of A (except on transitions into unreachable states in V where the output is m). Let $u^0 \in U$ be a state in A such that $v^\star_{\mathsf{B}_1}(u) = v^\star_{\mathsf{B}_2}(u) = 0$. Let $\mathsf{B} = (U, u_I, 2^{AP}, \Gamma, \tau, \sigma, F \cup \sqcap')$ be a PRM obtained by adding u^0 to the set of terminal states in A. Let $\pi^\star(s,u)$ be an optimal policy in the product MDP $M \times \mathsf{B}$. Then $\pi^\star(s,u)$ is also an optimal policy in the product MDP $M \times \mathsf{A}$.*

Proof Let π be a policy in $M \times \mathsf{B}$. We will show that $v_\pi^{M\times\mathsf{A}} = v_\pi^{M\times\mathsf{B}}$, i.e. that an arbitrary policy π has the same value in $M \times \mathsf{A}$. From there, it follows that if π is optimal in $M \times \mathsf{B}$, then it is optimal in $M \times \mathsf{A}$.

To make analysis easier, we will model terminal states as absorbing states (sinks with output 0). For easier notation, $\tilde{s}$ will refer to states in S^M, and $s = (\tilde{s}, u)$ will refer to states in $S^{M\times\mathsf{A}}$.

We will first show $v_\pi^{M\times\mathsf{A}}(s^0) = v_\pi^{M\times\mathsf{B}}(s^0)$ for every state $s^0 = (\tilde{s}, u^0)$. We have $v_\pi^{M\times\mathsf{B}}(s^0) = 0$ because s^0 is an absorbing state in $M \times \mathsf{B}$. We must show that $v_\pi^{M\times\mathsf{A}}(s^0) = 0$. Intuitively, this is the case because in $M \times \mathsf{A}$, the expected return when starting in $s^0 = (\tilde{s}, u^0)$ and following an arbitrary policy π is bounded with 0 from above and below. It is enough to show that $v_\pi^{M\times\mathsf{B}_1}(s^0) = 0$, because by the proof of Lemma 2 we have $v_\pi^{M\times\mathsf{A}}(s^0) = v_\pi^{M\times\mathsf{B}_1}(s^0)$. We will show this via Equation 5.

$$0 = -v^\star_{\mathsf{B}_2}(u^0) \le v_\pi^{M\times\mathsf{B}_1}(s^0) \le v^\star_{\mathsf{B}_1}(u^0) = 0 \tag{5}$$

Equalities in Equation 5 hold by assumption. The second inequality holds trivially. The first inequality holds because the "pessimistic" machine B_2 realizes the minimal value of every state (negated to obtain the discounted return in terms of B_1). More precisely, for a given state $s = (\tilde{s}, u)$ we have

$$\begin{aligned} v^\star_{\mathsf{B}_2}(u) &= \max_{\ell\in 2^{\mathrm{AP}}} \sum_{u'\in U} \tau^{\mathsf{B}_2}(u,\ell,u')(\sigma^{\mathsf{B}_2}(u,\ell,u') + \gamma v^\star_{\mathsf{B}_2}(u')) \\ &= (\star) \\ &= \max_{\substack{\ell\in 2^{\mathrm{AP}} \\ \tau(u,\ell,u')=0 \\ \forall u'\in V}} \sum_{u'\in U} \tau^{\mathsf{B}_2}(u,\ell,u')(\sigma^{\mathsf{B}_2}(u,\ell,u') + \gamma v^\star_{\mathsf{B}_2}(u')) \\ &= \max_{\substack{\ell\in 2^{\mathrm{AP}} \\ \tau(u,\ell,u')=0 \\ \forall u'\in V}} \sum_{u'\in U} \tau^{\mathsf{B}_1}(u,\ell,u')(-\sigma^{\mathsf{B}_1}(u,\ell,u') + \gamma v^\star_{\mathsf{B}_2}(u')) \end{aligned} \tag{6}$$

and similarly

$$\begin{aligned} v^\star_{\mathsf{B}_1}(u) &= \max_{\ell\in 2^{\mathrm{AP}}} \sum_{u'\in U} \tau^{\mathsf{B}_1}(u,\ell,u')(\sigma^{\mathsf{B}_1}(u,\ell,u') + \gamma v^\star_{\mathsf{B}_2}(u')) \\ &= \max_{\substack{\ell\in 2^{\mathrm{AP}} \\ \tau(u,\ell,u')=0 \\ \forall u'\in V}} \sum_{u'\in U} \tau^{\mathsf{B}_1}(u,\ell,u')(\sigma^{\mathsf{B}_1}(u,\ell,u') + \gamma v^\star_{\mathsf{B}_1}(u')) \end{aligned} \tag{7}$$

.

