Model-Free Reinforcement Learning for Symbolic Automata-encoded Objectives

Anand Balakrishnan [1], Stefan Jakšić [2], Edgar A. Aguilar [2], Dejan Ničković [2], and Jyotirmoy V. Deshmukh [1]

[1]University of Southern California, Los Angeles, California, USA

[2]AIT Austrian Institute of Technology GmbH, Vienna, Austria

Objectives

Our goal is to tackle the *reward engineering* problem for reinforcement learning (RL) such that any learned behavior is safe and interpretable. To this end, we propose the use of symbolic automata as a theoretically sound and practical framework to encode tasks for reinforcement learning agents.

Our main contributions are:

- A sound framework to design reward functions that ensures maximal probabilistic satisfaction of a given task.
- A symbolic potential function that uses spatial information in the symbolic automaton to speed up learning.

Introduction

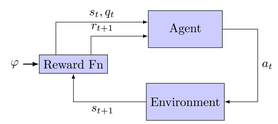


Figure 1. Overview of specification guided reinforcement learning.

- Reinforcement learning (RL) is a paradigm where an *agent* learns a controller by repeatedly interacting with an *environment* via a *reward feedback*.
- A good *reward function* must ensure correctness of any learned behavior.
  - Rewarding undesirable behavior can lead to unsafe consequences!
- Conservative (or sparse) rewarding strategies can lead to slow learning and poor convergence.

Symbolic Automata as Tasks

- We propose a novel approach to encode a finite sequence of tasks using symbolic automata [1].
- They can encode history-dependent goals along with rich quantitative information about the overall objective.
- There exists well-defined algorithms to encode Temporal Logic specifications as symbolic automata [2].

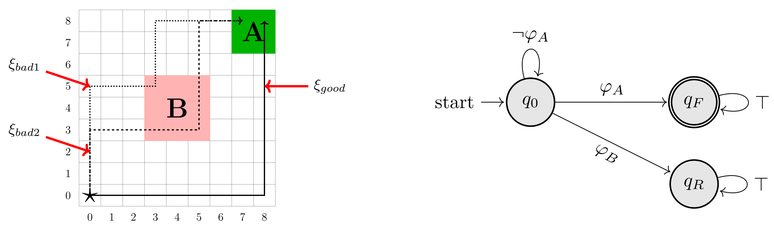
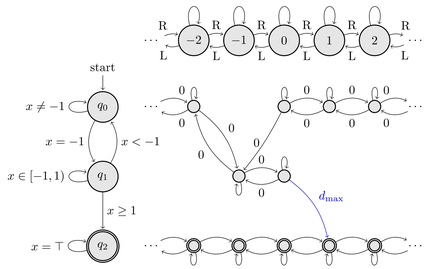


Figure 2. This symbolic automaton models the reach-avoid task $\varphi = \mathbf{F}\,\varphi_A \wedge \mathbf{G}\,\neg\varphi_B$ where $\varphi_A = (7 \le x \le 8) \wedge (7 \le y \le 8)$ denotes the region A, and $\varphi_B = (3 \le x \le 5) \wedge (3 \le y \le 5)$ denotes the region B.

Product Markov Decision Process

Given an MDP $\mathcal{M} = (S, s_{\text{init}}, A, P)$ and a symbolic automaton $\mathcal{A} = (X, Q, q_{\text{init}}, F, \Delta, \delta)$ — where F is the set of final states — we define a *product MDP*, $\mathcal{P} = \mathcal{M} \otimes \mathcal{A}$.



$$R((s,q),(s',q')) = \begin{cases} d_{\max}, & \text{if } q' \in F \\ & \text{and } q \notin F \\ 0, & \text{otherwise.} \end{cases}$$

