









Reinforcement Learning with Guarantees that Hold for Ever

Ernst Moritz Hahn¹, Mateo Perez², Sven Schewe³, Fabio Somenzi²,
Ashutosh Trivedi², and Dominik Wojtczak³

¹ University of Twente, Enschede, The Netherlands

² University of Colorado Boulder, Boulder, USA

³ University of Liverpool, Liverpool, UK

sven.schewe@liverpool.ac.uk


Abstract. Reinforcement learning is a successful explore-and-exploit approach, where a controller tries to learn how to navigate an unknown environment. The principle approach is for an intelligent agent to learn how to maximise expected rewards. But what happens if the objective refers to non-terminating systems? We can obviously not wait until an infinite amount of time has passed, assess the success, and update. But what can we do? This talk will tell.

1 Learning from Rewards

Model free reinforcement learning (RL) [16, 19] refers to a class of algorithms, where an intelligent agent (sometimes many, but we stick with basic case) receives rewards. Such rewards serve as feedback; they can be received after termination, after a fixed period of time, or after every action. Reaping high rewards reinforces a given behaviour of the agent (hence the name), while behaviour that leads to low rewards is avoided.

What makes RL algorithms popular is their flexibility and generality. They can work without being provided with a model of the environment dynamics and can handle probabilistic environment behaviour – that is, Markov decision processes (MDP) are their natural domain.

The goal of an agent in its interaction with its environment is to learn an optimal *strategy*, which describes how she chooses actions in a way that maximises the expected value of the overall reward, usually the total or discounted sum of individual rewards or a single reward at the end of a finite run. Discounted and average rewards are typical of infinite horizon problems.

This work is supported in part by the National Science Foundation (NSF) grant CCF-2009022 and by NSF CAREER award CCF-2146563, and by the Engineering and Physical Science Research Council through grant EP/V026887/1.  This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 864075 (CAESAR) and 956123 (FOCETA).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
J. F. Groote and M. Huisman (Eds.): FMICS 2022, LNCS 13487, pp. 3–7, 2022.
https://doi.org/10.1007/978-3-031-15008-1_1

The strategy of an agent describes what she does in each situation. Average, total, and discounted rewards usually lead to memoryless strategies, but strategies can take the history (i.e., the previous interaction with the environment) into account.

2 Learning with ω -Regular Objectives

Standard approaches to RL face natural difficulties when considering ω -regular properties [13, 17] that, for example, occur when specifications in linear time temporal logic (LTL) [12] are used.

This is because the reward for an overall run of a system is binary (does/does not satisfy the specification), and its binary value cannot be determined after a finite prefix.

So, how can we learn in this case?

We suggest a layered approach, which consists of the following:

1. Translate the property into a formal acceptor, like a Büchi [2] or parity automaton, such that maximising the likelihood of winning on the *syntactic* product with the MDP is good enough for maximising the chance of satisfying the property [3, 7, 18].
2. Translate the formal acceptor into a reachability objective, such that maximising the chance of reaching a goal state leads to maximising the chance of acceptance [4].
3. Wrap this reward structure into a standard RL approach [16].

2.1 Good-for-MDP Automata

Finding the right type of automata has two major ingredients:

1. a limited level of nondeterminism, such that the resulting automaton is *good for MDPs* (see below) and
2. a simple acceptance mechanism.

The first ingredient is a technical requirement. Formally(-sh), an automaton is **good-for-MDPs** if, for *all* finite MDPs, its syntactic product MDP (which is the syntactic product of the finite MDP with the automaton) has the same likelihood to satisfy the acceptance condition as an optimal control of the MDP has to satisfy the objective [4].

In this syntactic product, the agent has more to do: she has to resolve the nondeterminism of the automaton as well as the choices of the original MDP.

