Aleksandr Shevchenko

USING INHERITANCE DEPENDENCIES TO ACCELERATE ABSTRACTION-BASED SYNTHESIS OF FINITE-STATE CONTROLLERS FOR POMDPS

Supervisor: doc. RNDr. **Milan Češka**, Ph.D. Consultant: Ing. **Roman Andriushchenko**

Partially Observable Markov Decision Process (POMDP)

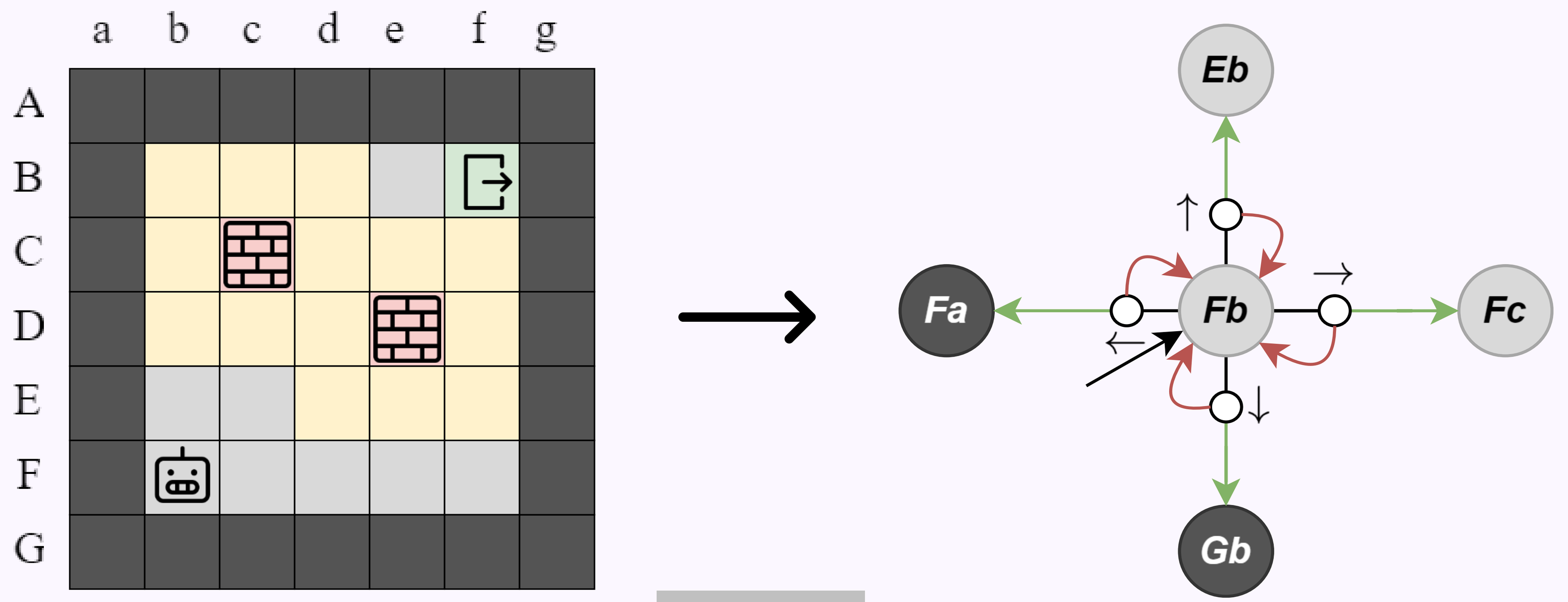

Figure 1

- A model for decision making and autonomous planning of agent navigation
- The goal is to find a strategy for given specification
- **Observations** correspond to the properties of the world, which can be detected by agent's sensor
- The agent lacks the knowledge of its current state and has information solely about the observation
- Possible specification: the robot must take as few steps as possible to the goal without hitting the wall

Finite-State Controller (FSC)

