Model-Free RL for Control Synthesis for MDPs and Stochastic Games

Alper Kamil Bozkurt, Yu Wang, Michael M. Zavlanos and Miroslav Pajic

Department of Electrical and Computer Engineering
Department of Computer Science
Pratt School of Engineering
Duke University

Duke PRATT SCHOOL of ENGINEERING

CPSL Duke

DUKE ROBOTICS

Preliminaries and Problem Statement

Model: (Labeled) Turn-Based Zero-Sum **Stochastic Games**

$$\mathcal{G} = \left(S, (S_\mu, S_\nu), A, P, s_0, AP, L\right)$$

- $S = S_\mu \cup S_\nu$ is a finite set of states; s_0 is an initial state
- S_μ, S_ν are the controller and the environment states
- A is a finite set of actions
- P is the transition probability function (unknown)
- AP is a set of labels/atomic propositions
- $L: S \rightarrow AP$ is a labeling function

Specification: Linear Temporal Logic (**LTL**)

$$\varphi \coloneqq \mathrm{true} \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 \mathrm{U} \varphi_2, \quad a \in AP$$

- $\varphi_1 \vee \varphi_2 \coloneqq \neg(\neg\varphi_1 \wedge \neg\varphi_2)$ | $\varphi_1 \rightarrow \varphi_2 \coloneqq \neg\varphi_1 \vee \varphi_2$
- $\Diamond\varphi \coloneqq \mathrm{true}\ \mathrm{U}\ \varphi$ | $\Box\varphi \coloneqq \neg(\Diamond\neg\varphi)$

Output: Finite-Memory **Strategy**

$$\pi = (M, \Delta, \alpha, m_0)$$

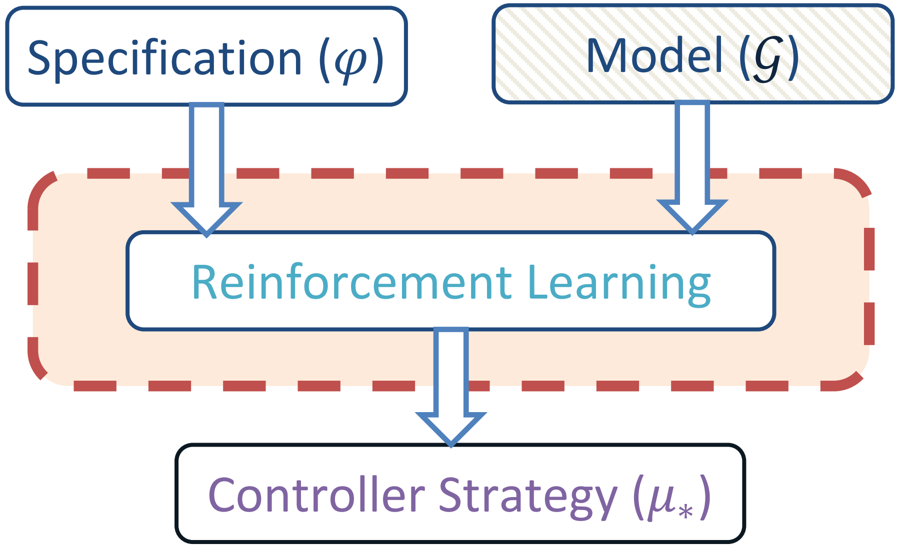
- M is a finite set of modes; m_0 is an initial state
- $\Delta: M \times S \rightarrow M$ is the transition function
- $\alpha: M \times S \rightarrow A$ maps the mode state pairs to actions

Problem Statement

Given a stochastic game $\boldsymbol{\mathcal{G}}$ where the transition probabilities and the topology is unknown and an LTL specification $\boldsymbol{\varphi}$, design a model-free RL algorithm that finds a finite-memory controller strategy $\boldsymbol{\mu}_*$ that satisfies

$$\mu_* = \operatorname{argmax}_\mu \min_\nu Pr_{\mu,\nu}(\mathcal{G} \vDash \varphi)$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are controller and environment strategies

