Learning Reward Machines for Partially Observable Reinforcement Learning

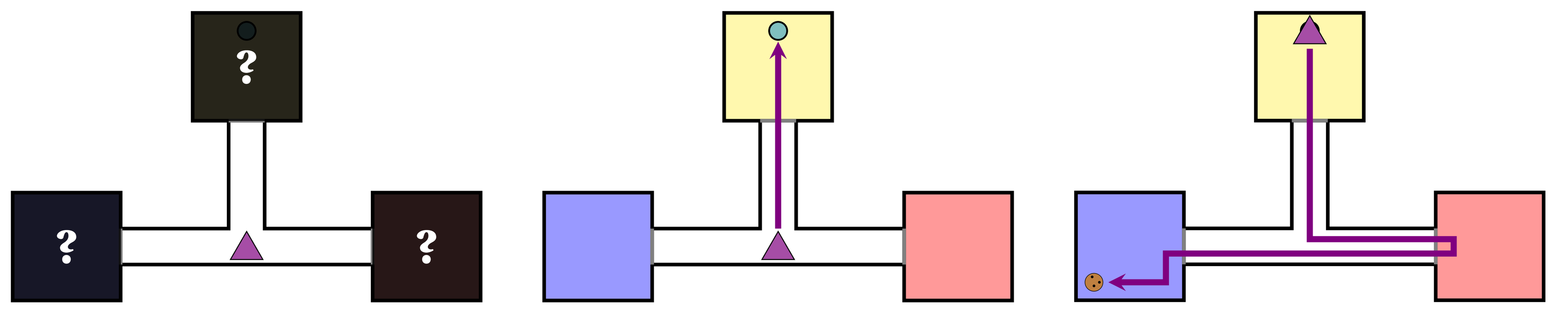
Rodrigo Toro Icarte[†‡] Ethan Waldie[†] Toryn Q. Klassen[†‡] Richard Valenzano[§] Margarita P. Castro[∥] Sheila A. McIlraith[†‡]

[†]Dept of Computer Science, University of Toronto [∥]Dept of Mechanical and Industrial Engineering, University of Toronto [‡]Vector Institute [§]Element AI

[†]{rntoro, ewaldie96, toryn, sheila}@cs.toronto.edu, [∥]mpcastro@mie.utoronto.ca, [§]rick.valenzano@elementai.com

Abstract. Reward Machines (RMs) provide a structured, automata-based representation of a reward function that enables a Reinforcement Learning (RL) agent to decompose an RL problem into structured subproblems that can be efficiently learned via off-policy learning. Here we show that RMs can be learned from experience, instead of being specified by the user, and that the resulting problem decomposition can be used to effectively solve partially observable RL problems. We pose the task of learning RMs as a discrete optimization problem where the objective is to find an RM that decomposes the problem into a set of subproblems such that the combination of their optimal memoryless policies is an optimal policy for the original problem. We show the effectiveness of this approach on three partially observable domains, where it significantly outperforms A3C, PPO, and ACER, and discuss its advantages, limitations, and broader potential.

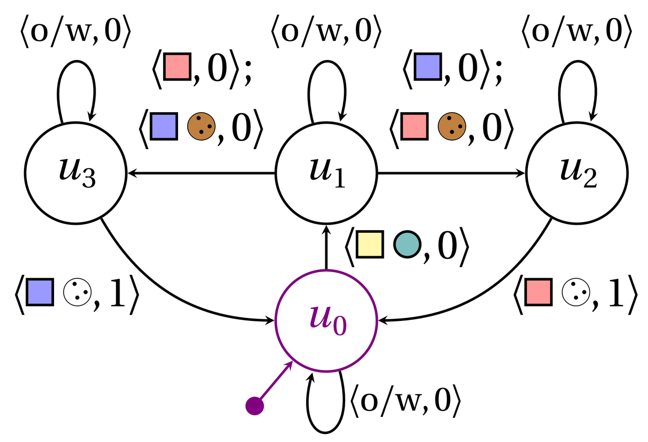
Partially Observable RL



Solving this problem requires **memory**!

$\pi^*(a|o_t) \neq \pi^*(a|o_0 \cdots o_t)$

Reward Machines (RMs) are **discrete memories**!