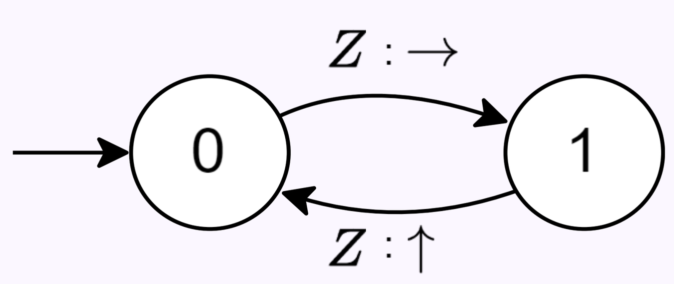

Figure 2

- A possible representation of POMDP's strategy (**policy**, scheduler)
- Mealy machine, determining the choice of actions based on a state of internal memory
- Generally, FSC with bigger number of nodes can represent more flexible strategies and yield better results
- A **family of FSCs** is a set of candidate FSCs for the given size of the memory
- The robot under this FSC ignores walls and alternates actions ↑ and →

INDUCTIVE SYNTHESIS of FSCs

- Analyses finite families of FSCs for specified memory size
- Gradually increases the design space by adding memory nodes
- Finding the optimal FSC is unsolvable

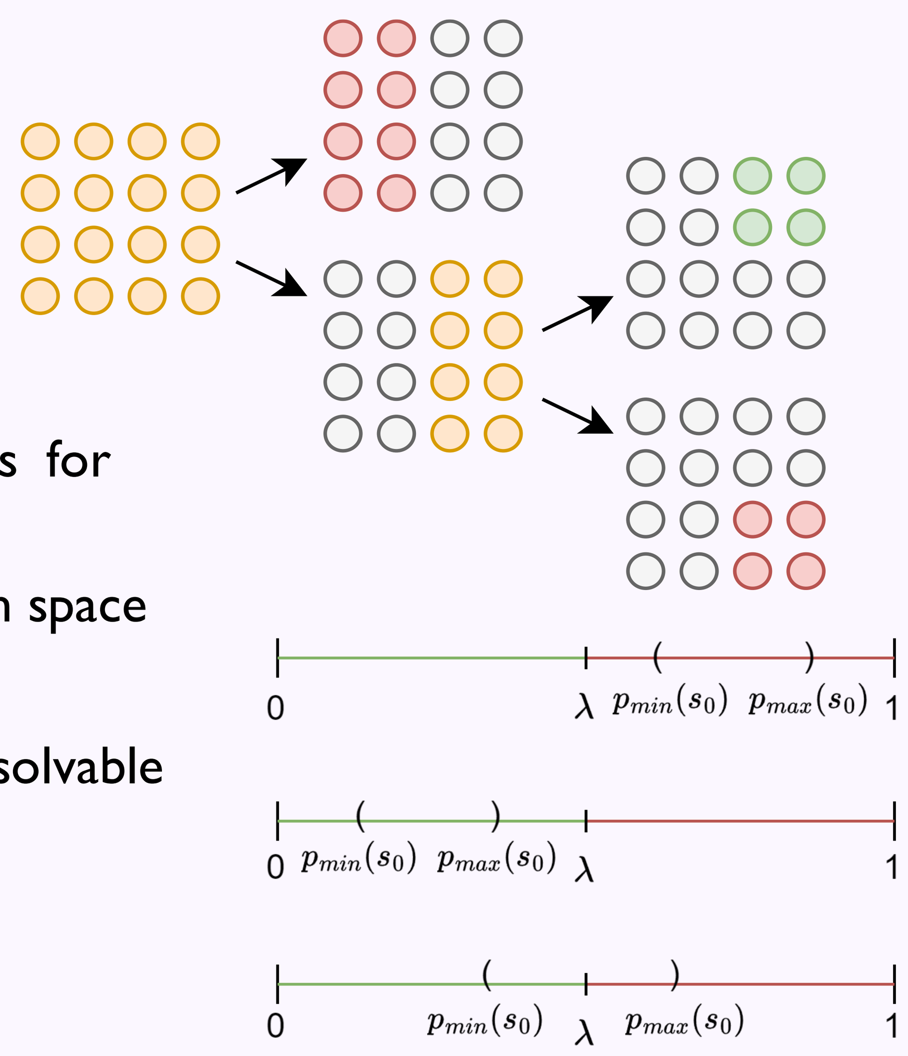

Figure 3

ABSTRACTION-REFINEMENT

- Operates with the abstraction of the family – **quotient MDP**
- **Splits** the family (**parent**) into subfamilies (**children**) and analyses them separately
