# Specification-Guided Reinforcement Learning

Suguman Bansal

Georgia Institute of Technology, USA

**Abstract.** The problem of reinforcement learning (RL) is to generate an optimal policy w.r.t. a given task in an unknown environment. Traditionally, the task is encoded in the form of a reward function which becomes cumbersome for long-horizon goals. An appealing alternate is to use logical specifications, opening the direction of RL from logical specifications. This paper summarizes the trials and triumphs in developing highly performant algorithms and obtain theoretical guarantees in RL from logical specifications.

## 1 Introduction

The problem of *Reinforcement Learning* (RL) is to generate a policy for a given task in an unknown environment by continuously interacting with it [30]. When combined with neural-networks (NN), RL has made remarkable strides in control synthesis in real-world domains, including challenging continuous (innite-state) environments with non-linear dynamics or unknown models. Few examples include tasks such as walking [6] and grasping [3], control of multi-agent systems [26,17,21], and control from visual inputs [23].

Yet, current RL approaches are poorly suited for control synthesis for long-horizon tasks. A critical challenge facing RL is that the desired task is encoded in the form of a reward. Specifying a long-horizon task in the form of a reward can be highly non-intuitive; Poor reward specification could hinder the performance and correctness of the learning algorithm. An appealing alternative is to express the task in the form of a high-level logical specification, such as a temporal logic [8,9,19,27], as opposed to a reward function. Logical specifications combine temporal operators with boolean connectives, enabling more natural encoding of a large class of desirable properties. Furthermore, logical specifications facilitate testing and verification, which could be used to rigorously evaluate the correctness of the learned policy.

This paper provides a brief overview of recent progress in RL from logical specifications. Formally, the problem is to learn a policy that optimizes the probability to satisfy the given specification in an environment modeled by a *Markov Decision Processs (MDP)*. The defining assumption in RL is that the transition probabilities of the MDP are unknown. Thus, the policy is learnt via exploration via repeated sampling of the environment. We briefly summarize performance of existing algorithms and known theoretical guarantees[1].

[1] Parts of the paper is based on joint work with Rajeev Alur, Osbert Bastani, and Kishor Jothimurugan.

## 2 Reinforcement Learning from Logical Specifications

**Markov Decision Process.** A Markov Decision Process (MDP) is a tuple $M = (S, A, \eta, P)$, where $S$ is a set of states, $\eta : S \to [0, 1]$ is the initial state distribution, $A$ is a finite set of actions, and $P : S \times A \times S \to [0, 1, ]$ is the transition probability function with $\Sigma_{s' \in S} P(s, a, s') = 1$ for all $s \in S$.

An infinite run $\zeta \in (S \times A)^\omega$ is the sequence $\zeta = s_0, a_0, s_1, a_1 \ldots$, where $s_0 \sim \eta$ and $P(s_i, a_i, s_{i+1}) > 0$ for all $i \geq 0$. Similarly, a finite run $\zeta \in (S \times A)^* \times S$ is a sequence $\zeta = s_0, a_0, s_1, a_1 \ldots a_{t-1} s_t$. For any run of length at least $j$, we let $\zeta_{i:j} = s_i, a_i, \ldots a_{j-1}, s_j$ be a sub-sequence of $\zeta$ for $i, j \in \mathbb{N}$.

Let $\mathsf{Runs}_f$ denote the set of all finite runs in the MDP. Let $D(A) = \{\Delta : A \to [0, 1] \text{ s.t. } \Sigma_{a \in A} \Delta(a) = 1\}$ be the set of all distributions on actions $A$. Then, a policy $\pi : \mathsf{Runs}_f \to D(A)$ maps a history of finite runs to a distribution on actions. Let $\Pi$ denote the set of all policies.

**Task Specifications.** There are different ways in which one can specify the objective of the learning algorithm. We define a *reinforcement learning task* to be a pair $(M, \phi)$ where $M$ is an MDP and $\phi$ is a specification for $M$. In general, a specification $\phi$ for $M = (S, A, s_0, P)$ defines a function $J_\phi^M : \Pi(S, A) \to \mathbb{R}$ and the reinforcement learning objective is to compute a policy $\pi$ that maximizes $J_\phi^M(\pi)$. Below, we define RL tasks traditionally using rewards and using logical specifications.