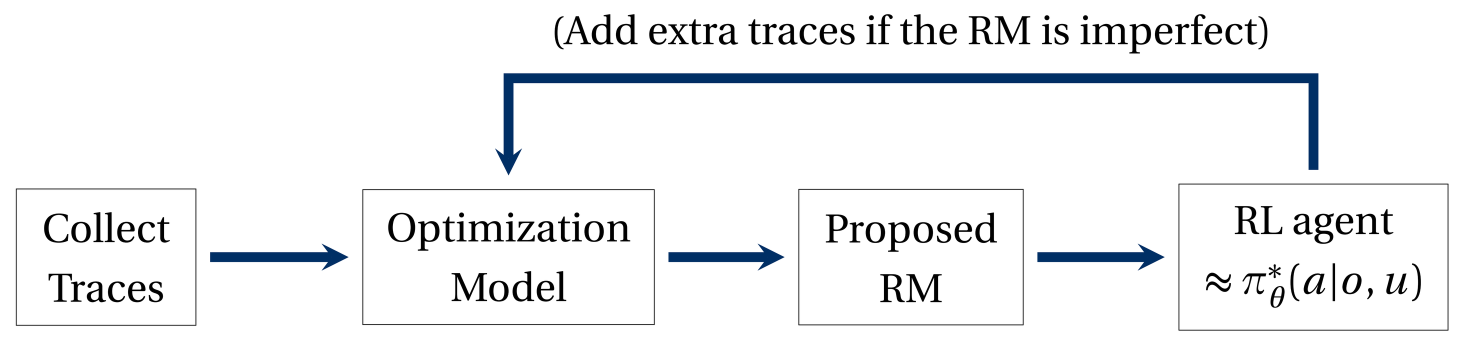
Hard problem $\xrightarrow{\text{RM}}$ Easy problem

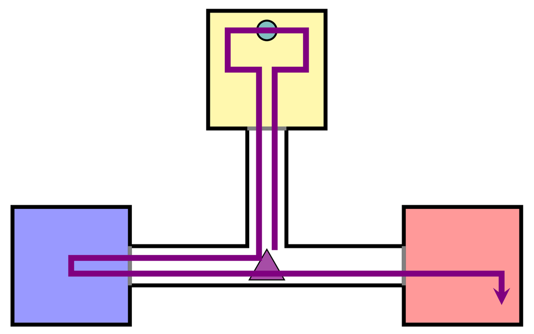
For this RM:

$\pi^*(a|o_0 \cdots o_t) = \pi^*(a|o_t, u_t)$

It is a **perfect memory**!

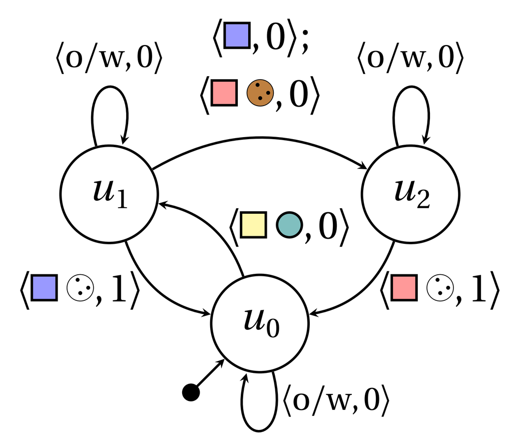
Our Approach (LRM)