- The number of splittings depends on bounds provided by model checking of quotient MDP. If the bounds are within the desired interval, the subfamily is accepted. If they are fully outside the interval, the subfamily is rejected. Otherwise we continue with splitting

QUOTIENT MDP

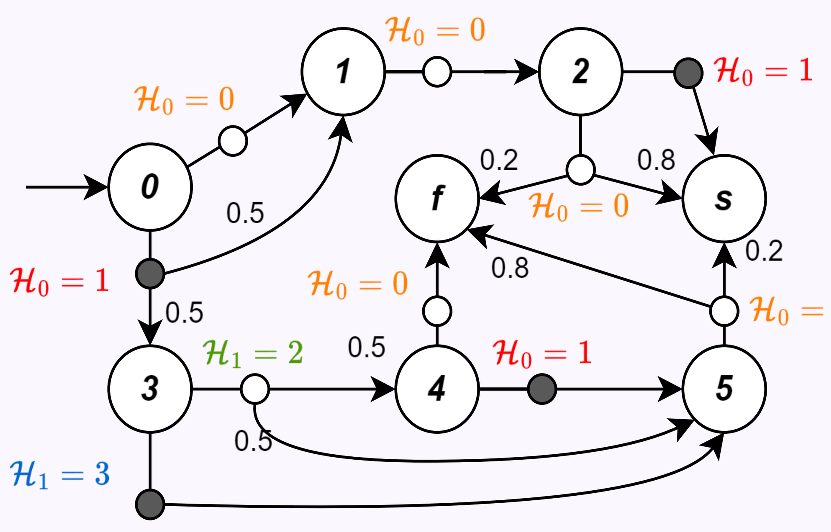

Figure 4

- Actions in the quotient MDP preserve the behaviour of **all individual realisations** (FSCs)
- Allows to switch between realisations and simulate the behaviour of an FSC, which is not originally presented in the family

PROPOSED IMPROVEMENT

- The main idea of **Inheritance Dependencies AR** is to minimise redundant model checkings
- Model-checking results of the parent can be preserved for children
- Marking a state as **vague** if it may have lost the optimal choice
- Predecessors of vague states are also vague