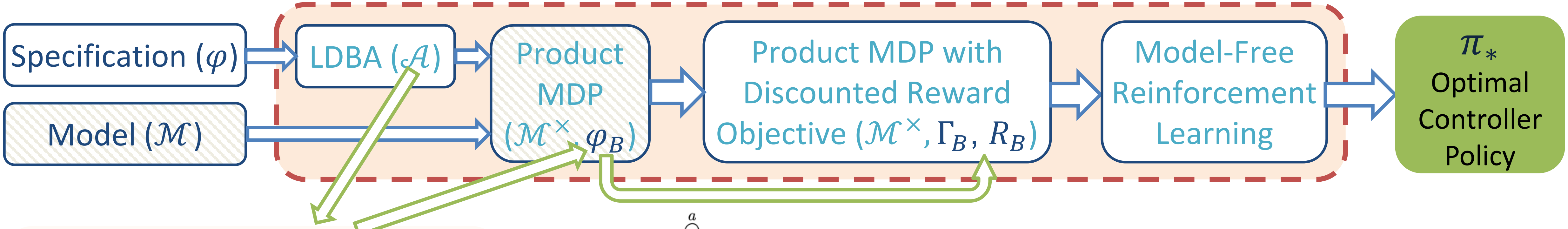


Control Synthesis via RL for MDPs

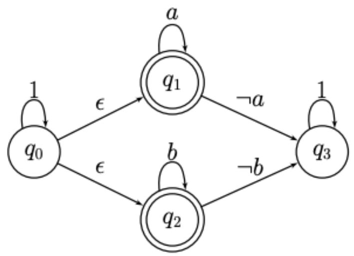
Problem Statement for MDPs

Given an MDP $\mathcal{M}$ where the transition probabilities and the topology are unknown and an LTL specification φ, design a model-free RL algorithm that finds a finite-memory objective policy π_* that satisfies

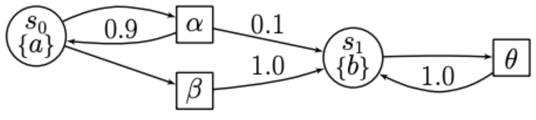
$$\pi_* = argmax_\pi Pr_\pi(\mathcal{M} \vDash \varphi)$$



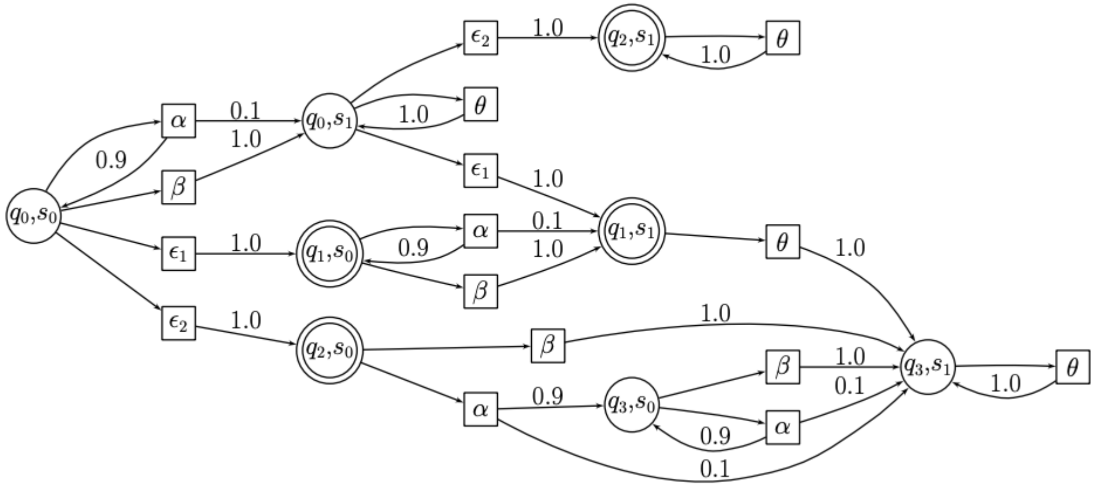
Limit-Deterministic <u>*Büchi*</u> Automata (LDBA) – consist of two deterministic components the ***initial*** and ***accepting***. The only nondeterministic transitions are the ϵ-moves from the initial component to the accepting components.



