Specification-Guided Reinforcement Learning

Rajeev Alur, Suguman Bansal, Osbert Bastani, Kishor Jothimurugan

University of Pennsylvania
{alur, suguman, obastani, kishor}@seas.upenn.edu

1 Goals of the Tutorial

This tutorial will introduce the AAAI audience to the emerging interdisciplinary research on *Reinforcement Learning (RL) from Logical Specifications*. The unprecedented proliferation of data-driven approaches, especially machine learning, has put the spotlight on building trustworthy AI through the combination of the contrasting characteristics of logical reasoning and machine learning. Reinforcement Learning from Logical Specifications is one such topic where formal logical constructs are utilized to overcome challenges faced by modern RL algorithms and their applications. Research on this topic is scattered across venues targeting subareas of AI. Foundational work has appeared at formal methods and artificial intelligence venues. Algorithmic development and applications have appeared at machine learning, robotics, and cyber-physical systems venues. Through this tutorial, we aim to consolidate recent progress in one capsule for a typical AI researcher. The tutorial will be designed to explain the importance of using formal specifications in RL and encourage researchers to apply existing techniques for RL from logical specifications as well as contribute to the growing body of work on this topic.

This tutorial will introduce reinforcement learning as a tool for automated synthesis of control policies and discuss the challenge of encoding long-horizon tasks using rewards. We will then formulate the problem of reinforcement learning from logical specifications and present recent progress in developing scalable algorithms as well as theoretical results demonstrating the hardness of learning in this context.

Keywords. Reinforcement Learning, Temporal Logic, Planning and Control, PAC Learning, Synthesis.

2 Overview

Overview of Tutorial. This tutorial will be organized around four themes covering major developments in reinforcement learning (RL) from logical specifications, described as follows:

1. Introduction to RL: Beginning with seminal results in RL, we will describe RL as a data-driven tool for automated synthesis of control policies and discuss few recent successes. We will then discuss the challenges faced by traditional RL in learning long-horizon tasks, where RL is defined as a reward-optimization problem.
2. Reinforcement Learning from Logical Specifications: Motivated by the success of logical specifications in (non data-driven) planning and control synthesis for long-horizon tasks, we will introduce the problem of RL from logical specifications as an approach to mitigate the shortcomings of traditional RL based on reward-optimization.
3. Practical Algorithms: We will summarize the leading approaches to development of practical algorithms to learn from logical specifications and describe an instance of each approach using the logical specification language SpectRL [19], and demonstrate few emerging tools.
4. Theoretical Foundations: In laying the theoretical foundations of learning from logical specifications, we will discuss the hardness of the problem and describe the theoretical guarantees obtained from the previously described practical algorithms.

History. While this tutorial will be prepared for the first time for AAAI 2023, the presenters have presented this work at several venues, including top-tier conferences such as NeurIPS, AISTATS, and CAV. Recently, the presenters have delivered parts of this tutorial as a black-board talk at the Dagstuhl Seminar on Machine Learning and Logical Reasoning: The New Frontier, as an invited talk at a workshop at the Simons Institute for the Theory of Computing, and as a round-table discussion at WOLVERINE at FLoC 2022. The presenters were also invited to contribute a survey on RL from Logical Specifications to an upcoming special issue journal.

Target Audience. This tutorial will be accessible to all AI researchers. We believe researchers working on topics related to control, planning, reinforcement learning, safe AI and formal synthesis would benefit the most from this tutorial. We estimate an audience size of 50-100 people.

Prerequisites. The tutorial will be self-contained, covering all the necessary background through its course. Background in reinforcement learning and/or logical reasoning is preferred, though not mandatory.

3 Outline

This is intended to be a short quarter-day tutorial, spanning 1 hrs 45 mins. Below, we provide a brief outline of the structure of the tutorial along with an estimate of time intended to be spent on each part.

