New Avenues in Inductive Synthesis of Finite-State Controllers for POMDPs

Filip Macák[1] R. Andriushchenko[1] M. Češka[1] S. Junges[2] J.P. Katoen[3]

[1]Brno University of Technology, Czech Republic

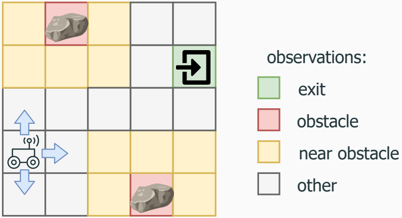
[2]Radboud University, The Netherlands

[3]RWTH Aachen University, Germany

First part of this presentation is based on paper: *Andriushchenko et al. Inductive Synthesis of Finite-State Controllers for POMDPs. In UAI'22.*

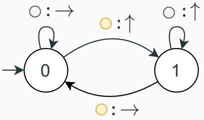
Inductive Synthesis of Finite-State Controllers for POMDPs

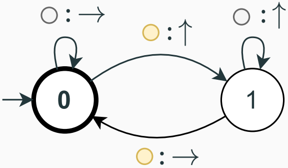
- partially observable Markov decision processes (**POMDPs**)
- indefinite-horizon specifications (no discounting)
- finite-state controllers (**FSCs**): compact automata-based representation of policies
- **goal:** quickly find a (small) admissible FSC and incrementally improve it by increasing its size

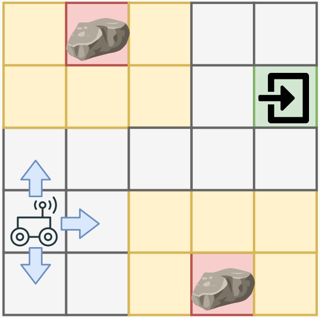


$\text{constraint}: \mathbb{P}_{\leq 0.1}\,[\mathrm{F}\ \mathit{crash}]$

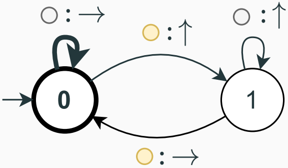
$\text{objective}: \text{minimize}\ \mathbb{R}^{\mathit{steps}}\,[\mathrm{F}\ \mathit{exit}]$