(a) A derived LDBA $\mathcal{A}$ for the LTL formula $\varphi = \Diamond\Box a \vee \Diamond\Box b$



(b) An example MDP $\mathcal{M}$; the circles denote MDP states, rectangles denote actions, and numbers transition probabilities



(c) The obtained product MDP

[1] A. K. Bozkurt, Y. Wang, M. M. Zavlanos, and M. Pajic, "Control Synthesis from LTL Specifications using Model-Free Reinforcement Learning", *IEEE International Conference on Robotics and Automation* (**ICRA**), 2020

Product Game Construction

Specification (φ) → DRA ($\mathcal{A}$) → Product Game ($\mathcal{G}^{\times}, \varphi_{Rabin}$)

Model ($\mathcal{G}$) → Product Game ($\mathcal{G}^{\times}, \varphi_{Rabin}$)

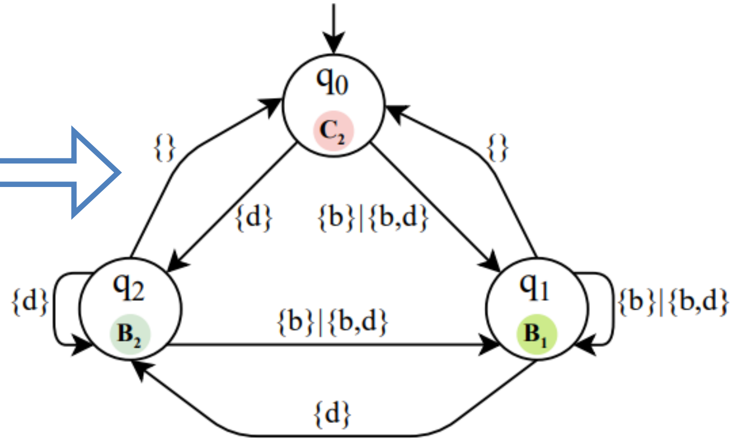
State Space
$S \mid \langle s \rangle$

→ Augmented State Space
$S^{\times} = S \times Q \mid \langle s, q \rangle$

LTL Specification
φ

→ Rabin Acceptance Condition
$$\varphi_{Rabin} = \bigvee_{i=1}^{k} \square\Diamond B_i \wedge \neg\square\Diamond C_i$$

$\varphi = \square\Diamond b \vee \Diamond\square d$ →



Rabin(1) Acceptance Condition

$$\varphi^{(1)}_{Rabin} = \square\Diamond B \wedge \neg\square\Diamond C$$

Pure and ***memoryless*** strategies suffice for both Player 1 (**Controller**) and Player 2 (**Environment**) for $\boldsymbol{\varphi^{(1)}_{Rabin}}$.

(Chatterjee et al.,2012)

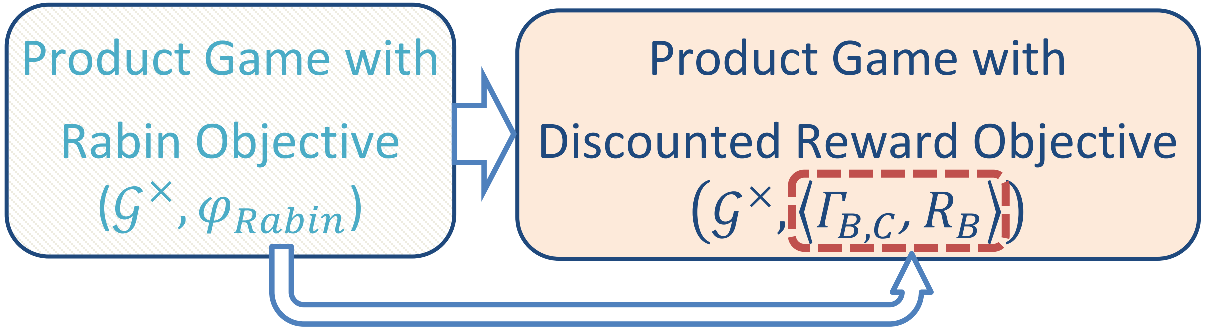
For pure and memoryless $\mu^{\times}$ and $\nu^{\times}$:

$$\mu^{\times}_{*} = argmax_{\mu^{\times}}\, min_{\nu^{\times}}\, Pr_{\mu^{\times},\nu^{\times}}\left(\mathcal{G}^{\times} \vDash \varphi^{(1)}_{Rabin}\right)$$

Optimal Finite-Memory Controller Strategy

$$\mu_{*} = argmax_{\mu}\, min_{\nu}\, Pr_{\mu,\nu}(\mathcal{G} \vDash \varphi)$$

Rabin(1) Acceptance Condition as Sum of Discounted Rewards

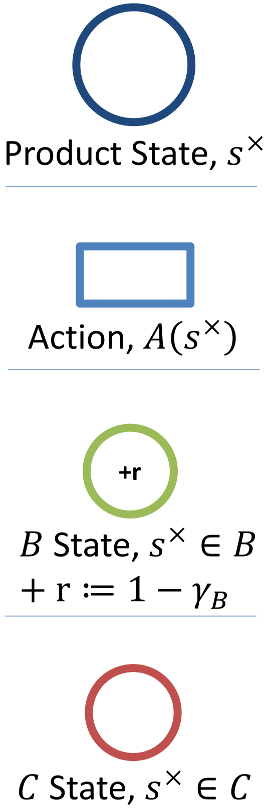


$$R_B(s) \coloneqq \begin{cases} 1-\gamma_B, & s^\times \in B \\ 0, & s^\times \notin B \end{cases}$$