3.1 RL and Current Challenges (10 mins)

We will begin with a general introduction to Reinforcement Learning (RL), briefly describing the rich history of work over the past few decades by theoreticians and practitioners [29]. We will discuss few success of RL in the development of AI systems through a data-driven approach across a wide ranging domains, including game-playing [27], healthcare [35] and (autonomous) control systems [17, 21, 22]. We will then motivate next-gen applications of RL. In particular, we will focus on the automated synthesis of controllers for performing long-horizon tasks such as navigation and manipulation [23, 24]. Prior advances in automated synthesis for long-horizon tasks have arrived from the planning and synthesis communities. However, these approaches often make assumptions that are violated in the real world and even in simulations, such as finite-state models, a priori knowledge of variables, bounded uncertainty and the like. In contrast, RL holds the potential of synthesis with minimal assumptions. Yet, state-of-the-art RL faces many hurdles in realizing the holy grail. We will describe these challenges, namely task-specification for long-horizon tasks, scalability of algorithms, and lack of guarantees for safety-critical systems.

3.2 RL from Logical Specifications (10 mins)

This section will motivate and formally define RL from logical specifications. We will begin with the traditional definition of RL as a reward-optimization problem in which optimal policy is synthesized by repeated sampling the environment to obtain local rewards. This formulation implies that the desired task for the control policy has to be encoded in rewards. We will present the challenges faced by reward-based task specification of long-horizon tasks, including difficulty in task expression, non-compositionality, reward-hacking [4] leading to generation of poor quality solutions, etc. Next, we will describe how task-specification utilizing temporal logic formulas can mitigate these issues. In particular, we will describe the logical specification language SpectRL [19] which encodes long-term behaviors combining reachability and safety tasks. Finally, we will define RL as a satisfaction-optimization problem where the goal is to generate policy that optimizes the satisfaction probability of the logical formula specifying the desired task.

3.3 Practical Algorithms (40 mins)

Recently, a myriad of RL algorithms [1, 6, 7, 8, 9, 10, 16, 14, 15, 36, 33, 18, 21, 30] have been proposed for learning from logical specifications. In this part of the tutorial, we categorize these methods into two broad classes and provide an overview of the high-level ideas along with one concrete algorithm for each class of algorithms, as elaborated below:

Specification to Rewards. We will begin with the natural approach explored by many early works on this topic. 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. We describe their advantage in learning stateful policies and also describe a few scalability issues. Finally, we will describe an algorithm [19] for generating rewards from SpectRL specifications in detail.

Compositional Algorithm. We will demonstrate through a family of examples that, despite early progress, the naïve approach of converting specifications to rewards scales poorly with complexity of specification due to the inherent greedy nature of RL algorithms. We then present a compositional approach for learning from specifications that 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. We will describe the details of the compositional approach [20] using SpectRL specifications.

Contemporary Work. Finally, we will briefly discuss other ways in which logical specifications have been incorporated in RL, including shielding for Safe RL [2], verification [5, 17] and generating interpretable and verifiable policies [31, 32].

3.4 Theoretical Foundations (35 mins)

In this part of the tutorial, we will present the theoretical foundations of RL from logical specifications. We will begin with describing the formal guarantes associated with the specification-to-reward approach of learning algorithms [11, 13, 28]. For this, we will present a theory of reductions in the context of RL formalizing the class of algorithms that convert specifications to rewards. We present two kinds of reductions, specification reduction and sampling-based reduction, and discuss when these reductions preserve optimal or near-optimal solutions. Finally, we will discuss PAC learning in the context of RL from logical specifications and briefly mention recent attempts [9] at obtaining PAC algorithms under addition assumptions. We then discuss recent results [3, 34] showing that PAC algorithms do not exist for Linear Temporal Logic (LTL) specifications and present a high-level overview of a proof. During this course, we will build the necessary theoretical background on LTL [25] and conversion of LTL formulas to automaton models that support RL algorithms [12, 26].

3.5 Conclusion and Open Problems (10 mins)

The tutorial will conclude with a small demonstration of how logical specifications can be use to learn robotics manipulation tasks in simulation, as an illustration of the utility of these algorithms. Finally, we conclude with a discussion on potential future work which includes designing more sample efficient algorithms for practical applications, use of discounting in LTL to obtain PAC algorithms, decomposition of temporal specifications into smaller ones for enabling compositional approaches, among many others.