Broadly speaking, this limits the type of nondeterminism the automaton can use: normally, an automaton can use unlimited look-ahead to resolve its nondeterministic choices, but this automaton has to take into account where the randomness of the MDP can take it. Moreover, it needs to react to almost all cases perfectly.

This is reminiscent of good-for-games automata [10], but they need to be able to deal with all (not merely almost all) interactions with their environment.

The relaxation to *almost all* allows for more automata. Indeed, one of the main differences between good-for-games and good-for-MDPs automata is that non-deterministic Büchi automata can always be used for the latter, while the former requires more complex acceptance mechanisms.

This brings us back to the second ingredient: a simple acceptance mechanism. This is a practical requirement rather than a technical necessity. It is due to the much higher cost of the further translation of complex acceptance mechanisms [8].

Standard translations to limit deterministic automata [7, 18] produce non-deterministic Büchi automata that are good-for-MDPs [6], but it is equally simple to produce other good-for-MDP automata with attractive alternative properties, like never offering more than two choices [6].

After learning, the automaton states (and structure) turn into a finite state memory.

2.2 From GFM Büchi Automata to Reachability and RL

A parameterised translation from good-for-MDPs automata to rewards proves to be quite simple: for a given parameter $\lambda \in]0, 1[$, whenever one passes an accepting state (or transition), go to an accepting sink with probability λ , and continue with a probability of $1 - \lambda$.

It is easy to see that the chance of reaching the accepting sink is at least the chance of satisfying the Büchi objective. But it also holds that, when λ goes to 0, the chance of reaching the accepting sink converges to the chance of satisfying the Büchi objective, and that optimal strategies for the reachability goal are stable, and optimal for the Büchi objective, for all sufficiently small values of λ .

This reachability objective can then be handled with standard RL techniques. We have used Q-learning, which would normally wrap the reachability objective into a discounted payoff objective to guarantee contraction.

3 Related Work

Reinforcement learning for ω -regular objectives has first been applied using deterministic Rabin automata [11, 14]. While there are small examples where this method does not produce optimal results [4], they have paved the way for further exploration.

The translation through reachability [4] has been complemented by an integrated approach to discounted payoff that uses different discount factors for accepting and non-accepting transitions [1]. This can be mimicked by replacing the reduction to reachability by replacing the transition to an accepting sink by obtaining a reward while discounting the rest of the game with a factor of $1 - \lambda$, while not discounting otherwise. Wrapping this approach into another discount scheme for RL (in order to guarantee contraction) leads to the same set of different discount factors [5].

Suitable limit deterministic automata have been replaced by good-for-MDP automata [6]. While current approaches to obtain them from general nondeterministic Büchi automata hinge on breakpoint constructions, it is also possible (but expensive: PSPACE hard and in EXPTIME) to check whether or not a nondeterministic Büchi automaton is already good-for-MDPs [15].

The learning approach extends to Markov games [8], which need good-for-games automata [10], and thus parity objectives. While using games as such does not seem to be problematic, handling more powerful acceptance conditions comes at a cost, broadly speaking by using the small parameter λ in different powers.

Being able to handle games also paves the way for using alternating automata (so long as they are good-for-MDPs) for ordinary MDPs, which has proven to allow for efficient translations from deterministic Streett to alternating Büchi automata that are good-for-MDPs, while their translation to nondeterministic Büchi automata (GFM or not) is expensive [9].