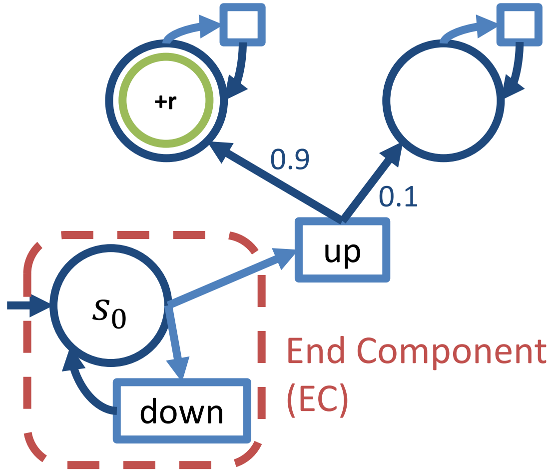
Reward Function

$$\Gamma_{B,C}(s^\times) \coloneqq \begin{cases} \gamma_B, & s^\times \in B \\ \gamma_C, & s^\times \in C \\ \gamma, & otherwise \end{cases}$$

Discount Function



Why γ?

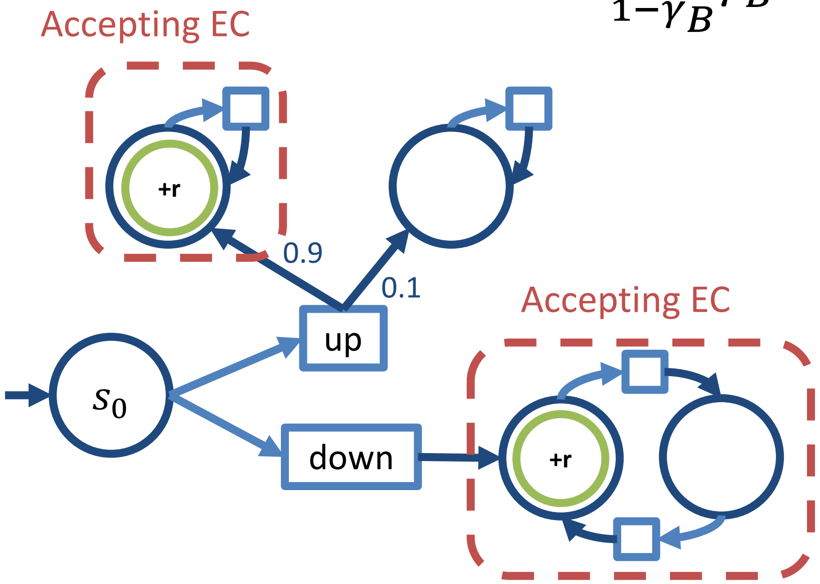


$\hat{Q}(down)$ cannot be decreased during model-free learning

Why γ_B?

$Q(up) = 0.9$

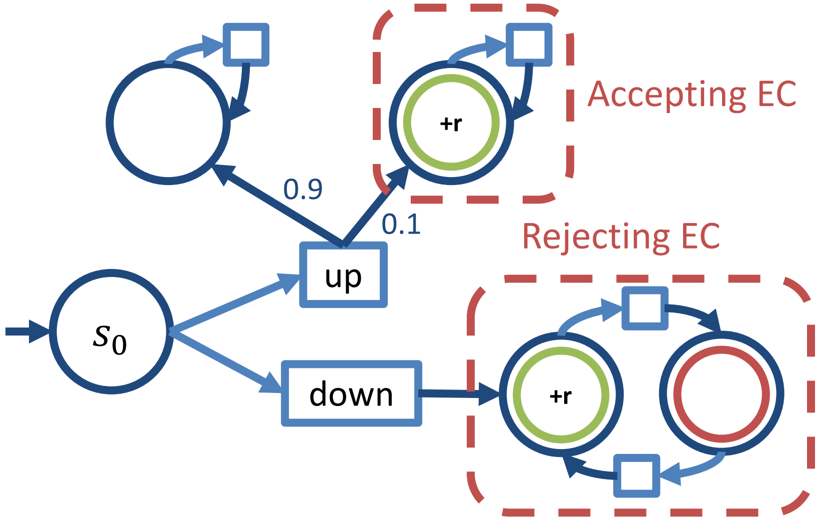
$$Q(down) = \frac{1}{1+\frac{1-\gamma}{1-\gamma_B}\gamma_B}$$



Why γ_C?