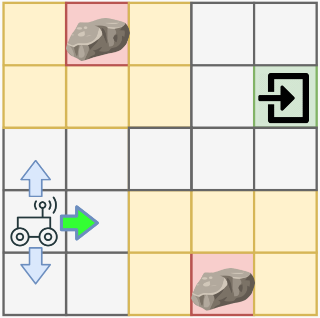



○ : →
○ : ↑
○ : ↑
0
1
○ : →

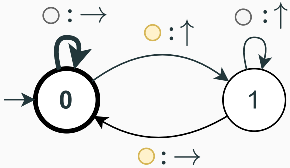


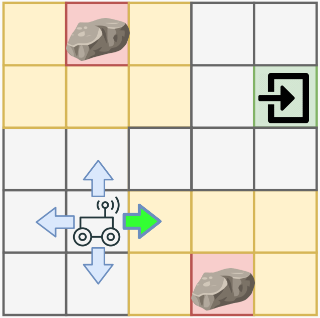
Finite-State Controller



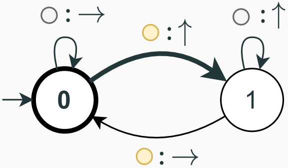


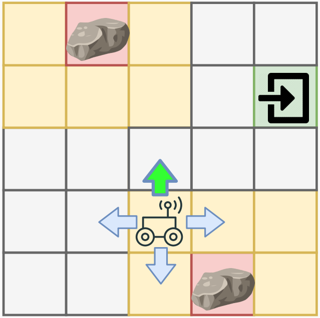
Finite-State Controller



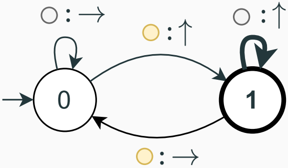


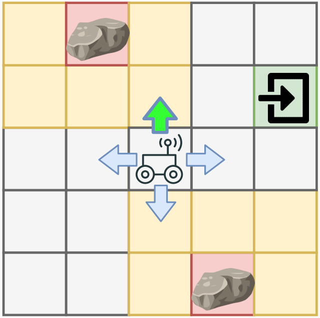
Finite-State Controller



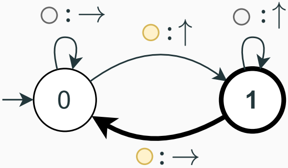


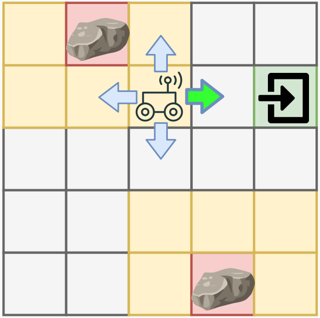
Finite-State Controller



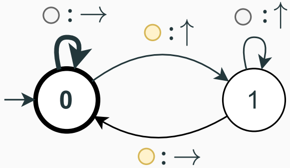


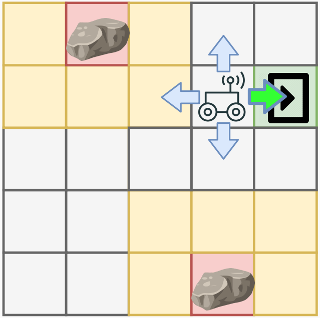
Finite-State Controller



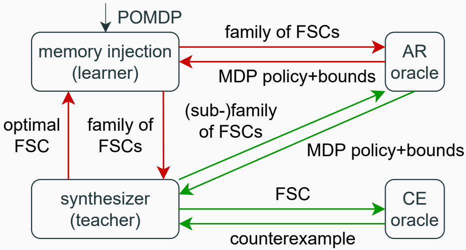


Finite-State Controller





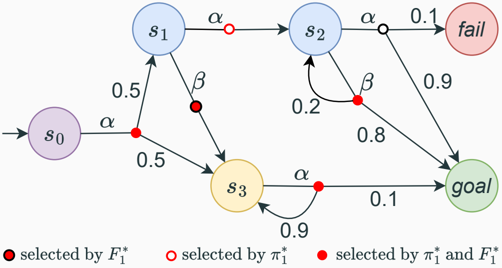
Inductive Synthesis of Finite-State Controllers for POMDPs



- **the inner synthesis loop** explores the given family of FSCs using two oracles and identifies the optimal solution
- **the outer synthesis loop** iteratively increases the size of the families explored in the inner loop by adding memory to the FSCs
- **abstraction refinement (AR) oracle** over-approximates all FSCs in the given (sub-)family using a quotient MDP
- **counterexample-based (CE) oracle** provides a counterexample in the form of a subfamily of unsatisfiable FSCs

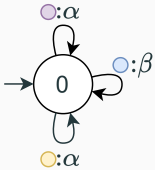
Inductive Synthesis of Finite-State Controllers for POMDPs

- simple POMDP example
- colors encode **observations**, blue states s_1 and s_2 are indistinguishable
- $\boldsymbol{F_1^*}$ represents 1-state FSC, $\boldsymbol{\pi_1^*}$ represents the optimizing policy for the fully-observable MDP
- 1-state FSC can only choose one action for each observation
- optimal 1-state FSC and the optimizing policy choose different actions in blue states

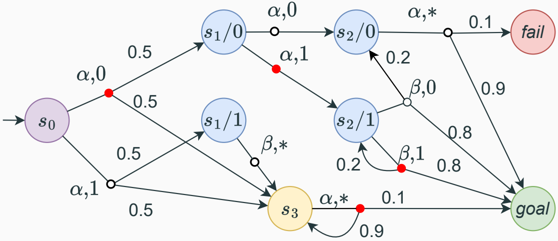


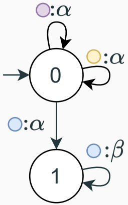
$$\text{constraint}: \mathbb{P}_{\leq 0}\,[\mathrm{F}\ \mathit{fail}]$$

$$\text{minimize}\ \mathbb{R}^{\mathit{steps}}\,[\mathrm{F}\ \mathit{goal}]$$



Inductive Synthesis of Finite-State Controllers for POMDPs





- to find an optimal FSC we have to add memory node for the blue observation
- adding a memory node creates a new family of FSCs
- in our example the family $\mathcal{F}_2$ is represented by the **quotient MDP** shown above (left figure)
- the FSC F_2^* (right figure) represents the optimal policy

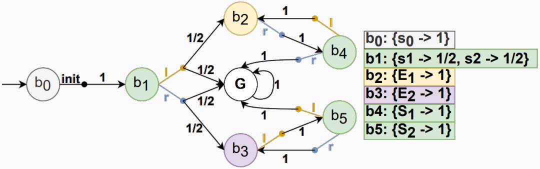
The inductive synthesis is implemented in a tool called PAYNT