4 Presenters

The tutorial presenters have expertise at the intersection of artificial intelligence, machine learning, and logical reasoning evident by strong publication records in AAAI/AISTATS, ICML/ICLR/NeurIPS, and CAV/LICS/POPL/PLDI/TACAS, respectively, and applications in EMSOFT/ICRA/RSS. The presenters have been involved in the development of frameworks and tools that will be discussed in detail in the tutorial. These frameworks are based on usage of techniques arising in formal methods and programming languages to reinforcement learning; areas on which tutorial presenters have deep expertise and have made fundamental contributions (Please see relevant recent publications by presenters below for details). The tutorial presenters have extensive experience of teaching at universities. A short biography of each presenter is attached below:

Rajeev Alur. Rajeev Alur is Zisman Family Professor of Computer and Information Science and the Founding Director of ASSET (Center for AI-Enabled Systems: Safe Explainable, and Trustworthy) at University of Pennsylvania. He obtained his bachelor's degree in computer science from IIT Kanpur in 1987 and PhD in computer science from Stanford University in 1991. Before joining Penn in 1997, he was with Computing Science Research Center at Bell Labs. His research is focused on formal methods for system design, and spans artificial intelligence, cyber-physical systems, distributed systems, logic in computer science, machine learning, and programming languages. He is a Fellow of the AAAS, a Fellow of the ACM, a Fellow of the IEEE, an Alfred P. Sloan Faculty Fellow, and a Simons Investigator. He was awarded the inaugural CAV (Computer-Aided Verification) award in 2008, ACM/IEEE Logic in Computer Science (LICS) Test-of-Time award in 2010, the inaugural Alonzo Church award by ACM SIGLOG / EATCS / EACSL / Kurt Goedel Society in 2016, Distinguished Alumnus Award by IIT Kanpur in 2017 for his work on timed automata. Prof. Alur has served as the chair of ACM SIGBED (Special Interest Group on Embedded Systems), the general chair of LICS, and the lead PI of the NSF Expeditions in Computing center ExCAPE (Expeditions in Computer Augmented Program Engineering). He is the author of the textbook Principles of Cyber-Physical Systems (MIT Press, 2015).

Suguman Bansal. Suguman Bansal is an NSF/CRA Computing Innovation Postdoctoral Fellow at the University of Pennsylvania, mentored by Prof. Rajeev Alur. Starting January 2023, she will be an Assistant Professor at Georgia Institute of Technology. Her research is focused on formal methods and its applications to artificial intelligence, programming languages, and machine learning. She is the recipient of the 2020 NSF CI Fellowship and has been named a 2021 MIT EECS Rising Star. She completed her Ph.D. in 2020, advised by Prof. Moshe Y. Vardi, from Rice University. She received B.S. with Honors in 2014 from Chennai Mathematical Institute.

Osbert Bastani. Osbert Bastani is an Assistant Professor of Computer and Information Science at the University of Pennsylvania. He leads the research group on Trustworthy Machine Learning, which broadly works on designing algorithms and techniques for improving the reliability of deep learning models; he is also a member of the ASSET Center for Trustworthy AI, the PRECISE Center for Safe AI, and the PRiML Center for Machine Learning. Prior to joining Penn, he received his Ph.D. in Computer Science from Stanford University in 2018, advised by Alex Aiken, and spent a year as a Postdoc at MIT with Armando Solar-Lezama.

Kishor Jothimurugan. Kishor Jothimurugan is a final-year PhD student at the University of Pennsylvania, advised by Prof. Rajeev Alur. His research focuses on applications of formal methods in reinforcement learning including RL from formal specifications, compositional RL algorithms and verification of neural network controllers. He received B.S. with Honors in 2017 from Chennai Mathematical Institute.

5 Selected Relevant Publications by Presenters

In reverse chronological order:

1. Rajeev Alur, Suguman Bansal, Osbert Bastani, and Kishor Jothimurugan. *A Framework for Transforming Specifications in Reinforcement Learning*. (To appear) Special Journal Issue Henzinger-60.
2. Kishor Jothimurugan, Suguman Bansal, Osbert Bastani, and Rajeev Alur. *Specification-Guided Learning of Nash Equilibria with High Social Welfare*. CAV 2022.
3. Kishor Jothimurugan, Suguman Bansal, Osbert Bastani, and Rajeev Alur. *Compositional Reinforcement Learning from Logical Specifications*. NeurIPS 2021.
4. Radoslav Ivanov, Kishor Jothimurugan, Steve Hsu, Vaidya Shaan, Rajeev Alur, and Osbert Bastani. *Compositional learning and verification of neural network controllers*. EMSOFT 2021.
5. Osbert Bastani, Shuo Li, and Anton Xue. *Safe reinforcement learning via statistical model predictive shielding* . RSS 2021.
6. Kishor Jothimurugan, Osbert Bastani, and Rajeev Alur. *Abstract value iteration for hierarchical deep reinforcement learning*. AISTATS 2021.
7. Shuo Li and Osbert Bastani. *Robust model predictive shielding for safe reinforcement learning with stochastic dynamics*. ICRA 2020.
8. Radoslav Ivanov, Taylor J Carpenter, James Weimer, Rajeev Alur, George J Pappas, and Insup Lee. *Verifying the safety of autonomous systems with neural network controllers* . TECS 2020.
9. Kishor Jothimurugan, Rajeev Alur, and Osbert Bastani. *Composable specifications for reinforcement learning*. NeurIPS, 2019.
10. Osbert Bastani, Yewen Pu, and Armando Solar-Lezama. *Verifiable reinforcement learning via policy extraction*. NeurIPS 2018.
11. Rajeev Alur and Tom Henzinger. *Reactive Modules*. FMSD 1999.

12. Rajeev Alur, Costas Courcoubetis, Tom Henzinger, P Ho, Xavier Nicollin, Alfredo Olivero, Joseph Sifakis, and Sergio Yovine. *The algorithmic analysis of hybrid systems.* Theoretical Computer Science 1995.

References

[1] Aksaray, D.; Jones, A.; Kong, Z.; Schwager, M.; and Belta, C. 2016. Q-learning for robust satisfaction of signal temporal logic specifications. In *Conference on Decision and Control (CDC)*, 6565–6570. IEEE.

[2] Alshiekh, M.; Bloem, R.; Ehlers, R.; Könighofer, B.; Niekum, S.; and Topcu, U. 2018. Safe reinforcement learning via shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[3] Alur, R.; Bansal, S.; Bastani, O.; and Jothimurugan, K. 2021. A Framework for Transforming Specifications in Reinforcement Learning. *arXiv preprint arXiv:2111.00272 (To appear in Henzinger-60)*.

[4] Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; and Mané, D. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.

[5] Bastani, O.; Pu, Y.; and Solar-Lezama, A. 2018. Verifiable reinforcement learning via policy extraction. In *Advances in Neural Information Processing Systems*.

[6] Bozkurt, A. K.; Wang, Y.; Zavlanos, M. M.; and Pajic, M. 2020. Control synthesis from linear temporal logic specifications using model-free reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 10349–10355. IEEE.

[7] Brafman, R.; De Giacomo, G.; and Patrizi, F. 2018. LTLf/LDLf non-markovian rewards. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

[8] De Giacomo, G.; Iocchi, L.; Favorito, M.; and Patrizi, F. 2019. Foundations for restraining bolts: Reinforcement learning with LTLf/LDLf restraining specifications. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, 128–136.

[9] Fu, J.; and Topcu, U. 2014. Probably Approximately Correct MDP Learning and Control With Temporal Logic Constraints. In *Robotics: Science and Systems*.

[10] Hahn, E. M.; Perez, M.; Schewe, S.; Somenzi, F.; Trivedi, A.; and Wojtczak, D. 2019. Omega-Regular Objectives in Model-Free Reinforcement Learning. In *Tools and Algorithms for the Construction and Analysis of Systems*, 395–412.

[11] Hahn, E. M.; Perez, M.; Schewe, S.; Somenzi, F.; Trivedi, A.; and Wojtczak, D. 2019. Omega-regular objectives in model-free reinforcement learning. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 395–412. Springer.

[12] Hahn, E. M.; Perez, M.; Schewe, S.; Somenzi, F.; Trivedi, A.; and Wojtczak, D. 2020. Good-for-MDPs automata for probabilistic analysis and reinforcement learning. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 306–323. Springer.

[13] Hahn, E. M.; Perez, M.; Schewe, S.; Somenzi, F.; Trivedi, A.; and Wojtczak, D. 2020. Reward Shaping for Reinforcement Learning with Omega-Regular Objectives. *arXiv preprint arXiv:2001.05977*.