$Q(up) = 0.1$

$$Q(down) = \frac{1}{1+\frac{1-\gamma_C}{1-\gamma_B}\gamma_B}$$



Main Theoretical Results

Reward Function

$$R_B^\gamma(s) \coloneqq \begin{cases} 1-\gamma_B, & s^\times \in B \\ 0, & s^\times \notin B \end{cases}$$

Discount Function

$$\Gamma_{B,C}^\gamma(s^\times) \coloneqq \begin{cases} \gamma_B(\gamma), & s^\times \in B \\ \gamma_C(\gamma), & s^\times \in C \\ \gamma, & otherwise \end{cases}$$

Discounted Rewards

$$G_{B,C}^\gamma(\sigma) = \sum_{i=0}^{\infty}\left(\prod_{j=0}^{i-1}\Gamma_{B,C}^\gamma(\sigma[j])\right)R_B^\gamma(\sigma[i])$$

Discount Constraints

$$\lim_{\gamma\to 1^-}\frac{1-\gamma}{1-\gamma_B(\gamma)} = \lim_{\gamma\to 1^-}\frac{1-\gamma_B(\gamma)}{1-\gamma_C(\gamma)} = 0$$

Theorem

$$\lim_{\gamma\to 1^-}\mathbb{E}_{\sigma\sim\mathcal{G}^\times_{\mu^\times,\nu^\times}}\left[G_{B,C}^\gamma(\sigma)\right] = Pr_{\mu^\times,\nu^\times}(\mathcal{G}^\times \vDash \Box\Diamond B \wedge \neg\Box\Diamond C)$$

Corollary

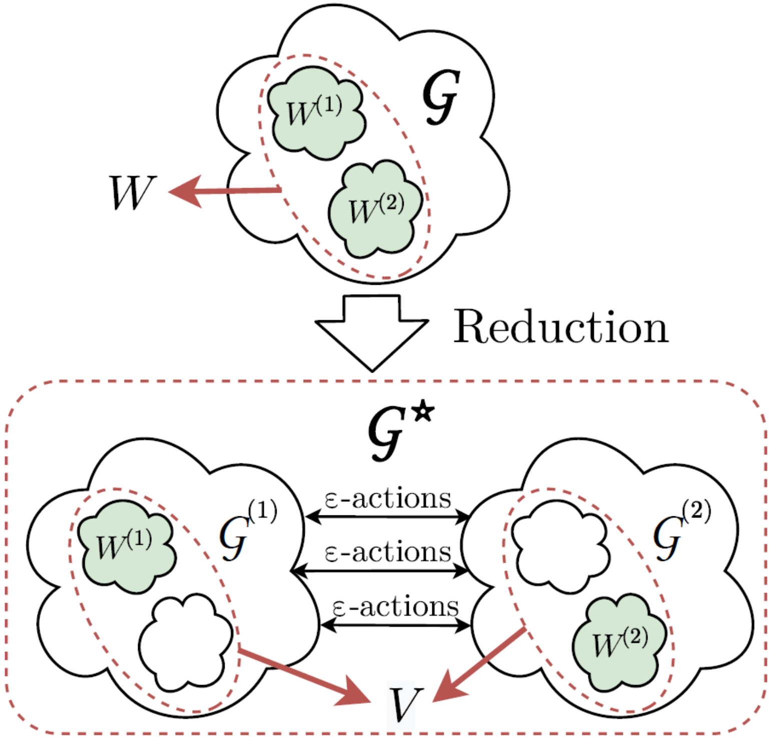
There exists $\gamma' < 1$ such that for all $\gamma \geq \gamma'$,

$$\mu_*^\times = argmax_{\mu^\times}\, min_{\nu^\times}\mathbb{E}_{\sigma\sim\mathcal{G}^\times_{\mu^\times,\nu^\times}}\left[G_{B,C}^\gamma(\sigma)\right]$$

$$= argmax_{\mu^\times}\, min_{\nu^\times} Pr_{\mu^\times,\nu^\times}(\mathcal{G}^\times \vDash \Box\Diamond B \wedge \neg\Box\Diamond C)$$

