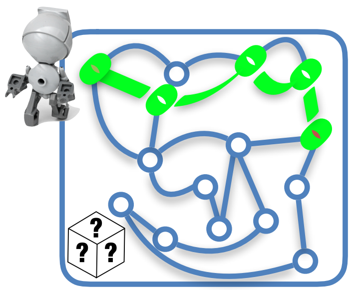
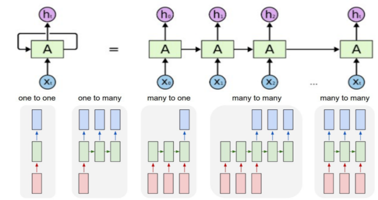
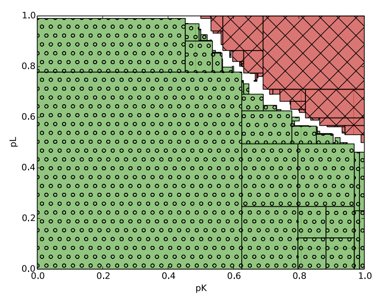
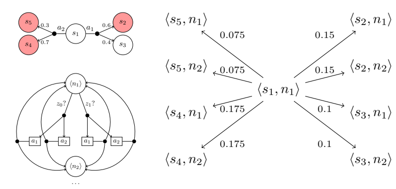
Planning under Partial Observability

A Betrothal of Formal Verification and Machine Learning









Nils Jansen

LearnAut, June 23, 2019

me

me

RWTH AACHEN UNIVERSITY

TEXAS
The University of Texas at Austin

Radboud University

joint work with:

Sebastian Junges, Joost-Pieter Katoen, Tim Quatmann, Bernd Becker, Ralf Wimmer, Leonore Winterer, Steve Carr, Ufuk Topcu, Mohamedreza Ahmadi, Roderick Bloem, Bettina Koenighofer, Alexandru Serban