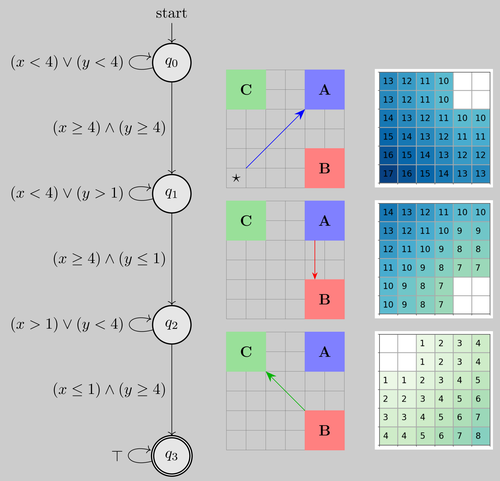
Figure 3. The (implicitly constructed) product of a simple MDP defined on the number line $\mathbb{Z}$ and a symbolic automaton encoding a task on the MDP. $R(\cdot,\cdot)$ is the *sparse* reward function for the product.

Symbolic Potential Functions

To better inform the RL agent on "good" paths in the MDP that satisfy the task at hand, we define a *shaped reward function*

$$\hat{R}((s,q),(s',q')) = R((s,q),(s',q')) + \Phi(s,q) - \Phi(s',q'),$$

where $\Phi(s,q)$ is a *symbolic potential function* defined on the product MDP, $\mathcal{P}$.



Note. The potential function only requires the current state of the system and $\mathcal{A}$ to generate a distance, i.e., it is model-free!

Experimental Results

We compare our proposed framework against some state-of-the-art methods with a similar problem statement, and significantly outperform them as the problem and tasks scale up.

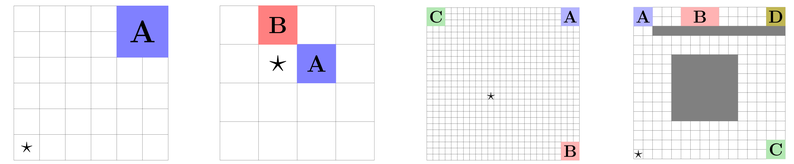


Figure 4. The above maps correspond to the environments where we evaluate each specification. Here, the $\star$ corresponds to the initial position of the RL agent, and the gray blocks correspond to obstacles or walls.

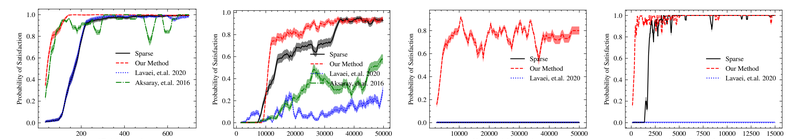


Figure 5. The above graphs plot the probability of a learned policy satisfying the given specification against the number of training epochs for each method.

Conclusion and Future work

- We present a RL framework with theoretical guarantees and the potential to scale up for complex tasks in large systems.
- We compare our method with some state-of-the-art methods proposed in literature.
- For more details, please refer to our preprint [3].

In future works, we hope to expand our theoretical guarantees to continuous space (state space and action space) systems, and generalize the framework to time-dependent tasks using timed automata abstractions.

References

[1] L. D'Antoni and M. Veanes, "The Power of Symbolic Automata and Transducers," in *Computer Aided Verification*, R. Majumdar and V. Kunčak, Eds., vol. 10426, Cham: Springer International Publishing, 2017, pp. 47–67, ISBN: 978-3-319-63386-2 978-3-319-63387-9. DOI: 10.1007/978-3-319-63387-9_3.

[2] S. Jakšić, E. Bartocci, R. Grosu, and D. Ničković, "An Algebraic Framework for Runtime Verification," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 11, pp. 2233–2243, Nov. 2018, ISSN: 1937-4151. DOI: 10.1109/TCAD.2018.2858460.

[3] A. Balakrishnan, S. Jaksic, E. A. Aguilar, D. Nickovic, and J. Deshmukh, "Model-Free Reinforcement Learning for Symbolic Automata-encoded Objectives," *arXiv:2202.02404 [cs]*, Feb. 2022. arXiv: 2202.02404 [cs].