μ_*: Optimal Finite-Memory Controller Strategy

Multiple Rabin Pairs

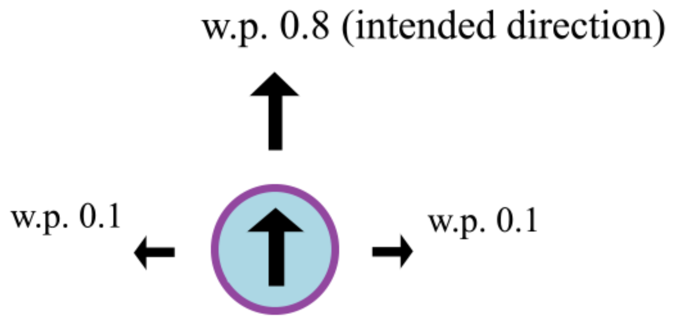


[1] A. K. Bozkurt, Y. Wang, M. M. Zavlanos, and M. Pajic, "Model-Free Reinforcement Learning for Stochastic Games with Linear Temporal Logic Objectives", arXiv:2010.01050, 2020

Case Study: Avoiding Adversary

Grid World:

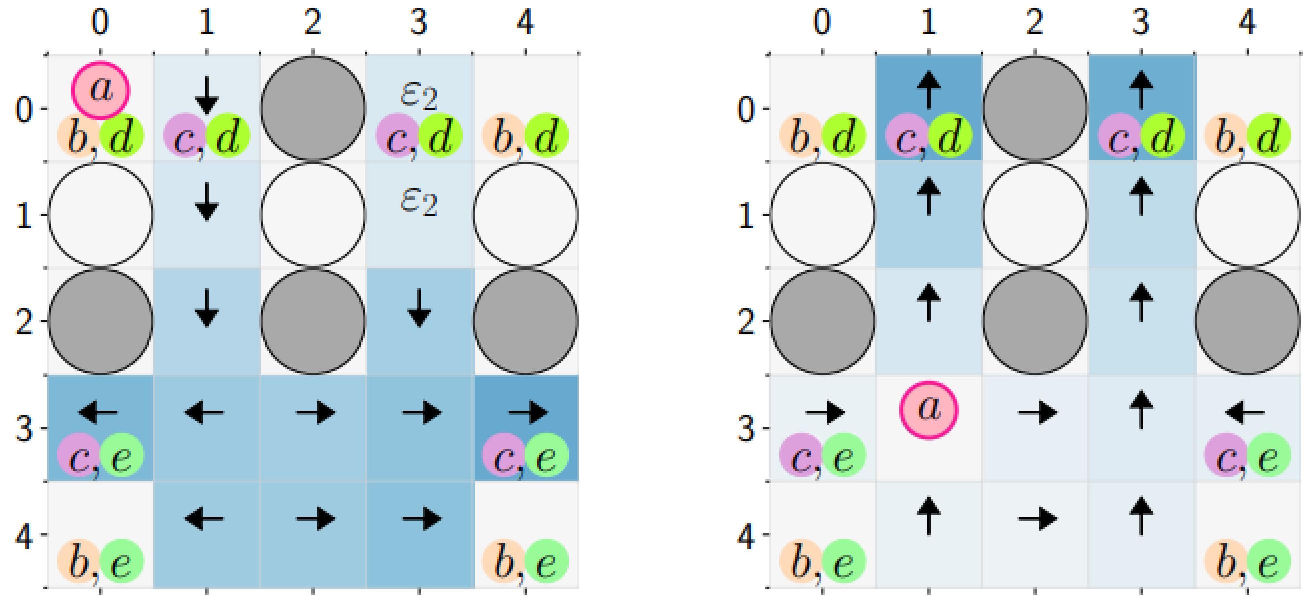
- The agent can take four actions:
  North, South, East, West
- The transition model :
  - The probability that the robot moves in the *intended* direction: **0.8**
  - The probability that the robot moves in a direction *orthogonal* to the intended direction: **0.2**
- Action: ***North***



Objective:

(1) Repeatedly visit a $\boldsymbol{b}$ and a $\boldsymbol{c}$ cell
(2) Eventually reach a safe region labeled with $\boldsymbol{d}$ or $\boldsymbol{e}$ and do not leave
(3) Avoid the adversary at all costs.

$$\varphi = \Box\Diamond \boldsymbol{b} \wedge \Box\Diamond \boldsymbol{c} \wedge (\Diamond\Box \boldsymbol{d} \vee \Diamond\Box \boldsymbol{e}) \wedge \Box\neg \boldsymbol{a}$$