[14] Hasanbeig, M.; Abate, A.; and Kroening, D. 2018. Logically-constrained reinforcement learning. *arXiv preprint arXiv:1801.08099*.

[15] Hasanbeig, M.; Kantaros, Y.; Abate, A.; Kroening, D.; Pappas, G. J.; and Lee, I. 2019. Reinforcement Learning for Temporal Logic Control Synthesis with Probabilistic Satisfaction Guarantees. In *Conference on Decision and Control (CDC)*, 5338–5343.

[16] Icarte, R. T.; Klassen, T.; Valenzano, R.; and McIlraith, S. 2018. Using reward machines for high-level task specification and decomposition in reinforcement learning. In *International Conference on Machine Learning*, 2107–2116. PMLR.

[17] Ivanov, R.; Jothimurugan, K.; Hsu, S.; Vaidya, S.; Alur, R.; and Bastani, O. 2021. Compositional Learning and Verification of Neural Network Controllers. *ACM Transactions on Embedded Computing Systems (TECS)*, 20(5s): 1–26.

[18] Jiang, Y.; Bharadwaj, S.; Wu, B.; Shah, R.; Topcu, U.; and Stone, P. 2020. Temporal-Logic-Based Reward Shaping for Continuing Learning Tasks. arXiv:2007.01498.

[19] Jothimurugan, K.; Alur, R.; and Bastani, O. 2019. A Composable Specification Language for Reinforcement Learning Tasks. In *Advances in Neural Information Processing Systems*, 13021–13030.

[20] Jothimurugan, K.; Bansal, S.; Bastani, O.; and Alur, R. 2021. Compositional Reinforcement Learning from Logical Specifications. *arXiv preprint arXiv:2106.13906*.

[21] Li, X.; Vasile, C.-I.; and Belta, C. 2017. Reinforcement learning with temporal logic rewards. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3834–3839. IEEE.

[22] Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

[23] Nachum, O.; Gu, S.; Lee, H.; and Levine, S. 2019. Near-optimal representation learning for hierarchical reinforcement learning. In *ICLR*.

[24] Nachum, O.; Gu, S. S.; Lee, H.; and Levine, S. 2018. Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, 3303–3313.

[25] Pnueli, A. 1977. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, 46–57. ieee.

[26] Sickert, S.; Esparza, J.; Jaax, S.; and Křetínskỳ, J. 2016. Limit-deterministic Büchi automata for linear temporal logic. In *International Conference on Computer Aided Verification*, 312–332. Springer.

[27] Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

[28] Somenzi, F.; and Trivedi, A. 2019. Reinforcement learning and formal requirements. In *International Workshop on Numerical Software Verification*, 26–41. Springer.

[29] Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

[30] Vaezipoor, P.; Li, A. C.; Icarte, R. A. T.; and Mcilraith, S. A. 2021. Ltl2action: Generalizing ltl instructions for multi-task rl. In *International Conference on Machine Learning*, 10497–10508. PMLR.

[31] Verma, A.; Le, H. M.; Yue, Y.; and Chaudhuri, S. 2019. Imitation-projected programmatic reinforcement learning. In *Advances in Neural Information Processing Systems*.

[32] Verma, A.; Murali, V.; Singh, R.; Kohli, P.; and Chaudhuri, S. 2018. Programmatically interpretable reinforcement learning. In *International Conference on Machine Learning*, 5045–5054. PMLR.

[33] Xu, Z.; and Topcu, U. 2019. Transfer of Temporal Logic Formulas in Reinforcement Learning. In *International Joint Conference on Artificial Intelligence*, 4010–4018.

[34] Yang, C.; Littman, M. L.; and Carbin, M. 2021. Reinforcement Learning for General LTL Objectives Is Intractable. *CoRR*, abs/2111.12679.

[35] Yu, C.; Liu, J.; Nemati, S.; and Yin, G. 2021. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1): 1–36.

[36] Yuan, L. Z.; Hasanbeig, M.; Abate, A.; and Kroening, D. 2019. Modular deep reinforcement learning with temporal logic specifications. *arXiv preprint arXiv:1909.11591*.