*Disocunted-sum rewards.* Traditionally, the specifications is a reward function that maps transitions in $M$ to real values. The specification consists of a reward function $R : S \times A \times S \to \mathbb{R}$ and a discount factor $\gamma \in ]0, 1[$—i.e., $\phi = (R, \gamma)$. The value of a policy $\pi$ is

$$J_\phi^M(\pi) = \mathbb{E}_{\zeta \sim D_\pi^M}\Big[\sum_{i=0}^{\infty} \gamma^i R(s_i, a_i, s_{i+1})\Big],$$

where $s_i$ and $a_i$ denote the state and the action at the $i^{\text{th}}$ step of $\zeta$, respectively. Limit-average instead of discounted-sum is also commonly used.

*Logical specifications.* Rewards are defined w.r.t. a given set of states $S$ and actions $A$, and can only be interpreted over MDPs with the same state and action spaces. Logical specifications are defined independently of $S$ and $A$. To achieve this, a common assumption is that there is a fixed set of propositions $\mathsf{Prop}$, and a labeling function $L : S \to 2^{\mathsf{Prop}}$ denoting which propositions hold in a given state. Given a run $\zeta = s_0 a_0 s_1 a_1 \ldots$, we let $L(\zeta)$ denote the corresponding sequence of labels $L(\zeta) = L(s_0)L(s_1)\ldots$. A *labeled MDP* is a tuple $M = (S, A, s_0, P, L)$. Wlog, we only consider labeled MDPs in the rest of the paper.

Formal languages are used to specify qualitative properties about runs of the system. A logical specification $\phi = \mathcal{L} \subseteq (2^{\mathsf{Prop}})^\omega$ is a set of "desirable" sequences of labels. The value of a policy $\pi$ is the probability of generating a sequence in $\mathcal{L}$—i.e.,

$$J^M_\phi(\pi) = D^M_\pi\big(\{\zeta \in \mathsf{Runs}(S, A) \mid L(\zeta) \in \mathcal{L}\}\big).$$

Examples of logical specifications are reachability properties, safety properties, and temporal logics such as LTL [27], LTL over finite-traces [8], or SpectRL [19].

**Reinforcement Learning.** Given an RL task $(M, \phi)$, assuming the transition probabilities of $M$ are unknown, the problem of RL is to generate a policy

$$\pi^* = \arg\max_{\pi \in \Pi} J^M_\phi(\pi).$$

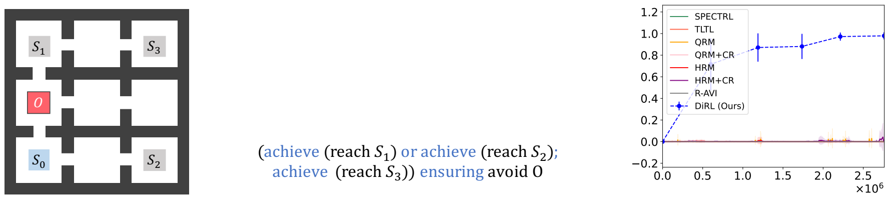
## 3 Algorithms

Recently, a myriad of algorithms have been proposed for learning from logical specifications [1,4,5,7,10,12,16,15,14,34,32,18,24,31]. These methods can be categorized into two broad classes, as described below:

*Specification to Rewards.* Many early works on RL from logical specifications took a natural approach to solve the problem. Here, the goal is to automatically synthesize rewards from a given formal specification and then to use a traditional RL algorithm to learn an optimal policy from the synthesized rewards. The simplest reward function would be to assign a positive reward to executions that satisfy the specification and a zero reward to the rest. This naive reward function results in poor performance of the learning algorithm due to the sparsity of rewards. Thus, most algorithms are modified to provide informative intermediate rewards that also guide the search of an optimal policy. Few methods include policy-preserving reward shaping and rewards based on distance metrics.