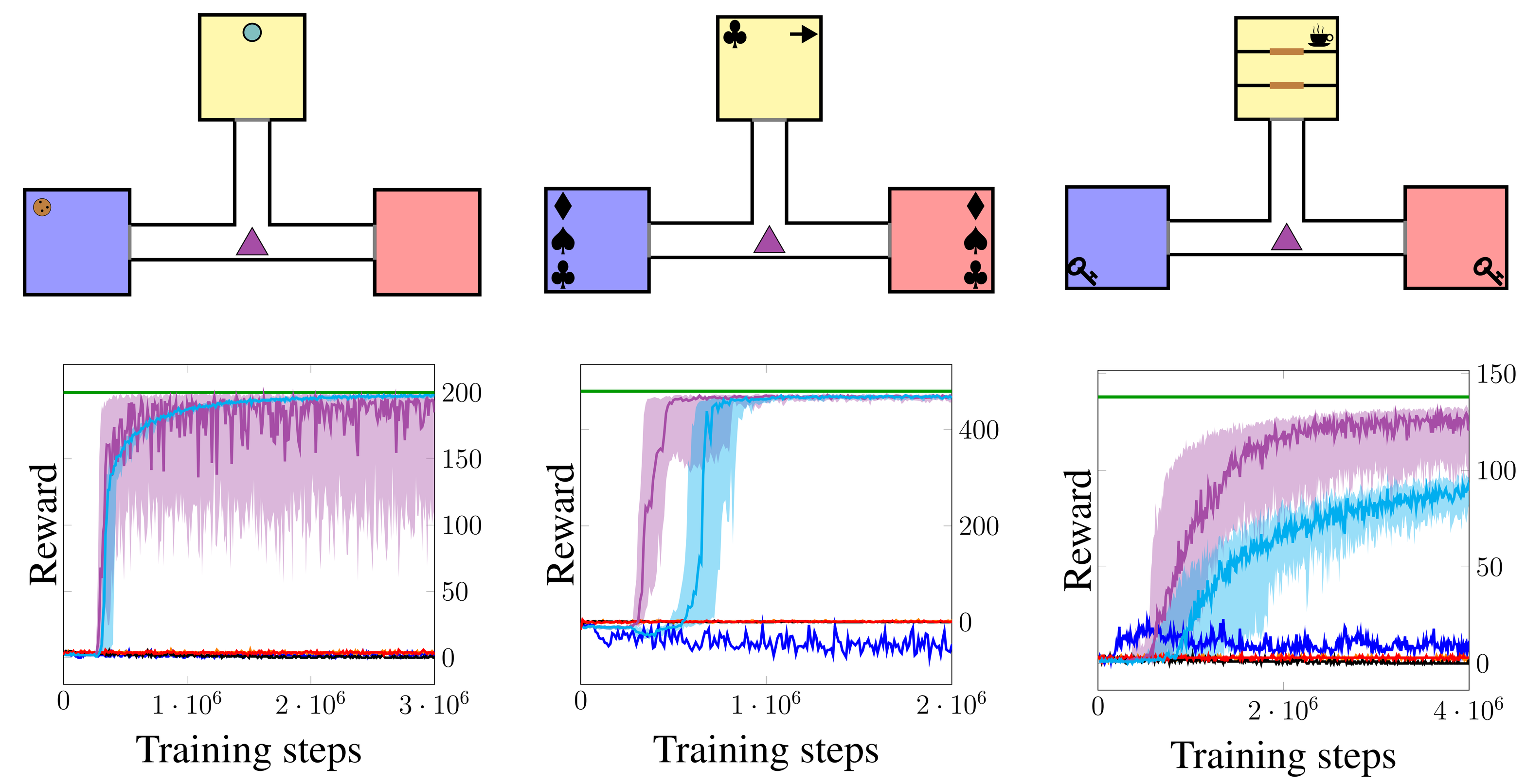
(many traces!)

