An Automata-Theoretic Approach to . . . .

Mateo Perez, Fabio Somenzi, Ashutosh Trivedi

University of Colorado Boulder

August 1, 2022

Joint work with Ernst Moritz Hahn, Sven Schewe, and Dominik Wojtczak

An Automata-Theoretic Approach to Reinforcement Learning

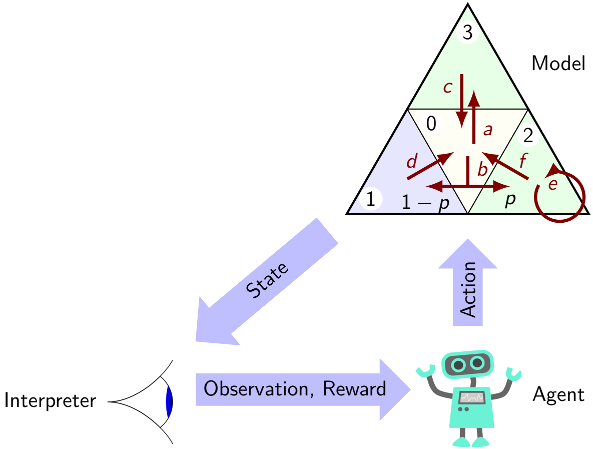
Mateo Perez, Fabio Somenzi, Ashutosh Trivedi

University of Colorado Boulder

August 1, 2022

Joint work with Ernst Moritz Hahn, Sven Schewe, and Dominik Wojtczak

Reinforcement Learning



The problem

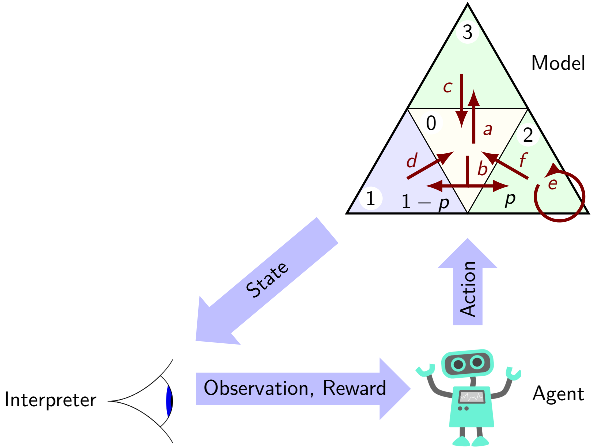
Specifying objectives via reward simplifies the development of new algorithms.
However, it is tedious and error-prone to specify reward manually.

The problem

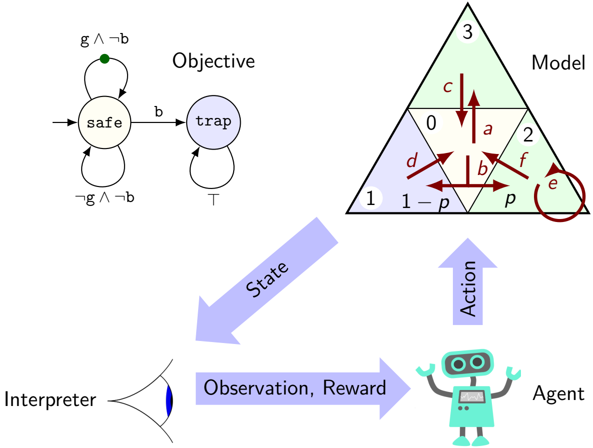
Specifying objectives via reward simplifies the development of new algorithms.
However, it is tedious and error-prone to specify reward manually.

Let's specify a formal requirement and have it "compiled" to the representation used by RL.
We can use Linear Temporal Logic and ideas from probabilistic model checking!

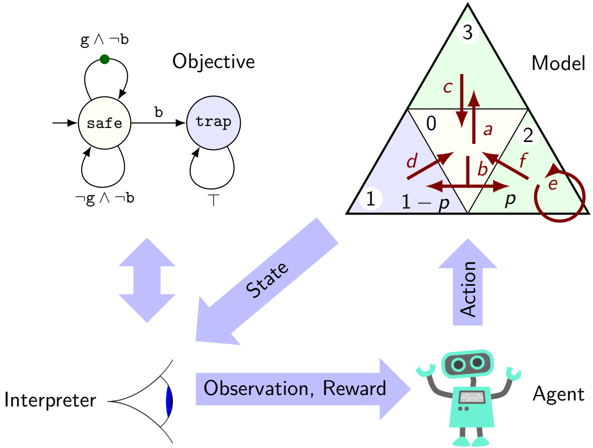
Model-free reward translation



Model-free reward translation



Model-free reward translation



Objective

Model

State

Action

Observation, Reward

Interpreter

Agent

Rabin to discounted reward[1]

Can we use Rabin automata? No correct translation has been proposed.