RWTH AACHEN UNIVERSITY

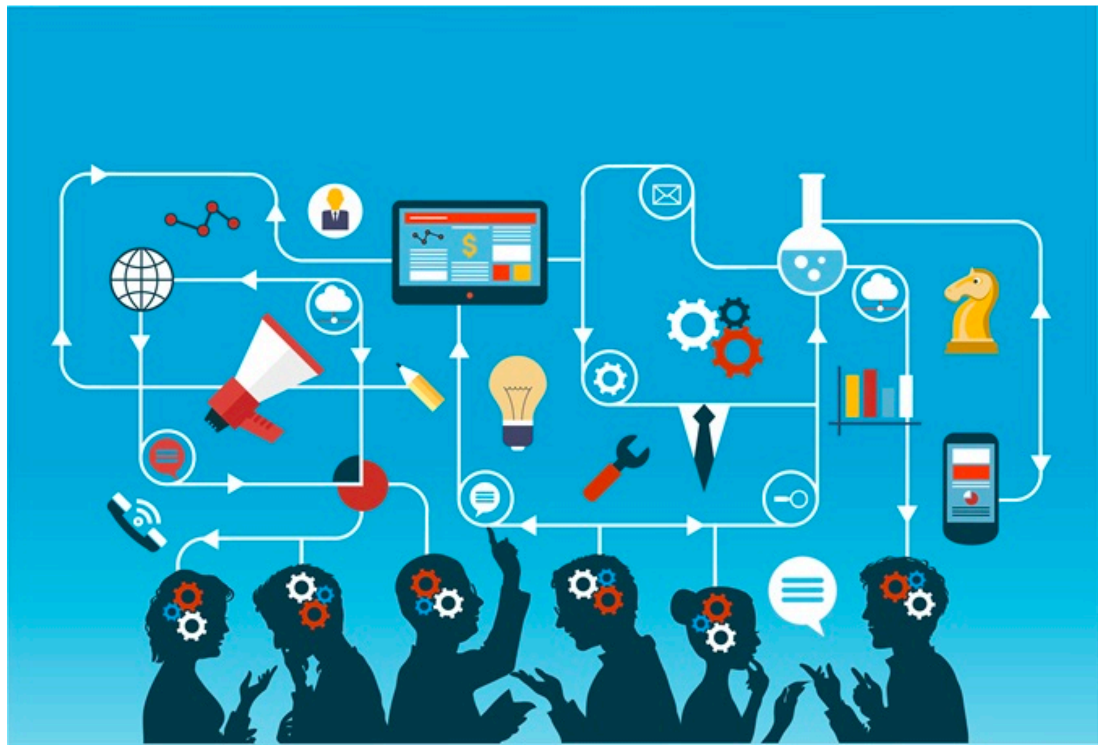
UNI FREIBURG

TEXAS The University of Texas at Austin

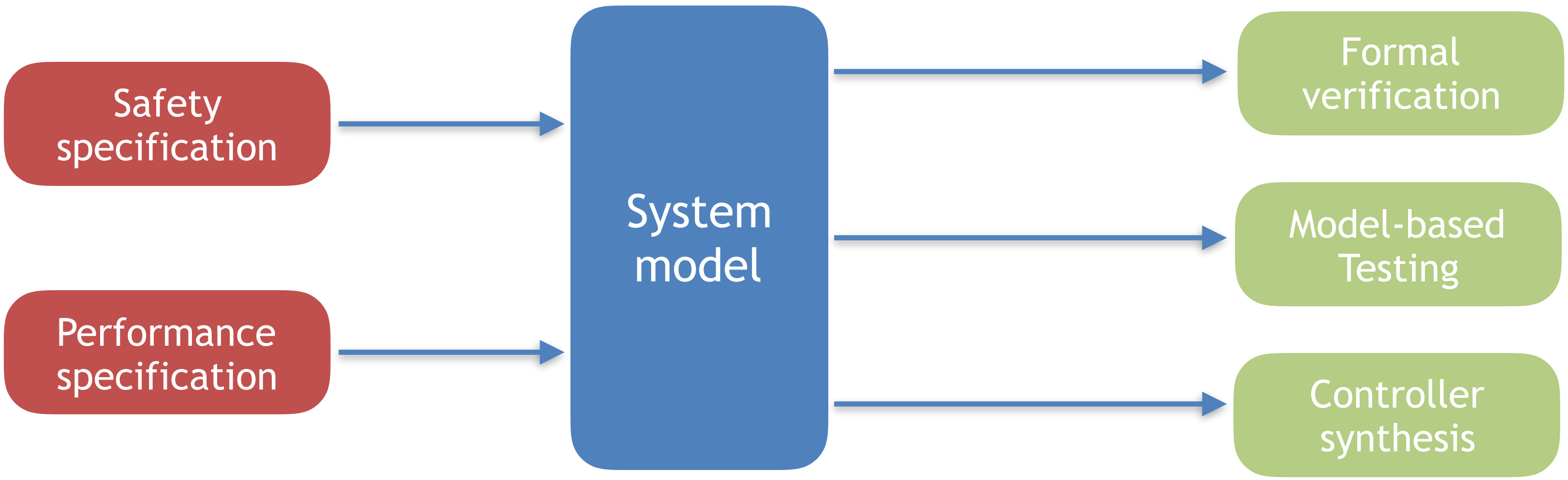
Caltech

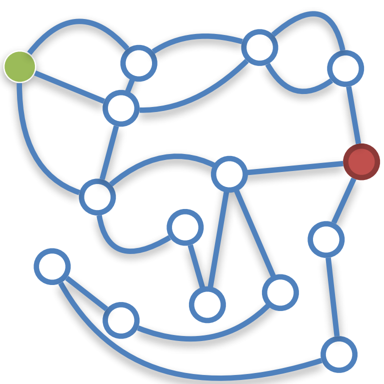
TU Graz

Radboud University

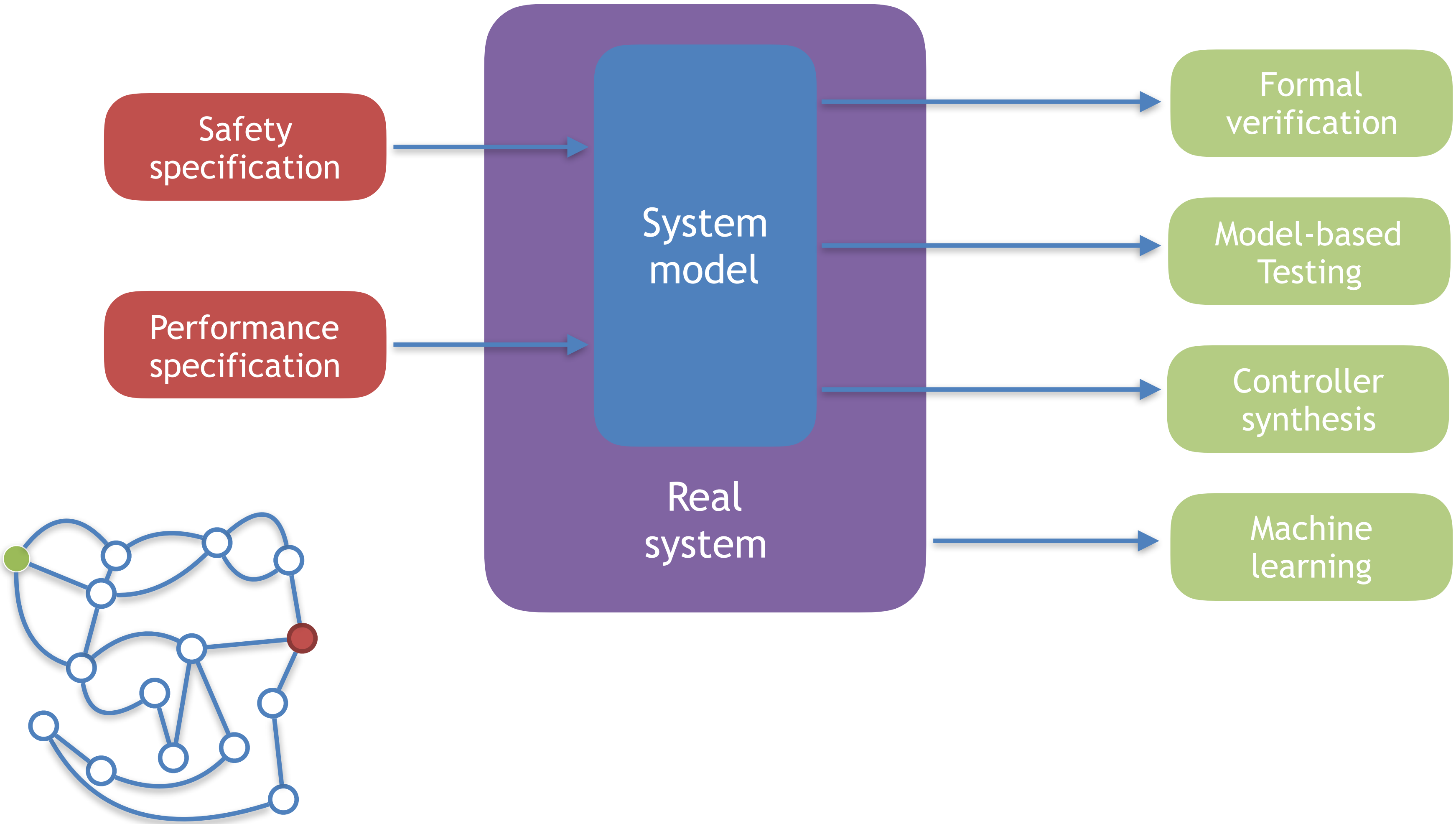


Autonomous (Cyber-physical) Systems

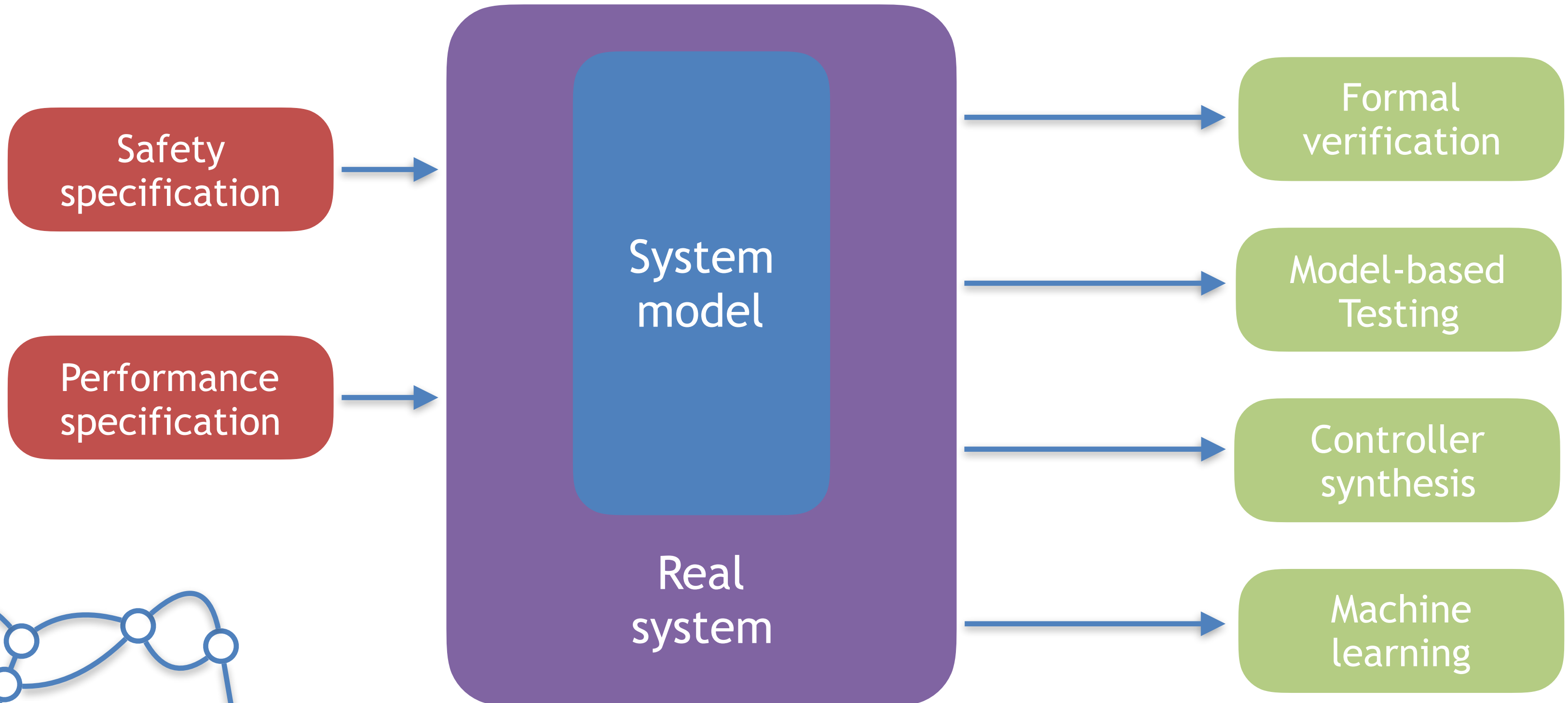


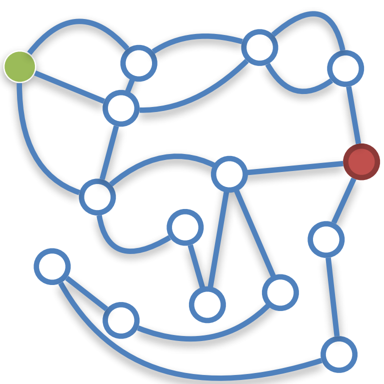


Autonomous (Cyber-physical) Systems



Autonomous (Cyber-physical) Systems





Autonomous (Cyber-physical) Systems

Safety specification

Performance specification

System model

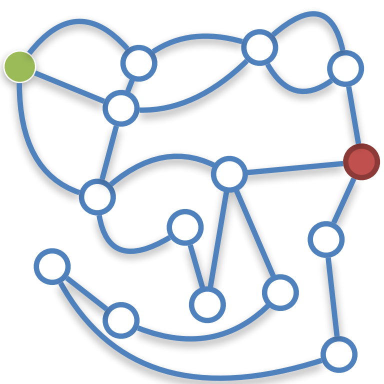
Real system

Formal verification

Model-based Testing

Controller synthesis

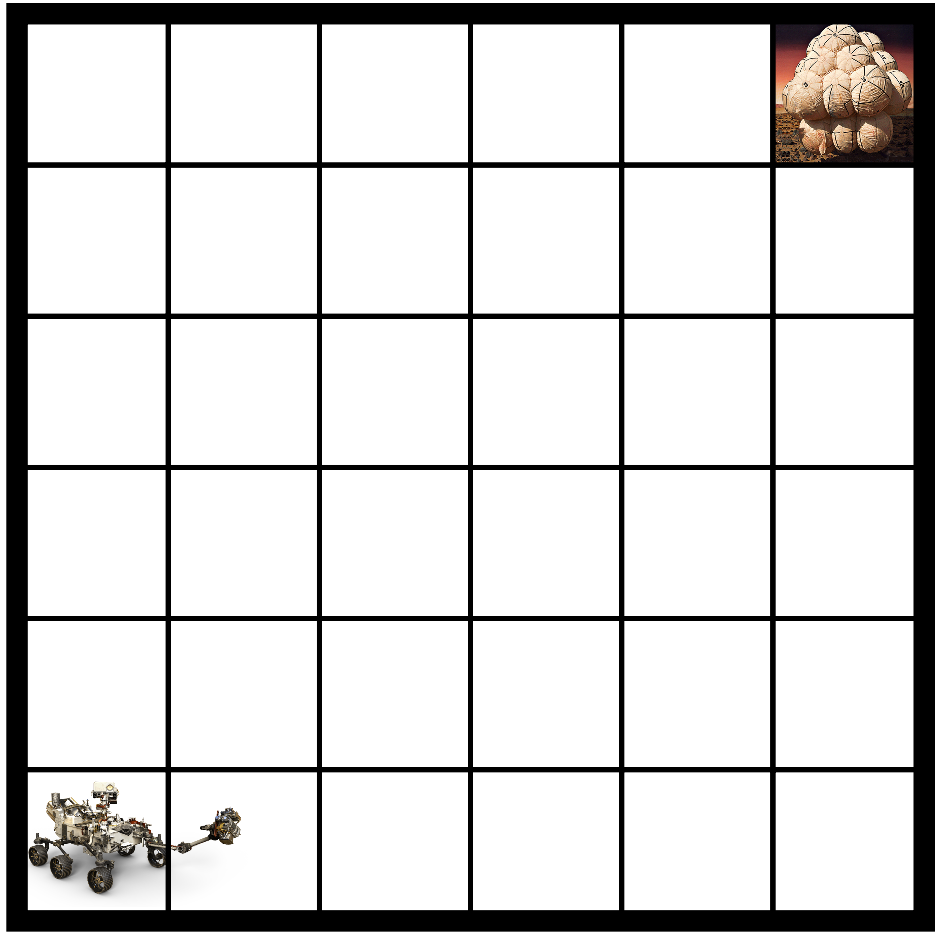
Machine learning



Solutions at the interfaces of domains

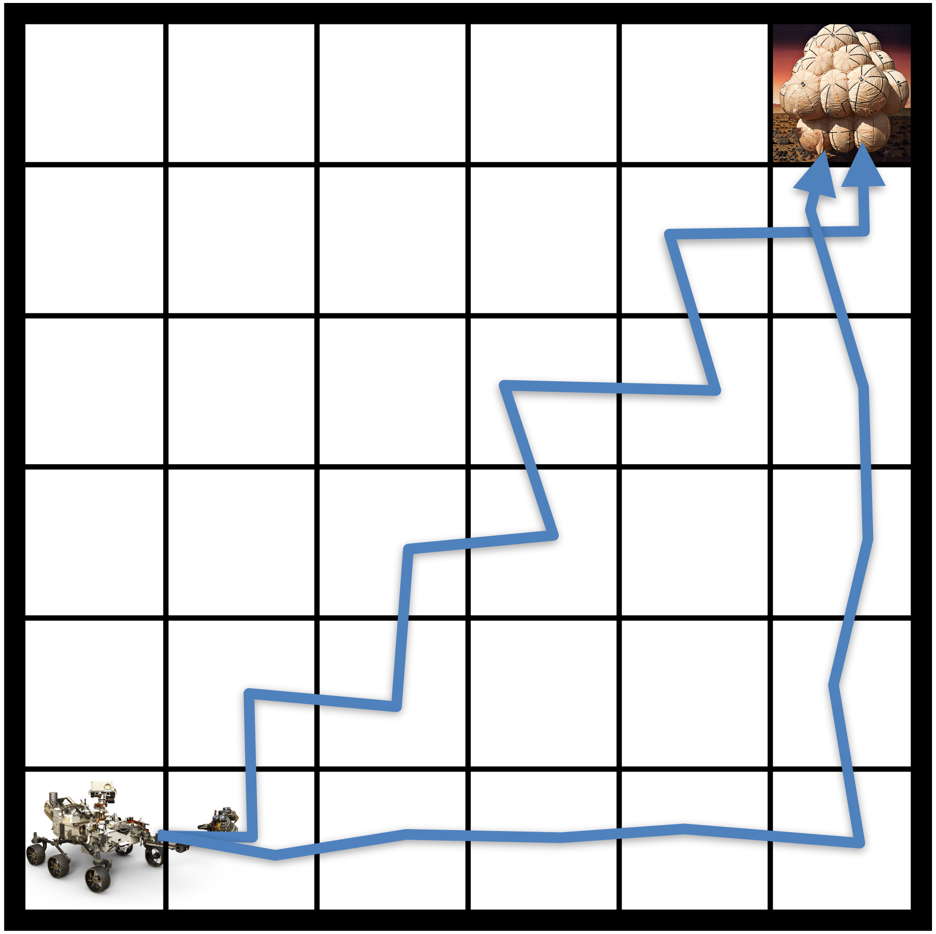
Help the Robot

Find the best way to the airbag



Help the Robot

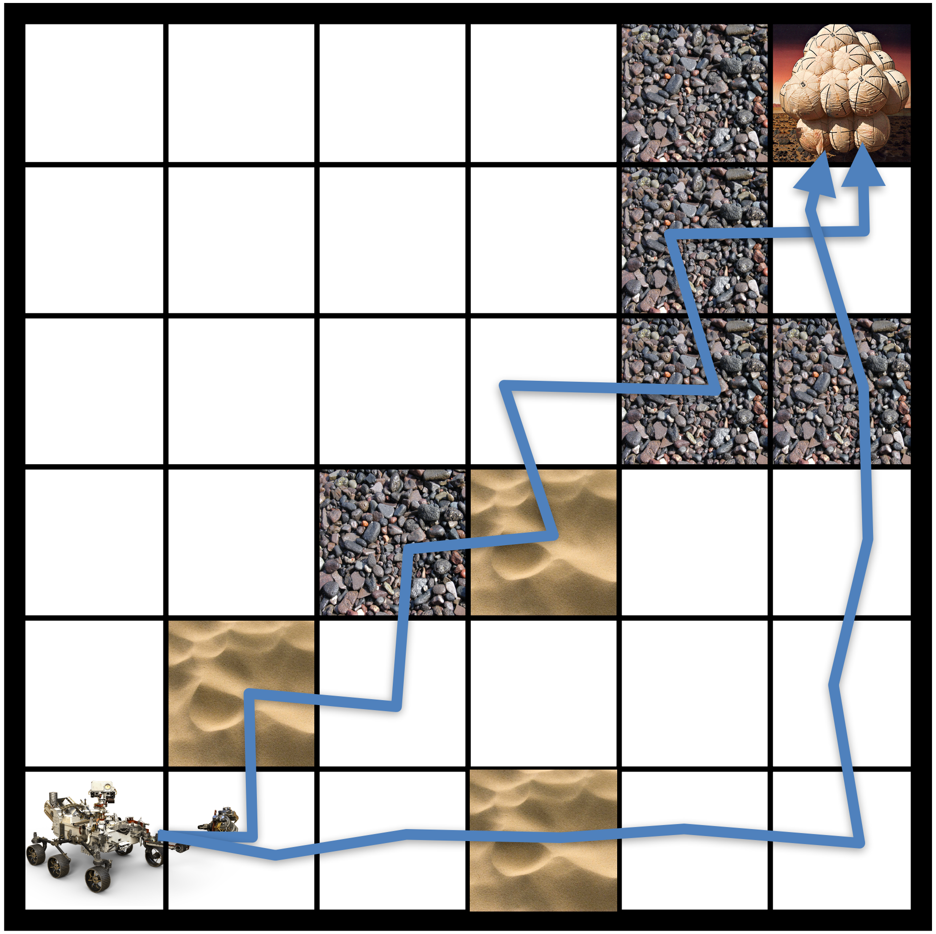
Find the best way to the airbag



Help the Robot

Find the best way to the airbag

Take expensive surfaces into account

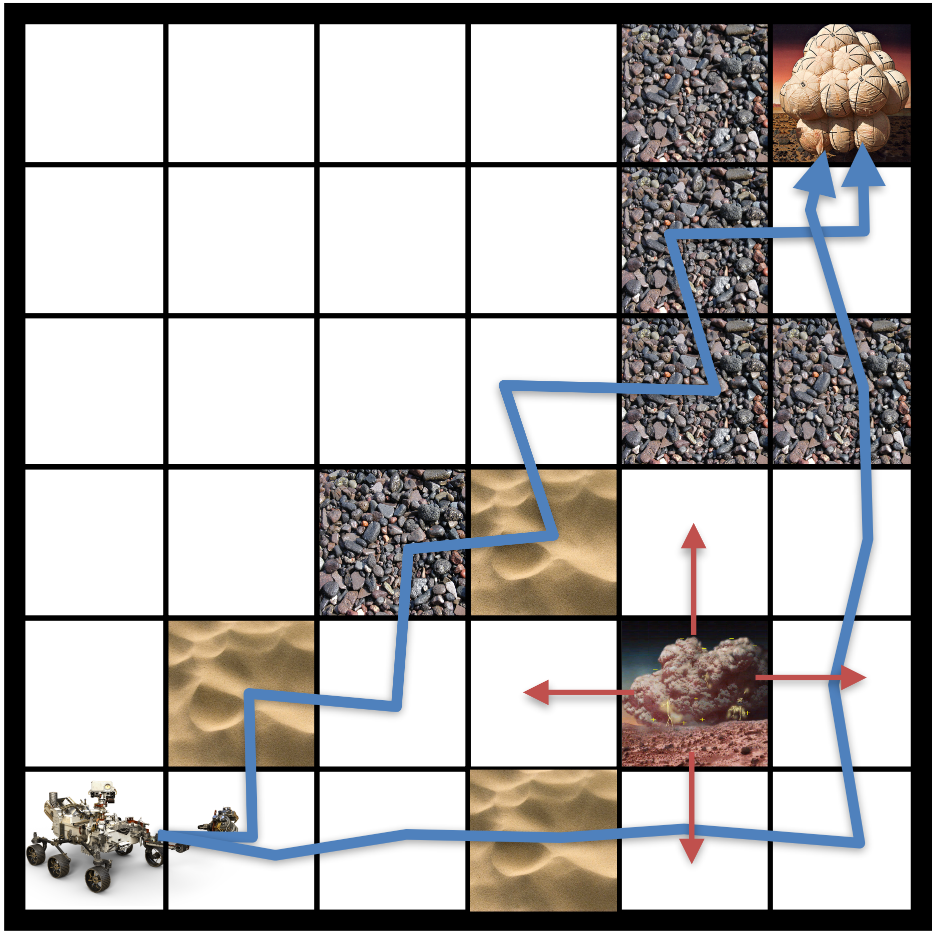


Help the Robot

Find the best way to the airbag

Take expensive surfaces into account

Avoid randomly moving dust storm

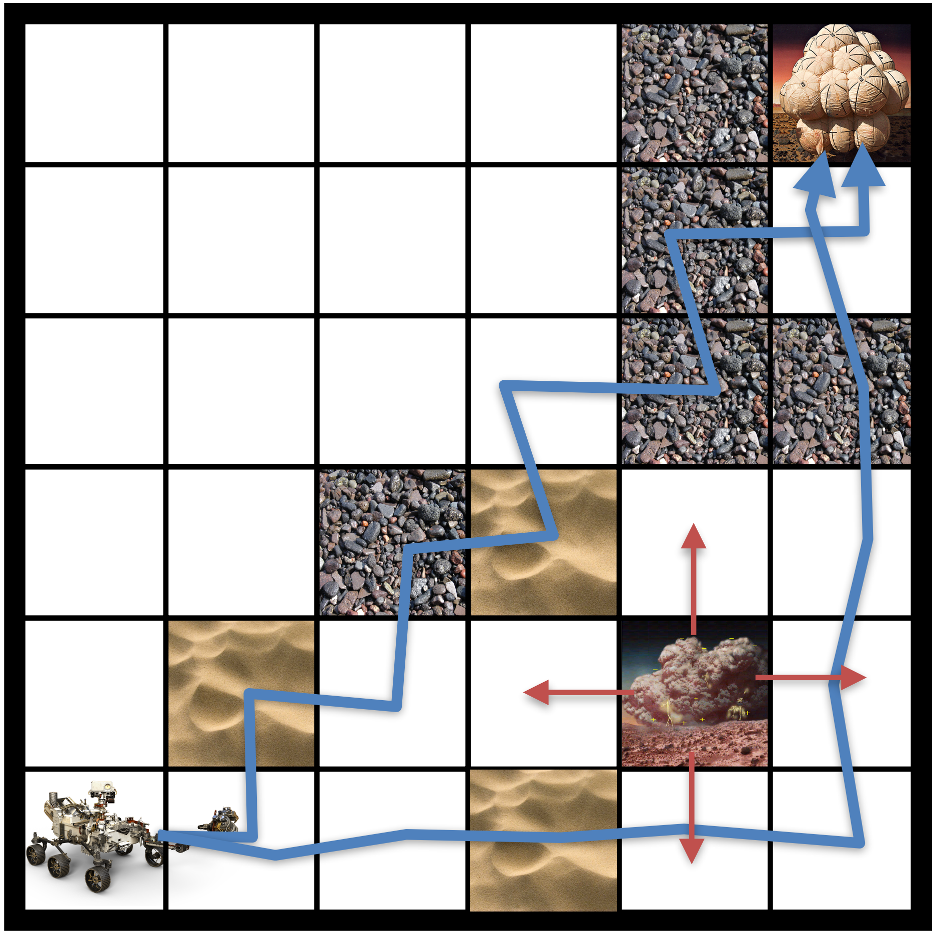


Help the Robot

Find the best way to the airbag

Take expensive surfaces into account

Avoid randomly moving dust storm



Find **safe and/ or cost-optimal** policy to get to the airbag

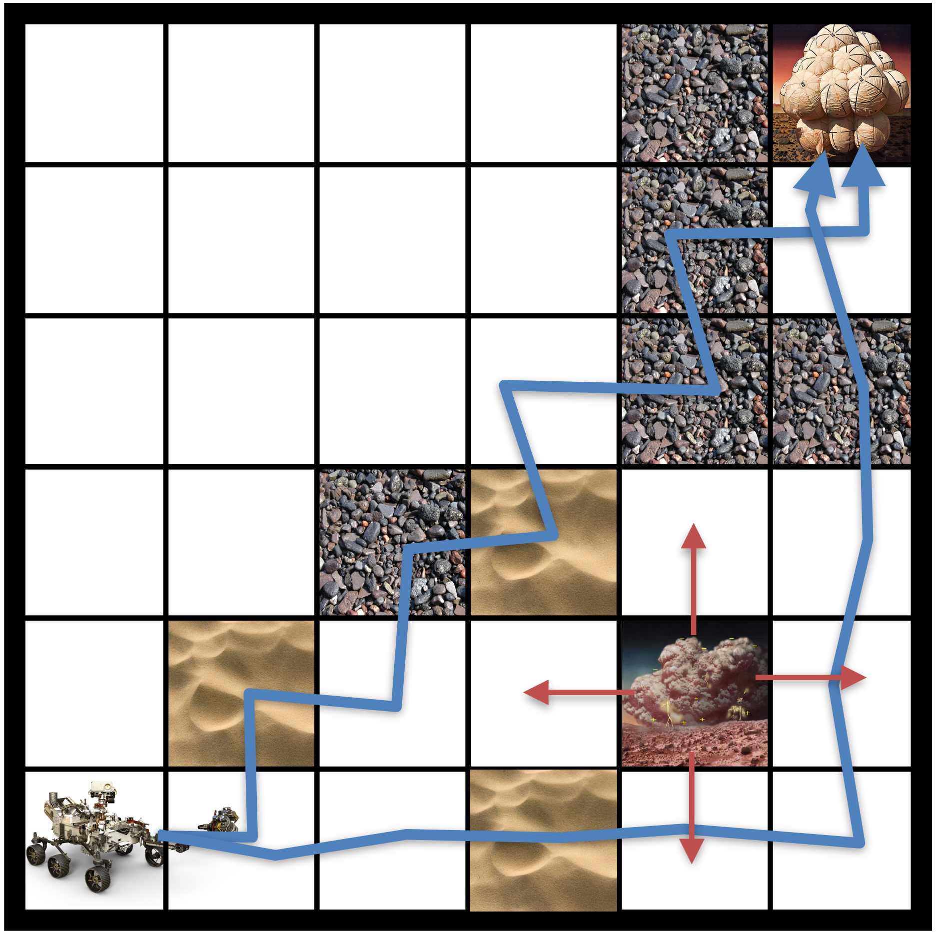
Help the Robot

multi-objective model checking for MDPs

Find the best way to the airbag

Take expensive surfaces into account

Avoid randomly moving dust storm



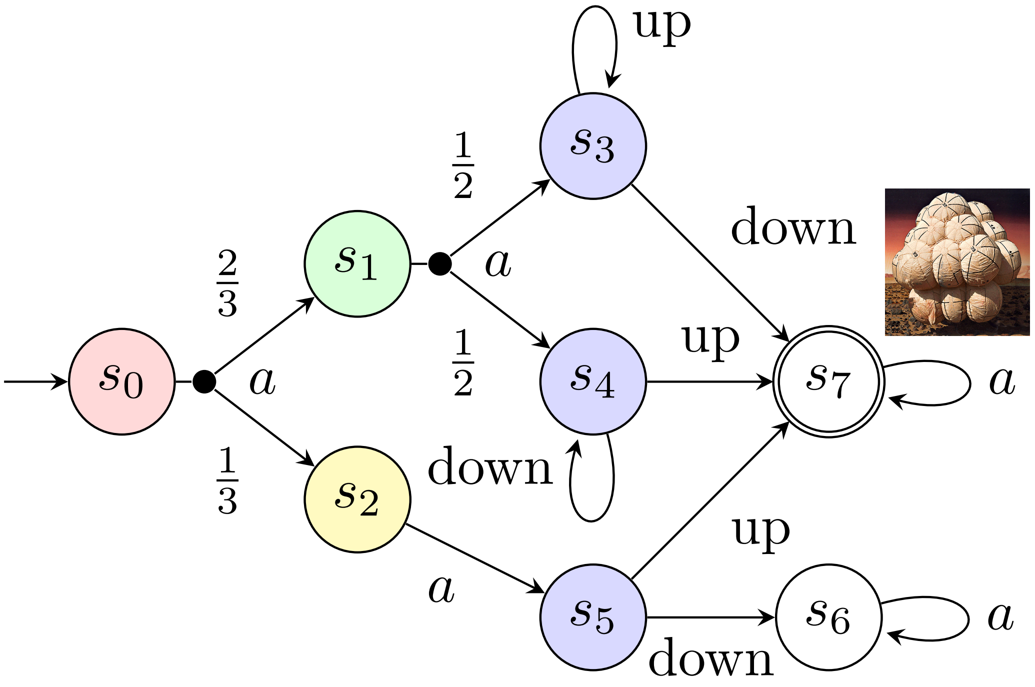
Find **safe and/ or cost-optimal** policy to get to the airbag

Underlying Model: Markov Decision Process

MDPs

$$Pr_{max}(\Diamond s_7)$$

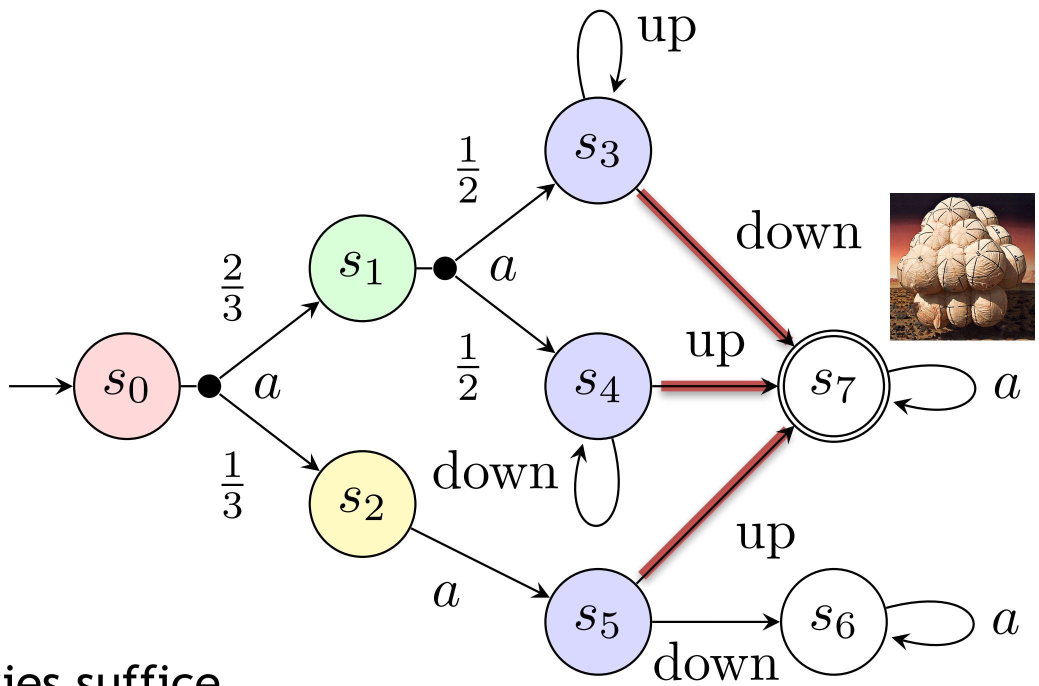
$$EC_{min}(\Diamond s_7)$$



MDPs

$$Pr_{max}(\Diamond s_7)$$

$$EC_{min}(\Diamond s_7)$$

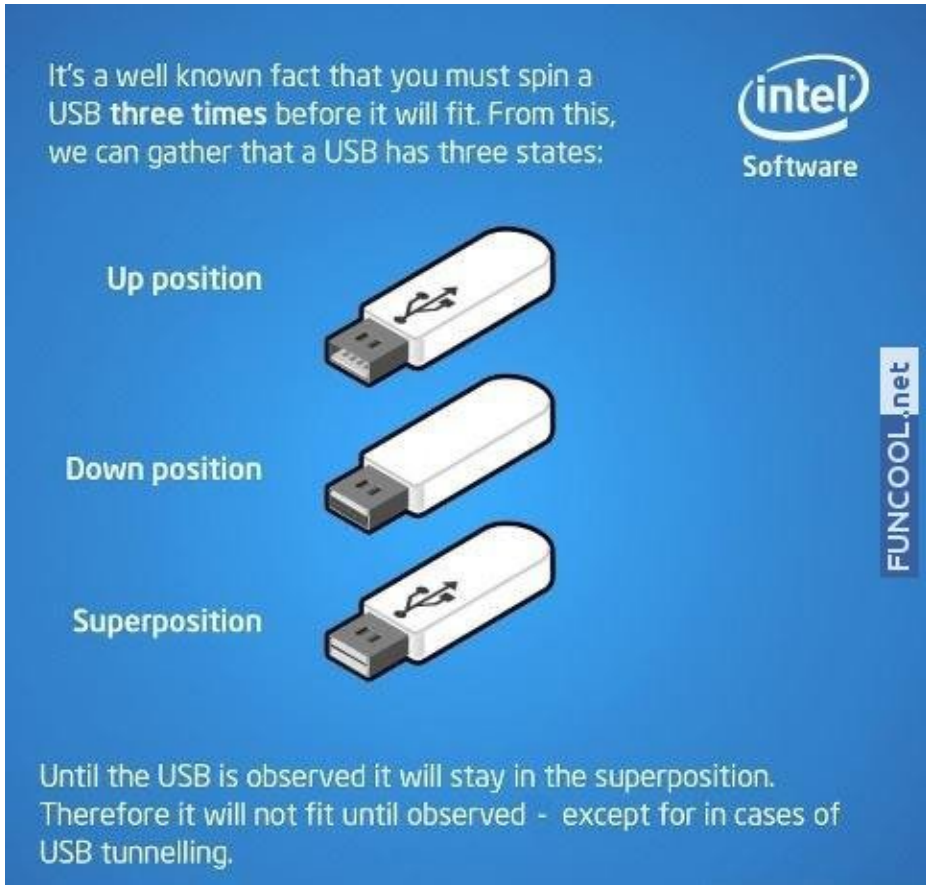


- efficient model checking

  memoryless deterministic strategies suffice for single objectives

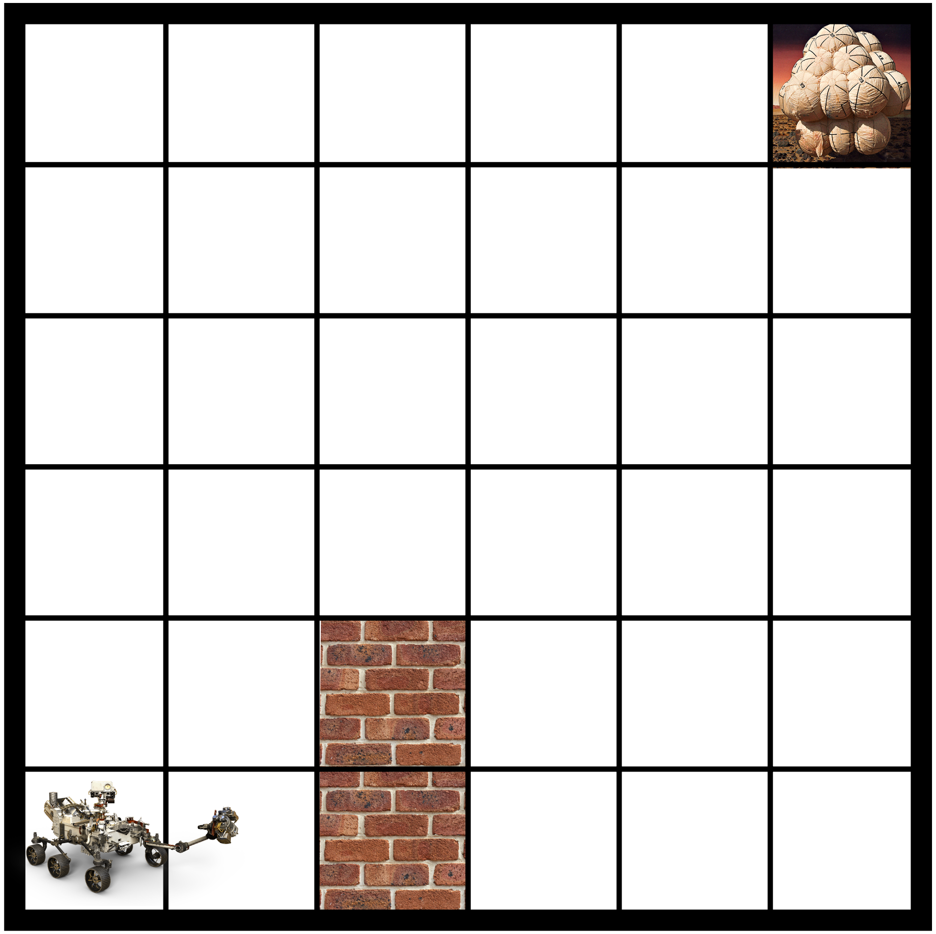
  randomized strategies may be needed for multiple objectives

Partial Observability



Help the Robot with Partial Observability

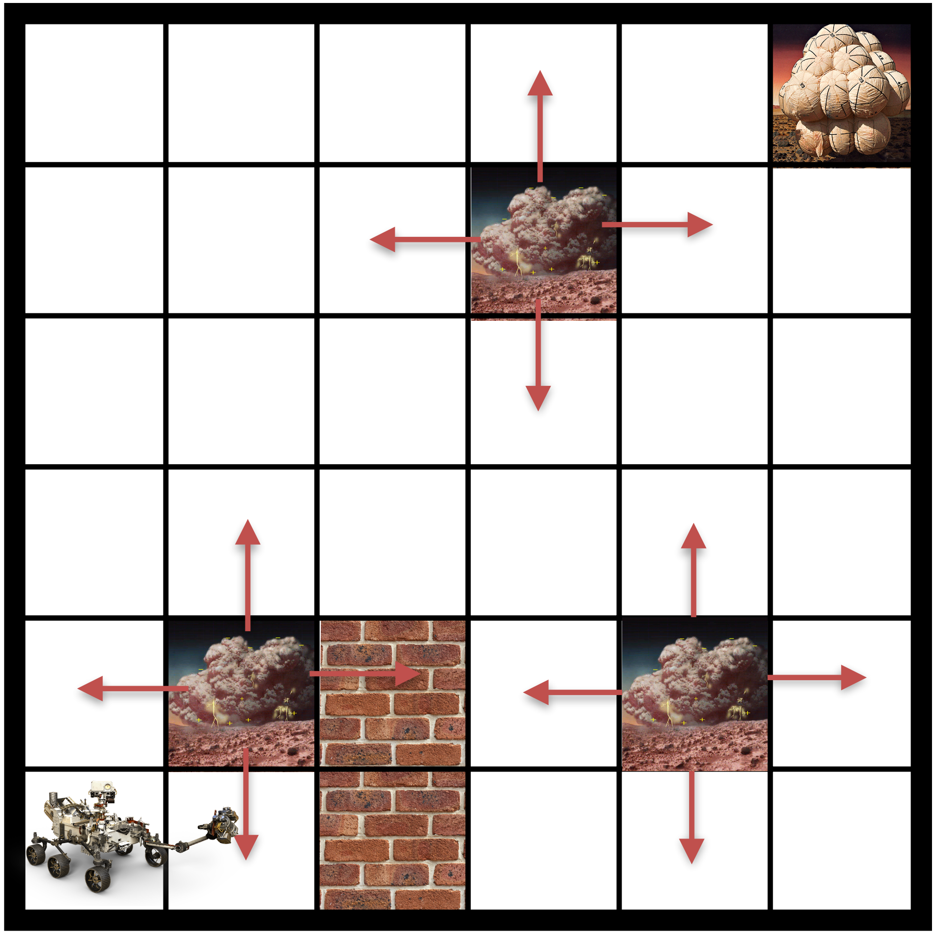
Robot has restricted range of vision



Help the Robot with Partial Observability

Robot has restricted range of vision

Storm is only **observable** when near

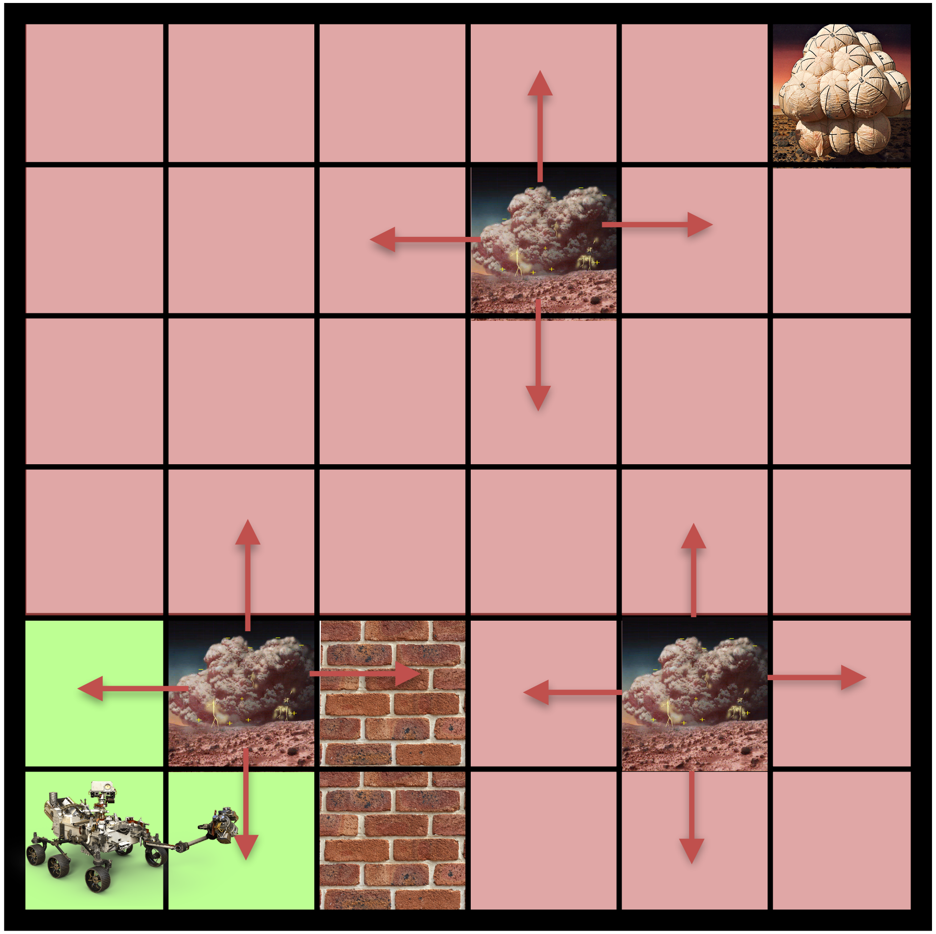


Help the Robot with Partial Observability

Robot has restricted range of vision

Storm is only **observable** when near

For robot, storm is either **near** or **far**

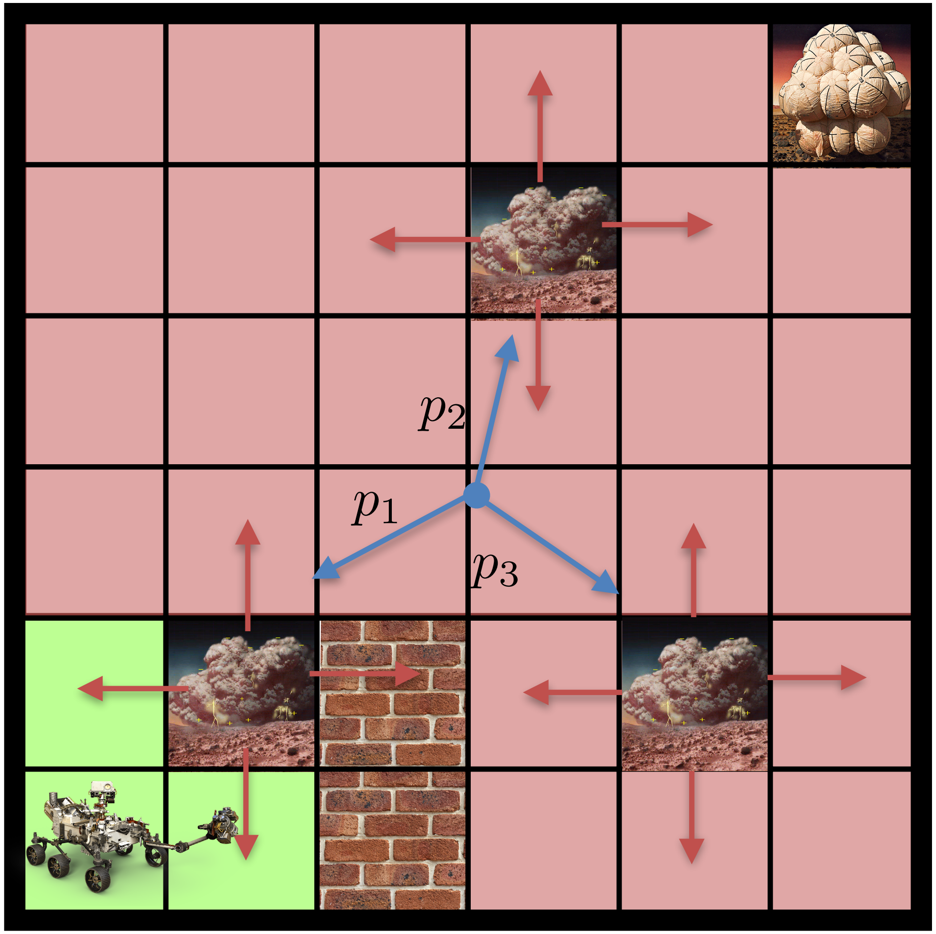


Help the Robot with Partial Observability

Robot has restricted range of vision

Storm is only **observable** when near

For robot, storm is either **near** or **far**



Belief state: Likelihood of the actual position of the storm

infinite belief MDP

Help the Robot with Partial Observability

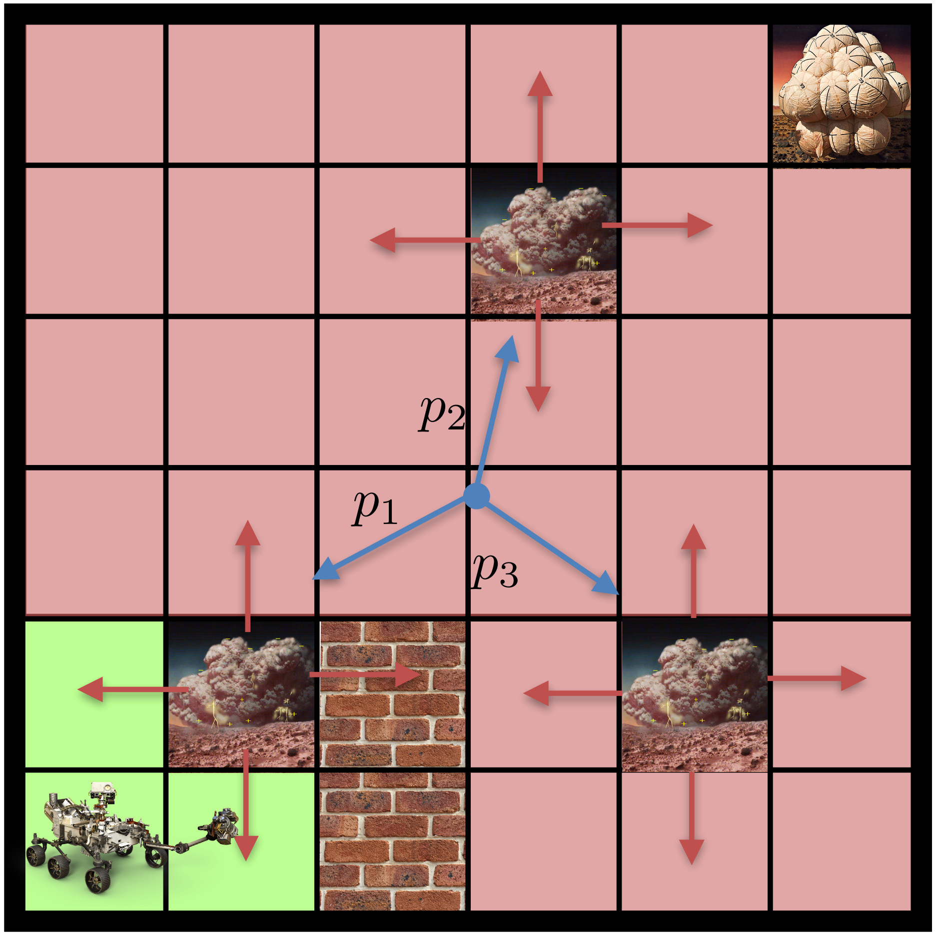
Robot has restricted range of vision

Storm is only **observable** when near

For robot, storm is either **near** or **far**

Find strategy that induces

$$Pr_{max}(\neg B\, U\, G)$$



Find **safe and/or cost-optimal** policy to get to the airbag

Belief state: Likelihood of the actual position of the storm

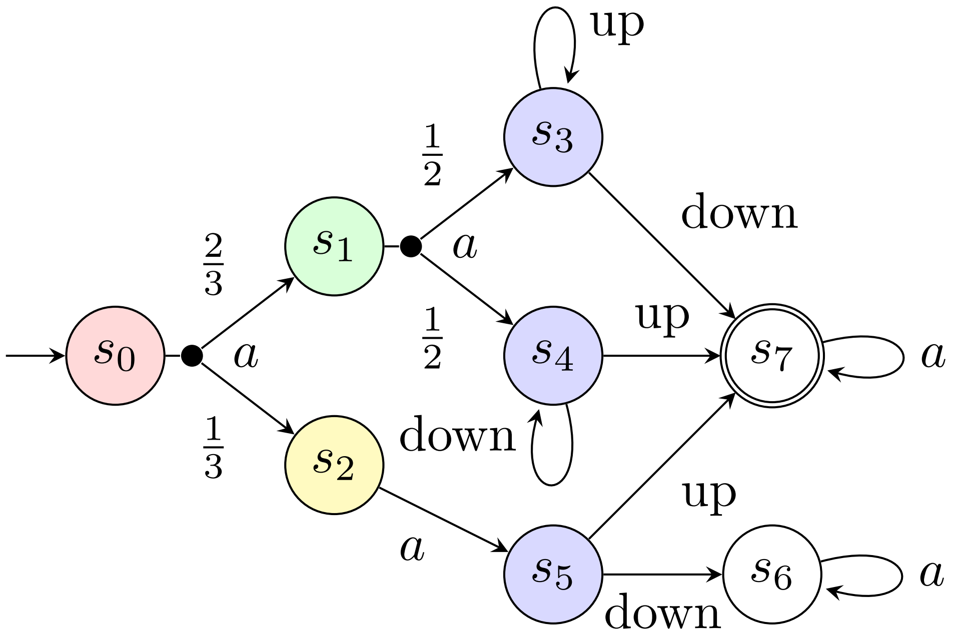
infinite belief MDP

POMDPs

$$Pr_{max}(\Diamond s_7)$$

$$EC_{min}(\Diamond s_7)$$

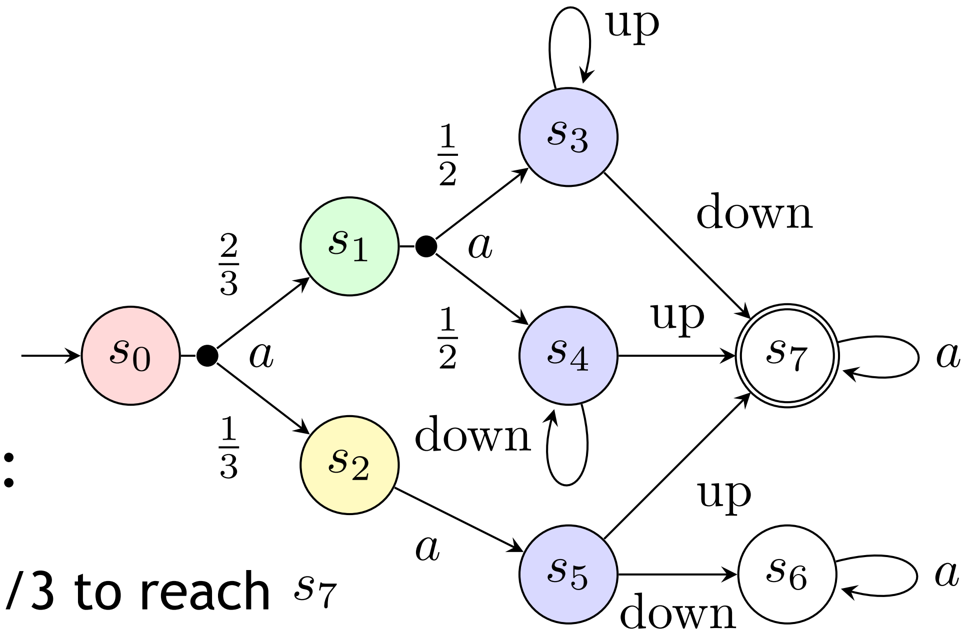
Choices at observation ‘blue’:



POMDPs

$$Pr_{max}(\Diamond s_7)$$

$$EC_{min}(\Diamond s_7)$$



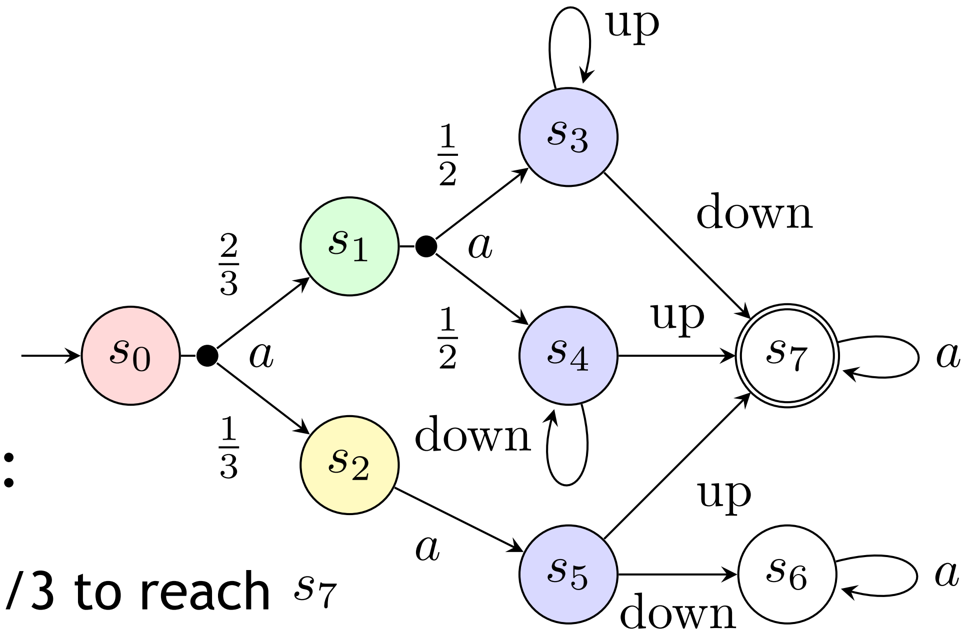
Choices at observation ‘blue’:

- Choose ‘up’ at each state: prob 2/3 to reach s_7
  memoryless deterministic

POMDPs

$$Pr_{max}(\Diamond s_7)$$

$$EC_{min}(\Diamond s_7)$$



Choices at observation 'blue':

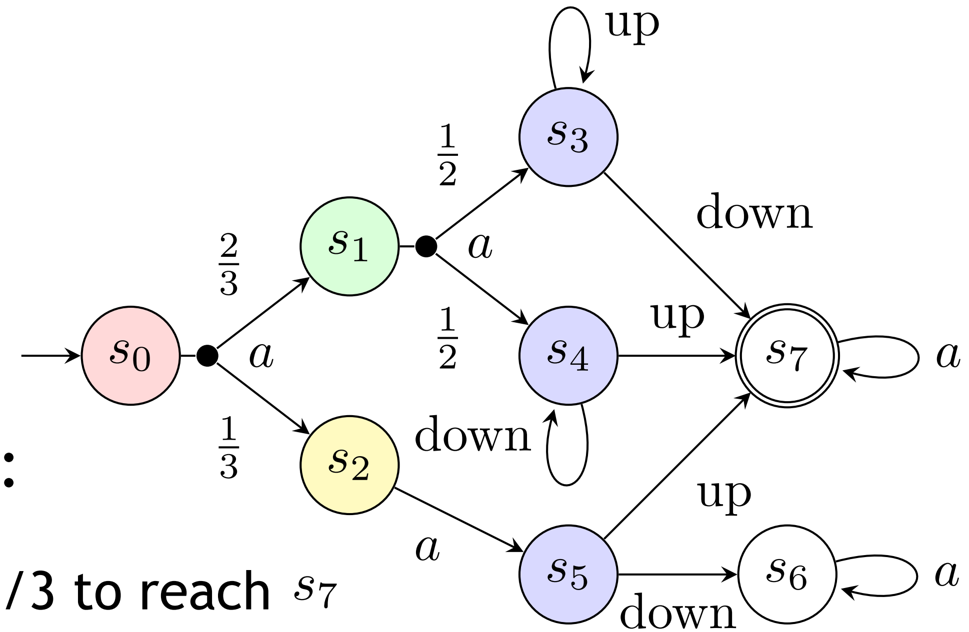
- Choose 'up' at each state: prob 2/3 to reach s_7
  memoryless deterministic

- Choose 'up' with prob $0 < p < 1$ and 'down' with prob $1 - p$: $\frac{2}{3} + \frac{1}{3}p < 1$
  memoryless randomized

POMDPs

$$Pr_{max}(\Diamond s_7)$$

$$EC_{min}(\Diamond s_7)$$



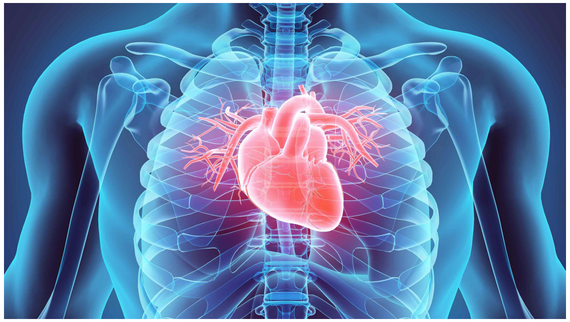
Choices at observation 'blue':

- Choose 'up' at each state: prob 2/3 to reach s_7
  memoryless deterministic
- Choose 'up' with prob $0 < p < 1$ and 'down' with prob $1-p$: $\frac{2}{3} + \frac{1}{3}p < 1$
  memoryless randomized
- Choose 'up' if predecessor is 'yellow'. Otherwise, choose 'up' if 'blue' observed even number of times, 'down' otherwise: prob 1
  deterministic with memory

POMDPs - Applications

Stock

Surveying



Health

Wireless

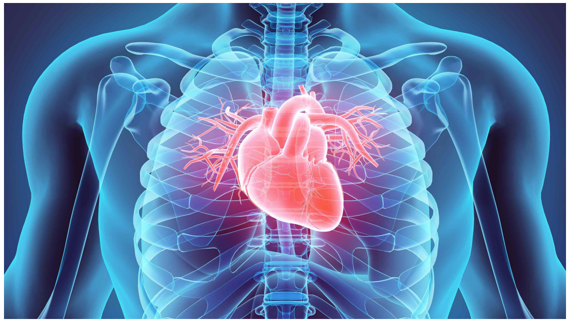
Autonomous

Machine

POMDPs - Applications

Stock

Surveying



Health

Wireless

Autonomous

Machine

And by the way: POMDPs and their subclasses form the operational semantics to probabilistic program with discrete probability distributions.

Computing Policies for POMDPs

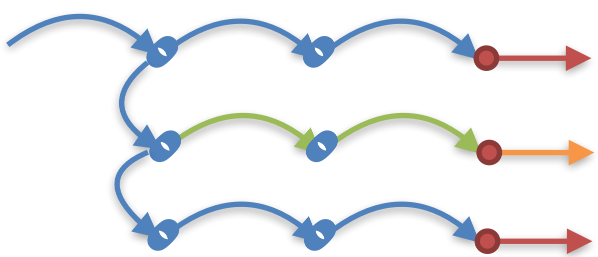
- Randomized with infinite memory: undecidable, optimal results.

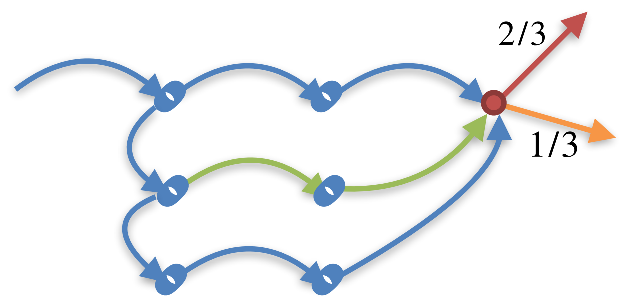
Computing Policies for POMDPs

- Randomized with infinite memory: undecidable, optimal results.
- Randomized with finite memory: NP-hard, SQRT-SUM-hard, in PSPACE, not optimal in general, but sufficient for many applications.

Computing Policies for POMDPs

- Randomized with infinite memory: undecidable, optimal results.
- Randomized with finite memory: NP-hard, SQRT-SUM-hard, in PSPACE, not optimal in general, but sufficient for many applications.
- Intuitively: Randomization can often trade off memory.





Stories - Policy Synthesis for POMDPs

1. Game-based abstraction
2. Finite-memory controllers
3. Recurrent neural networks
4. Fun: Humans in the loop

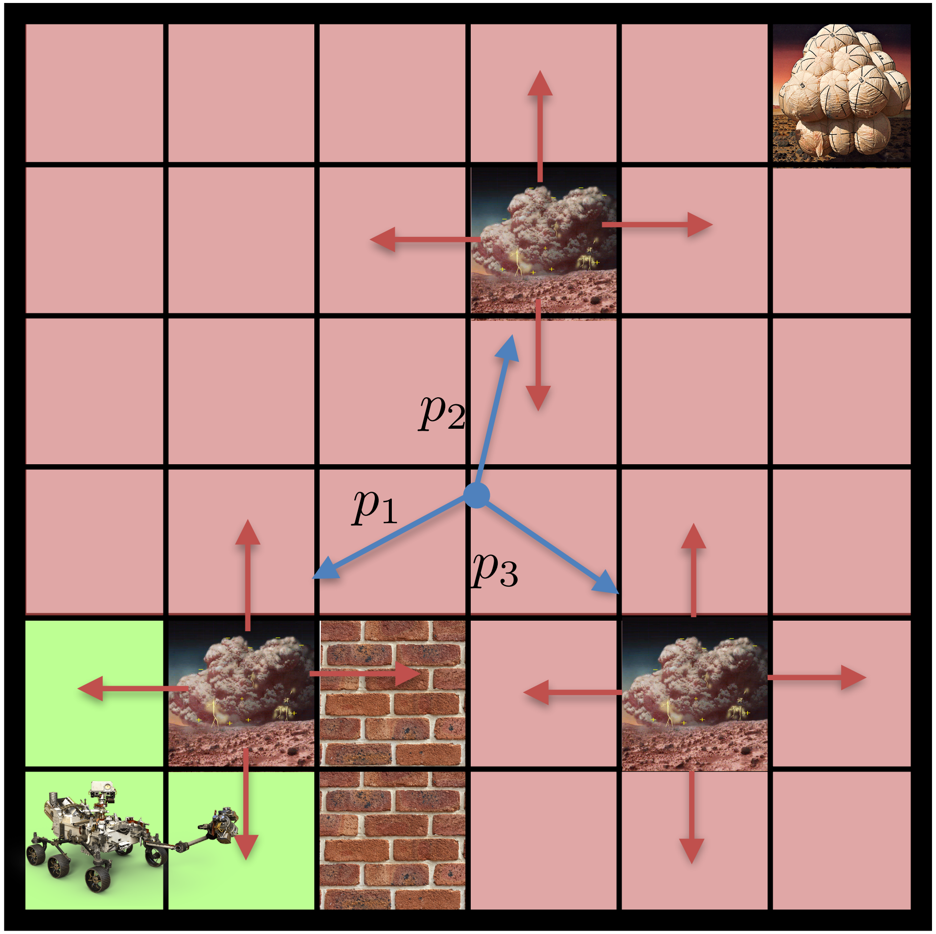
If time permits: Teaser on safe reinforcement learning.

Game-based Abstraction for POMDPs

Robot has restricted range of vision

Storm is only **observable** when near

For robot, storm is either **near** or **far**

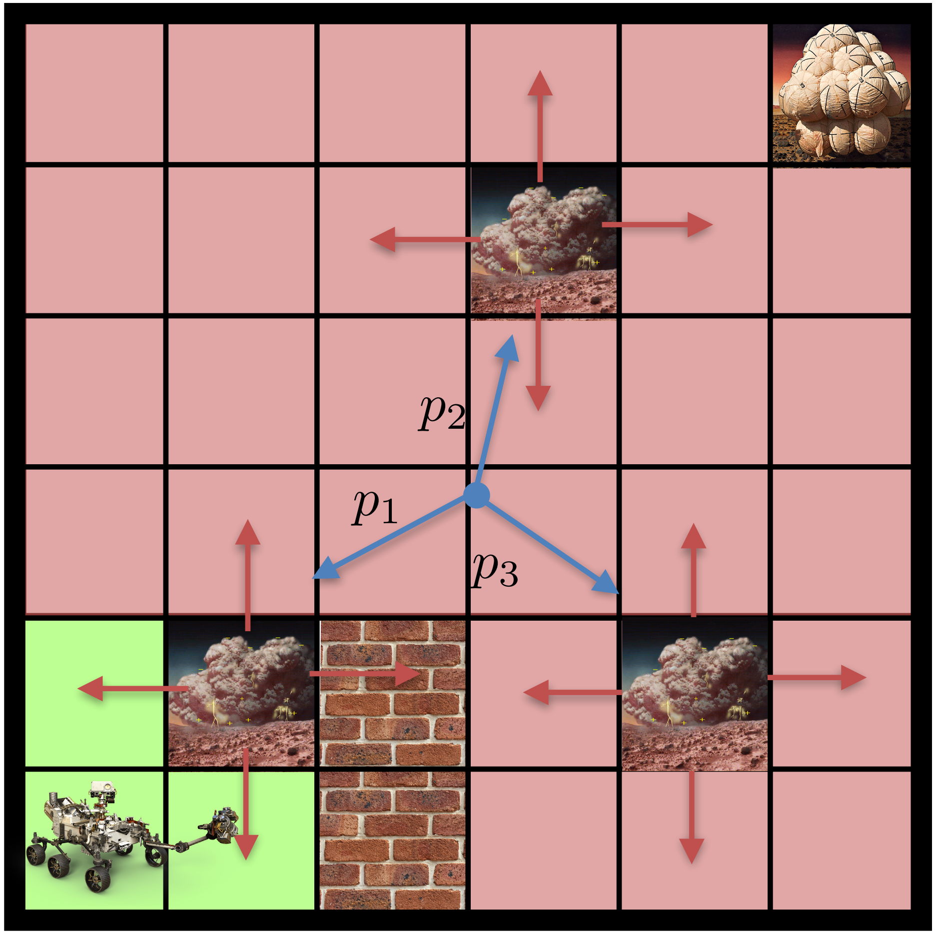


Game-based Abstraction for POMDPs

Robot has restricted range of vision

Storm is only **observable** when near

For robot, storm is either **near** or **far**



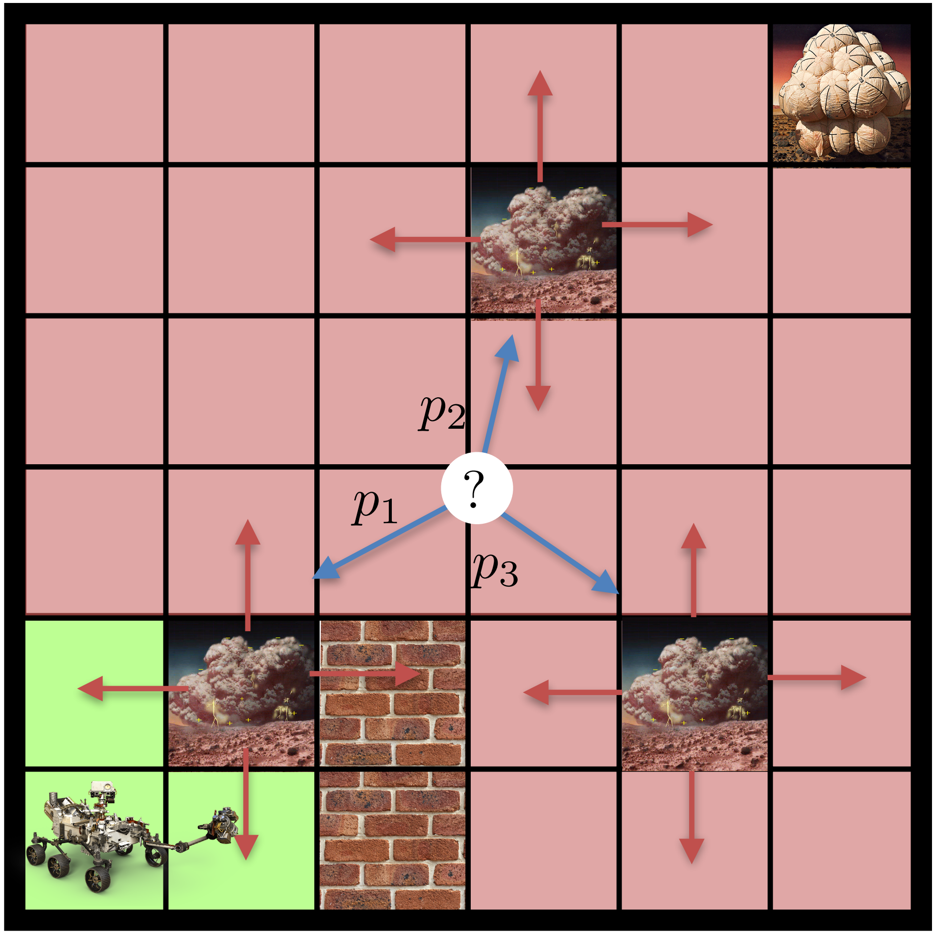
Abstract possible positions into nondeterministic choices

Game-based Abstraction for POMDPs

Robot has restricted range of vision

Storm is only **observable** when near

For robot, storm is either **near** or **far**



Abstract possible positions into nondeterministic choices

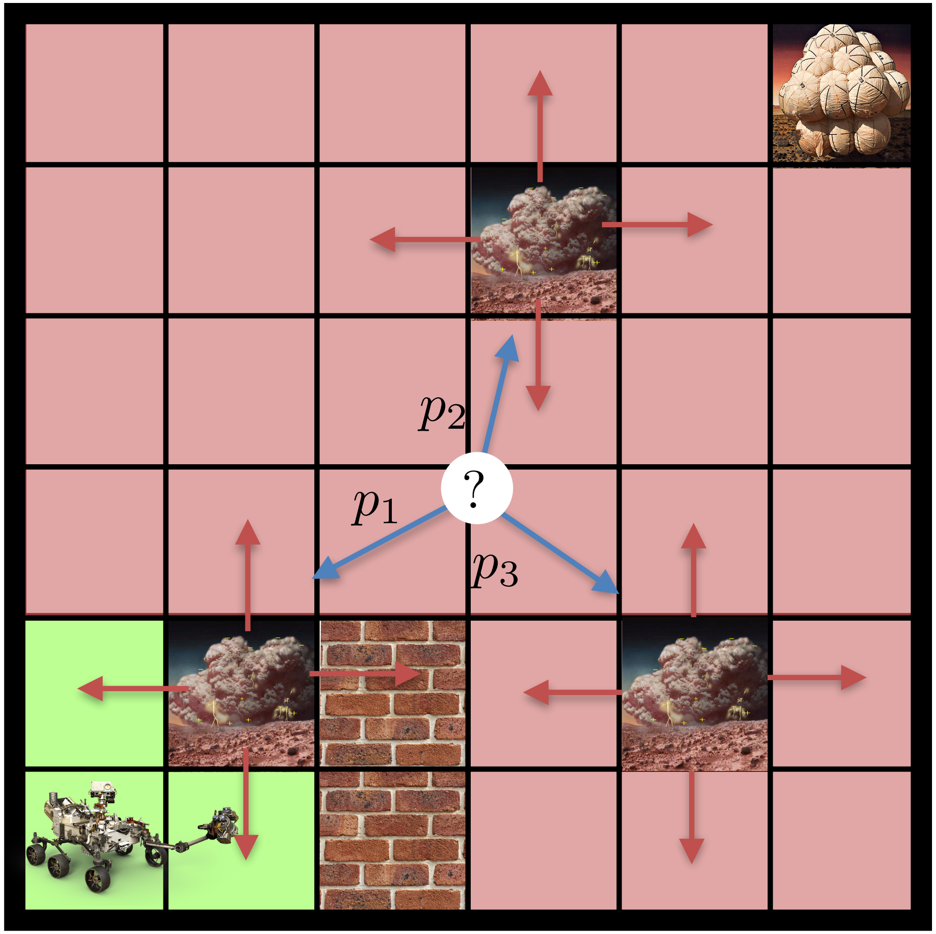
Instead of infinitely many distributions, finite number of choices

Game-based Abstraction for POMDPs

Robot has restricted range of vision

Storm is only observable when near

For robot, storm is either near or far



Abstract possible positions into nondeterministic choices

Instead of infinitely many distributions, finite number of choices

Probabilistic Two-Player Game

Concept: Game-based Abstraction for POMDPs

Concept: Game-based Abstraction for POMDPs

- **Merge states** that share an observation into an abstract state
  - probabilistic movements of storm outside of the visible area

Concept: Game-based Abstraction for POMDPs

- **Merge states** that share an observation into an abstract state
  - probabilistic movements of storm outside of the visible area

- **Introduce choice** over those states
  - position of storm is now determined nondeterministically

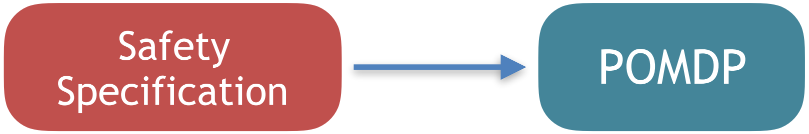
Concept: Game-based Abstraction for POMDPs

- **Merge states** that share an observation into an abstract state
  - probabilistic movements of storm outside of the visible area
- **Introduce choice** over those states
  - position of storm is now determined nondeterministically
- Additional level of nondeterminism: **2-Player game**
  - player 2 chooses position of storm

Concept: Game-based Abstraction for POMDPs

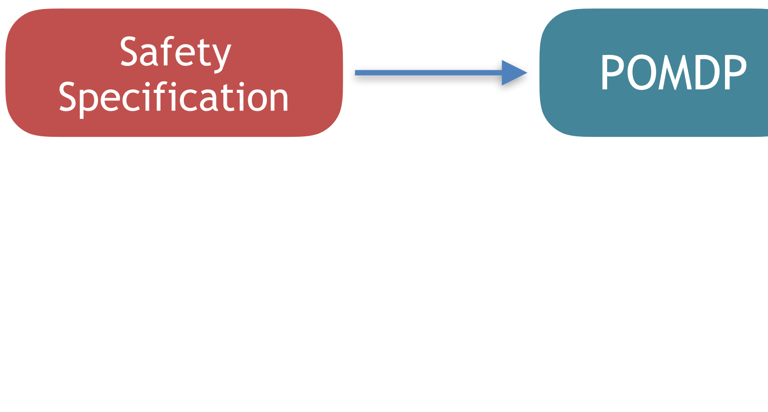
- **Merge states** that share an observation into an abstract state
  - probabilistic movements of storm outside of the visible area
- **Introduce choice** over those states
  - position of storm is now determined nondeterministically
- Additional level of nondeterminism: **2-Player game**
  - player 2 chooses position of storm
- **Worst case** analysis
  - opponent can jump —> storm is strengthened, spurious movements

Story - Game-based Abstraction for POMDPs

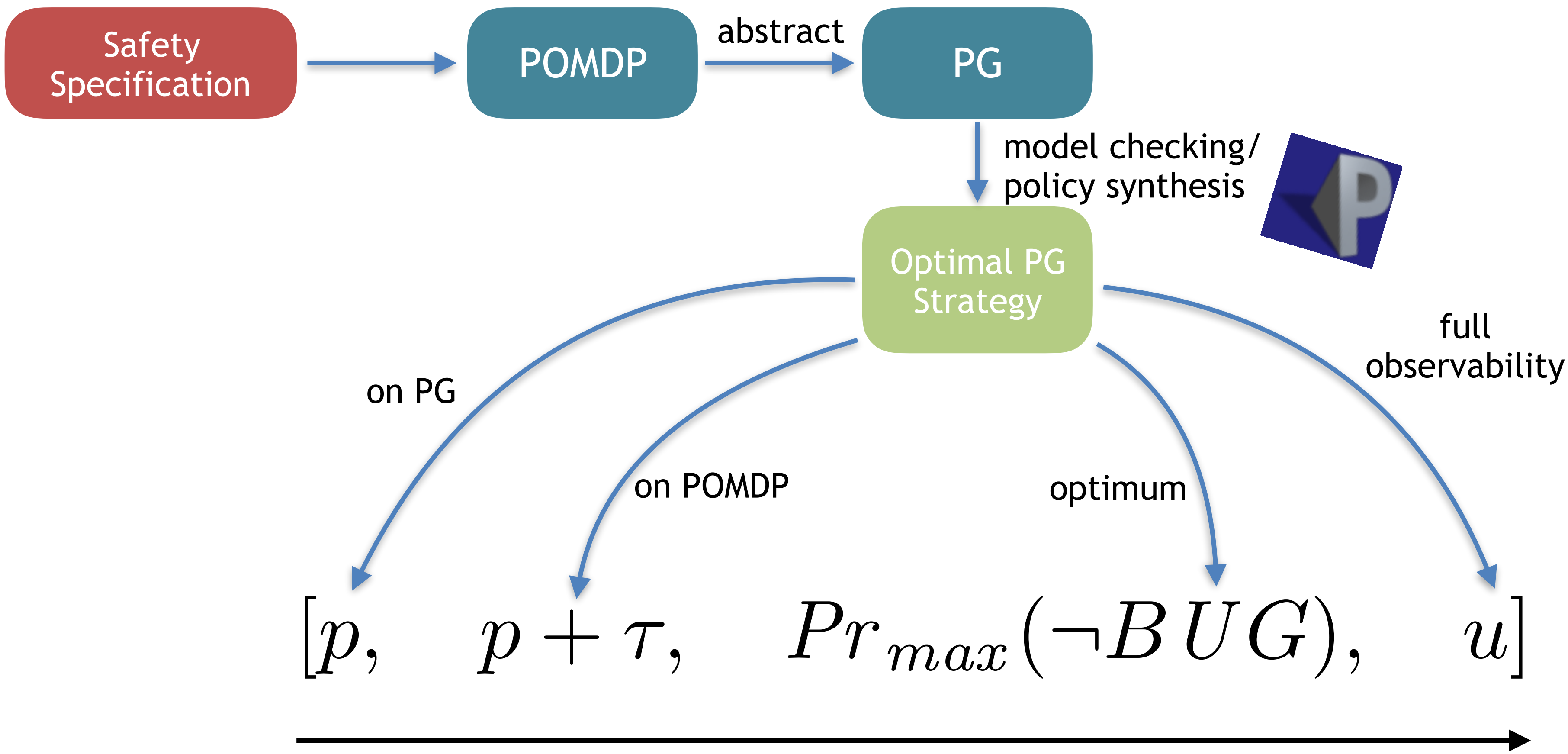


Story - Game-based Abstraction for POMDPs

Story - Game-based Abstraction for POMDPs



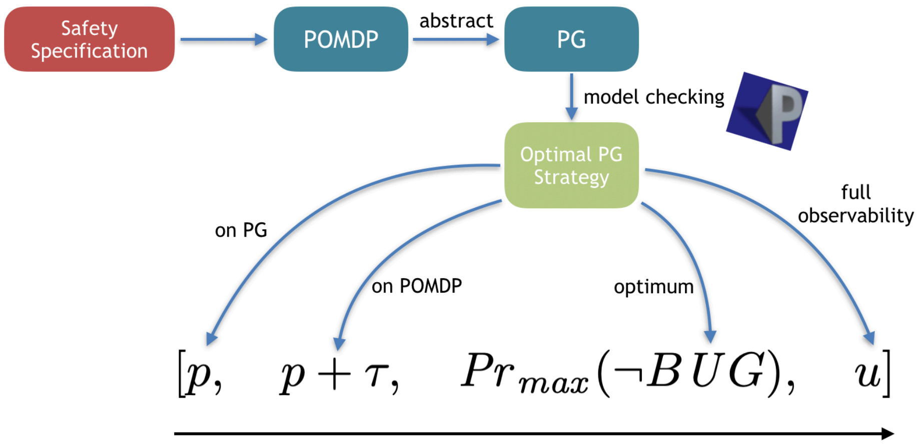
Story - Game-based Abstraction for POMDPs



Refinement

- Usual state splitting as for MDPs is not possible
  - we need a one-to-one correspondence with the POMDP
- Remove spurious movements
- History-based refinement
  - 1-step, multi-step
  - region-based, magnifying lens abstraction
- Alternative: Refine environment —> increase range of vision

Correctness and Completeness



$$[p, \quad p+\tau, \quad Pr_{max}(\neg BUG), \quad u]$$

Correct, as (refined) PG strategy is an overapproximation.

Not complete, as abstraction may be too coarse.