Tabu Search



1) DDQN
2) DQRM

Results

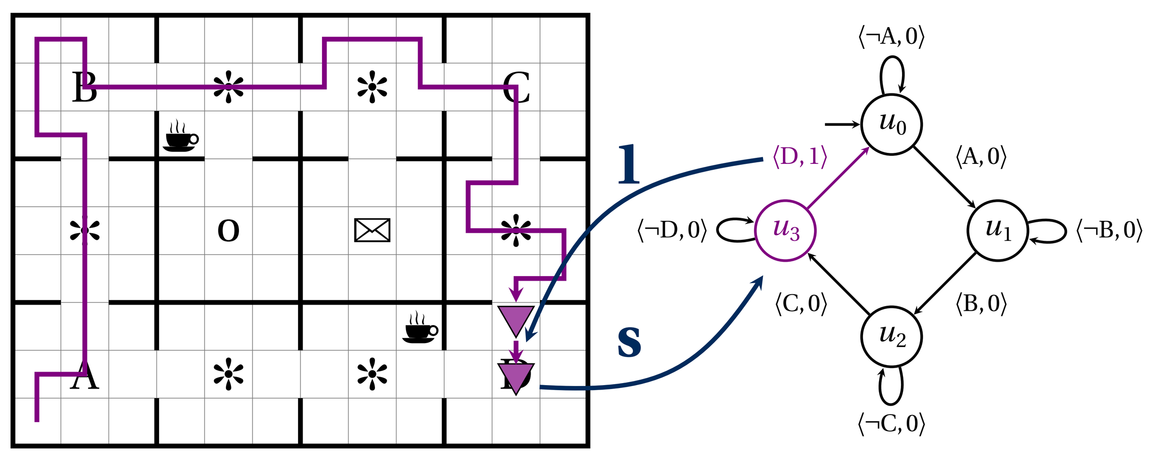


LRM-DQRM **LRM-DDQN** Optimal DDQN A3C ACER PPO

Summary ⇑ / ⇓ Details

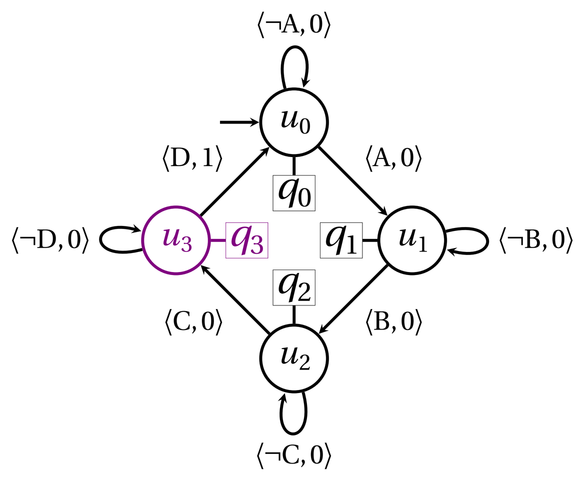
Reward Machines (RMs)

RMs are automata-based representations of a reward function that allow RL agents to learn policies faster [1].



Q-Learning for Reward Machines (QRM)

- Learn one policy (q-function) per state in the RM.
- Select actions using the policy of the current RM state.
- Reuse experience to update all the q-values at the same time.



Q-updates

$q_0(s,a) \xleftarrow{\alpha} 0 + \gamma \max_{a'} q_0(s',a')$

$q_1(s,a) \xleftarrow{\alpha} 0 + \gamma \max_{a'} q_1(s',a')$

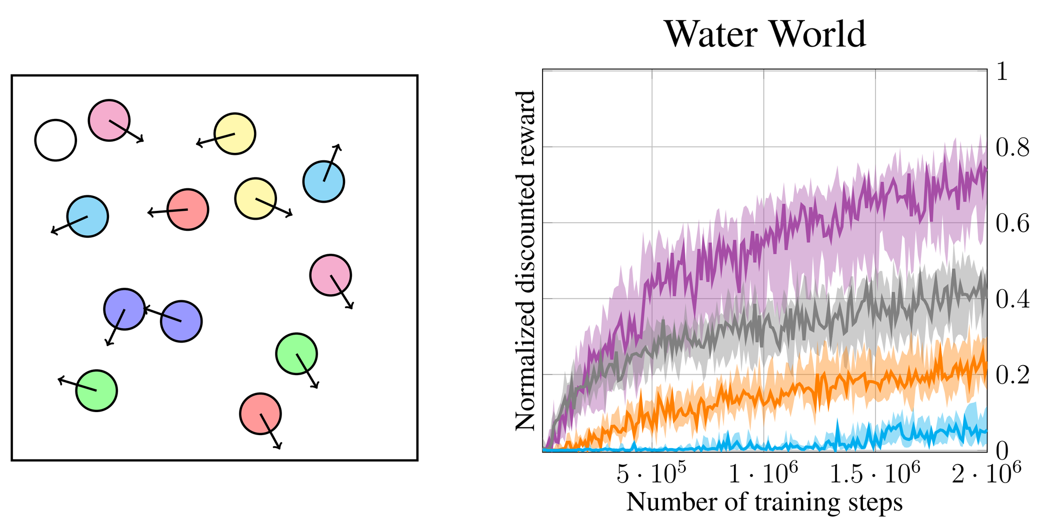
$q_2(s,a) \xleftarrow{\alpha} 0 + \gamma \max_{a'} q_2(s',a')$

$q_3(s,a) \xleftarrow{\alpha} 1 + \gamma \max_{a'} q_0(s',a')$

RMs in Previous Work

- RMs were specified by the user (directly [1] or via LTL [2]).
- QRM did not work under partial observability.

Our 2018 RM Results



Legend: DQRM DHRL-RM DHRL DDQN

Main Contributions

1. First approach for learning RMs from experience.
2. Extended RMs and QRM to work under partial observability.
3. Developed a theory for 1 and 2.

RMs under Partial Observability

RMs are defined over a set of propositional symbols $\mathcal{P}$ that correspond to high-level events (e.g., $\mathcal{P} = \{\dots\}$) that the agent can detect using a *labelling function* $L : O \times A \times O \to 2^{\mathcal{P}}$.

Def. A Reward Machine is a tuple $R_{\mathcal{P}} = \langle U, u_0, \delta_u, \delta_r \rangle$ where U is a finite set of states, $u_0 \in U$ is an initial state, δ_u is the state-transition function, $\delta_u : U \times 2^{\mathcal{P}} \to U$, and δ_r is the reward-transition function, $\delta_r : U \times 2^{\mathcal{P}} \to \mathbb{R}$.