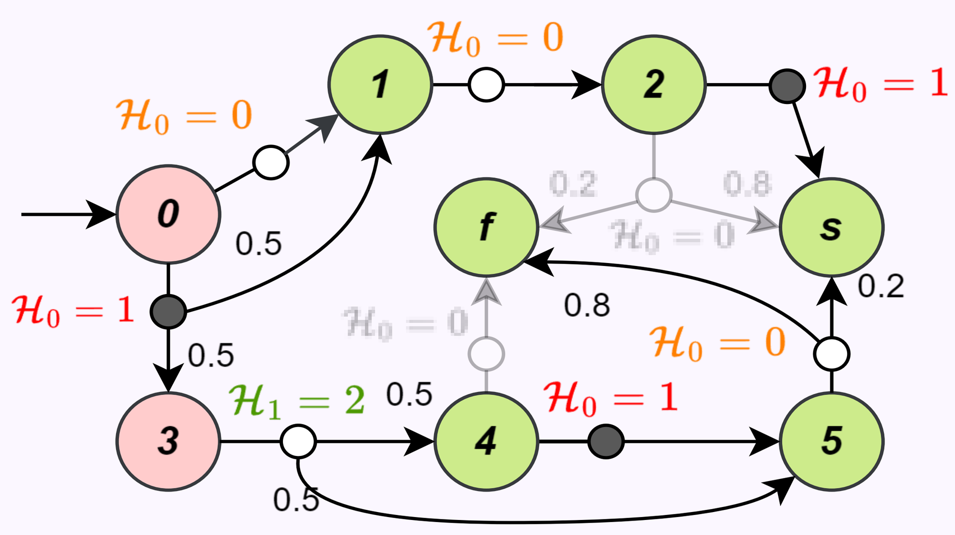

Figure 5

- Creating a **mask** of choices based on received classification of states and optimal choices of the parent
- **Extended IDAR** searches for **affected** states, which for certain lost their optimal choice compared to parent's scheduler
- There are less affected states than vague ones, which results in the smaller mask and in more accelerated model checking

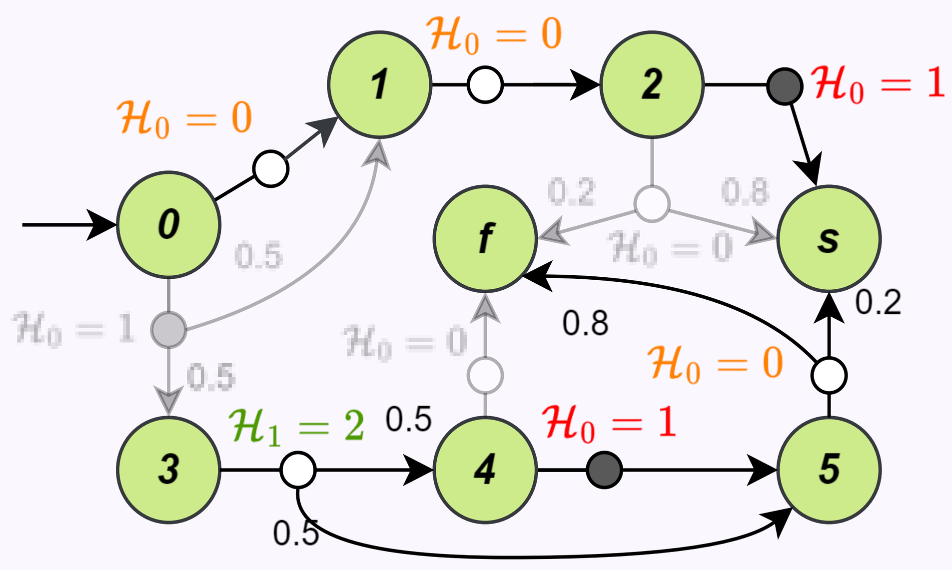

Figure 6

- **Smart EIDAR** combines AR and EIDAR for greater efficiency
- Switches between these methods based on parameters obtained after running the first few iterations

RESULTS

1) How much was the analysis of families accelerated?
2) What impact did our methods have on the synthesis?

Model / memory size / num. of iterations / overall AR time	Method	Overall speedup*	MB speedup*	MC speedup*
rocks12 / 1 / 1.000 14.67 s.	IDAR	1.08	1.61	1.73
	EIDAR	1.59	3.32	3.73
	SEIDAR	1.5	3.15	3.57
rocks12 / 1 / 10.000 139.64 s.	IDAR	1.16	1.76	1.82
	EIDAR	1.7	3.6	4.09
	SEIDAR	1.59	3.41	3.92
rocks12 / 3 / 800 175.02s.	IDAR	1.52	1.63	1.81
	EIDAR	1.56	1.94	2.3
	SEIDAR	1.54	1.9	2.23
refuel20 / 1 / 1.000 22.04 s.	IDAR	0.78	1.01	1
	EIDAR	0.75	0.75	0.63
	SEIDAR	1.32	1.06	0.95
refuel20 / 1 / threshold 0.000829 18.54 s.	IDAR	0.82	1.09	1.05
	EIDAR	6.82	6.99	3.47
	SEIDAR	6.68	6.89	3.47

Table 1

- Proposed method improves the scalability of the inductive synthesis

*Speedup is provided in comparison with classic AR

BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2024