Results

Benchmark		Size		PRISM	Storm		Inductive synthesis		Upper-bounds
Model	Spec.	S/Act	Z		first	best	fastest	best	
Grid-av 4-0.1	$P_{\max}$	17 59	4	[0.21,1] $< 1s$	0.82 $< 1s$	0.85 124s	0.92 (4) $< 1s$	**0.93 (5f)** **53s***	≤ 0.99
Grid 30-sl	$R_{\min}$	900 3587	37	TO/MO	121 1s	- -	**119 (6)** **150s***	- -	≥ 116.1
Maze sl	$R_{\min}$	15 54	8	[**7.09**,7.09] **2s**	7.67 $< 1s$	- -	**7.14 (3)** $< 1s$	**7.09 (3f)** **1s***	≥ 7.08
Crypt 4	$P_{\max}$	1972 4612	510	[0.33,0.79] 6s	**0.33** $< 1s$	- -	**0.33 (0)** $< 1s$	- -	≤ 0.33
Nrp 8	$P_{\max}$	125 161	41	[0.13,0.24] 3s	**0.13** $< 1s$	- -	**0.13 (0)** $< 1s$	- -	≤ 0.13
Hallway	$R_{\min}$	61 301	23	TO/MO	19.3 $< 1s$	19.2 $< 1s$	**16.3 (1)** $< 1s$	**14.9 (4f)** **218s***	≥ 12.4
Drone 4-2	$P_{\max}$	1226 3026	761	TO/MO	0.86 $< 1s$	0.91 138s	0.94 (0) $< 1s$	**0.97 (2)** **326s**	≤ 0.97
Refuel 6	$P_{\max}$	208 565	50	[0.67,0.72] 136s	**0.67** $< 1s$	- -	0.44 (2) $< 1s$	0.67 (2f) 45s*	≤ 0.69
Netw-p 2-8-20	$R_{\max}$	$2 \cdot 10^4$ $3 \cdot 10^4$	4909	[**557**,557] **1099s**	537 $< 1s$	- -	540 (0) 105s	- -	≤ 558
Rocks 12	$R_{\min}$	6553 $3 \cdot 10^4$	1645	TO/MO	38 $< 1s$	**20** **47s**	42 (0) 1s	- -	≥ 20

Belief-Based Methods

- most of the state-of-the-art approaches use **beliefs**
- probabilistic model checker **Storm**
- constructing **Belief MDP**
- construction techniques: **cut-offs**, **clipping**
- they provide good bounds, but the results are not as easy to interpret as **FSCs**
- can we use them to enhance the synthesis loop?



- finding **memoryless** scheduler in **Belief MDP** gives us **observation-based** scheduler for the initial POMDP

Belief-Based Methods Integration

- **what** information should we extract from the Belief MDP and it's scheduler?
- can we use the extracted information to enhance both inner and outer synthesis loop? And **how**?
- **when** should we use the extracted information?

- actions taken in belief MDP scheduler, bound values, belief MDP structure
- **inner synthesis loop**: improved exploration of the family of FSCs, using bounds to prune the design space
- **outer synthesis loop**: memory injection improvement based on belief MDP scheduler
- proof of concept experiments show promising results

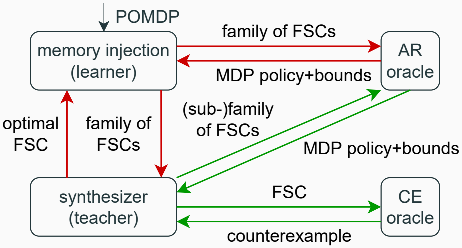
Integration experiments

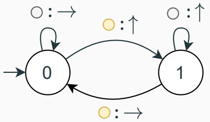
Benchmark Model	Spec.	Storm clipping	Storm cut-offs	Storm over-app	PAYNT	PAYNT + Storm
Grid-av 4-0.1	P_{max}	0.85 124s	**0.93** 66s	0.94	0.92 $< 1s$	**0.93** 78s
Network 2-8-20	R_{min}	6.56 3s	**3.2** 13s	3.2	4.24 188s	4.19 183s
Network 3-8-20	R_{min}	11.88 403s	11.15 62s	6.26	11.01 188s	**10.91** 256s
Rocks 12	R_{min}	**20** 51s	33.24 66s	20	42 $< 1s$	**20** 51s
Rocks 16	R_{min}	**26** 86s	43.06 68s	25.88	46 2s	**26** 94s
drone 4-1	P_{max}	0.79 1044s	0.86 64s	0.97	**0.87** 254s	0.81 160s

Time differences are not important here as we are trying to find small FSCs, which represent the solution more efficiently compared to belief-based approach!

Thank You for Your Attention!

- novel learning framework to obtain finite-state controllers (**FSCs**) for **POMDPs**
- enhancing the inductive synthesis loop by extracting information from belief-based methods
- future research: **online analysis**, **discounting**





See: *Andriushchenko et al. Inductive Synthesis of Finite-State Controllers for POMDPs*