Perfect Reward Machines

Perfect RMs make the environment Markovian w.r.t. $O \times U$, i.e.:

$$\Pr(o_{t+1}, r_{t+1} | o_0, a_0, \dots, o_t, a_t) = \Pr(o_{t+1}, r_{t+1} | o_t, u_t, a_t)$$

for every possible trace $o_0, a_0, \dots, o_t, a_t$ generated by any policy.

Properties: Given an environment $\mathcal{E}$: (i) If the set of belief states of $\mathcal{E}$ is finite, then there exists a perfect RM for $\mathcal{E}$, and (ii) Optimal policies over $O \times U$ for perfect RMs are also optimal for $\mathcal{E}$.

Optimization Model

Inputs:

1. A set of traces: $\mathcal{T} = \{\mathcal{T}_0, \dots, \mathcal{T}_n\}$ where each trace $\mathcal{T}_i$ is:

$$\mathcal{T}_i = (o_{i,0}, a_{i,0}, r_{i,0}, \dots, a_{i,t_i-1}, r_{i,t_i-1}, o_{i,t_i}).$$

2. A labelling function $L : O \times A \times O \to 2^{\mathcal{P}}$.

Model:

$$\underset{\langle U, u_0, \delta_u, \delta_r \rangle}{\text{minimize}} \sum_{i \in I} \sum_{t \in T_i} \log(|N_{x_{i,t}, L(e_{i,t})}|) \qquad (\texttt{LRM})$$

$$\text{s.t.}\ \langle U, u_0, \delta_u, \delta_r \rangle \in \mathcal{R}_{\mathcal{P}} \qquad (1)$$

$$|U| \leq u_{\max} \qquad (2)$$

$$x_{i,t} \in U \qquad \forall i \in I, t \in T_i \cup \{t_i\} \qquad (3)$$

$$x_{i,0} = u_0 \qquad \forall i \in I \qquad (4)$$

$$x_{i,t+1} = \delta_u(x_{i,t}, L(e_{i,t+1})) \qquad \forall i \in I, t \in T_i \qquad (5)$$

$$N_{u,l} \subseteq 2^{2^{\mathcal{P}}} \qquad \forall u \in U, l \in 2^{\mathcal{P}} \qquad (6)$$

$$L(e_{i,t+1}) \in N_{x_{i,t}, L(e_{i,t})} \qquad \forall i \in I, t \in T_i \qquad (7)$$

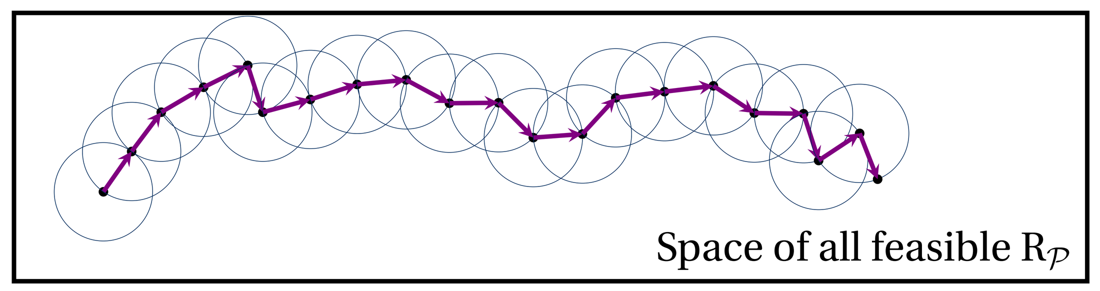
Outputs:

1. A Reward Machine: $\langle U, u_0, \delta_u, \delta_r \rangle \in \mathcal{R}_{\mathcal{P}}$.
2. A set of possible next high-level observations $N_{x_{i,t}, L(e_{i,t})}$.

Theorem (necessary conditions):

When $\mathcal{T} \to \infty$, any perfect RM is an optimal solution of $\texttt{LRM}$.

Tabu Search



Experimental Remarks

- The binary classifiers were used by all the approaches.
- LRM works because Tabu search finds high-quality RMs.
- DQRM pays off in domains with sparse rewards.
- **Code**: `bitbucket.org/RToroIcarte/lrm`.

References

[1] R. Toro Icarte, T. Klassen, R. Valenzano, and S. McIlraith. Using reward machines for high-level task specification and decomposition in reinforcement learning. In *ICML18*.

[2] A. Camacho, R. Toro Icarte, T. Klassen, R. Valenzano, and S. McIlraith. LTL and beyond: Formal languages for reward function specification in reinforcement learning. In *IJCAI19*.

[3] R. Toro Icarte, E. Waldie, T. Klassen, R. Valenzano, M. Castro, and S. McIlraith. Learning reward machines for partially observable reinforcement learning. In *NeurIPS19*.