Experiments - Comparison to PRISM-POMDP

	POMDP solution			PG solution			Lifting	MDP
Grid size	States	Result	Sol. Time	States	Result	Sol. Time	Result	Result
3×3	299	**0.8323**	**0.26**	396	**0.8323**	**0.040**	0.8323	0.8323
4×4	983	**0.9556**	**1.81**	1344	**0.9556**	**0.078**	0.9556	0.9556
5×5	2835	**0.9882**	**175.94**	6016	**0.9740**	**0.452**	0.9825	0.9882
5×6	4390	**0.9945**	**4215.06**	7986	**0.9785**	**0.534**	0.9893	0.9945
6×6	6705	**?**	**– MO –**	10544	**0.9830**	**1.414**	0.9933	0.9970
8×8	24893	**?**	**– MO –**	23128	**0.9897**	**6.349**	0.9992	0.9998
10×10	66297	**?**	**– MO –**	40464	**0.9914**	**12.652**	0.9999	0.9999
20×20	– Time out while modelling –			199144	**0.9921**	**127.356**	0.9999	0.9999
30×30	– Time out while modelling –			477824	**0.9921**	**489.369**	– MO –	0.9999
40×40	– Time out while modelling –			876504	**0.9921**	**1726.489**	– MO –	0.9999
50×50	– Time out while modelling –			1395184	**0.9921**	**3963.281**	– MO –	– MO –

Experiments - Comparison to PRISM-POMDP

	POMDP solution			PG solution			Lifting	MDP
Grid size	States	Result	Sol. Time	States	Result	Sol. Time	Result	Result
3×3	299	**0.8323**	**0.26**	396	**0.8323**	**0.040**	0.8323	0.8323
4×4	983	**0.9556**	**1.81**	1344	**0.9556**	**0.078**	0.9556	0.9556
5×5	2835	**0.9882**	**175.94**	6016	**0.9740**	**0.452**	0.9825	0.9882
5×6	4390	0.9945	**4215.06**	7986	0.9785	**0.534**	0.9893	0.9945
6×6	6705	**?**	**– MO –**	10544	**0.9830**	**1.414**	0.9933	0.9970
8×8	24893	**?**	**– MO –**	23128	**0.9897**	**6.349**	0.9992	0.9998
10×10	66297	**?**	**– MO –**	40464	**0.9914**	**12.652**	0.9999	0.9999
20×20	– Time out while modelling –			199144	**0.9921**	**127.356**	0.9999	0.9999
30×30	– Time out while modelling –			477824	**0.9921**	**489.369**	– MO –	0.9999
40×40	– Time out while modelling –			876504	**0.9921**	**1726.489**	– MO –	0.9999
50×50	– Time out while modelling –			1395184	**0.9921**	**3963.281**	– MO –	– MO –

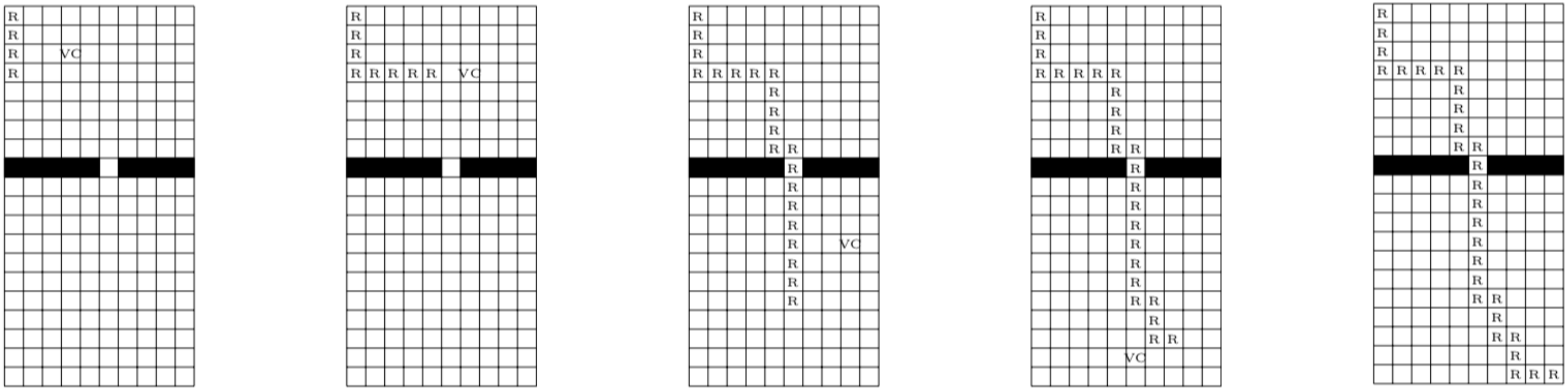
Experiments - Comparison to PRISM-POMDP

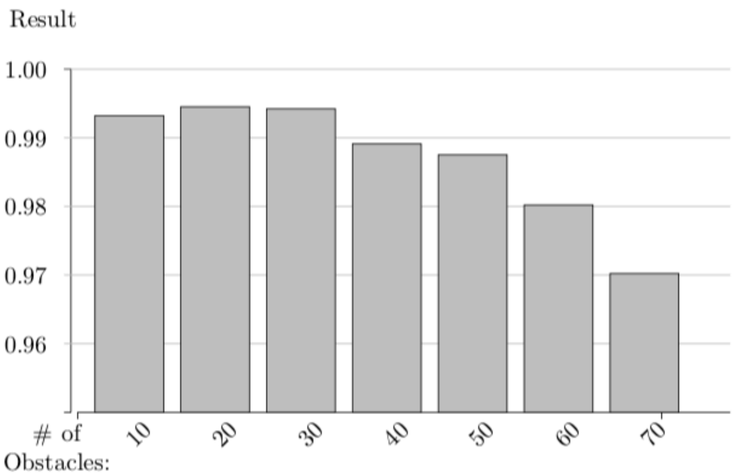
	POMDP solution			PG solution			Lifting	MDP
Grid size	States	Result	Sol. Time	States	Result	Sol. Time	Result	Result
3×3	299	**0.8323**	**0.26**	396	**0.8323**	**0.040**	0.8323	0.8323
4×4	983	**0.9556**	**1.81**	1344	**0.9556**	**0.078**	0.9556	0.9556
5×5	2835	**0.9882**	**175.94**	6016	**0.9740**	**0.452**	0.9825	0.9882
5×6	4390	**0.9945**	**4215.06**	7986	**0.9785**	**0.534**	0.9893	0.9945
6×6	6705	**?**	**– MO –**	10544	**0.9830**	**1.414**	0.9933	0.9970
8×8	24893	**?**	**– MO –**	23128	**0.9897**	**6.349**	0.9992	0.9998
10×10	66297	**?**	**– MO –**	40464	**0.9914**	**12.652**	0.9999	0.9999
20×20	– Time out while modelling –			199144	**0.9921**	**127.356**	0.9999	0.9999
30×30	– Time out while modelling –			477824	**0.9921**	**489.369**	– MO –	0.9999
40×40	– Time out while modelling –			876504	**0.9921**	**1726.489**	– MO –	0.9999
50×50	– Time out while modelling –			1395184	**0.9921**	**3963.281**	– MO –	– MO –

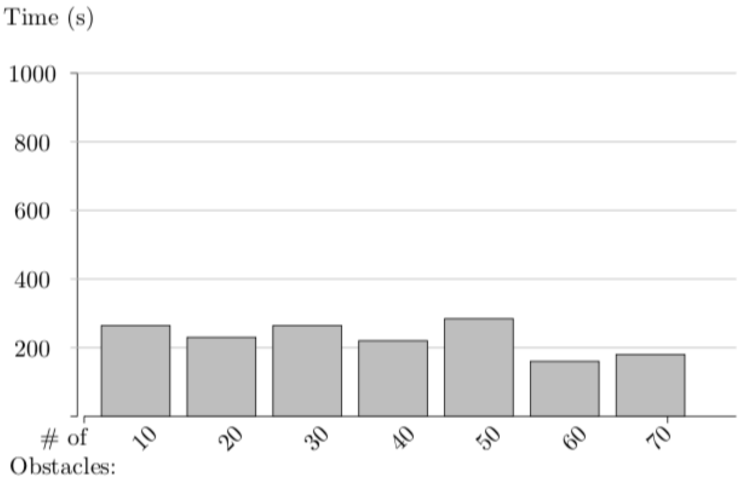
Experiments - Comparison to PRISM-POMDP

	POMDP solution			PG solution			Lifting	MDP
Grid size	States	Result	Sol. Time	States	Result	Sol. Time	Result	Result
3×3	299	**0.8323**	**0.26**	396	**0.8323**	**0.040**	0.8323	0.8323
4×4	983	**0.9556**	**1.81**	1344	**0.9556**	**0.078**	0.9556	0.9556
5×5	2835	**0.9882**	**175.94**	6016	**0.9740**	**0.452**	0.9825	0.9882
5×6	4390	**0.9945**	**4215.06**	7986	**0.9785**	**0.534**	0.9893	0.9945
6×6	6705	**?**	**– MO –**	10544	**0.9830**	**1.414**	0.9933	0.9970
8×8	24893	**?**	**– MO –**	23128	**0.9897**	**6.349**	0.9992	0.9998
10×10	66297	**?**	**– MO –**	40464	**0.9914**	**12.652**	0.9999	0.9999
20×20	– Time out while modelling –			199144	**0.9921**	**127.356**	0.9999	0.9999
30×30	– Time out while modelling –			477824	**0.9921**	**489.369**	– MO –	0.9999
40×40	– Time out while modelling –			876504	**0.9921**	**1726.489**	– MO –	0.9999
50×50	– Time out while modelling –			1395184	**0.9921**	**3963.281**	– MO –	– MO –

Experiments - Policies

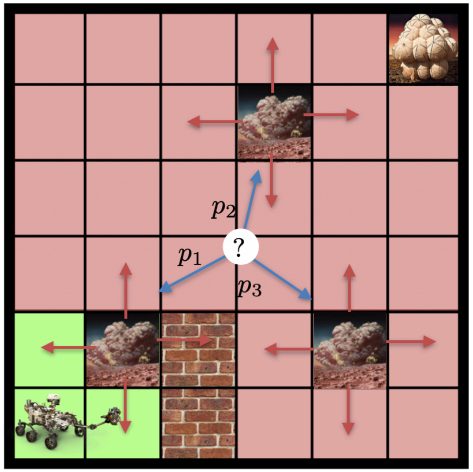


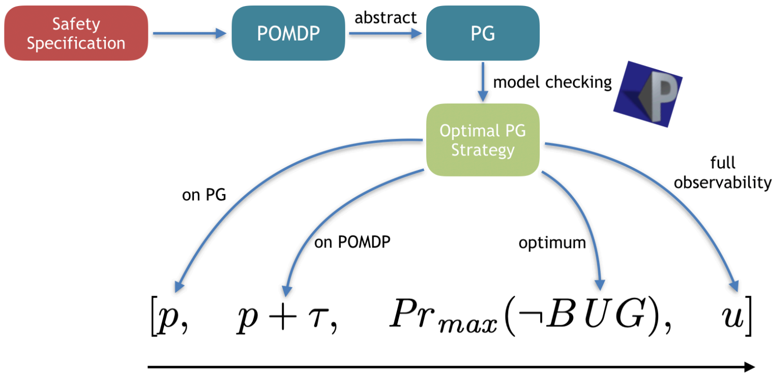




Conclusion - Story 1

- First approach to game-based abstraction for POMDPs
- Superior scalability, no completeness
- Future: Automatic refinement





Leonore Winterer, Sebastian Junges, Ralf Wimmer, Nils Jansen, Ufuk Topcu, Joost-Pieter Katoen, Bernd Becker: Motion planning under partial observability using game-based abstraction. CDC 2017: 2201-2208

Leonore Winterer, Sebastian Junges, Ralf Wimmer, Nils Jansen, Ufuk Topcu, Joost-Pieter Katoen, Bernd Becker: Motion Planning under Partial Observability using Game-Based Abstraction. CoRR abs/1708.04236 (2019)

Stories - Policy Synthesis for POMDPs

1. Game-based abstraction
2. Finite-memory controllers
3. Recurrent neural networks
4. Fun: Humans in the loop

Story - Finite-state Controllers (FSCs)

POMDP

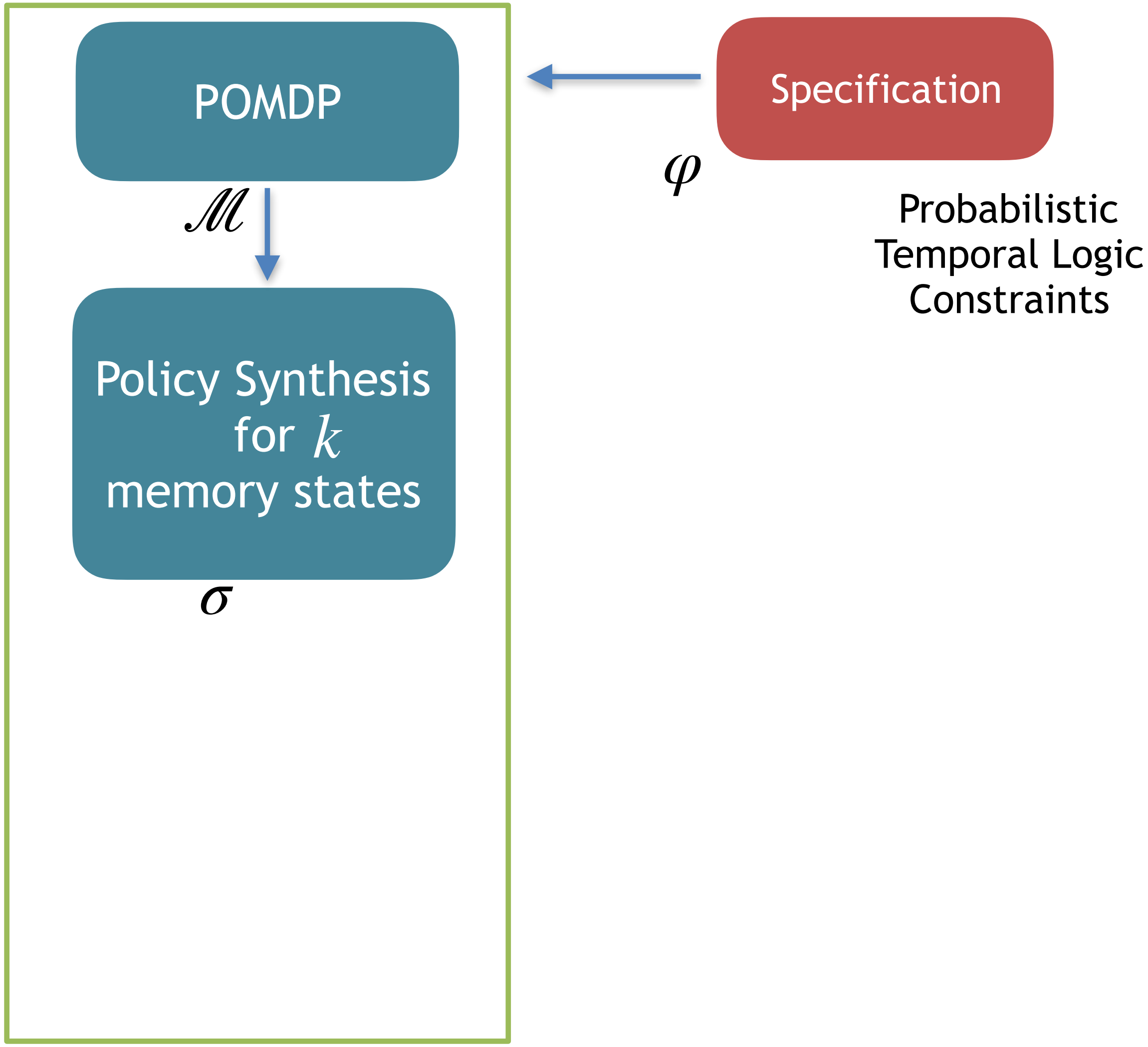
$\mathcal{M}$

Specification

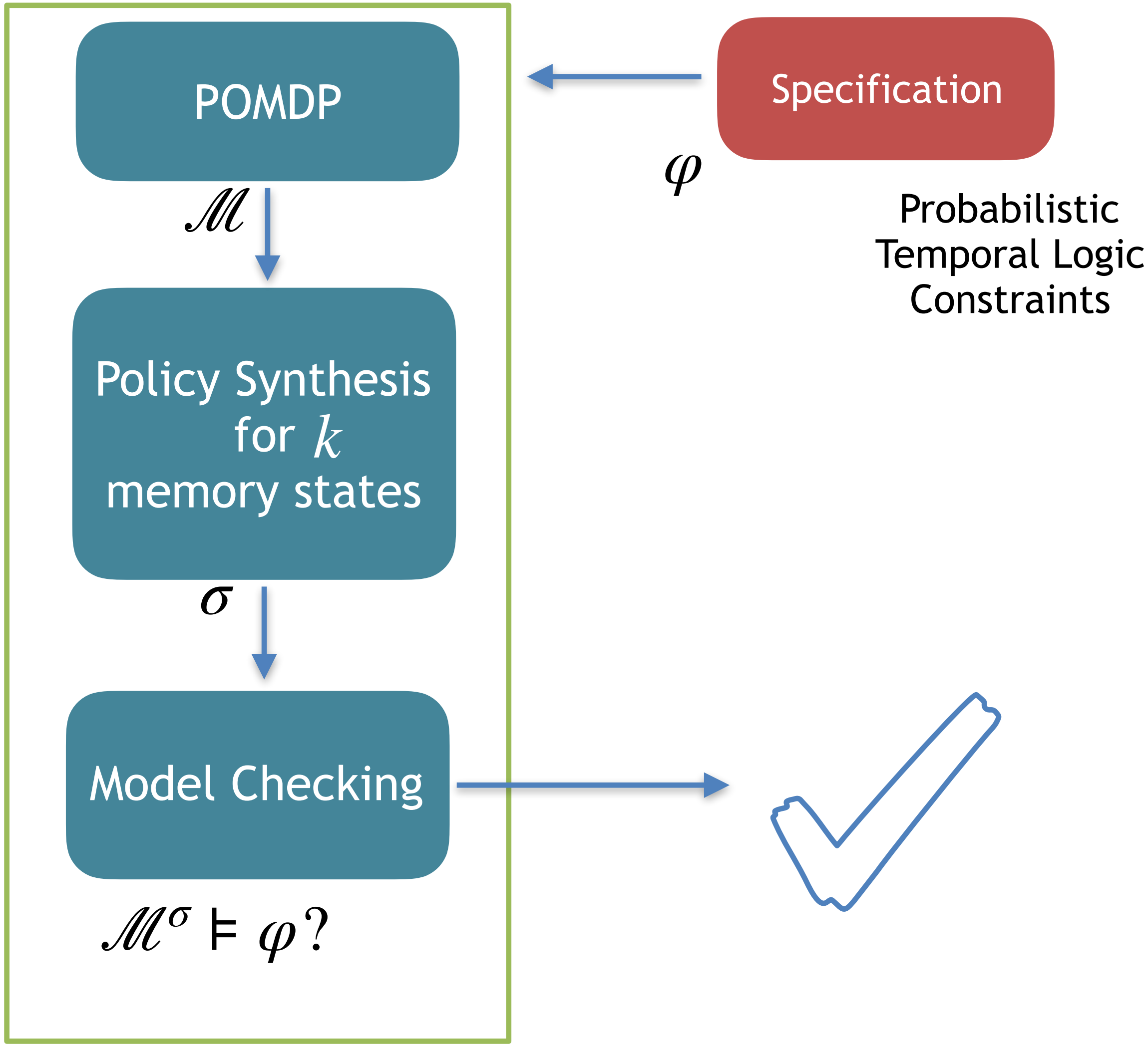
φ

Probabilistic
Temporal Logic
Constraints

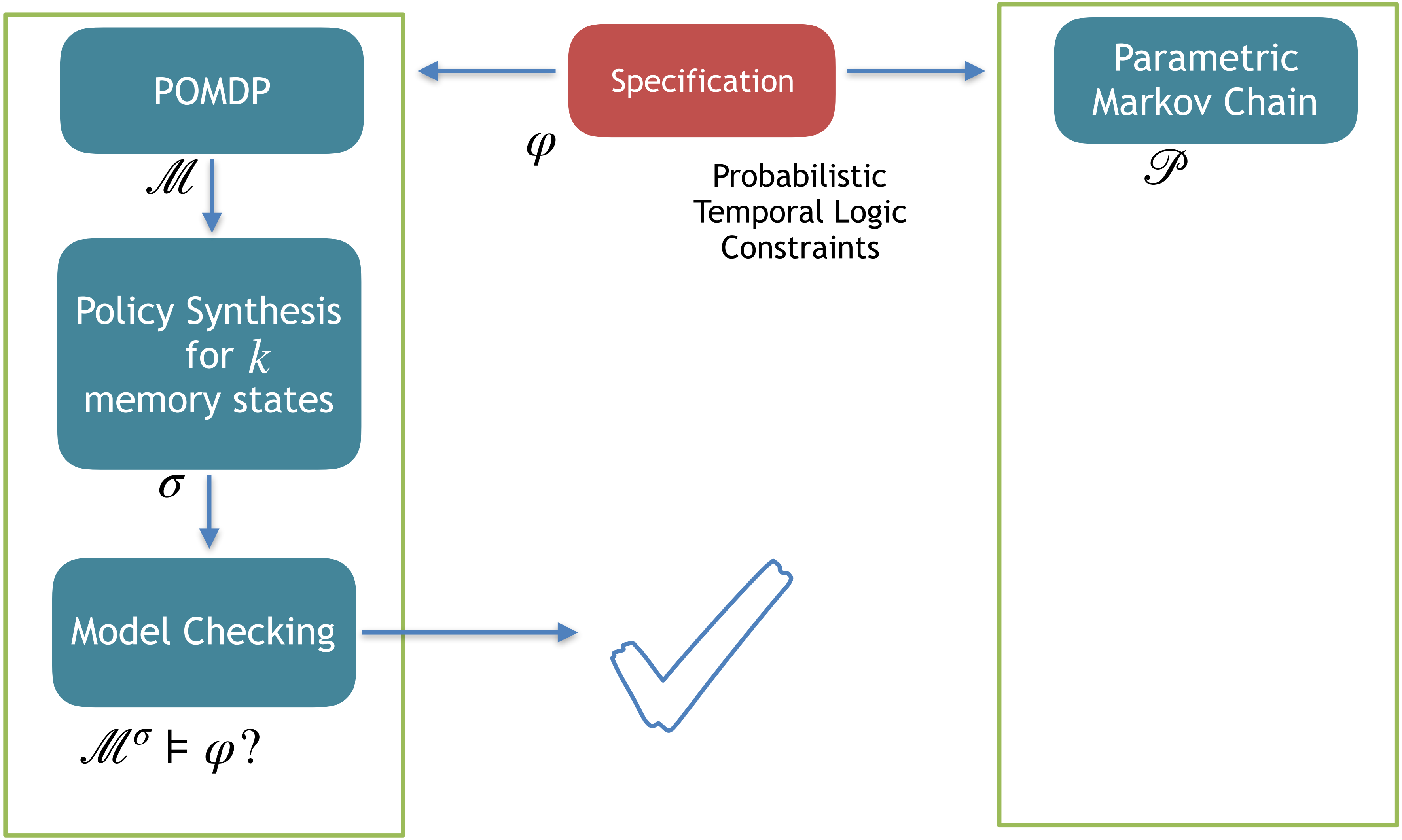
Story - Finite-state Controllers (FSCs)



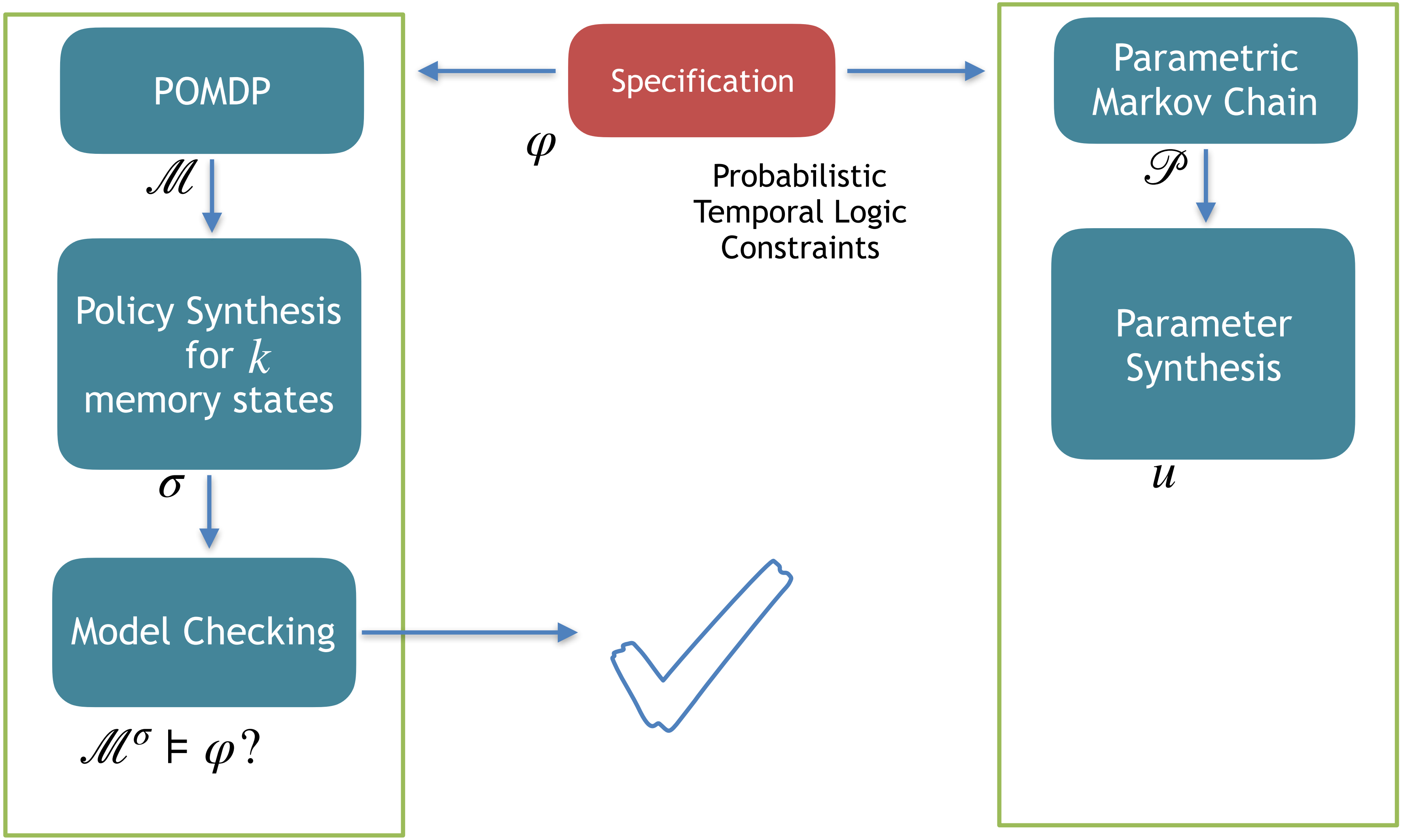
Story - Finite-state Controllers (FSCs)



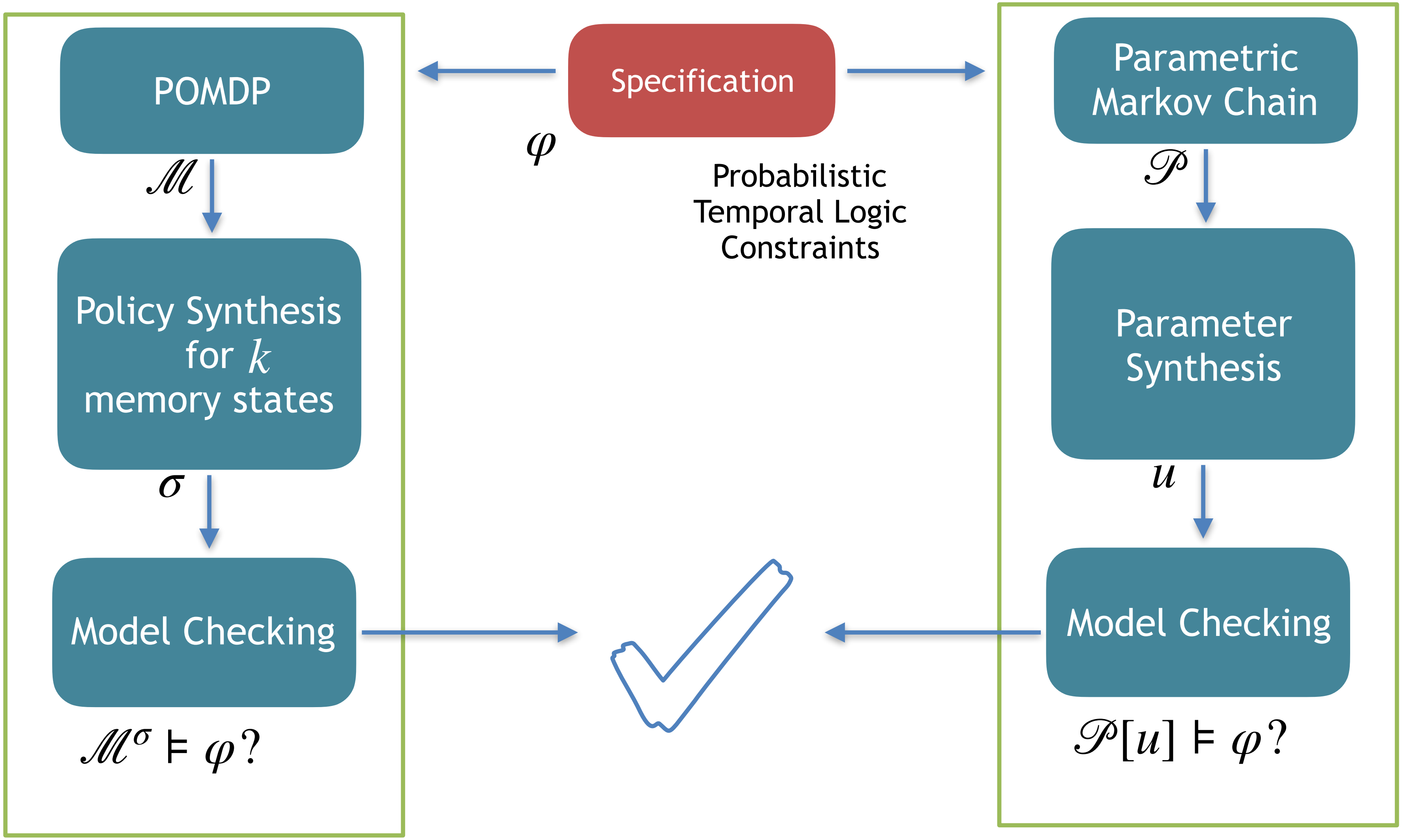
Story - Finite-state Controllers (FSCs)



Story - Finite-state Controllers (FSCs)

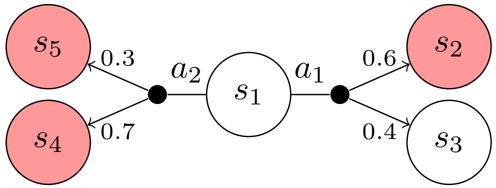


Story - Finite-state Controllers (FSCs)

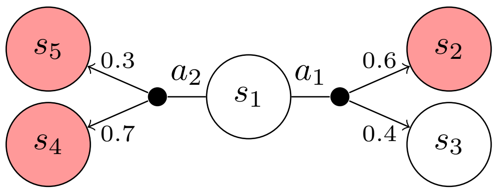


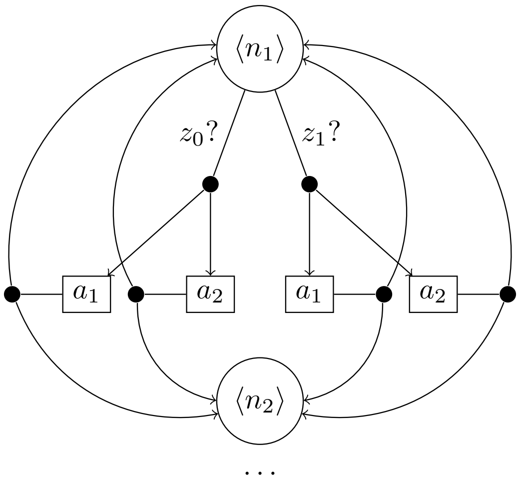
Finite-Memory Strategies for POMDPs

Finite-Memory Strategies for POMDPs

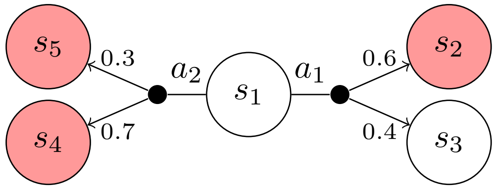


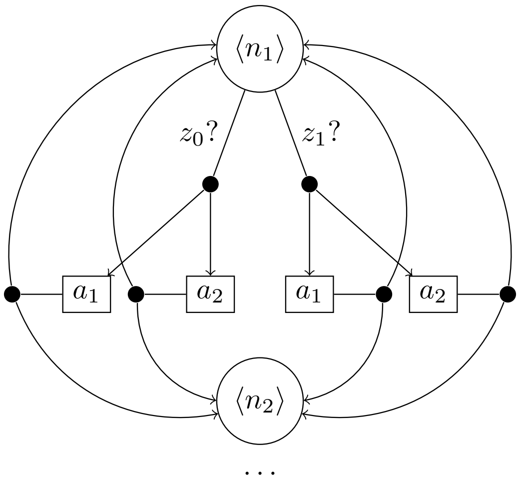
Finite-Memory Strategies for POMDPs

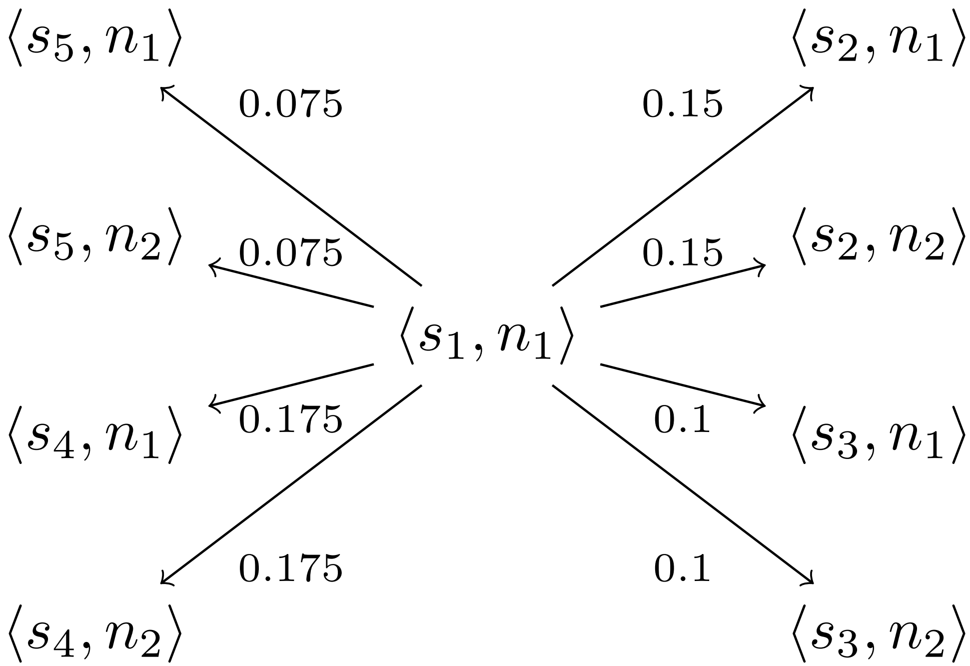




Finite-Memory Strategies for POMDPs

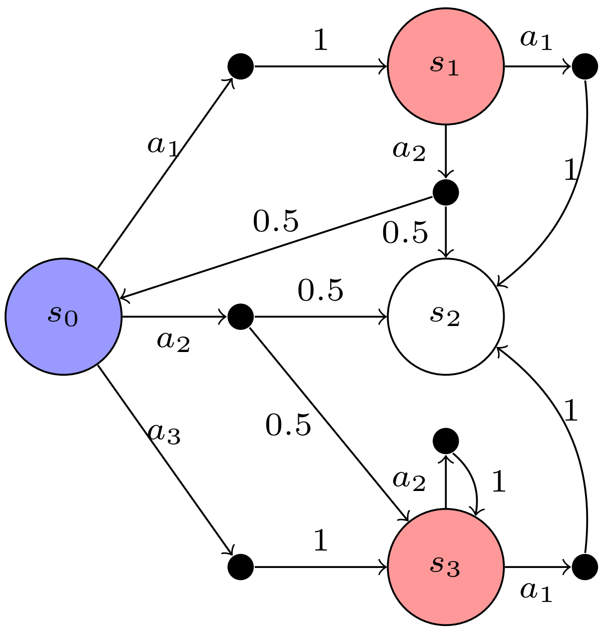






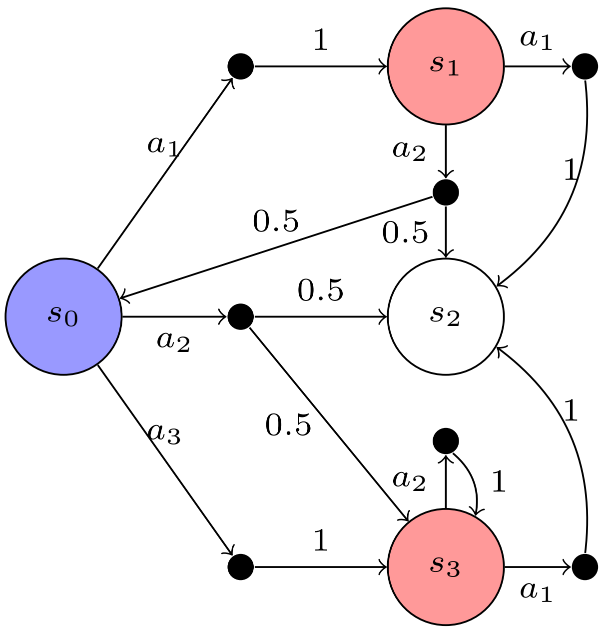
- Encode memory using finite state controller (FSCs)
- On the product, policy is memoryless