Mostly, these algorithms demonstrate merit in learning *stateful policies*. For example, if the task was to learn to return to the initial state after visiting a goal, then the movement of the agent along the path connecting the initial state to the goal state depends on whether the goal has been visited or not. Since most specification-to-reward conversion schemes are *stateful*, such information automatically gets encoded in the learnt policy. By and large, these methods have shown to learn high-quality policies in finite-state environments, and for simple tasks in infinite-state environments.

*Compositional Algorithm.* These methods build on early progress of naïve approach of converting specifications to rewards with the objective to scale to complex long-horizon specifications in complex infinite-state environments. An example of a task that proves to be too complex for early works is illustrated in Figure 1. Specification-to-reward based approaches fail on such examples due to the inherent greedy nature of RL algorithms. In this example, the algorithms learn to reach $S_2$ instead of $S_1$ as the former is easier to learn due to the absence of any obstacle. However, this is not fruitful towards satisfaction of the original specification as there is no direct path from $S_2$ to $S_3$.



(achieve (reach $S_1$) or achieve (reach $S_2$);
achieve (reach $S_3$)) ensuring avoid O

Fig. 1: Left: The 9-rooms environment, with initial region $S_0$ in the bottom-left, an obstacle $O$ in the middle-left, and three subgoal regions $S_1, S_2, S_3$ in the remaining corners. Middle: A user-provided specification $\phi$. Right: Learning curves for compositional approach DiRL [20] and some baselines; $x$-axis is number of steps and $y$-axis is probability of achieving $\phi$.

Compositional approaches for learning from specifications leverage the structure of a given specification to first decompose the original task into several simpler and easier-to-learn tasks and then compose the policies learnt for these subtasks to obtain a policy that maximizes satisfaction of the original specification. This way these algorithms combine planning on the high-specification with learning on low-level tasks to scale to large and complex specifications. The structure of the specification could be exploited further for significant improvements in the performance of learning algorithms along the metrics of scalability in specification, sample complexity, and quality of solutions.

## 4 Theoretical guarantees

We present the theoretical foundations of RL from logical specifications. The formal guarantees associated with the specification-to-reward approach of learning algorithms have been studied [11,13,28]. We will discuss PAC learning in the context of RL from logical specifications. We then discuss recent results [2,33] showing that PAC algorithms do not exist for Linear Temporal Logic (LTL) specifications and present a high-level overview of a proof.

For this section, we assume an MDP is has finitely many states $S$ and actions $A$, and there is a unique initial state $s_0$, i.e., $\eta(s_0) = 1$ and $\eta(s) = 0$ for all $s \in S \setminus \{s_0\}$. A learning algorithm $\mathcal{A}$ can be thought of as an iterative process that in each iteration (i) either resets the MDP state to an initial state or takes a step in $M$, and (ii) outputs its current estimate of an optimal policy $\pi$. A learning algorithm $\mathcal{A}$ induces a random sequence of output policies $\{\pi_n\}_{n=1}^{\infty}$ where $\pi_n$ is the policy output in the $n^{\text{th}}$ iteration.

Let $\mathcal{J}^*(M, \phi) = \sup_\pi J_\phi^M(\pi)$ denote the maximum value of $J_\phi^M$. We let $\Pi_{opt}(M, \phi)$ denote the set of all optimal policies in $M$ w.r.t. $\phi$—i.e., $\Pi_{opt}(M, \phi) = \{\pi | J_\phi^M(\pi) = \mathcal{J}^*(M, \phi)\}$. In many cases, it is sufficient to compute an $\varepsilon$-optimal policy $\tilde{\pi}$ with $J_\phi^M(\tilde{\pi}) \geq \mathcal{J}^*(M, \phi) - \varepsilon$; we let $\Pi_{opt}^{\varepsilon}(M, \phi)$ denote the set of all $\varepsilon$-optimal policies in $M$ w.r.t. $\phi$.