References

1. 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 2020, Paris, France, May 31–August 31, 2020, pp. 10349–10355. IEEE (2020). <https://doi.org/10.1109/ICRA40945.2020.9196796>
2. Büchi, J.R.: On a decision method in restricted second order arithmetic. In: Proceedings of the International Congress on Logic, Methodology, and Philosophy of Science, 1960, Berkeley, California, USA, pp. 1–11. Stanford University Press (1962)
3. Courcoubetis, C., Yannakakis, M.: The complexity of probabilistic verification. *J. ACM* **42**(4), 857–907 (1995)
4. Hahn, E.M., Perez, M., Schewe, S., Somenzi, F., Trivedi, A., Wojtczak, D.: Omega-regular objectives in model-free reinforcement learning. In: Vojnar, T., Zhang, L. (eds.) TACAS 2019. LNCS, vol. 11427, pp. 395–412. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17462-0_27
5. Hahn, E.M., Perez, M., Schewe, S., Somenzi, F., Trivedi, A., Wojtczak, D.: Faithful and Effective Reward Schemes for Model-Free Reinforcement Learning of Omega-Regular Objectives. In: Hung, D.V., Sokolsky, O. (eds.) ATVA 2020. LNCS, vol. 12302, pp. 108–124. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59152-6_6
6. Hahn, E.M., Perez, M., Schewe, S., Somenzi, F., Trivedi, A., Wojtczak, D.: Good-for-MDPs automata for probabilistic analysis and reinforcement learning. In: Tools and Algorithms for the Construction and Analysis of Systems, pp. 306–323 (2020)
7. Hahn, E.M., Li, G., Schewe, S., Turrini, A., Zhang, L.: Lazy probabilistic model checking without determinisation. In: Proceedings of the 26th Conference on Concurrency Theory (CONCUR 2015), September 1–4, Madrid. LIPIcs, vol. 42, pp. 354–367. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany (2015)
8. Trivedi, A., Wojtczak, D.: Model-free reinforcement learning for stochastic parity games. In: Konnov, I., Kovács, L. (eds.) 31st International Conference on Concurrency Theory, CONCUR 2020, 1–4 September 2020, Vienna, Austria (Virtual Conference). LIPIcs, vol. 171, pp. 21:1–21:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020)

9. Hahn, E.M., Perez, M., Schewe, S., Somenzi, F., Trivedi, A., Wojtczak, D.: Model-free reinforcement learning for branching Markov decision processes. In: Silva, A., Leino, K.R.M. (eds.) CAV 2021. LNCS, vol. 12760, pp. 651–673. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-81688-9_30
10. Henzinger, T.A., Piterman, N.: Solving games without determinization. In: Ésik, Z. (ed.) CSL 2006. LNCS, vol. 4207, pp. 395–410. Springer, Heidelberg (2006). https://doi.org/10.1007/11874683_26
11. Hiromoto, M., Ushio, T.: Learning an optimal control policy for a Markov decision process under linear temporal logic specifications. In: Symposium Series on Computational Intelligence, pp. 548–555, December 2015
12. Manna, Z., Pnueli, A.: The Temporal Logic of Reactive and Concurrent Systems *Specification*. Springer, New York (1991). <https://doi.org/10.1007/978-1-4612-0931-7>
13. Perrin, D., Pin, J.É.: Infinite Words: Automata, Semigroups. Logic and Games. Elsevier, Amsterdam (2004)
14. Sadigh, D., Kim, E., Coogan, S., Sastry, S.S., Seshia, S.A.: A learning based approach to control synthesis of Markov decision processes for linear temporal logic specifications. In: IEEE Conference on Decision and Control (CDC), pp. 1091–1096, December 2014
15. Schewe, S., Tang, Q., Zhanabekova, T.: Deciding what is good-for-MDPS. CoRR abs/2202.07629 (2022), <https://arxiv.org/abs/2202.07629>
16. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, 2nd edn. MIT Press, London (2018)
17. Thomas, W.: Handbook of Theoretical Computer Science, Chap. Automata on Infinite Objects, pp. 133–191. The MIT Press/Elsevier, London (1990)
18. Vardi, M.Y.: Automatic verification of probabilistic concurrent finite-state programs. In: 26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21–23 October 1985. pp. 327–338. IEEE Computer Society (1985)
19. Wiering, M., van Otterlo, M. (eds.): Reinforcement Learning: State of the Art. Springer, Heidelberg (2012). <https://doi.org/10.1007/978-3-642-27645-3>