Dependencies for Randomized Policies



Dependencies for Randomized Policies

Map observation-action pairs to randomized choices



$a_1 \longrightarrow 1-q$

$a_2 \longrightarrow q$

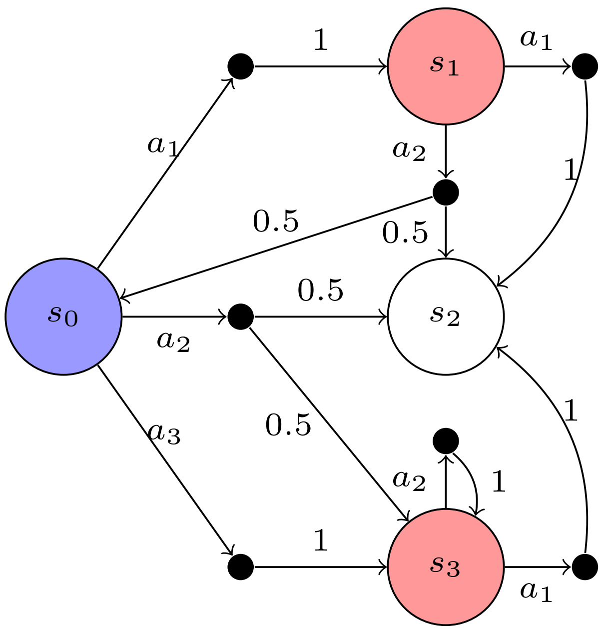
$a_1 \longrightarrow p_1$

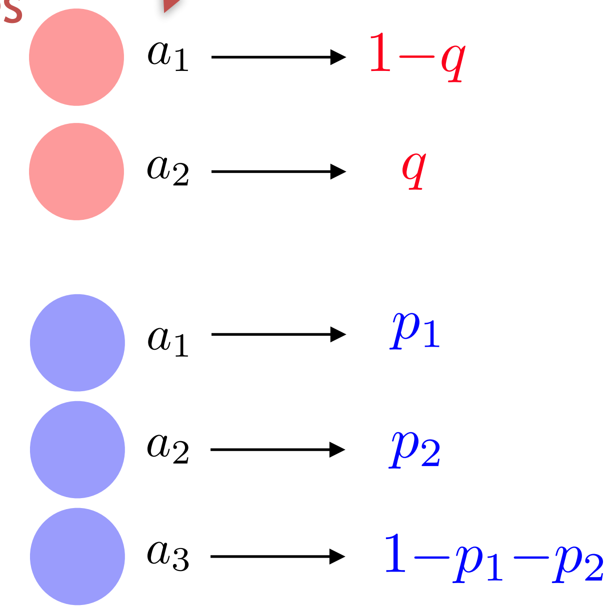
$a_2 \longrightarrow p_2$

$a_3 \longrightarrow 1-p_1-p_2$

Dependencies for Randomized Policies

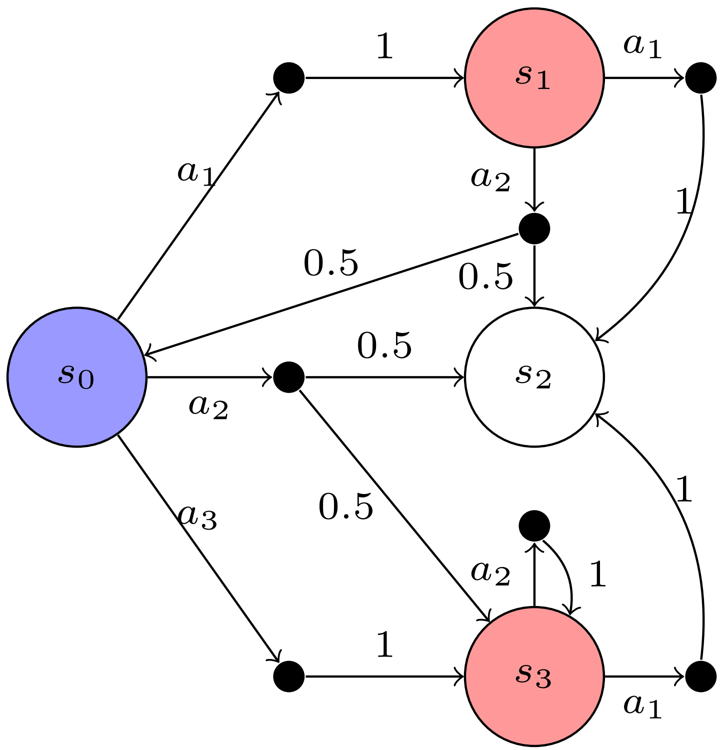
Map observation-action pairs to randomized choices



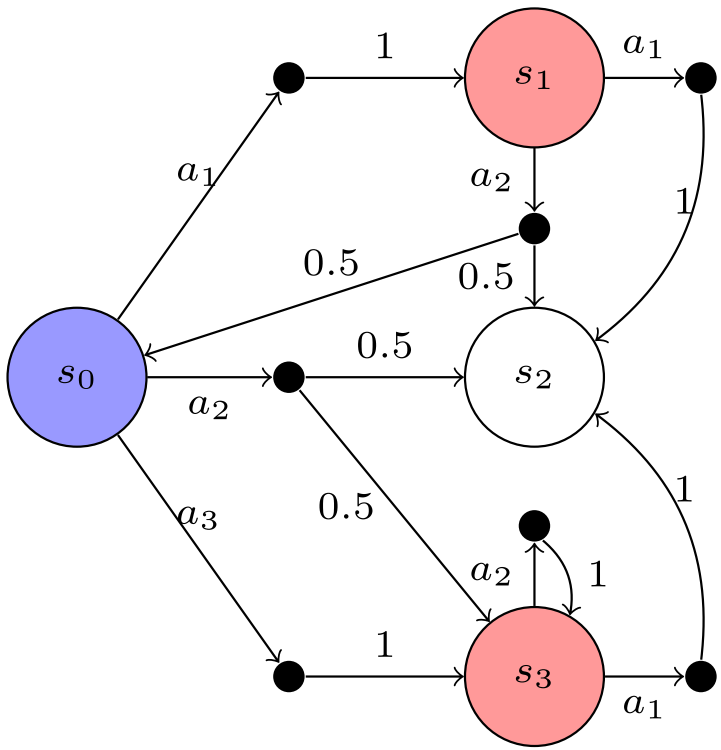


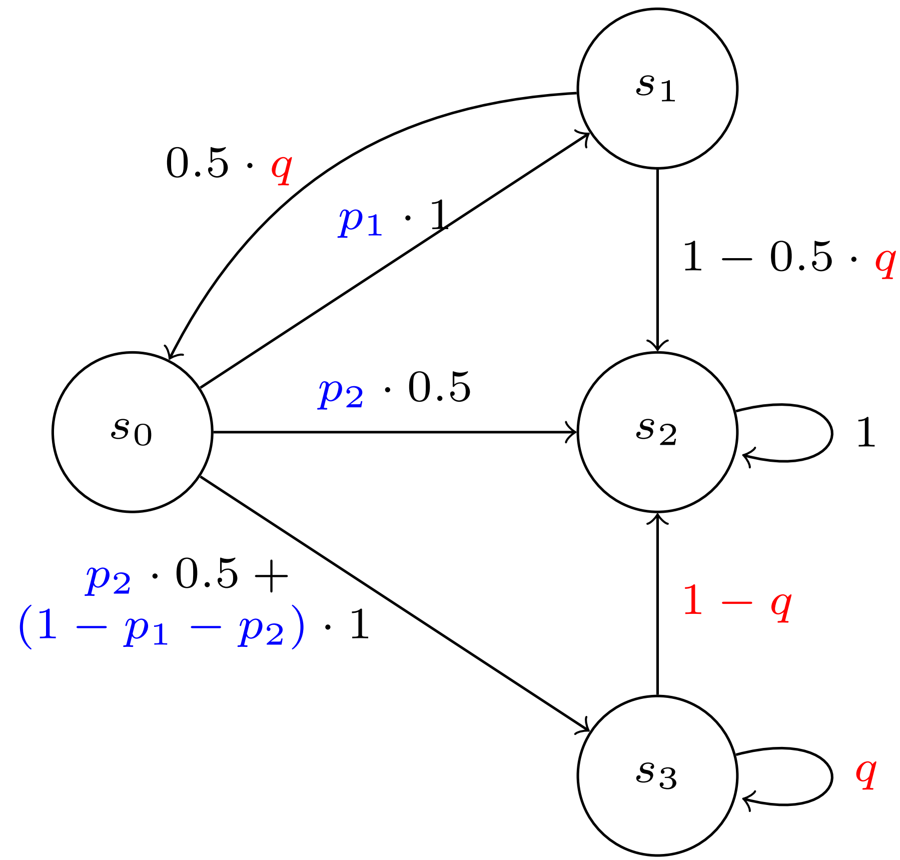
Randomized observation-based policy is sufficiently described by these probabilities!

Randomized Policies as Parametric MCs

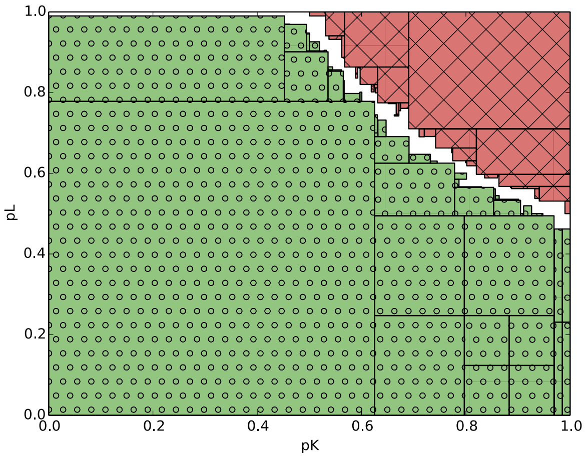


Randomized Policies as Parametric MCs





Parameter Synthesis - Outputs



Parameter Space Partitioning

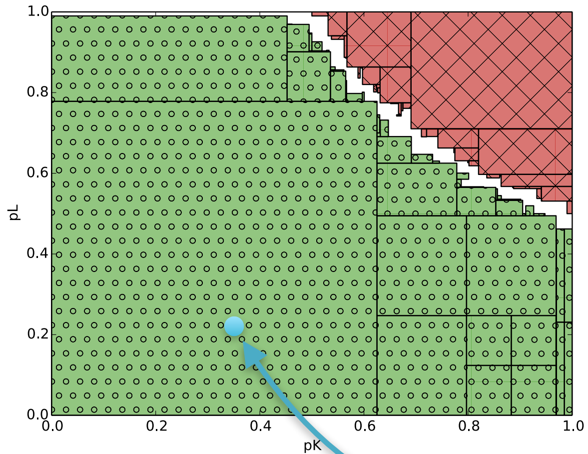
- concise description of parameter values that yield (un)satisfactory results

$$p \cdot (1-p) \cdot \frac{1-q}{1-pq}$$

Rational Function

- generalization of non-parametric model checking

Parameter Synthesis - Outputs



Parameter Space Partitioning

- concise description of parameter values that yield (un)satisfactory results

Feasible Solution

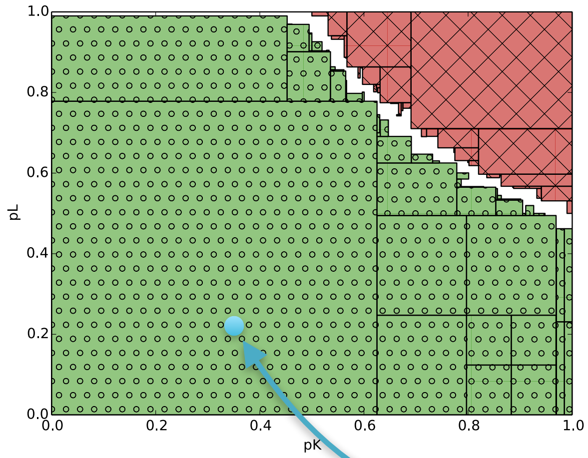
- one parameter valuation that is satisfactory

$$p \cdot (1-p) \cdot \frac{1-q}{1-pq}$$

Rational Function

- generalization of non-parametric model checking

Parameter Synthesis - Outputs



Parameter Space Partitioning

- concise description of parameter values that yield (un)satisfactory results

Feasible Solution

- one parameter valuation that is satisfactory

$$p \cdot (1-p) \cdot \frac{1-q}{1-pq}$$

Rational Function

- generalization of non-parametric model checking

Already the feasibility problem is ETR-hard! (existential theory of the reals)

Winkler T., Junges S., Pérez, G. A., Katoen, J.-P.: On the Complexity of Reachability in Parametric Markov Decision Processes

Parameter Synthesis Approaches

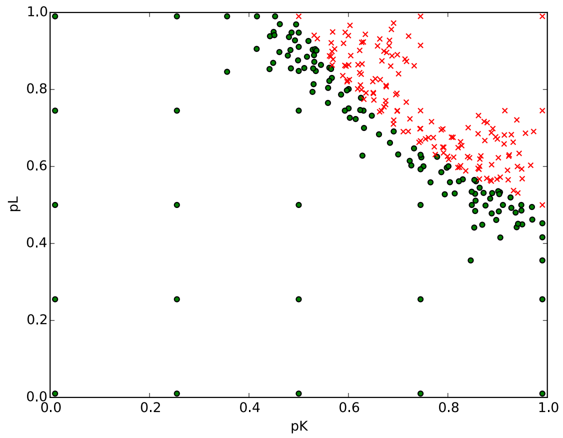
Parameter Synthesis Approaches

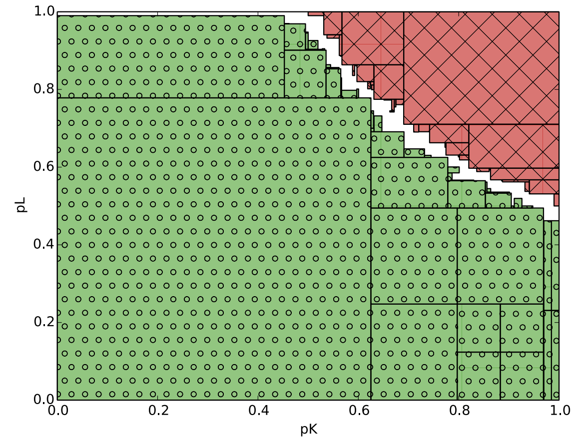
QEST 2014

CAV 2015

ATVA 2016

- Tool support: PRISM, PARAM, PROPhESY, Storm





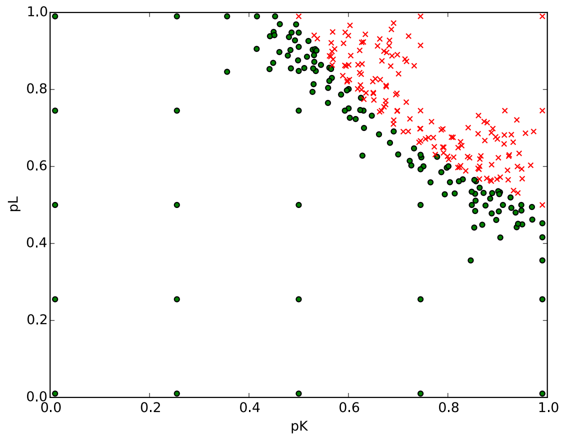
Parameter Synthesis Approaches

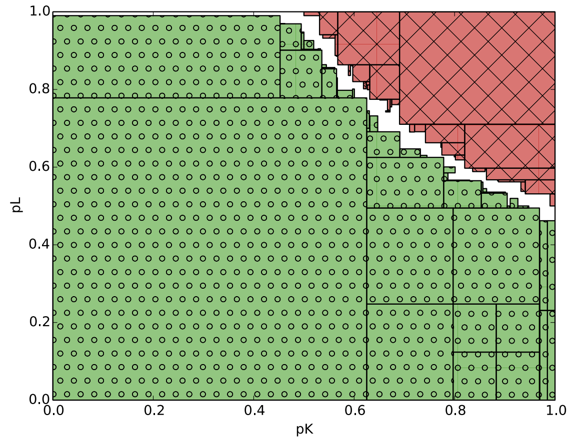
QEST 2014

CAV 2015

ATVA 2016

- Tool support: PRISM, PARAM, PROPhESY, Storm



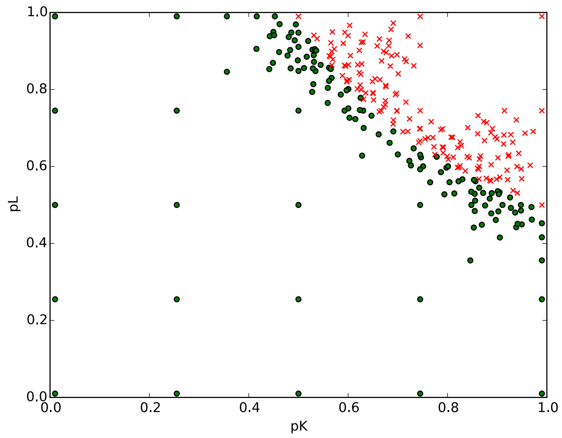


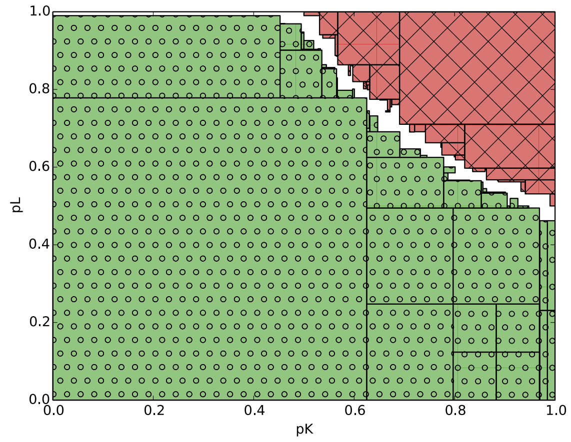
- Used to be restricted to a few parameters

Parameter Synthesis Approaches

- QEST 2014
- CAV 2015
- ATVA 2016
- TACAS 2017
- ATVA 2018

- Tool support: PRISM, PARAM, PROPhESY, Storm





- Used to be restricted to a few parameters
- Utilize convex optimization:
  - Sequential Convex Optimization for the Efficient Verification of Parametric MDPs
  - Synthesis in pMDPs: A Tale of 1001 Parameters

Nonlinear Program - Parameter Synthesis

$$
\begin{aligned}
\text{minimize} \quad & p_{s_I} \\
\text{subject to} \quad & \\
\forall s \in T. \quad & p_s = 1 \\
\forall s, s' \in S. \forall \alpha \in Act. \quad & \mathcal{P}(s, \alpha, s') \geq \epsilon \\
\forall s \in S. \forall \alpha \in Act. \quad & \sum_{s' \in S} \mathcal{P}(s, \alpha, s') = 1 \\
& \lambda \geq p_{s_I} \\
\forall s \in S \setminus T. \forall \alpha \in Act. \quad & p_s \geq \sum_{s' \in S} \mathcal{P}(s, \alpha, s') \cdot p_{s'}
\end{aligned}
$$

Nonlinear Program - Parameter Synthesis

$$
\begin{aligned}
&\text{minimize} && p_{s_I} \\
&\text{subject to} && \\
&\forall s \in T. && p_s = 1 \\
&\forall s, s' \in S. \forall \alpha \in Act. && \mathcal{P}(s, \alpha, s') \geq \epsilon \\
&\forall s \in S. \forall \alpha \in Act. && \sum_{s' \in S} \mathcal{P}(s, \alpha, s') = 1 \\
& && \lambda \geq p_{s_I} \\
&\forall s \in S \setminus T. \forall \alpha \in Act. && p_s \geq \sum_{s' \in S} \mathcal{P}(s, \alpha, s') \cdot p_{s'}
\end{aligned}
$$

probability at target states

graph-preserving parameter instantiations

well-defined distributions

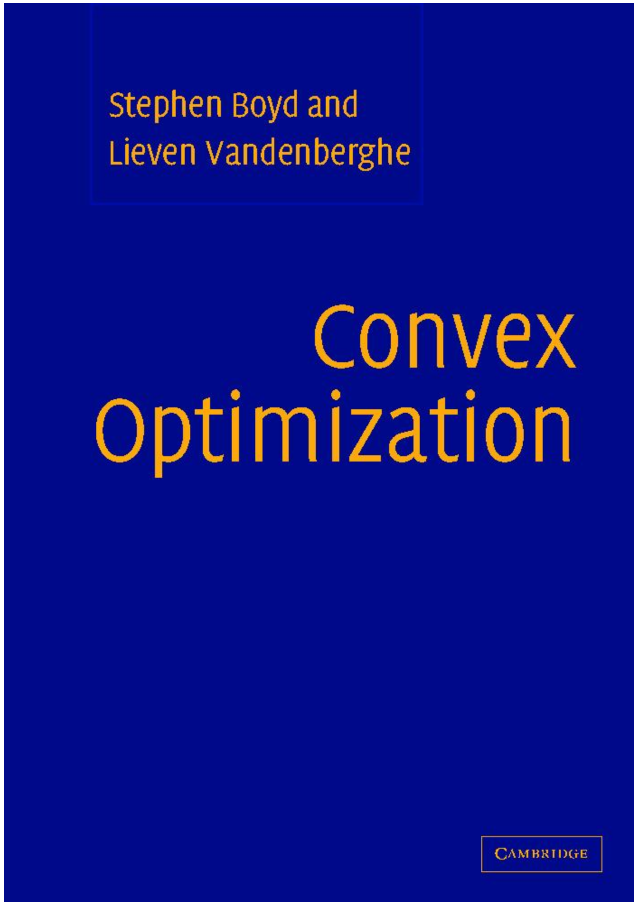
satisfaction of specification

probability computation

affine function

real-valued variable

Convexify it!



Story - Convex-concave Procedure

Specification → General parametric MDP

↓

Parametric MDP restricted to affine functions

↓

Nonlinear Program

PARAM

Story - Convex-concave Procedure

Specification

General parametric MDP

Parametric MDP restricted to affine functions

PARAM

Nonlinear Program

Quadratically Constrained Quadratic Program (QCQP)

Story - Convex-concave Procedure

Specification → General parametric MDP

General parametric MDP → Parametric MDP restricted to affine functions

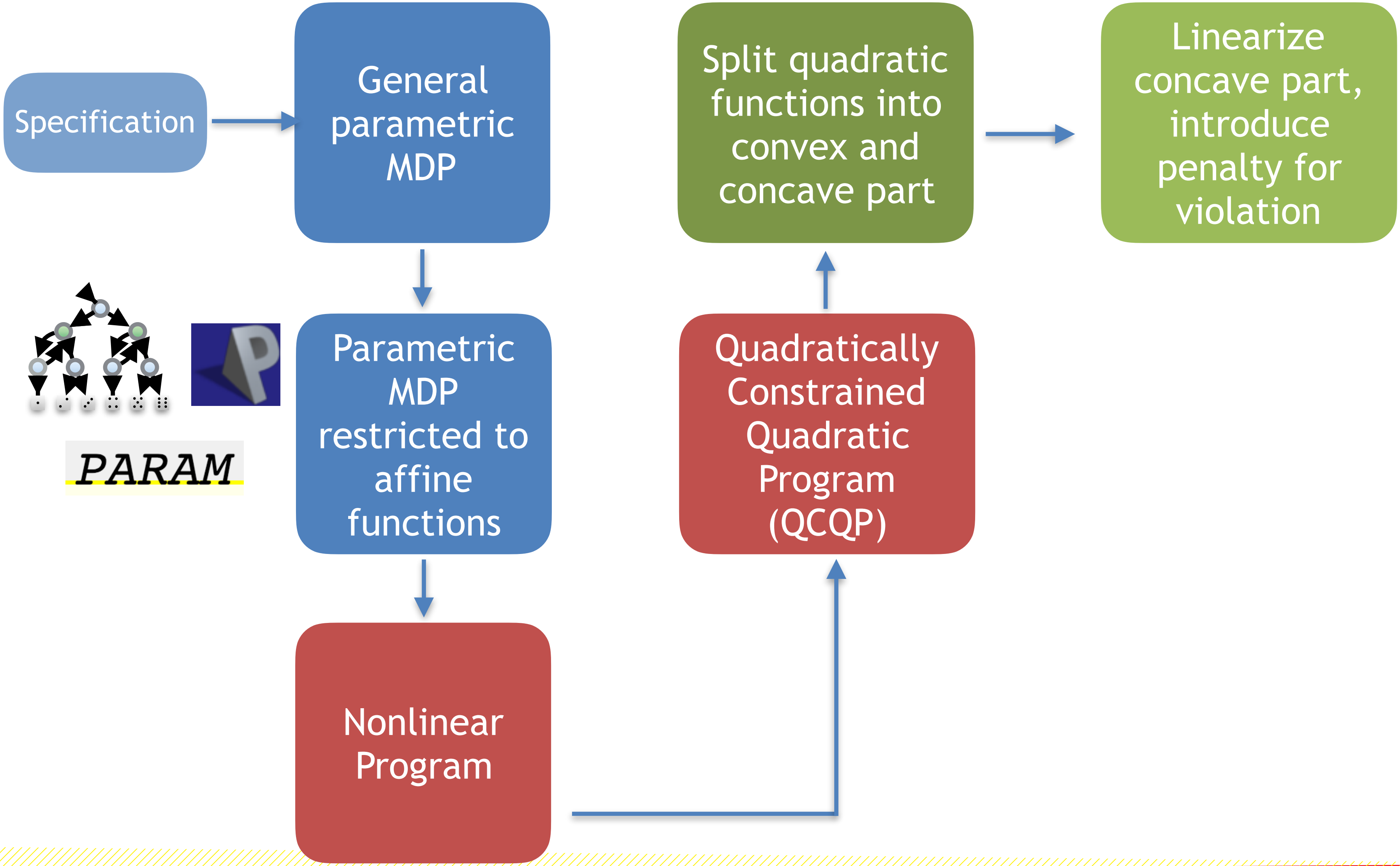
PARAM

Parametric MDP restricted to affine functions → Nonlinear Program

Nonlinear Program → Quadratically Constrained Quadratic Program (QCQP)

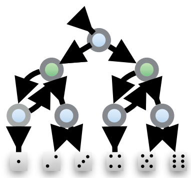
Quadratically Constrained Quadratic Program (QCQP) → Split quadratic functions into convex and concave part

Story - Convex-concave Procedure



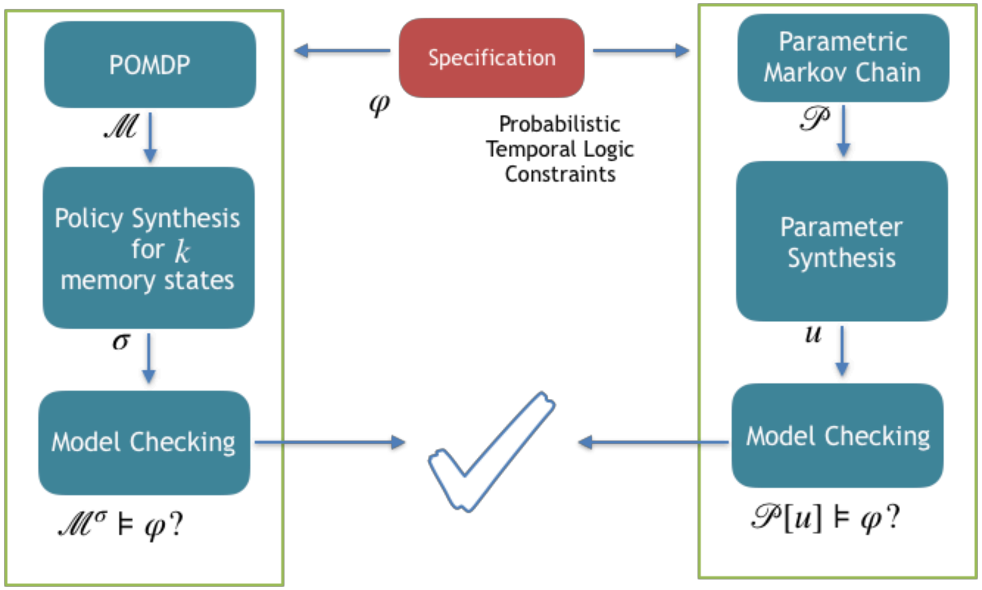
Story - Convex-concave Procedure

Specification → General parametric MDP → Parametric MDP restricted to affine functions → Nonlinear Program → Quadratically Constrained Quadratic Program (QCQP) → Split quadratic functions into convex and concave part → Linearize concave part, introduce penalty for violation → Minimize violations using Model Checking → Feasible solution

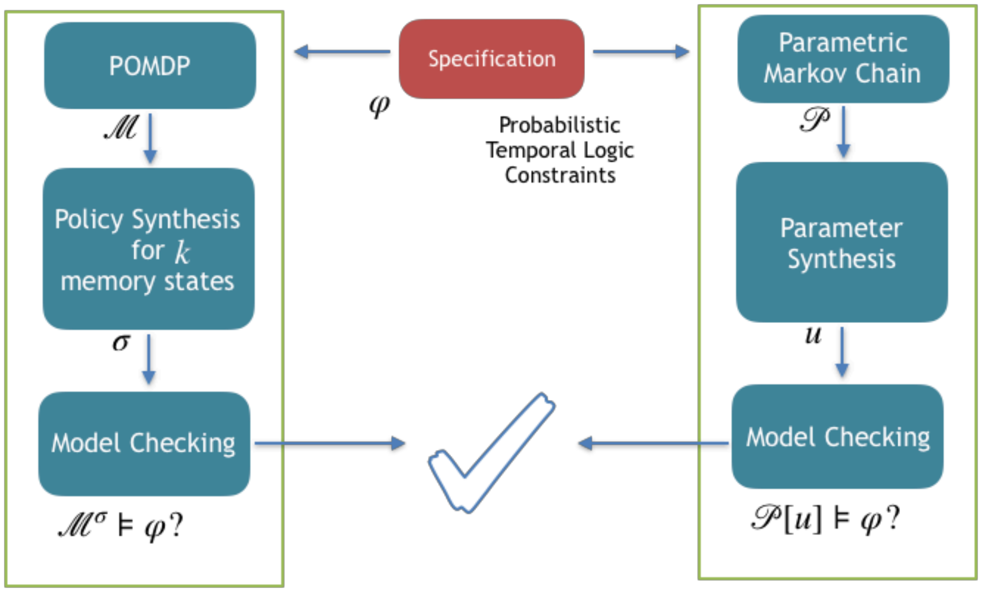


PARAM

Why is This Helpful?