(a) Adversary is at $(0,0)$ and $i{=}1$ (b) Adversary is at $(3,1)$ and $i{=}2$

The darker blue, the higher estimated satisfaction probability

Secure Planning Against Stealthy Attacks

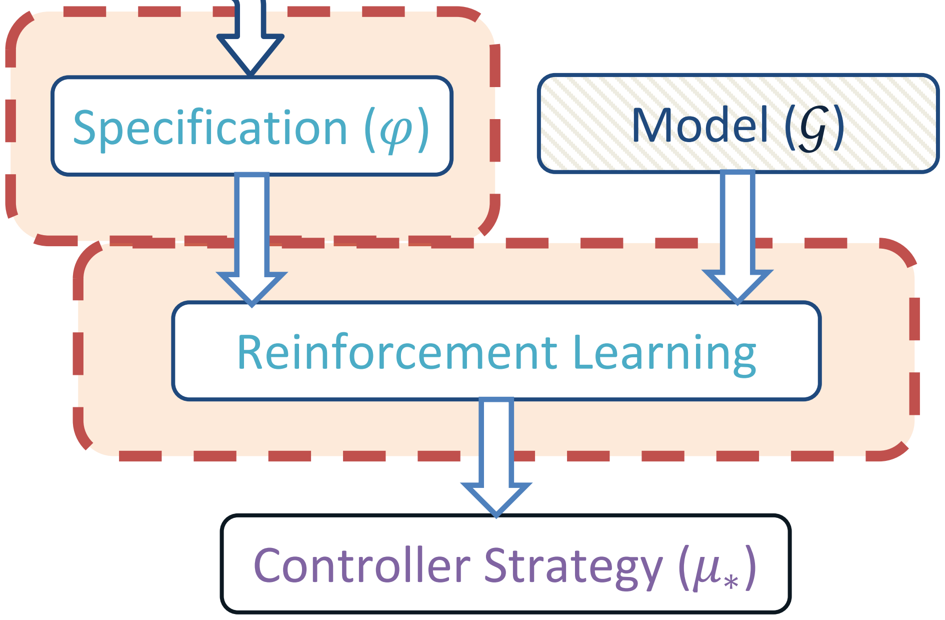
Duke
PRATT SCHOOL of ENGINEERING

Controller:

- aims to perform a given **task**
- does **not have a model** of the environment
- has a perfect knowledge of the current state
- has an intrusion-detection system (**IDS**) that monitors anomalies
- can **detect** attacks only when the IDS raises an **alarm**

Attacker:

- aims to prevent the controller from performing the given task
- has a **perfect knowledge** of the current state, the controller strategy and the IDS mechanism
- can attack on **actuators** unless detected
- tends to stay **stealthy**



LTL Formulation of Controller Objective φ

- captures the controller task and the IDS mechanism
- reflects the behavior of stealthy attackers
- translates into a small DRA

$\varphi = \varphi_{IDS} \vee \varphi_{TASK}$, where φ_{IDS} is a **reachability objective**

Example: *Counting-Based IDS*

$\varphi_{IDS} = \Diamond\big(\text{anomaly} \wedge \bigcirc(\text{anomaly} \wedge \bigcirc\Diamond^{\leq 1}(\text{attack} \wedge \bigcirc\Diamond attack))\big)$

Secure Planning Case Studies

Duke
PRATT SCHOOL *of* ENGINEERING

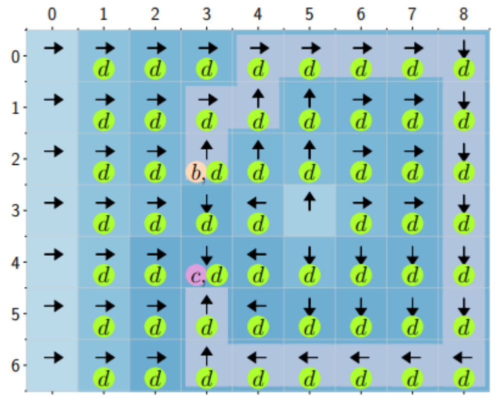
$\varphi = \varphi_{IDS} \vee \varphi_{TASK}$

$\varphi_{IDS} = \Diamond\big(\text{anomaly} \wedge \bigcirc(\text{anomaly} \wedge \bigcirc\Diamond^{\leq 1}(\text{attack} \wedge \bigcirc\Diamond \mathit{attack}))\big)$