[1] An Impossibility Result in Automata-Theoretic Reinforcement Learning. ATVA 2022.

Rabin to discounted reward[1]

Can we use Rabin automata? No correct translation has been proposed.

Optimal strategies in RL mix.

$$Q^*(s, a_0) = 5, Q^*(s, a_1) = 5, Q^*(s, a_2) = 3$$

Any strategy that mixes a_0 and a_1 in s maintains optimality.

[1] An Impossibility Result in Automata-Theoretic Reinforcement Learning. ATVA 2022.

Rabin to discounted reward[1]

Can we use Rabin automata? No correct translation has been proposed.

Optimal strategies in RL mix.

$$Q^*(s, a_0) = 5, Q^*(s, a_1) = 5, Q^*(s, a_2) = 3$$

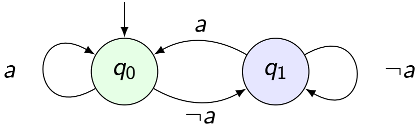
Any strategy that mixes a_0 and a_1 in s maintains optimality.

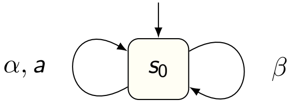
Optimal strategies for Rabin may not mix!

[1] An Impossibility Result in Automata-Theoretic Reinforcement Learning. ATVA 2022.

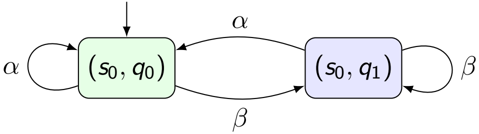
Rabin to discounted reward

$\mathbf{FG}a \vee \mathbf{FG}\neg a$





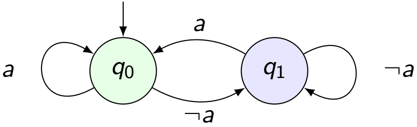
$\{\langle q_0, q_1\rangle, \langle q_1, q_0\rangle\}$

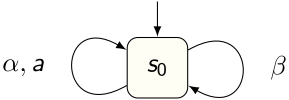


α is optimal, β is optimal, but mixing α and β is not.

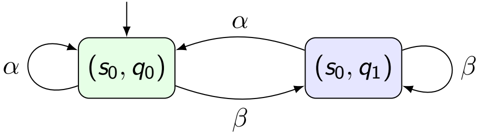
Rabin to discounted reward

$\mathbf{FG}a \vee \mathbf{FG}\neg a$





$\{\langle q_0, q_1\rangle, \langle q_1, q_0\rangle\}$



α is optimal, β is optimal, but mixing α and β is not.

We can not reduce a Rabin automaton directly to reward without additional memory.

Büchi to discounted reward

To use Büchi automata, we may require nondeterminism.

[1] Automatic Verification of Probabilistic Concurrent Finite-State Programs. Moshe Y. Vardi. FOCS 1985.

[2] Limit-Deterministic Büchi Automata for Linear Temporal Logic. Sickert et al. 2016

[3] Good-for-MDPs Automata for Probabilistic Analysis and Reinforcement Learning. TACAS 2020

Büchi to discounted reward

To use Büchi automata, we may require nondeterminism.

For an automata-theoretic approach to model-checking of probabilistic programs "we eliminate the need for a complete determinization of the given automaton."[1] – Moshe Vardi

[1] Automatic Verification of Probabilistic Concurrent Finite-State Programs. Moshe Y. Vardi. FOCS 1985.

[2] Limit-Deterministic Büchi Automata for Linear Temporal Logic. Sickert et al. 2016

[3] Good-for-MDPs Automata for Probabilistic Analysis and Reinforcement Learning. TACAS 2020

Büchi to discounted reward

To use Büchi automata, we may require nondeterminism.

For an automata-theoretic approach to model-checking of probabilistic programs "we eliminate the need for a complete determinization of the given automaton."[1] – Moshe Vardi

We can use suitable limit-deterministic Büchi automata[2] and more generally Good-for-MDPs (GFM) automata.[3]

[1] Automatic Verification of Probabilistic Concurrent Finite-State Programs. Moshe Y. Vardi. FOCS 1985.

[2] Limit-Deterministic Büchi Automata for Linear Temporal Logic. Sickert et al. 2016

[3] Good-for-MDPs Automata for Probabilistic Analysis and Reinforcement Learning. TACAS 2020

GFM Büchi to discounted reward[1]

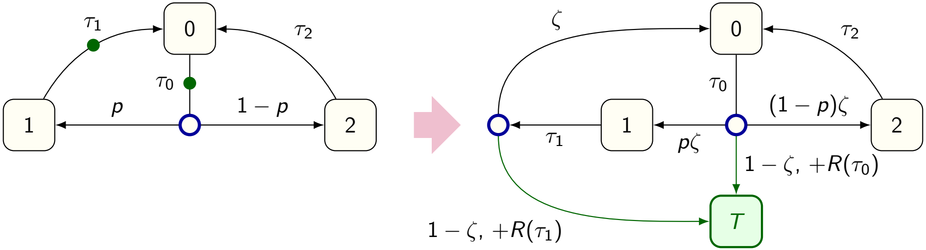
How do we assign the reward?