Why is This Helpful?



- All algorithms and complexity results carry over
- Extensive and mature tool-support for parameter synthesis
- Superior performance to state-of-the-art POMDP solvers

http://stormchecker.org

https://github.com/moves-rwth/prophesy

Experiments

Problem			Info			PSO			SMT	CCP		
Set	Inst	Spec	States	Trans.	Par.	tmin	tmax	**tavg**	**t**	**t**	solv	iter
Brp	16,2	$\mathbb{P}_{\leq 0.1}$	98	194	**2**	0	0	**0**	40	0	30%	3
Brp	512,5	$\mathbb{P}_{\leq 0.1}$	6146	12290	**2**	24	36	**28**	TO	33	24%	3
Crowds	10,5	$\mathbb{P}_{\leq 0.1}$	42	82	**2**	4	5	5	8	**4**	2%	4
Nand	5,10	$\mathbb{P}_{\leq 0.05}$	10492	20982	**2**	21	51	28	TO	**22**	21%	2
Zeroconf	10000	$\mathbb{E}_{\leq 10010}$	10003	20004	**2**	2	4	**3**	TO	57	81%	3
GridA	4	$\mathbb{P}_{\geq 0.84}$	1026	2098	**72**	11	11	**11**	TO	22	81%	11
GridB	8,5	$\mathbb{P}_{\geq 0.84}$	8653	17369	**700**	409	440	427	TO	**213**	84%	8
GridB	10,6	$\mathbb{P}_{\geq 0.84}$	16941	33958	**1290**	533	567	553	TO	**426**	84%	7
GridC	6	$\mathbb{E}_{\leq 4.8}$	1665	305	**168**	261	274	267	TO	**169**	90%	23
Maze	5	$\mathbb{E}_{\leq 14}$	1303	2658	**590**	213	230	219	TO	**67**	89%	8
Maze	5	$\mathbb{E}_{\leq 6}$	1303	2658	**590**	–	–	TO	TO	**422**	85%	97
Maze	7	$\mathbb{E}_{\leq 6}$	2580	5233	**1176**	–	–	TO	TO	**740**	90%	60
Netw	5,2	$\mathbb{E}_{\leq 11.5}$	21746	63158	**2420**	312	523	359	TO	**207**	39%	3
Netw	5,2	$\mathbb{E}_{\leq 10.5}$	21746	63158	**2420**	–	–	TO	TO	**210**	38%	4
Netw	4,3	$\mathbb{E}_{\leq 11.5}$	38055	97335	**4545**	–	–	TO	TO	MO	-	-
Repud	8,5	$\mathbb{P}_{\geq 0.1}$	1487	3002	**360**	16	22	18	TO	**4**	36%	2
Repud	8,5	$\mathbb{P}_{\leq 0.05}$	1487	3002	**360**	273	324	293	TO	**14**	72%	4
Repud	16,2	$\mathbb{P}_{\leq 0.01}$	790	1606	**96**	–	–	TO	TO	**15**	78%	9
Repud	16,2	$\mathbb{P}_{\geq 0.062}$	790	1606	**96**	–	–	TO	TO	TO	-	-

pMCs

POMDPs

Numerical Experiments

Performance analysis

Simple Grid.

POMDP with **17** states, **62** branches, **3** observations

Numerical Experiments

Actual optimum (arbitrary K) 4.13

Performance analysis

Simple Grid.

POMDP with **17** states, **62** branches, **3** observations

Find a policy such that we arrive at destination within T steps

Numerical Experiments

Actual optimum (arbitrary K) 4.13

Performance analysis

Simple Grid.

POMDP with **17** states, **62** branches, **3** observations

Find a policy such that we arrive at destination within T steps

K	states	parameters	time for T=4.15	time for T=5.5
K=1	47	3	not possible	<1 s
K=2	183	15	7.4 s	<1 s

Nils Jansen

Radboud University

Numerical Experiments

Actual optimum (arbitrary K) 4.13

Performance analysis

Simple Grid.

POMDP with **17** states, **62** branches, **3** observations

Find a policy such that we arrive at destination within T steps

K	states	parameters	time for T=4.15	time for T=5.5
K=1	47	3	not possible	<1 s
K=2	183	15	7.4 s	<1 s

Automatically proven: For K=1, 5 is a lower bound. (<1 s)

Larger Numerical Experiments

Practical implications

Network protocol:
Optimally assign packets to slots.

Actual optimum
(arbitrary K) around 9

POMDP with **2729** states, **4937** branches, **361** observations

Find a policy such that the expected packet loss is below T packets

Larger Numerical Experiments

Practical implications

Network protocol:
Optimally assign packets to slots.

Actual optimum (arbitrary K) around 9

POMDP with **2729** states, **4937** branches, **361** observations

Find a policy such that the expected packet loss is below T packets

K	states	parameters	time for T=10	time for T=15
K=1	3268	276	43 s	4 s
K=2	16004	1783	877 s	28 s

SolvePOMDP and PRISM-POMDP: Time outs (> 3600 seconds)

Larger Numerical Experiments

Practical implications

Network protocol:
Optimally assign packets to slots.

Actual optimum (arbitrary K) around 9

POMDP with **2729** states, **4937** branches, **361** observations

Find a policy such that the expected packet loss is below T packets

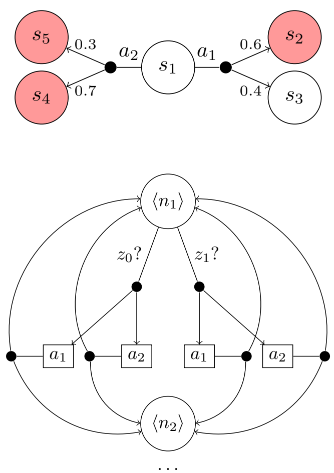
K	states	parameters	time for T=10	time for T=15
K=1	3268	276	43 s	4 s
K=2	16004	1783	877 s	28 s

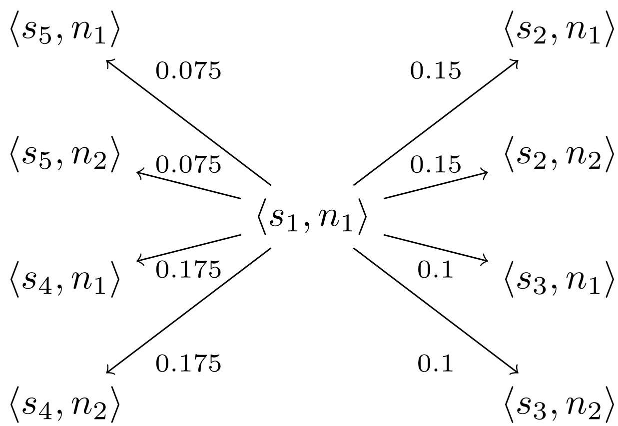
SolvePOMDP and PRISM-POMDP: Time outs (> 3600 seconds)

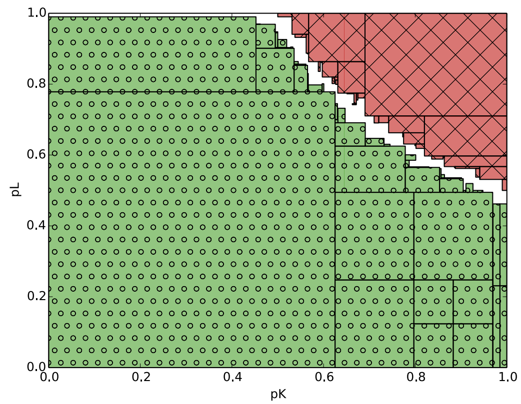
For K=4, 5 is a lower bound. (183 s)

Conclusion to Story 2

- Novel way to generate provably correct POMDP strategies
- Good scalability, not optimal
- Future: More principled approach to permissive strategies







Sebastian Junges, Nils Jansen, Ralf Wimmer, Tim Quatmann, Leonore Winterer, Joost-Pieter Katoen, Bernd Becker: Finite-State Controllers of POMDPs using Parameter Synthesis. UAI 2018: 519-529

Stories - Policy Synthesis for POMDPs

1. Game-based abstraction
2. Finite-memory controllers
3. Recurrent neural networks
4. Fun: Humans in the loop

Be Lazy: Guess a Policy and Verify!

$\mathscr{M}$ POMDP

φ Specification

Probabilistic
Temporal Logic
Constraints

Be Lazy: Guess a Policy and Verify!

POMDP $\mathscr{M}$

Specification φ

Probabilistic Temporal Logic Constraints

Guess Candidate Policy σ

Be Lazy: Guess a Policy and Verify!

POMDP

$\mathcal{M}$

Specification

φ

Probabilistic Temporal Logic Constraints

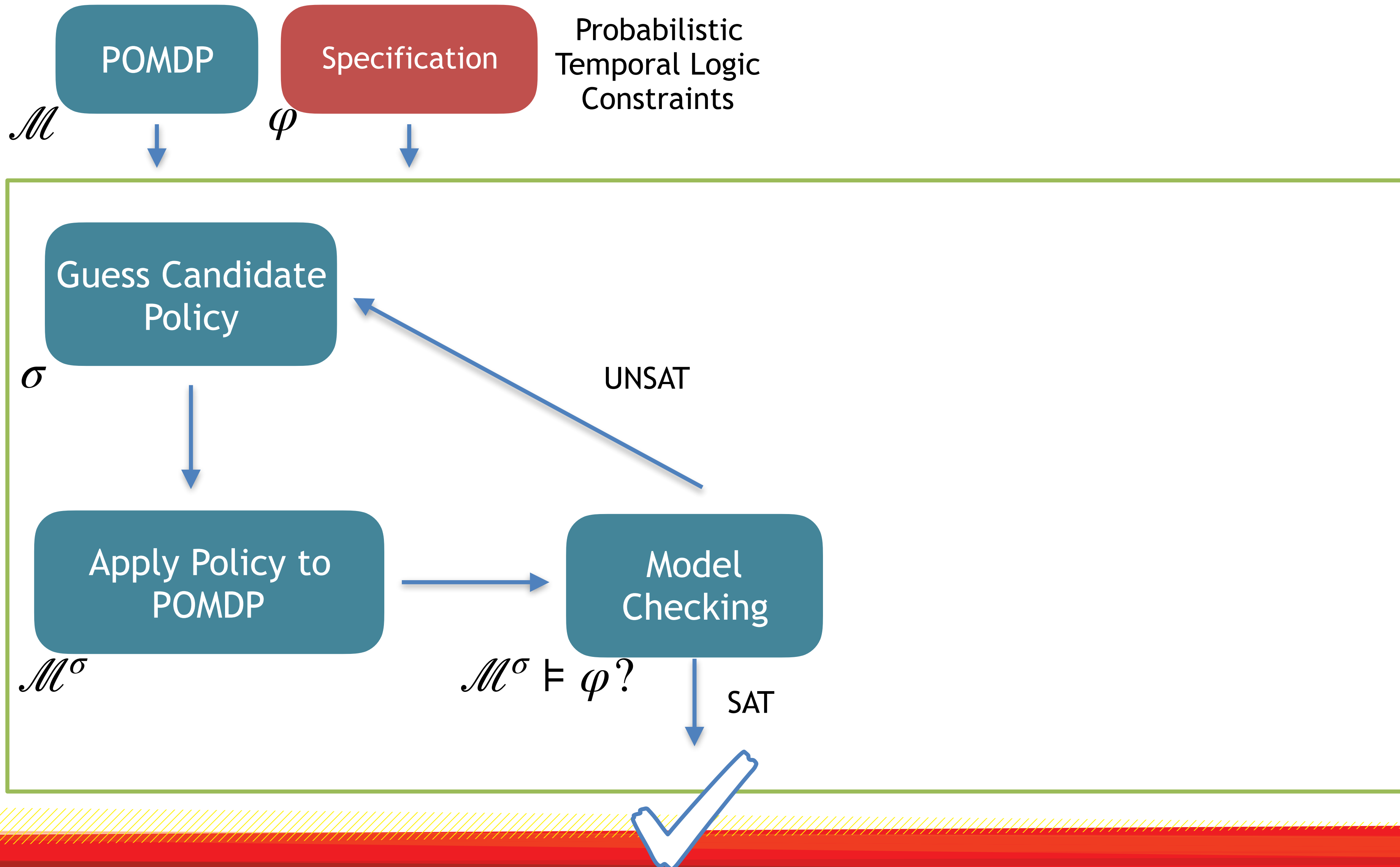
Guess Candidate Policy

σ

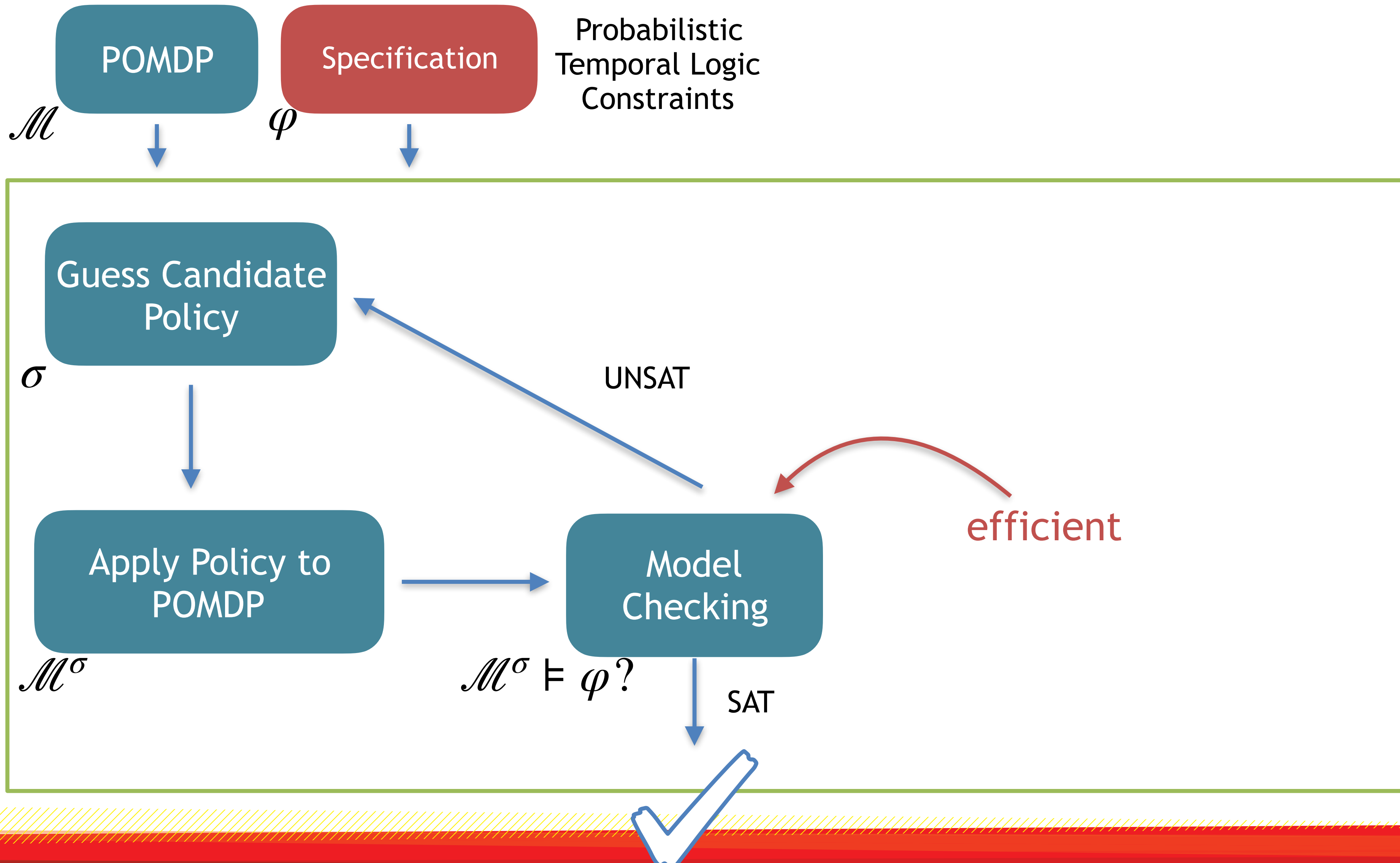
Apply Policy to POMDP

$\mathcal{M}^\sigma$

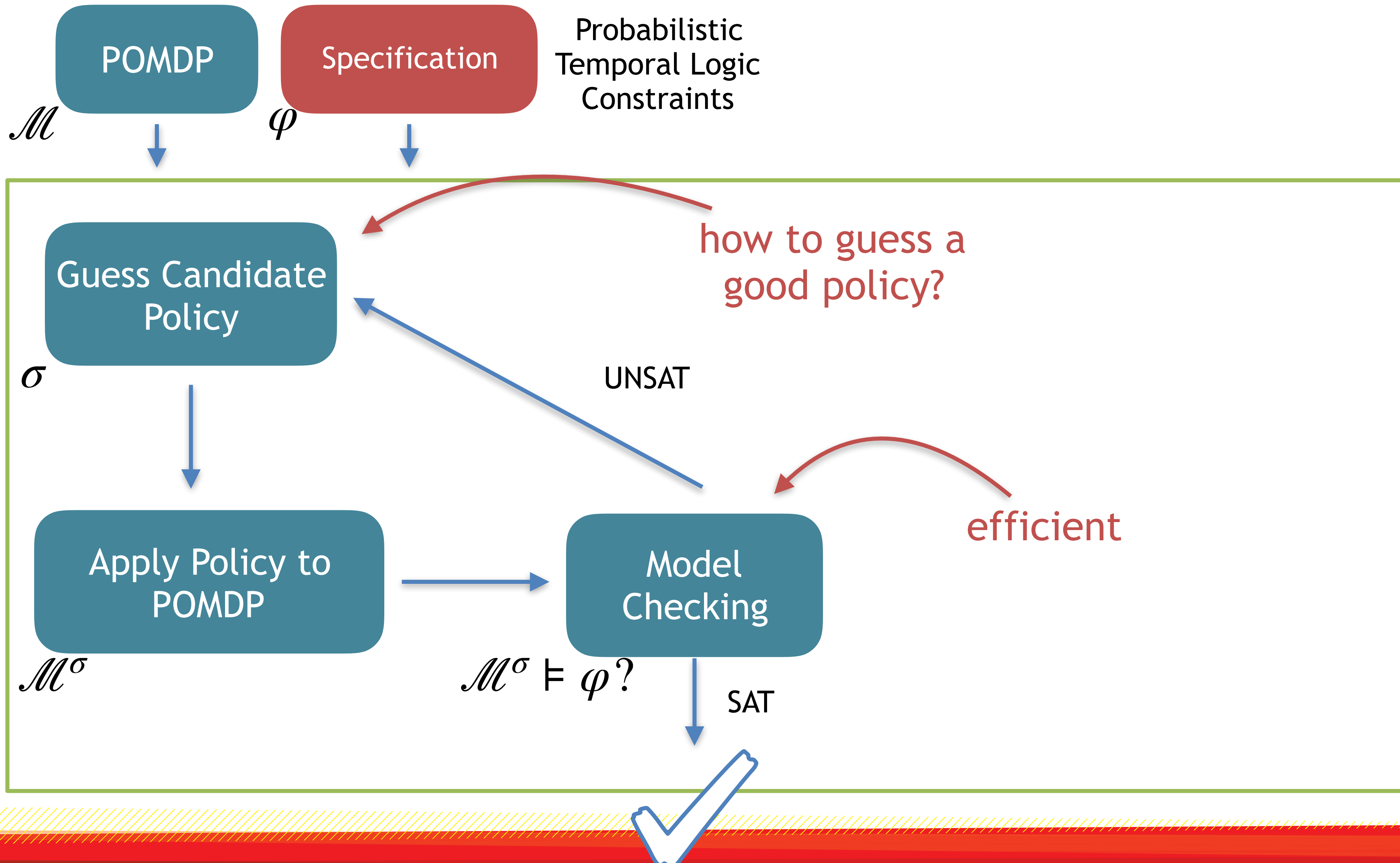
Be Lazy: Guess a Policy and Verify!



Be Lazy: Guess a Policy and Verify!



Be Lazy: Guess a Policy and Verify!

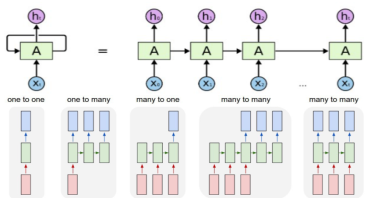


Let Machine Learning do the Guessing?

POMDP

Specification

$\mathscr{M}$ φ



σ policy network

UNSAT

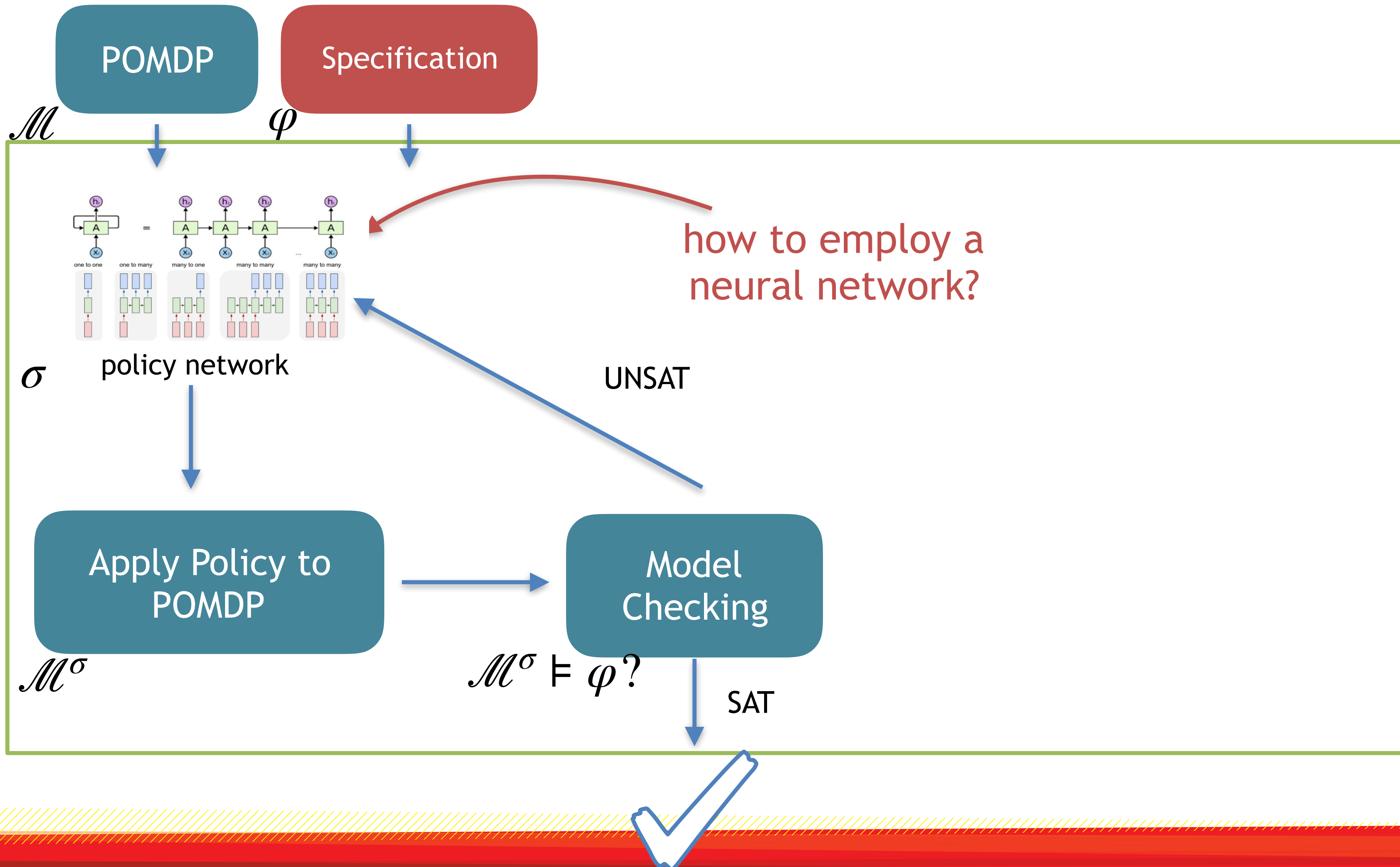
Apply Policy to POMDP

Model Checking

$\mathscr{M}^\sigma$ $\mathscr{M}^\sigma \models \varphi$?

SAT

Let Machine Learning do the Guessing?