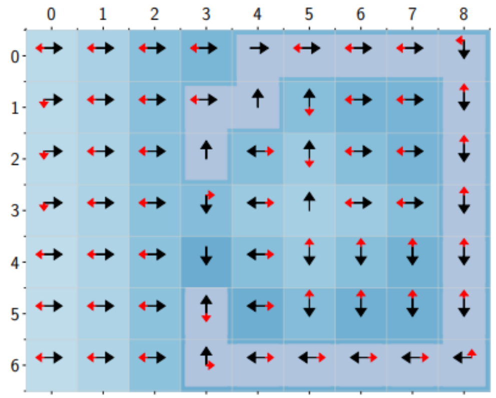
Repeated Coverage:

(1) Repeatedly visit a $\boldsymbol{b}$ and a $\boldsymbol{c}$ cell
(2) Eventually reach a safe region labeled with $\boldsymbol{d}$ and do not leave

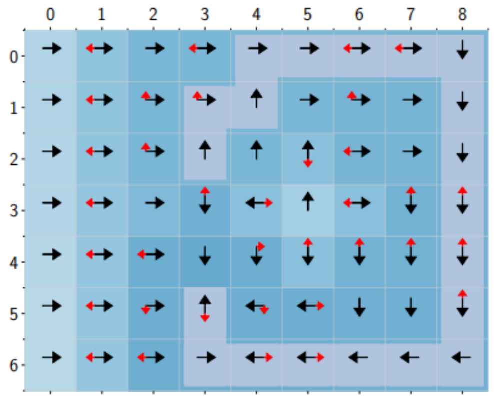
$\varphi_{TASK} = \Box\Diamond \boldsymbol{b} \wedge \Box\Diamond \boldsymbol{c} \wedge \Diamond\Box \boldsymbol{d}$



(a) The controller strategy from b to c and the labels of the cells



(b) The controller and the attacker strategies from b to c before any anomaly



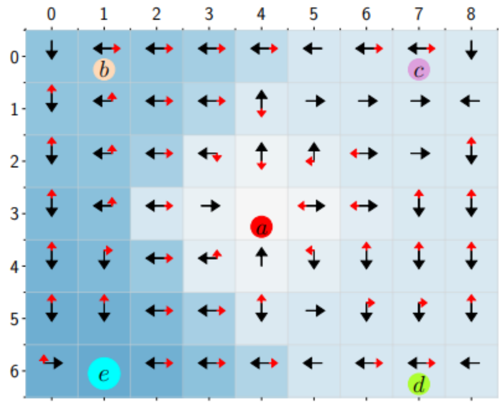
(c) The controller and the attacker strategies from b to c after one anomaly

Sequence of Tasks:

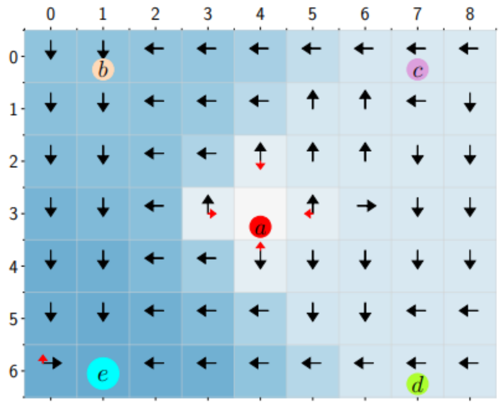
(1) Visit $\boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}$ in order
(2) Avoid the danger zone $\boldsymbol{a}$ at all costs

$\varphi_{TASK} = \Diamond\big(\boldsymbol{b} \wedge \Diamond(\boldsymbol{c} \wedge \Diamond(\boldsymbol{d} \wedge \Diamond \boldsymbol{e}))\big) \wedge \Box\neg \boldsymbol{a}$

(a) The controller strategy from d to e and the labels of the cells



(b) The controller and the attacker strategies from d to e right after an anomaly happens



(c) The controller and the attacker strategies from d to e right after an alarm is raised

Conclusion

Summary:

- We convert a control synthesis problem in stochastic games to a reinforcement learning problem
- A controller strategy maximizing the return maximizes the satisfaction probability
- Our method does not require (or learn) the transition probabilities or the topology
- Convergence of reinforcement learning is ensured

Future Work:

- More practical algorithms that converge to the desired strategy faster
- The use of approximate reinforcement learning to handle large state spaces

Thank you

Duke UNIVERSITY | PRATT SCHOOL of ENGINEERING