($\star$): By assumption, when $\tau^{\mathsf{B}_2}(u,\ell,u') > 0$ for any $u' \in V$ then $\tau^{\mathsf{B}_2}(u,\ell,u'') = 0$ for all $u'' \notin V$. Intuitively, if reading input ℓ from state u induces a transition to $u' \in V$ with positive probability in B_2, since the transitions of the causal DFA are *not* probabilistic, every other state u'' such that $\tau^{\mathsf{B}_2}(u,\ell,u'') > 0$ must also transition into the same rejecting sink state in the causal DFA. In that case, $\sigma^{\mathsf{B}_2}(u,\ell,u') = m$ for all u', which is lower than any possible immediate reward and resulting state value. Therefore, the maximum is not attained for the input ℓ. Similar reasoning is applied in Equation 7.

Equation 6 and Equation 7 show that $-v^\star_{\mathsf{B}_2}$ bounds the value of any policy in $M \times \mathsf{B}_1$ from below, that is $-v^\star_{\mathsf{B}_2}(u) \le v_\pi^{M\times\mathsf{B}_1}(s)$. The argument is that $v^\star_{\mathsf{B}_2}$ is a solution to the Bellman optimality equation in B_1 with negated rewards. In particular, Equation 6 and Equation 7 show that one can

disregard transitions into unreachable states in V when computing state values. In that case, $-v^{\star}_{\mathsf{B}_2}$ is the pessimal state value in B_1. Intuitively, one attains a discounted return of $-v^{\star}_{\mathsf{B}_2}(u)$ when starting in u and minimizing the expected discounted sum of rewards along transitions in B_1, while ignoring transitions into V.

We proceed to show $v_\pi^{M\times\mathsf{A}}(s) = v_\pi^{M\times\mathsf{B}}(s)$ in all components (not just for $s = s^0$).

When we fix an arbitrary policy π, we obtain the immediate reward vector $\mathcal{R}_s^{\mathsf{A};\pi}$ ($\mathcal{R}_s^{\mathsf{B};\pi}$) and probabilistic transition matrix $\mathcal{P}_{s,s'}^{\mathsf{A};\pi}$ ($\mathcal{P}_{s,s'}^{\mathsf{B};\pi}$) for $M \times \mathsf{A}$ ($M \times \mathsf{B}$).

The state value Bellman equation for π in $M \times \mathsf{A}$ can then be expressed in matrix form as in Equation 8 (and similarly for $M \times \mathsf{B}$).

$$v_\pi^{\mathsf{A}} = \mathcal{R}^{\mathsf{A};\pi} + \gamma\mathcal{P}^{\mathsf{A};\pi}v_\pi^{\mathsf{A}} \tag{8}$$

We proceed to show that v_π^{B} also solves Equation 8, that is that Equation 9 holds.

$$v_\pi^{\mathsf{B}} = \mathcal{R}^{\mathsf{A};\pi} + \gamma\mathcal{P}^{\mathsf{A};\pi}v_\pi^{\mathsf{B}} \tag{9}$$

We already know that the Equation 9 holds in rows corresponding to $s^0 \in (\tilde{\jmath}, \sqcap') : \tilde{\jmath} \in \mathcal{S}^{\mathcal{M}}$, as we have shown $v_\pi^{\mathsf{A}}(s^0) = v_\pi^{\mathsf{B}}(s^0) = 0$. For $s \in S \setminus (\tilde{\jmath}, \sqcap') : \tilde{\jmath} \in \mathcal{S}^{\mathcal{M}}$, we have $\mathcal{R}_s^{\mathsf{A};\pi} = \mathcal{R}_s^{\mathsf{B};\pi}$ (B does not change the immediate reward on transitions from $s \neq s^0$). However, we also have $\mathcal{P}_{s,:}^{\mathsf{A};\pi} = \mathcal{P}_{s,:}^{\mathsf{B};\pi}$ ($M \times \mathsf{B}$ transitions in the same way as $M \times \mathsf{A}$ from $s \neq s^0$).

■

Now we can prove Theorem 1.