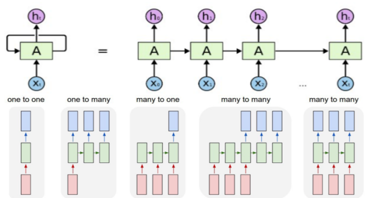
RNN Strategy Improvement

POMDP

Specification

$\mathscr{M}$

φ



σ strategy network

Apply Policy to POMDP

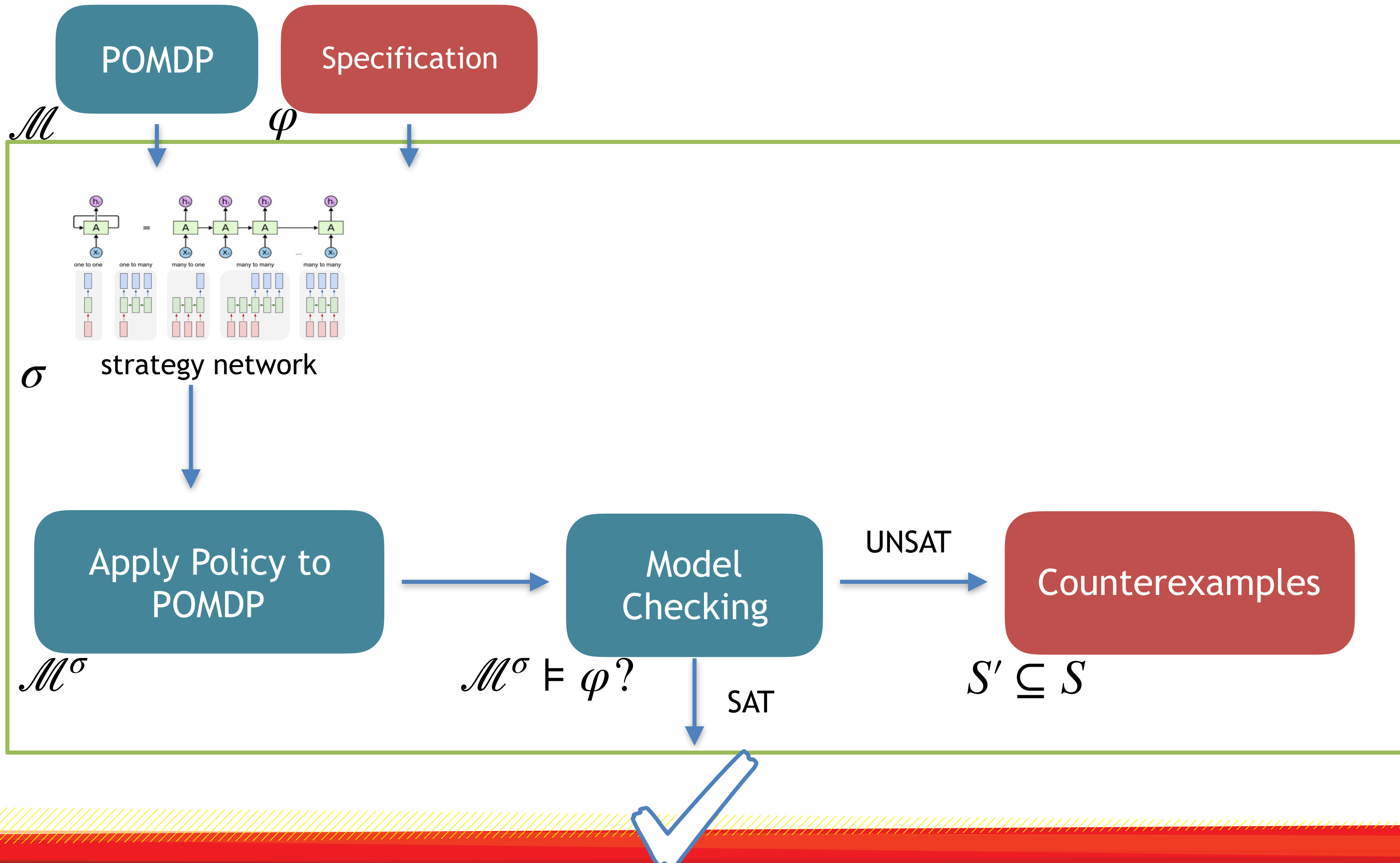
$\mathscr{M}^\sigma$

Model Checking

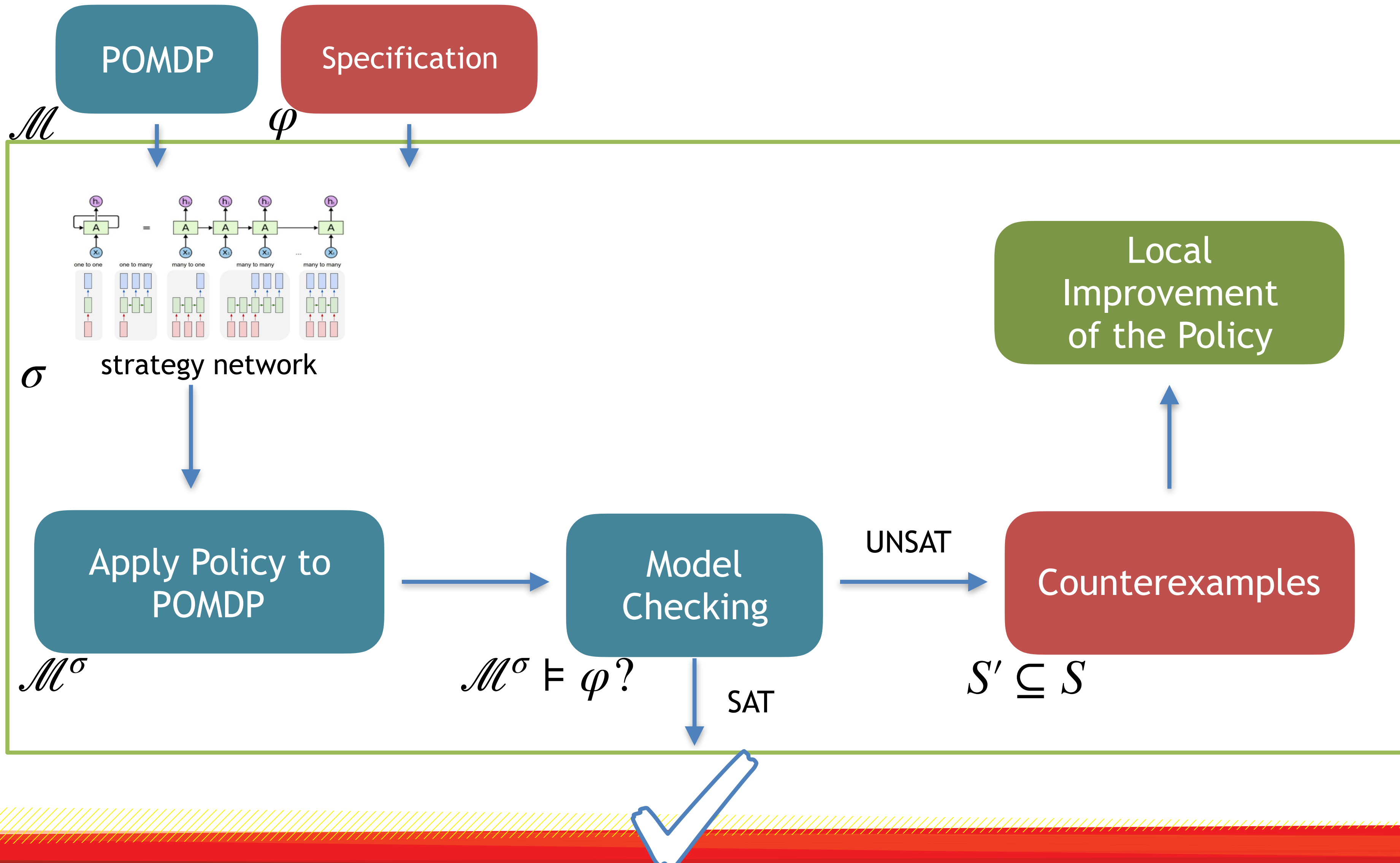
$\mathscr{M}^\sigma \models \varphi$?

SAT

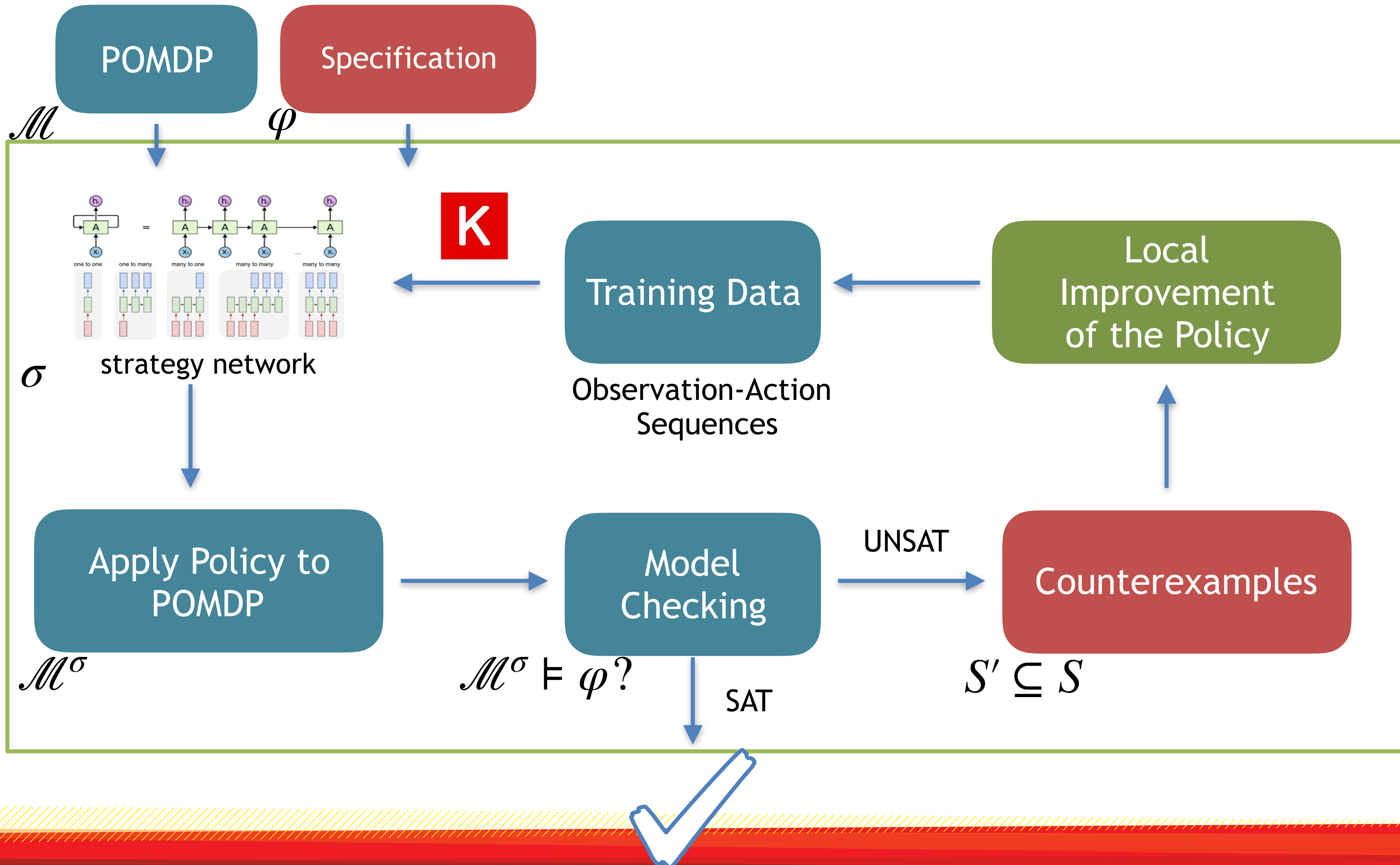
RNN Strategy Improvement



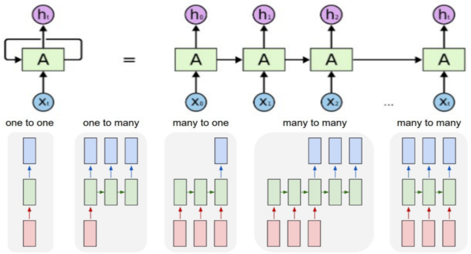
RNN Strategy Improvement



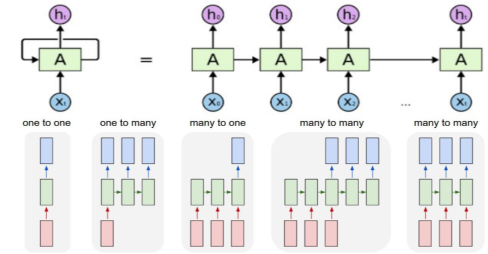
RNN Strategy Improvement



Learning Strategies with RNNs



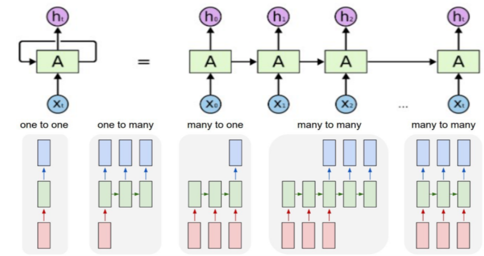
Learning Strategies with RNNs



Recurrent Neural Network

- long short-term memory (LSTM) architecture to learn dependencies in sequential data
- trained with observation-action sequences $ObsSeq_{fin}^{\mathcal{M}}$
- policy network $\sigma\colon ObsSeq_{fin}^{\mathcal{M}} \to Distr(Act)$

Learning Strategies with RNNs

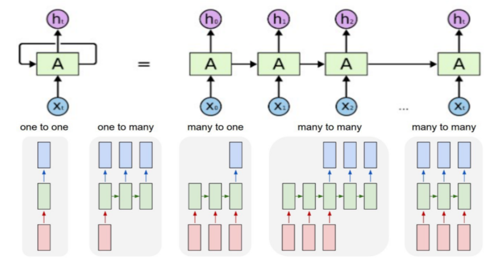


Recurrent Neural Network

- long short-term memory (LSTM) architecture to learn dependencies in sequential data
- trained with observation-action sequences $ObsSeq^{\mathscr{M}}_{fin}$
- policy network $\sigma\colon ObsSeq^{\mathscr{M}}_{fin} \rightarrow Distr(Act)$

predictor for a (memoryless) randomized policy

Learning Strategies with RNNs



Recurrent Neural Network

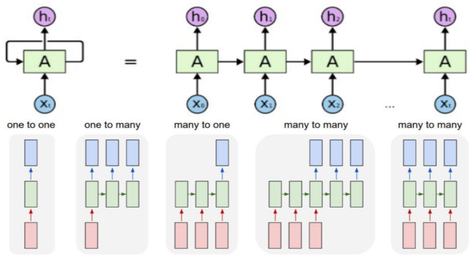
- long short-term memory (LSTM) architecture to learn dependencies in sequential data
- trained with observation-action sequences $ObsSeq^{\mathcal{M}}_{fin}$
- policy network $\sigma\colon ObsSeq^{\mathcal{M}}_{fin} \to Distr(Act)$

predictor for a (memoryless) randomized policy

Training

- Compute optimal MDP policy
- Generate (possible) observation-action sequences
- Observations are input labels, actions are output labels

Learning Strategies with RNNs



Recurrent Neural Network

- long short-term memory (LSTM) architecture to learn dependencies in sequential data
- trained with observation-action sequences $ObsSeq_{fin}^{\mathcal{M}}$
- policy network $\sigma\colon ObsSeq_{fin}^{\mathcal{M}} \to Distr(Act)$

predictor for a (memoryless) randomized policy

Training

- Compute optimal MDP policy
- Generate (possible) observation-action sequences
- Observations are input labels, actions are output labels

Large Environments

- Train on smaller environments that share observations and actions

Improving the Policy

Improving the Policy

Identify critical decisions $\sigma(z)(\alpha) > 0$ that lead to states with high probability of violating the specification.

Improving the Policy

Identify critical decisions $\sigma(z)(\alpha) > 0$ that lead to states with high probability of violating the specification.

For each observation $z \in (O)$ with critical decision, minimize the number of different critical actions.

Improving the Policy

Identify critical decisions $\sigma(z)(\alpha) > 0$ that lead to states with high probability of violating the specification.

For each observation $z \in (O)$ with critical decision, minimize the number of different critical actions.

Retrain with the new (locally improved) policy.

Improving the Policy

Identify critical decisions $\sigma(z)(\alpha) > 0$ that lead to states with high probability of violating the specification.

For each observation $z \in (O)$ with critical decision, minimize the number of different critical actions.

Retrain with the new (locally improved) policy.

Local linear program

$$\max_{\gamma(z)(a), a \in Act} \min_{s \in S} p_s \tag{1}$$

subject to

$$\forall s \in O^{-1}(z). \quad p_s = \sum_{a \in Act} \gamma(z)(a) \cdot \sum_{s' \in S} \mathscr{P}(s, a, s') \cdot p^*(s')$$

Improving the Policy

Identify critical decisions $\sigma(z)(\alpha) > 0$ that lead to states with high probability of violating the specification.

For each observation $z \in (O)$ with critical decision, minimize the number of different critical actions.

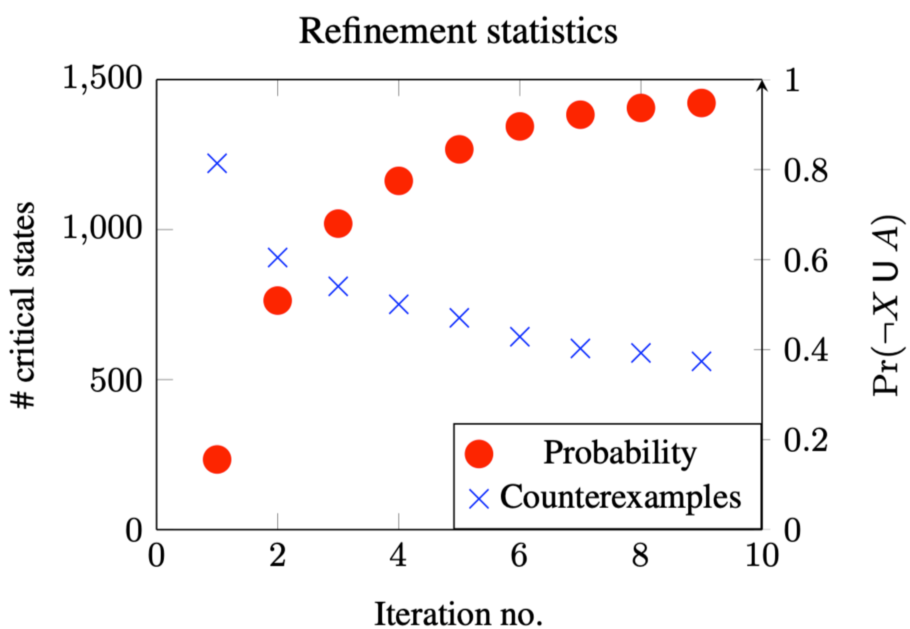
Retrain with the new (locally improved) policy.

Local linear program

$$\max_{\gamma(z)(a), a \in Act} \; \min_{s \in S} p_s \qquad (1)$$

subject to

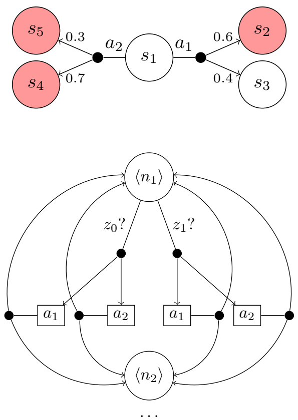
$$\forall s \in O^{-1}(z). \quad p_s = \sum_{a \in Act} \gamma(z)(a) \cdot \sum_{s' \in S} \mathscr{P}(s,a,s') \cdot p^*(s')$$

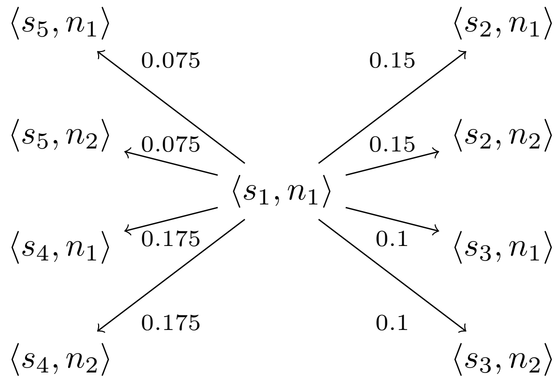


Even if specification is satisfied,
there may be critical states and decisions!

Finite-memory Strategies

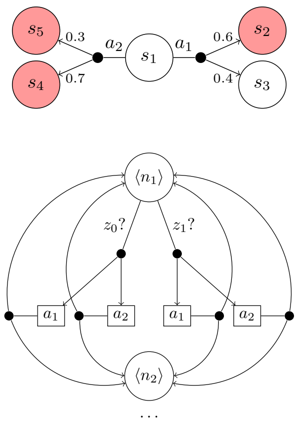
- Encode finite memory:

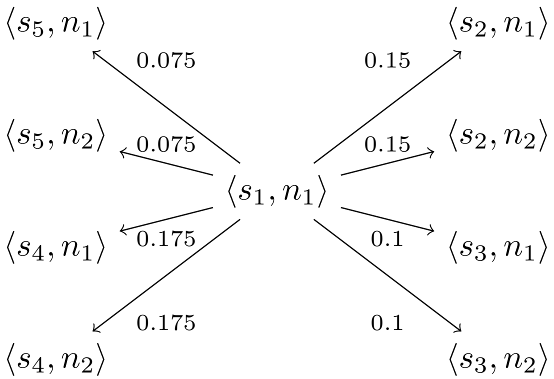




Finite-memory Strategies

- Encode finite memory:

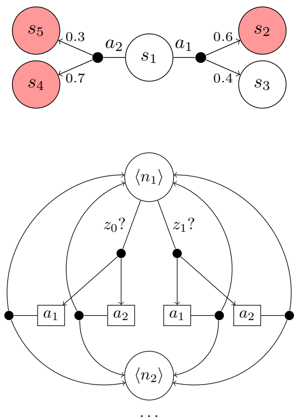


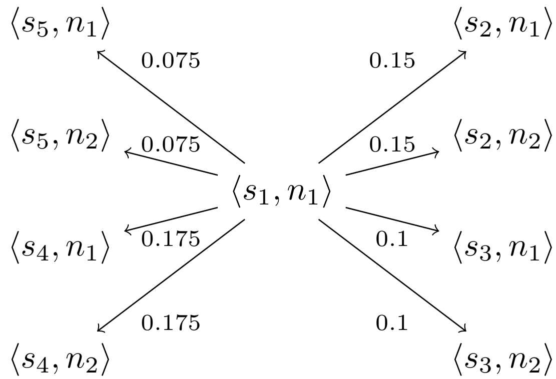


- Policy network is of the form $\sigma\colon ObsSeq_{fin}^{\mathscr{M}} \rightarrow Distr(Act)$

Finite-memory Strategies

- Encode finite memory:

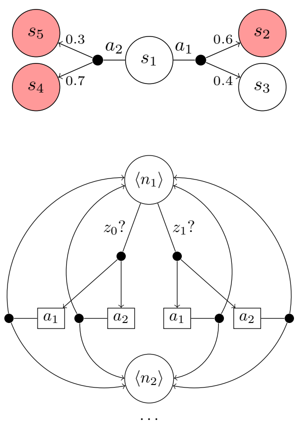


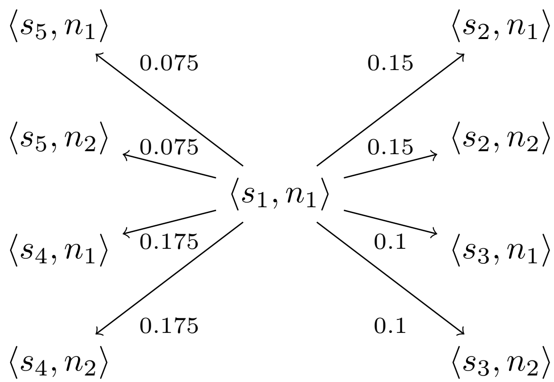


- Policy network is of the form $\sigma\colon ObsSeq_{fin}^{\mathscr{M}} \rightarrow Distr(Act)$
- But: How to infer a memory-update function to construct a finite-state controller?

Finite-memory Strategies

- Encode finite memory:

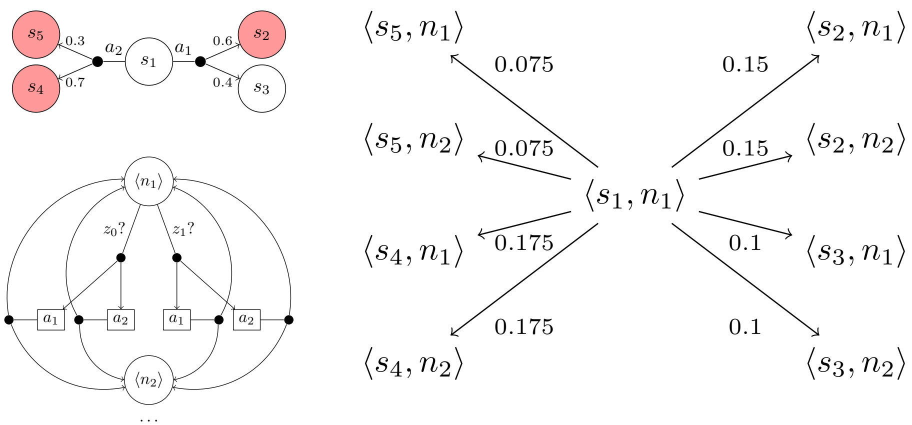




- Policy network is of the form $\sigma\colon ObsSeq_{fin}^{\mathscr{M}} \to Distr(Act)$
- But: How to infer a memory-update function to construct a finite-state controller?
- First solution: Predefine memory update, for instance (deterministic) transition upon repetition of an observation.

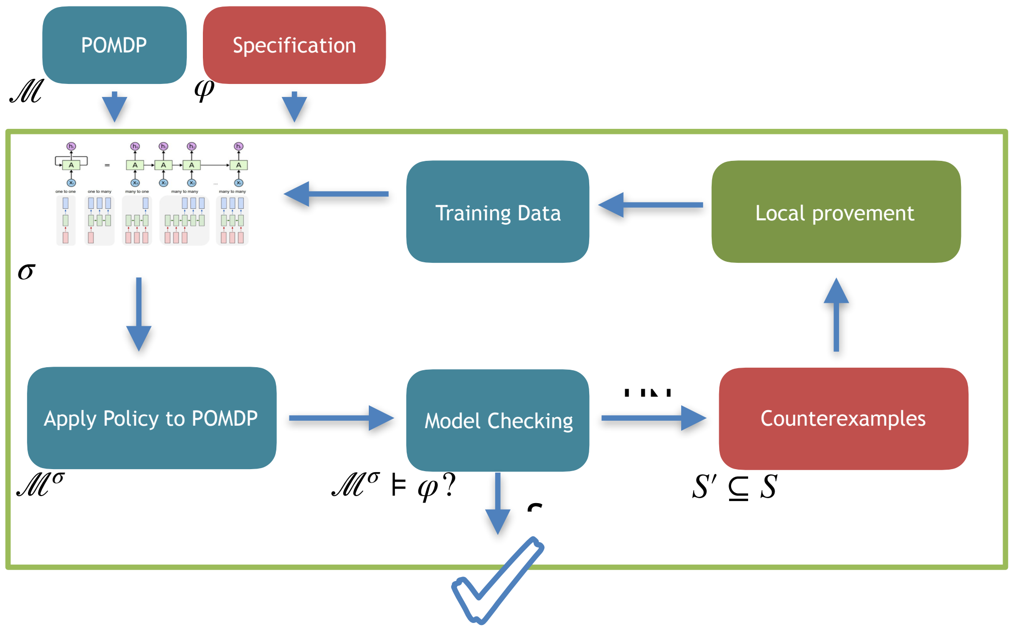
Finite-memory Strategies

- Encode finite memory:

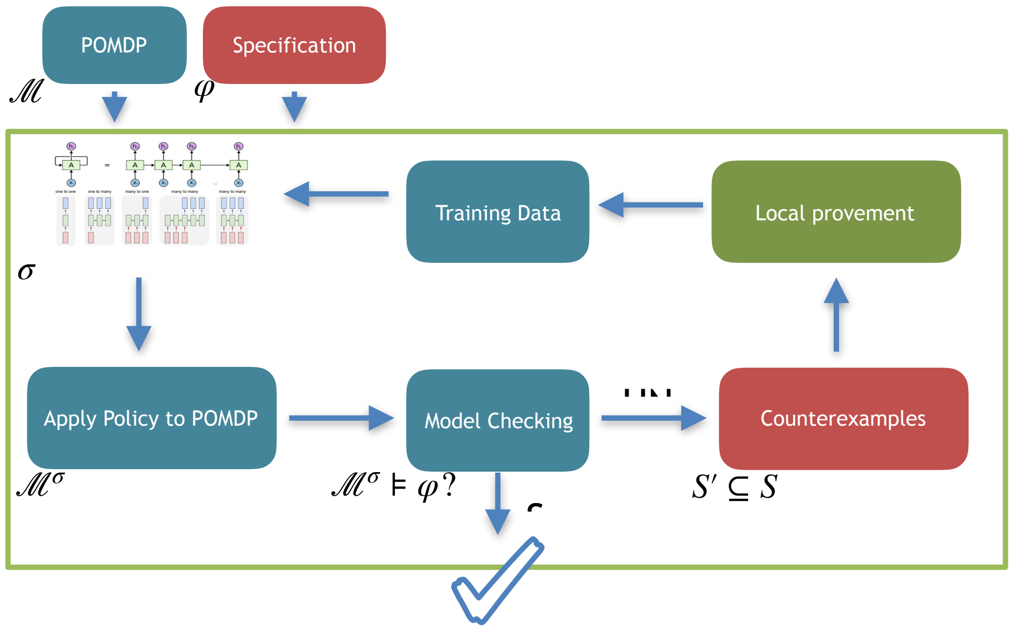


- Policy network is of the form $\sigma\colon ObsSeq^{\mathscr{M}}_{fin} \to Distr(Act)$
- But: How to infer a memory-update function to construct a finite-state controller?
- First solution: Predefine memory update, for instance (deterministic) transition upon repetition of an observation.
- Compute product of FSC and POMDP and compute memoryless policy as before.

Correctness and Completeness?

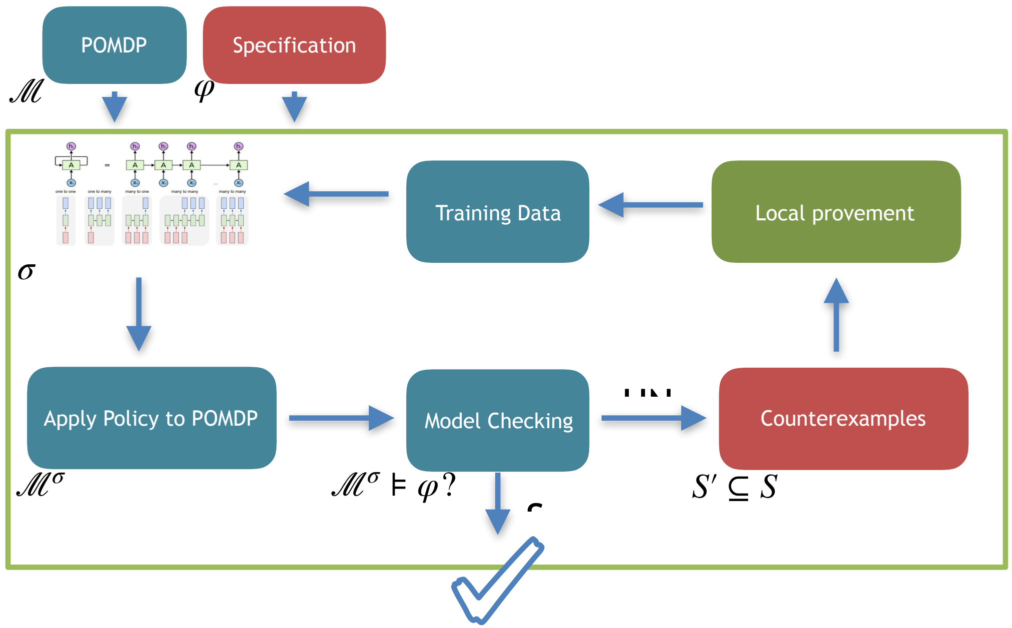


Correctness and Completeness?



Correct, as each policy prediction is evaluated using model checking.

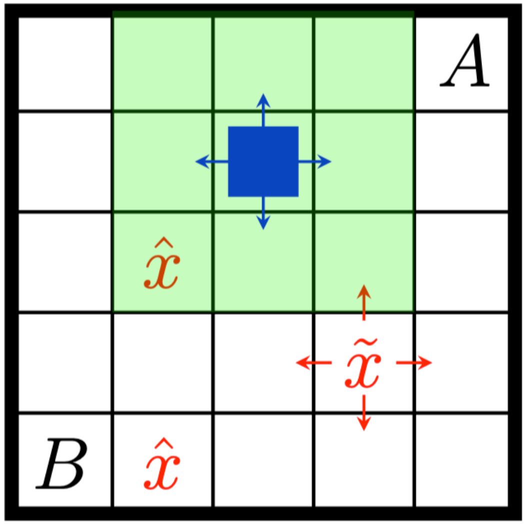
Correctness and Completeness?



Correct, as each policy prediction is evaluated using model checking.

Not complete, as we may never find a feasible policy. Also, problem is undecidable (or hard) anyways :).