- $+1$ reward on accepting edges and 0 otherwise does not work. Why?
- maximize expected frequency of accepting edges $\neq$ maximize probability that the frequency is positive
- Seeing accepting edges on every other step with probability 1 is valued lower than seeing accepting edges on every step with probability $2/3$.

[1] Omega-Regular Objectives in Model-Free Reinforcement Learning. TACAS 2019

GFM Büchi to discounted reward

- Instead, let's introduce an additional parameter $\zeta \in (0,1)$.
- On accepting edges with probability $1-\zeta$ assign $+1$ reward and terminate.



- Under total reward, satisfying traces are given a value of 1.
- Under total reward, traces that are not satisfying are given a value of ε with $\lim_{\zeta\uparrow 1}\varepsilon = 0$.

GFM Büchi to discounted reward

Theorem (Limit reachability)

For a given MDP, there exists a threshold for $\zeta' \in (0,1)$ *and for* $\gamma' \in (0,1)$ *such that for any* $\zeta > \zeta'$ *and* $\gamma > \gamma'$ *maximizing the discounted reward from the construction above maximizes the probability of satisfaction of the Büchi objective.*

Summary

Instead of assigning reward manually, perform a translation from a high-level objective.
For omega-regular objectives (LTL):

[1] Model-Free Reinforcement Learning for Stochastic Parity Games. CONCUR 2020.
[2] Model-Free Reinforcement Learning for Lexicographic Omega-Regular Objectives. FM 2021.

Summary

Instead of assigning reward manually, perform a translation from a high-level objective.
For omega-regular objectives (LTL):

- Rabin: Not possible without additional memory. There is a simple on-the-fly translation to Büchi.

[1] Model-Free Reinforcement Learning for Stochastic Parity Games. CONCUR 2020.
[2] Model-Free Reinforcement Learning for Lexicographic Omega-Regular Objectives. FM 2021.

Summary

Instead of assigning reward manually, perform a translation from a high-level objective.
For omega-regular objectives (LTL):

- Rabin: Not possible without additional memory. There is a simple on-the-fly translation to Büchi.
- GFM Büchi: Simply rewarding accepting edges isn't correct. Instead, flip a weighted coin after each accepting edge to reach an accepting sink.

[1] Model-Free Reinforcement Learning for Stochastic Parity Games. CONCUR 2020.
[2] Model-Free Reinforcement Learning for Lexicographic Omega-Regular Objectives. FM 2021.

Summary

Instead of assigning reward manually, perform a translation from a high-level objective.
For omega-regular objectives (LTL):

- Rabin: Not possible without additional memory. There is a simple on-the-fly translation to Büchi.
- GFM Büchi: Simply rewarding accepting edges isn't correct. Instead, flip a weighted coin after each accepting edge to reach an accepting sink.
- Parity[1]: Needed for games. Have a set of increasingly weighted coins with accepting and rejecting sinks.

[1] Model-Free Reinforcement Learning for Stochastic Parity Games. CONCUR 2020.
[2] Model-Free Reinforcement Learning for Lexicographic Omega-Regular Objectives. FM 2021.

Summary

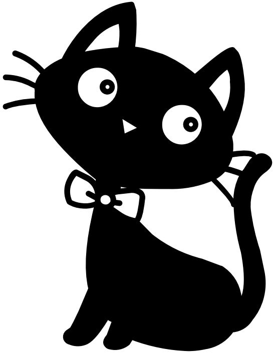
Instead of assigning reward manually, perform a translation from a high-level objective.
For omega-regular objectives (LTL):

- Rabin: Not possible without additional memory. There is a simple on-the-fly translation to Büchi.
- GFM Büchi: Simply rewarding accepting edges isn't correct. Instead, flip a weighted coin after each accepting edge to reach an accepting sink.
- Parity[1]: Needed for games. Have a set of increasingly weighted coins with accepting and rejecting sinks.
- Lexicographic[2]: Add a memory gadget. Then, use large enough weights to separate the associated Büchi rewards.

[1] Model-Free Reinforcement Learning for Stochastic Parity Games. CONCUR 2020.

[2] Model-Free Reinforcement Learning for Lexicographic Omega-Regular Objectives. FM 2021.

Mungojerrie[1]



MUNGOJERRIE

Formal Reinforcement Learning

https://plv.colorado.edu/mungojerrie/

[1] Mungojerrie: Reinforcement Learning of Linear-Time Objectives. Preprint 2021

Thank you!