**Definition 1.** *A learning algorithm $\mathcal{A}$ is said to be PAC-MDP for a class of specifications $\mathcal{C}$ if, there is a function $h$ such that for any $p > 0$, $\varepsilon > 0$, and any RL task $(M, \phi)$ with $M = (S, A, s_0, P)$ and $\phi \in \mathcal{C}$, taking $N = h(|S|, |A|, |\phi|, \frac{1}{p}, \frac{1}{\varepsilon})$, with probability at least $1 - p$, we have*

$$\left|\left\{n \mid \pi_n \notin \Pi^{\varepsilon}_{opt}(M, \phi)\right\}\right| \leq N.$$

We say a PAC-MDP algorithm is *efficient* if the *sample complexity* function $h$ is polynomial in $|S|, |A|, \frac{1}{p}$ and $\frac{1}{\varepsilon}$. There are efficient PAC-MDP algorithms for discounted-sum rewards [22,29]. Unfortunately, we show that that it is not possible to obtain PAC-MDP algorithms for safety specifications.

**Theorem 1.** *[2] There does not exist a PAC-MDP algorithm for the class of safety specifications.*

Intuitively, Theorem 1 follows from that fact that, when learning from simulation, it is highly likely that the learning algorithm will encounter identical transitions when the underlying MDP is modified slightly. This makes it impossible to infer an $\varepsilon$-optimal policy using a number of samples that is independent of the transition probabilities since safety specifications are not robust [25]. For a full proof, see [2].

## References

1. Aksaray, D., Jones, A., Kong, Z., Schwager, M., Belta, C.: Q-learning for robust satisfaction of signal temporal logic specifications. In: Conference on Decision and Control (CDC). pp. 6565–6570. IEEE (2016)
2. Alur, R., Bansal, S., Bastani, O., Jothimurugan, K.: A framework for transforming specifications in reinforcement learning. https://arxiv.org/abs/2111.00272 (2021)
3. Andrychowicz, O.M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., et al.: Learning dexterous in-hand manipulation. The International Journal of Robotics Research **39**(1), 3–20 (2020)
4. Bozkurt, A.K., Wang, Y., Zavlanos, M.M., Pajic, M.: Control synthesis from linear temporal logic specifications using model-free reinforcement learning. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 10349–10355. IEEE (2020)
5. Brafman, R., De Giacomo, G., Patrizi, F.: Ltlf/ldlf non-markovian rewards. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
6. Collins, S., Ruina, A., Tedrake, R., Wisse, M.: Efficient bipedal robots based on passive-dynamic walkers. Science **307**(5712), 1082–1085 (2005)
7. De Giacomo, G., Iocchi, L., Favorito, M., Patrizi, F.: Foundations for restraining bolts: Reinforcement learning with ltlf/ldlf restraining specifications. In: Proceedings of the International Conference on Automated Planning and Scheduling. vol. 29, pp. 128–136 (2019)
8. De Giacomo, G., Vardi, M.Y.: Linear temporal logic and linear dynamic logic on finite traces. In: IJCAI'13 Proceedings of the Twenty-Third international joint conference on Artificial Intelligence. pp. 854–860. Association for Computing Machinery (2013)