Proof Algorithm 1 starts with a PRM A and applies a series transformations in order to obtain a new PRM B. Then, it runs QRM for (M, B) instead of (M, A). Lemmas 1, 2, and 3 show that the optimal policy either remains the same when the transformations are applied (Transformation 2, 3), or that the optimal policy for the initial PRM can be easily recovered from the transformed PRM (Transformation 1). Line 1 applies Transformation 1 and 2. Lines 2-6 apply Transformation 3. Finally, convergence to optimal policy of Algorithm 1 then follows from the convergence to optimal policy of QRM.

■

Appendix C. Four Doors Case Study

Figure 11 depicts the PRM for the 4-door task. As can be seen, increasing the number of doors leads to an exponential increase in the number of states.

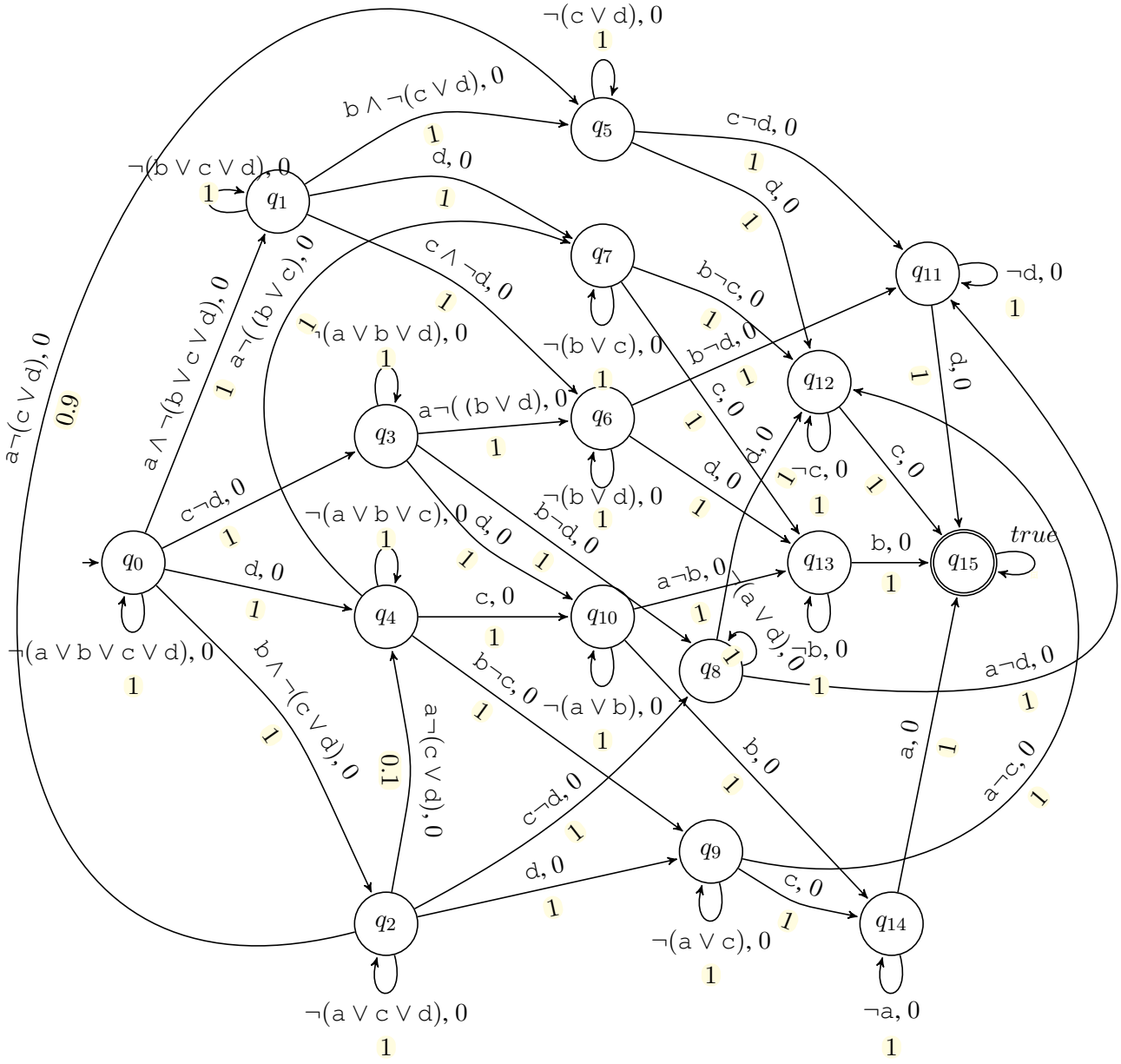


Figure 11: The PRM without causal info about the four-door task