9. Donzé, A.: On signal temporal logic. In: International Conference on Runtime Verification. pp. 382–383. Springer (2013)
10. Fu, J., Topcu, U.: Probably approximately correct MDP learning and control with temporal logic constraints. In: Robotics: Science and Systems (2014)
11. Hahn, E.M., Perez, M., Schewe, S., Somenzi, F., Trivedi, A., Wojtczak, D.: Reward shaping for reinforcement learning with omega-regular objectives. arXiv preprint arXiv:2001.05977 (2020)
12. Hahn, E.M., Perez, M., Schewe, S., Somenzi, F., Trivedi, A., Wojtczak, D.: Omega-regular objectives in model-free reinforcement learning. In: Tools and Algorithms for the Construction and Analysis of Systems. pp. 395–412 (2019)
13. Hahn, E.M., Perez, M., Schewe, S., Somenzi, F., Trivedi, A., Wojtczak, D.: Omega-regular objectives in model-free reinforcement learning. In: International Conference on Tools and Algorithms for the Construction and Analysis of Systems. pp. 395–412. Springer (2019)
14. Hasanbeig, M., Kantaros, Y., Abate, A., Kroening, D., Pappas, G.J., Lee, I.: Reinforcement learning for temporal logic control synthesis with probabilistic satisfaction guarantees. In: Conference on Decision and Control (CDC). pp. 5338–5343 (2019)
15. Hasanbeig, M., Abate, A., Kroening, D.: Logically-constrained reinforcement learning. arXiv preprint arXiv:1801.08099 (2018)
16. Icarte, R.T., Klassen, T., Valenzano, R., McIlraith, S.: Using reward machines for high-level task specification and decomposition in reinforcement learning. In: International Conference on Machine Learning. pp. 2107–2116. PMLR (2018)
17. Inala, J.P., Yang, Y., Paulos, J., Pu, Y., Bastani, O., Kumar, V., Rinard, M., Solar-Lezama, A.: Neurosymbolic transformers for multi-agent communication. arXiv preprint arXiv:2101.03238 (2021)
18. Jiang, Y., Bharadwaj, S., Wu, B., Shah, R., Topcu, U., Stone, P.: Temporal-logic-based reward shaping for continuing learning tasks (2020)
19. Jothimurugan, K., Alur, R., Bastani, O.: A composable specification language for reinforcement learning tasks. Advances in Neural Information Processing Systems **32** (2019)
20. Jothimurugan, K., Bansal, S., Bastani, O., Alur, R.: Compositional reinforcement learning from logical specifications. In: Advances in Neural Information Processing Systems (2021)
21. Jothimurugan, K., Bansal, S., Bastani, O., Alur, R.: Specification-guided learning of nash equilibria with high social welfare (2022)
22. Kearns, M., Singh, S.: Near-optimal reinforcement learning in polynomial time. Machine learning **49**(2), 209–232 (2002)
23. Levine, S., Finn, C., Darrell, T., Abbeel, P.: End-to-end training of deep visuomotor policies. The Journal of Machine Learning Research **17**(1), 1334–1373 (2016)
24. Li, X., Vasile, C.I., Belta, C.: Reinforcement learning with temporal logic rewards. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 3834–3839. IEEE (2017)
25. Littman, M.L., Topcu, U., Fu, J., Isbell, C., Wen, M., MacGlashan, J.: Environment-independent task specifications via GLTL. arXiv preprint arXiv:1704.04341 (2017)
26. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. arXiv preprint arXiv:1706.02275 (2017)
27. Pnueli, A.: The temporal logic of programs. In: 18th Annual Symposium on Foundations of Computer Science. pp. 46–57. IEEE (1977)

28. Somenzi, F., Trivedi, A.: Reinforcement learning and formal requirements. In: International Workshop on Numerical Software Verification. pp. 26–41. Springer (2019)
29. Strehl, A.L., Li, L., Wiewiora, E., Langford, J., Littman, M.L.: PAC model-free reinforcement learning. In: Proceedings of the 23rd international conference on Machine learning. pp. 881–888 (2006)
30. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)
31. Vaezipoor, P., Li, A.C., Icarte, R.A.T., Mcilraith, S.A.: Ltl2action: Generalizing ltl instructions for multi-task rl. In: International Conference on Machine Learning. pp. 10497–10508. PMLR (2021)
32. Xu, Z., Topcu, U.: Transfer of temporal logic formulas in reinforcement learning. In: International Joint Conference on Artificial Intelligence. pp. 4010–4018 (7 2019)
33. Yang, C., Littman, M.L., Carbin, M.: Reinforcement learning for general LTL objectives is intractable. CoRR **abs/2111.12679** (2021), `https://arxiv.org/abs/2111.12679`
34. Yuan, L.Z., Hasanbeig, M., Abate, A., Kroening, D.: Modular deep reinforcement learning with temporal logic specifications. arXiv preprint arXiv:1909.11591 (2019)