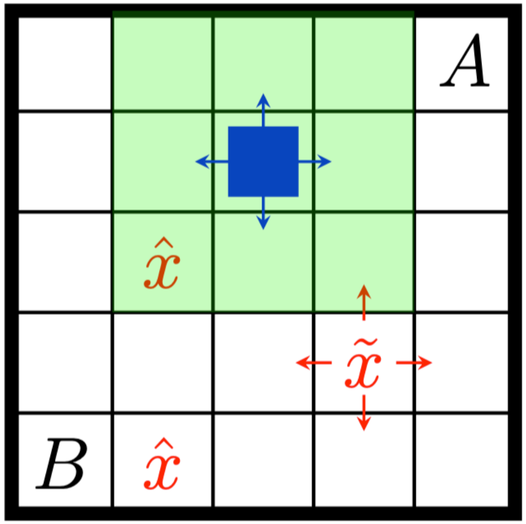
Experiments - LTL



Problem	$\|S\|$	$\|Act\|$	$\|Z\|$
Navigation (c)	c^4	4	256
Delivery (c)	c^2	4	256
Slippery (c)	c^2	4	256
Maze(c)	$3c+8$	4	7
Grid(c)	c^2	4	2
RockSample$[4,4]$	257	9	2
RockSample$[5,5]$	801	10	2
RockSample$[7,8]$	12545	13	2

			RNN-based Synthesis		PRISM-POMDP	
Problem	States	Type, φ	Res.	Time (s)	Res.	Time (s)
Navigation (3)	333	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.74	**14.16**	**0.84**	73.88
Navigation (4)	1088	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.82	**22.67**	**0.93**	1034.64
Navigation (4) [2-FSC]	13373	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.91	47.26	–	–
Navigation (4) [4-FSC]	26741	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.92	59.42	–	–
Navigation (4) [8-FSC]	53477	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.92**	85.26	–	–
Navigation (5)	2725	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.91	**34.34**	MO	MO
Navigation (5) [2-FSC]	33357	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.92	115.16	–	–
Navigation (5) [4-FSC]	66709	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.92	159.61	–	–
Navigation (5) [8-FSC]	133413	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.92**	250.91	–	–
Navigation (10)	49060	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.79	**822.87**	MO	MO
Navigation (10) [2-FSC]	475053	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.83	1185.41	–	–
Navigation (10) [4-FSC]	950101	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.85**	1488.77	–	–
Navigation (10) [8-FSC]	1900197	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.81	1805.22	–	–
Navigation (15)	251965	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.91**	**1271.80***	MO	MO
Navigation (20)	798040	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.96**	**4712.25***	MO	MO
Navigation (30)	4045840	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.95**	**25191.05***	MO	MO
Navigation (40)	–	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	TO	TO	MO	MO
Delivery (4) [2-FSC]	80	$\mathbb{E}^{\mathcal{M}}_{\min}, \varphi_2$	6.02	35.35	**6.0**	**28.53**
Delivery (5) [2-FSC]	125	$\mathbb{E}^{\mathcal{M}}_{\min}, \varphi_2$	8.11	**78.32**	**8.0**	102.41
Delivery (10) [2-FSC]	500	$\mathbb{E}^{\mathcal{M}}_{\min}, \varphi_2$	**18.13**	**120.34**	MO	MO
Slippery (4) [2-FSC]	460	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_3$	0.78	67.51	**0.90**	**5.10**
Slippery (5) [2-FSC]	730	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_3$	0.89	84.32	**0.93**	**83.24**
Slippery (10) [2-FSC]	2980	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_3$	**0.98**	**119.14**	MO	MO
Slippery (20) [2-FSC]	11980	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_3$	**0.99**	**1580.42**	MO	MO

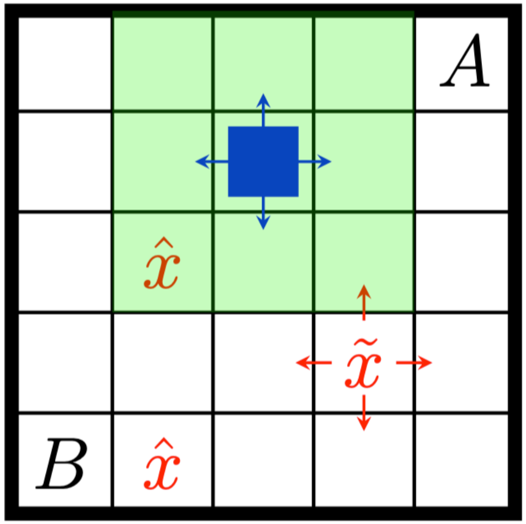
Experiments - LTL



Problem	$\lvert S\rvert$	$\lvert Act\rvert$	$\lvert Z\rvert$
Navigation (c)	c^4	4	256
Delivery (c)	c^2	4	256
Slippery (c)	c^2	4	256
Maze(c)	$3c+8$	4	7
Grid(c)	c^2	4	2
RockSample[4, 4]	257	9	2
RockSample[5, 5]	801	10	2
RockSample[7, 8]	12545	13	2

			RNN-based Synthesis		PRISM-POMDP	
Problem	States	Type, φ	Res.	Time (s)	Res.	Time (s)
Navigation (3)	333	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.74	**14.16**	**0.84**	73.88
Navigation (4)	1088	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.82	**22.67**	**0.93**	1034.64
Navigation (4) [2-FSC]	13373	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.91	47.26	–	–
Navigation (4) [4-FSC]	26741	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.92	59.42	–	–
Navigation (4) [8-FSC]	53477	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.92**	85.26	–	–
Navigation (5)	2725	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.91	**34.34**	MO	MO
Navigation (5) [2-FSC]	33357	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.92	115.16	–	–
Navigation (5) [4-FSC]	66709	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.92	159.61	–	–
Navigation (5) [8-FSC]	133413	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.92**	250.91	–	–
Navigation (10)	49060	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.79	**822.87**	MO	MO
Navigation (10) [2-FSC]	475053	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.83	1185.41	–	–
Navigation (10) [4-FSC]	950101	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.85**	1488.77	–	–
Navigation (10) [8-FSC]	1900197	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.81	1805.22	–	–
Navigation (15)	251965	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.91**	**1271.80***	MO	MO
Navigation (20)	798040	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.96**	**4712.25***	MO	MO
Navigation (30)	4045840	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.95**	**25191.05***	MO	MO
Navigation (40)	–	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	TO	TO	MO	MO
Delivery (4) [2-FSC]	80	$\mathbb{E}^{\mathcal{M}}_{\min}, \varphi_2$	6.02	35.35	**6.0**	**28.53**
Delivery (5) [2-FSC]	125	$\mathbb{E}^{\mathcal{M}}_{\min}, \varphi_2$	8.11	**78.32**	**8.0**	102.41
Delivery (10) [2-FSC]	500	$\mathbb{E}^{\mathcal{M}}_{\min}, \varphi_2$	**18.13**	**120.34**	MO	MO
Slippery (4) [2-FSC]	460	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_3$	0.78	67.51	**0.90**	**5.10**
Slippery (5) [2-FSC]	730	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_3$	0.89	84.32	**0.93**	**83.24**
Slippery (10) [2-FSC]	2980	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_3$	**0.98**	**119.14**	MO	MO
Slippery (20) [2-FSC]	11980	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_3$	**0.99**	**1580.42**	MO	MO

Experiments - LTL



Problem	$\|S\|$	$\|Act\|$	$\|Z\|$
Navigation (c)	c^4	4	256
Delivery (c)	c^2	4	256
Slippery (c)	c^2	4	256
Maze(c)	$3c+8$	4	7
Grid(c)	c^2	4	2
RockSample$[4,4]$	257	9	2
RockSample$[5,5]$	801	10	2
RockSample$[7,8]$	12545	13	2

			RNN-based Synthesis		PRISM-POMDP	
Problem	States	Type, φ	Res.	Time (s)	Res.	Time (s)
Navigation (3)	333	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.74	**14.16**	**0.84**	73.88
Navigation (4)	1088	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.82	**22.67**	**0.93**	1034.64
Navigation (4) [2-FSC]	13373	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.91	47.26	–	–
Navigation (4) [4-FSC]	26741	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.92	59.42	–	–
Navigation (4) [8-FSC]	53477	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.92**	85.26	–	–
Navigation (5)	2725	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.91	**34.34**	MO	MO
Navigation (5) [2-FSC]	33357	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.92	115.16	–	–
Navigation (5) [4-FSC]	66709	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.92	159.61	–	–
Navigation (5) [8-FSC]	133413	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.92**	250.91	–	–
Navigation (10)	49060	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.79	**822.87**	MO	MO
Navigation (10) [2-FSC]	475053	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.83	1185.41	–	–
Navigation (10) [4-FSC]	950101	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.85**	1488.77	–	–
Navigation (10) [8-FSC]	1900197	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	0.81	1805.22	–	–
Navigation (15)	251965	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.91**	**1271.80***	MO	MO
Navigation (20)	798040	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.96**	**4712.25***	MO	MO
Navigation (30)	4045840	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	**0.95**	**25191.05***	MO	MO
Navigation (40)	–	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_1$	TO	TO	MO	MO
Delivery (4) [2-FSC]	80	$\mathbb{E}^{\mathcal{M}}_{\min}, \varphi_2$	6.02	35.35	**6.0**	**28.53**
Delivery (5) [2-FSC]	125	$\mathbb{E}^{\mathcal{M}}_{\min}, \varphi_2$	8.11	**78.32**	**8.0**	102.41
Delivery (10) [2-FSC]	500	$\mathbb{E}^{\mathcal{M}}_{\min}, \varphi_2$	**18.13**	**120.34**	MO	MO
Slippery (4) [2-FSC]	460	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_3$	0.78	67.51	**0.90**	**5.10**
Slippery (5) [2-FSC]	730	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_3$	0.89	84.32	**0.93**	**83.24**
Slippery (10) [2-FSC]	2980	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_3$	**0.98**	**119.14**	MO	MO
Slippery (20) [2-FSC]	11980	$\mathbb{P}^{\mathcal{M}}_{\max}, \varphi_3$	**0.99**	**1580.42**	MO	MO

Experiments - Standard POMDPs

Problem	Type	RNN-based Synthesis States	RNN-based Synthesis Res	RNN-based Synthesis Time (s)	PRISM-POMDP Res	PRISM-POMDP Time (s)	pomdpSolve Res	pomdpSolve Time (s)
Maze (1)	$\mathbb{E}^{\mathcal{M}}_{\mathrm{min}}$	68	4.31	31.70	**4.30**	**0.09**	4.30	0.30
Maze (2)	$\mathbb{E}^{\mathcal{M}}_{\mathrm{min}}$	83	5.31	46.65	5.23	2.176	**5.23**	**0.67**
Maze (3)	$\mathbb{E}^{\mathcal{M}}_{\mathrm{min}}$	98	8.10	58.75	7.13	38.82	**7.13**	**2.39**
Maze (4)	$\mathbb{E}^{\mathcal{M}}_{\mathrm{min}}$	113	11.53	58.09	8.58	543.06	**8.58**	**7.15**
Maze (5)	$\mathbb{E}^{\mathcal{M}}_{\mathrm{min}}$	128	14.40	**68.09**	13.00	4110.50	**12.04**	132.12
Maze (6)	$\mathbb{E}^{\mathcal{M}}_{\mathrm{min}}$	143	22.34	**71.89**	MO	MO	**18.52**	1546.02
Maze (10)	$\mathbb{E}^{\mathcal{M}}_{\mathrm{min}}$	203	100.21	**158.33**	MO	MO	MO	MO
Grid (3)	$\mathbb{E}^{\mathcal{M}}_{\mathrm{min}}$	165	2.90	38.94	2.88	2.332	**2.88**	**0.07**
Grid (4)	$\mathbb{E}^{\mathcal{M}}_{\mathrm{min}}$	381	4.32	79.99	4.13	1032.53	**4.13**	**0.77**
Grid (5)	$\mathbb{E}^{\mathcal{M}}_{\mathrm{min}}$	727	6.623	91.42	MO	MO	**5.42**	**1.94**
Grid (10)	$\mathbb{E}^{\mathcal{M}}_{\mathrm{min}}$	5457	**13.630**	**268.40**	MO	MO	MO	MO
RockSample[4, 4]	$\mathbb{E}^{\mathcal{M}}_{\mathrm{max}}$	2432	17.71	35.35	N/A	N/A	**18.04**	**0.43**
RockSample[5, 5]	$\mathbb{E}^{\mathcal{M}}_{\mathrm{max}}$	8320	18.40	**43.74**	N/A	N/A	**19.23**	621.28
RockSample[7, 8]	$\mathbb{E}^{\mathcal{M}}_{\mathrm{max}}$	166656	20.32	**860.53**	N/A	N/A	**21.64**	20458.41

Experiments - Standard POMDPs

Problem	Type	RNN-based Synthesis States	RNN-based Synthesis Res	RNN-based Synthesis Time (s)	PRISM-POMDP Res	PRISM-POMDP Time (s)	pomdpSolve Res	pomdpSolve Time (s)
Maze (1)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	68	4.31	31.70	**4.30**	**0.09**	4.30	0.30
Maze (2)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	83	5.31	46.65	5.23	2.176	**5.23**	**0.67**
Maze (3)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	98	8.10	58.75	7.13	38.82	**7.13**	**2.39**
Maze (4)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	113	11.53	58.09	8.58	543.06	**8.58**	**7.15**
Maze (5)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	128	14.40	**68.09**	13.00	4110.50	**12.04**	132.12
Maze (6)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	143	22.34	**71.89**	MO	MO	**18.52**	1546.02
Maze (10)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	203	100.21	**158.33**	MO	MO	MO	MO
Grid (3)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	165	2.90	38.94	2.88	2.332	**2.88**	**0.07**
Grid (4)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	381	4.32	79.99	4.13	1032.53	**4.13**	**0.77**
Grid (5)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	727	6.623	91.42	MO	MO	**5.42**	**1.94**
Grid (10)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	5457	**13.630**	**268.40**	MO	MO	MO	MO
RockSample[4, 4]	$\mathbb{E}^{\mathcal{M}}_{\mathtt{max}}$	2432	17.71	35.35	N/A	N/A	**18.04**	**0.43**
RockSample[5, 5]	$\mathbb{E}^{\mathcal{M}}_{\mathtt{max}}$	8320	18.40	**43.74**	N/A	N/A	**19.23**	621.28
RockSample[7, 8]	$\mathbb{E}^{\mathcal{M}}_{\mathtt{max}}$	166656	20.32	**860.53**	N/A	N/A	**21.64**	20458.41

Experiments - Standard POMDPs

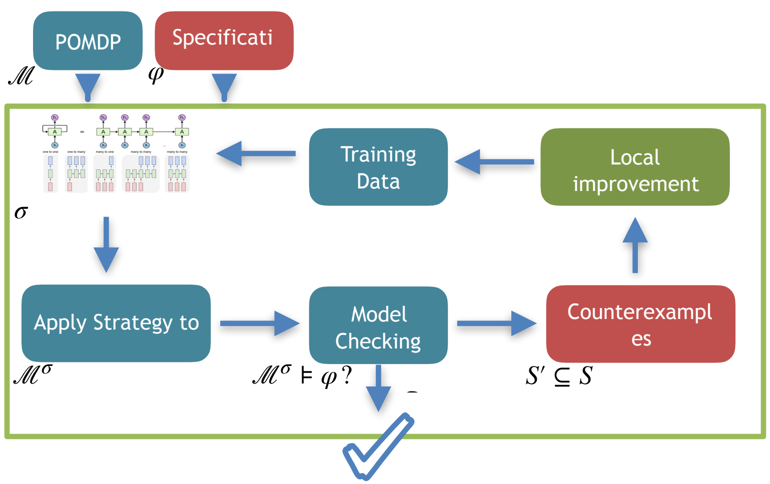
Problem	Type	RNN-based Synthesis States	RNN-based Synthesis Res	RNN-based Synthesis Time (s)	PRISM-POMDP Res	PRISM-POMDP Time (s)	pomdpSolve Res	pomdpSolve Time (s)
Maze (1)	$\mathbb{E}^{\mathcal{M}}_{\min}$	68	4.31	31.70	**4.30**	**0.09**	4.30	0.30
Maze (2)	$\mathbb{E}^{\mathcal{M}}_{\min}$	83	5.31	46.65	5.23	2.176	**5.23**	**0.67**
Maze (3)	$\mathbb{E}^{\mathcal{M}}_{\min}$	98	8.10	58.75	7.13	38.82	**7.13**	**2.39**
Maze (4)	$\mathbb{E}^{\mathcal{M}}_{\min}$	113	11.53	58.09	8.58	543.06	**8.58**	**7.15**
Maze (5)	$\mathbb{E}^{\mathcal{M}}_{\min}$	128	14.40	**68.09**	13.00	4110.50	**12.04**	132.12
Maze (6)	$\mathbb{E}^{\mathcal{M}}_{\min}$	143	22.34	**71.89**	MO	MO	**18.52**	1546.02
Maze (10)	$\mathbb{E}^{\mathcal{M}}_{\min}$	203	100.21	**158.33**	MO	MO	MO	MO
Grid (3)	$\mathbb{E}^{\mathcal{M}}_{\min}$	165	2.90	38.94	2.88	2.332	**2.88**	**0.07**
Grid (4)	$\mathbb{E}^{\mathcal{M}}_{\min}$	381	4.32	79.99	4.13	1032.53	**4.13**	**0.77**
Grid (5)	$\mathbb{E}^{\mathcal{M}}_{\min}$	727	6.623	91.42	MO	MO	**5.42**	**1.94**
Grid (10)	$\mathbb{E}^{\mathcal{M}}_{\min}$	5457	**13.630**	**268.40**	MO	MO	MO	MO
RockSample[4, 4]	$\mathbb{E}^{\mathcal{M}}_{\max}$	2432	17.71	35.35	N/A	N/A	**18.04**	**0.43**
RockSample[5, 5]	$\mathbb{E}^{\mathcal{M}}_{\max}$	8320	18.40	**43.74**	N/A	N/A	**19.23**	621.28
RockSample[7, 8]	$\mathbb{E}^{\mathcal{M}}_{\max}$	166656	20.32	**860.53**	N/A	N/A	**21.64**	20458.41

Experiments - Standard POMDPs

Problem	Type	RNN-based Synthesis States	RNN-based Synthesis Res	RNN-based Synthesis Time (s)	PRISM-POMDP Res	PRISM-POMDP Time (s)	pomdpSolve Res	pomdpSolve Time (s)
Maze (1)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	68	4.31	31.70	**4.30**	**0.09**	4.30	0.30
Maze (2)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	83	5.31	46.65	5.23	2.176	**5.23**	**0.67**
Maze (3)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	98	8.10	58.75	7.13	38.82	**7.13**	**2.39**
Maze (4)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	113	11.53	58.09	8.58	543.06	**8.58**	**7.15**
Maze (5)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	128	14.40	**68.09**	13.00	4110.50	**12.04**	132.12
Maze (6)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	143	22.34	**71.89**	MO	MO	**18.52**	1546.02
Maze (10)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	203	100.21	**158.33**	MO	MO	MO	MO
Grid (3)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	165	2.90	38.94	2.88	2.332	**2.88**	**0.07**
Grid (4)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	381	4.32	79.99	4.13	1032.53	**4.13**	**0.77**
Grid (5)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	727	6.623	91.42	MO	MO	**5.42**	**1.94**
Grid (10)	$\mathbb{E}^{\mathcal{M}}_{\mathtt{min}}$	5457	**13.630**	**268.40**	MO	MO	MO	MO
RockSample[4, 4]	$\mathbb{E}^{\mathcal{M}}_{\mathtt{max}}$	2432	17.71	35.35	N/A	N/A	**18.04**	**0.43**
RockSample[5, 5]	$\mathbb{E}^{\mathcal{M}}_{\mathtt{max}}$	8320	18.40	**43.74**	N/A	N/A	**19.23**	621.28
RockSample[7, 8]	$\mathbb{E}^{\mathcal{M}}_{\mathtt{max}}$	166656	20.32	**860.53**	N/A	N/A	**21.64**	20458.41

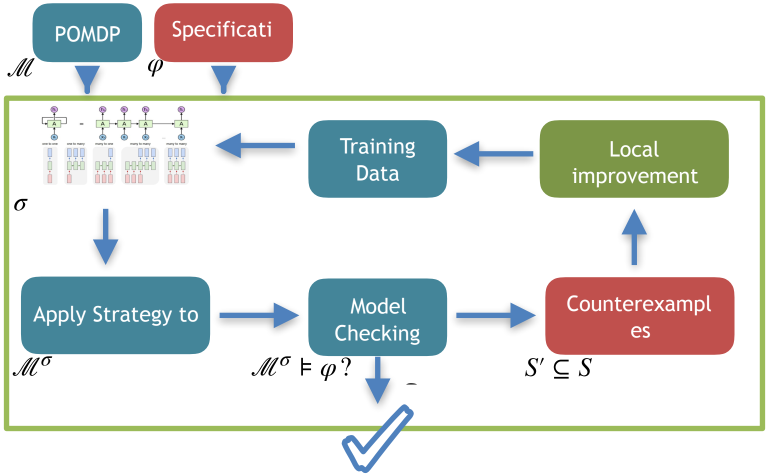
Conclusion Story 3

- Novel way to generate provably correct POMDP policies
- Good scalability, not optimal
- Results transferrable
- Future work: More principled approach to finite-memory strategies



Conclusion Story 3

- Novel way to generate provably correct POMDP policies
- Good scalability, not optimal
- Results transferrable
- Future work: More principled approach to finite-memory strategies

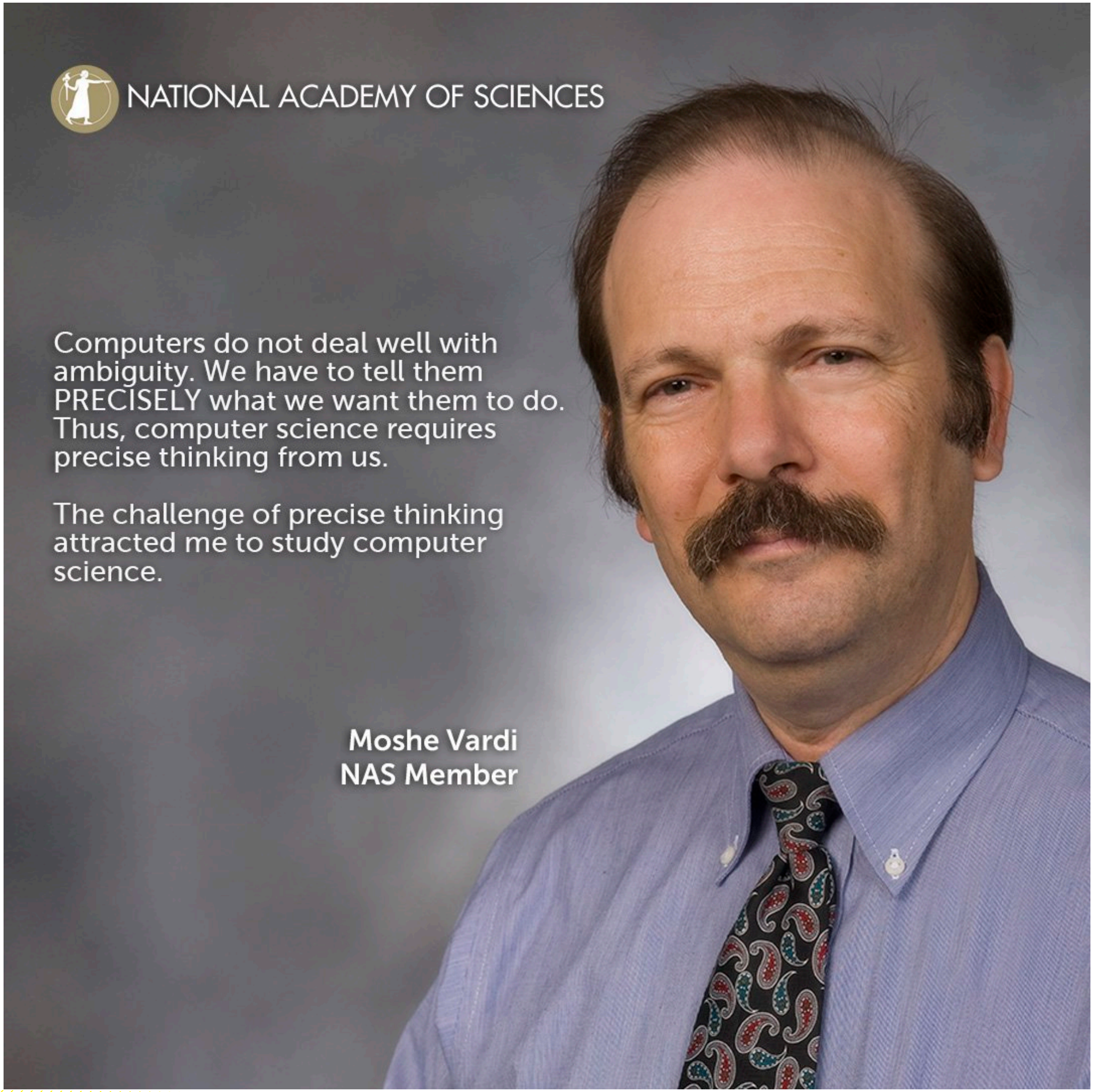


Steven Carr, Nils Jansen, Ralf Wimmer, Alexandru Constantin Serban, Bernd Becker, Ufuk Topcu: Counterexample-Guided Strategy Improvement for POMDPs Using Recurrent Neural Networks. IJCAI (2019)

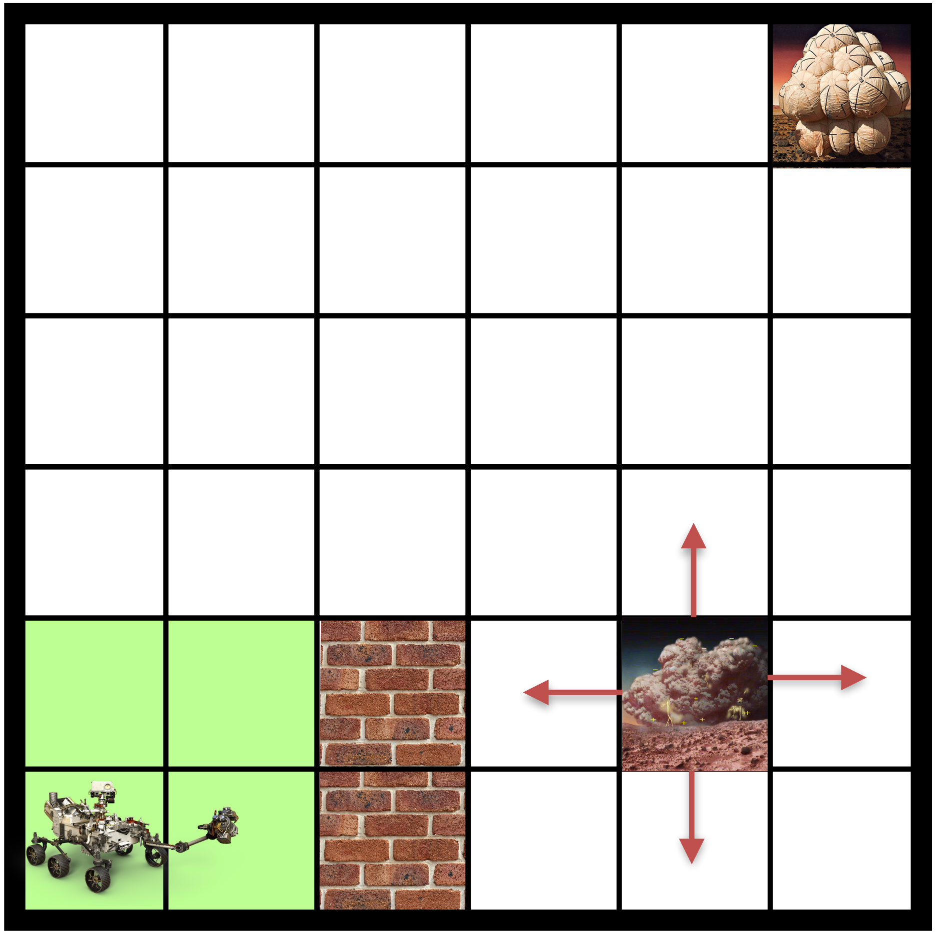
Steven Carr, Nils Jansen, Ralf Wimmer, Alexandru Constantin Serban, Bernd Becker, Ufuk Topcu: Counterexample-Guided Strategy Improvement for POMDPs Using Recurrent Neural Networks. CoRR abs/1903.08428 (2019)

Stories - Policy Synthesis for POMDPs

1. Game-based abstraction
2. Finite-memory controllers
3. Recurrent neural networks
4. Fun: Humans in the loop

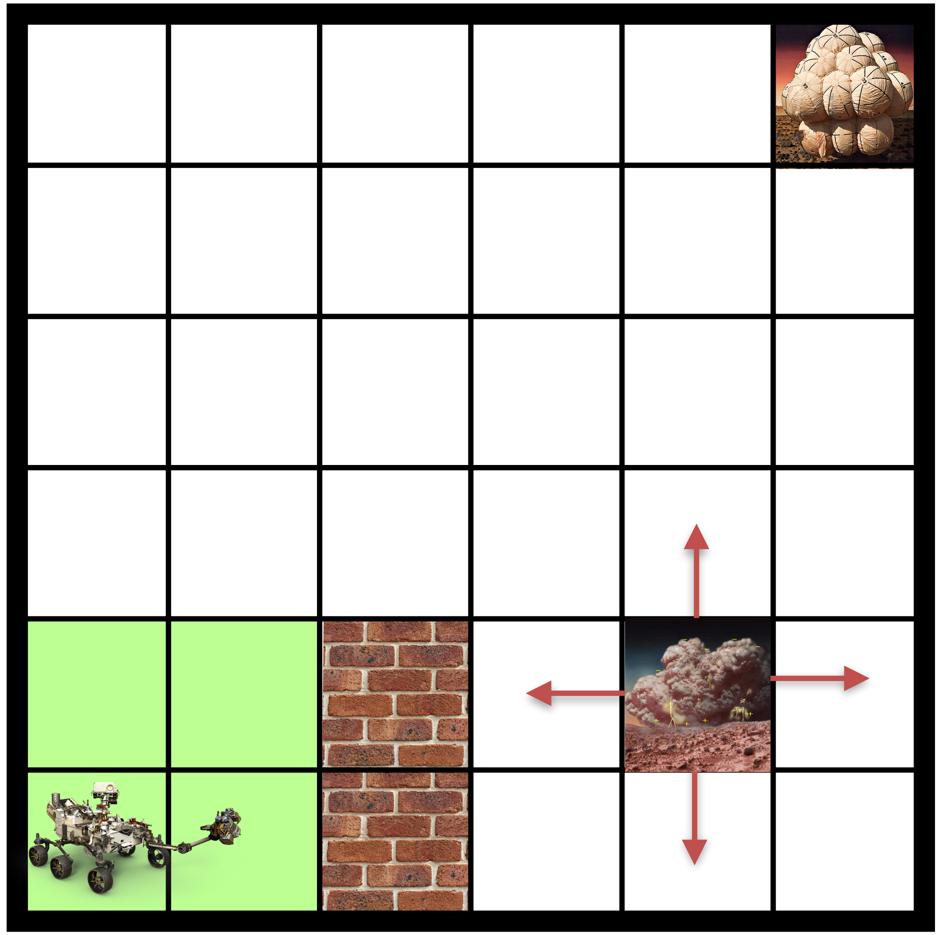

NATIONAL ACADEMY OF SCIENCES
Computers do not deal well with ambiguity. We have to tell them PRECISELY what we want them to do. Thus, computer science requires precise thinking from us.
The challenge of precise thinking attracted me to study computer science.
Moshe Vardi
NAS Member

Idea: Human-in-the-loop Synthesis for POMDPs



Idea: Human-in-the-loop Synthesis for POMDPs

Turn scenario into an arcade game

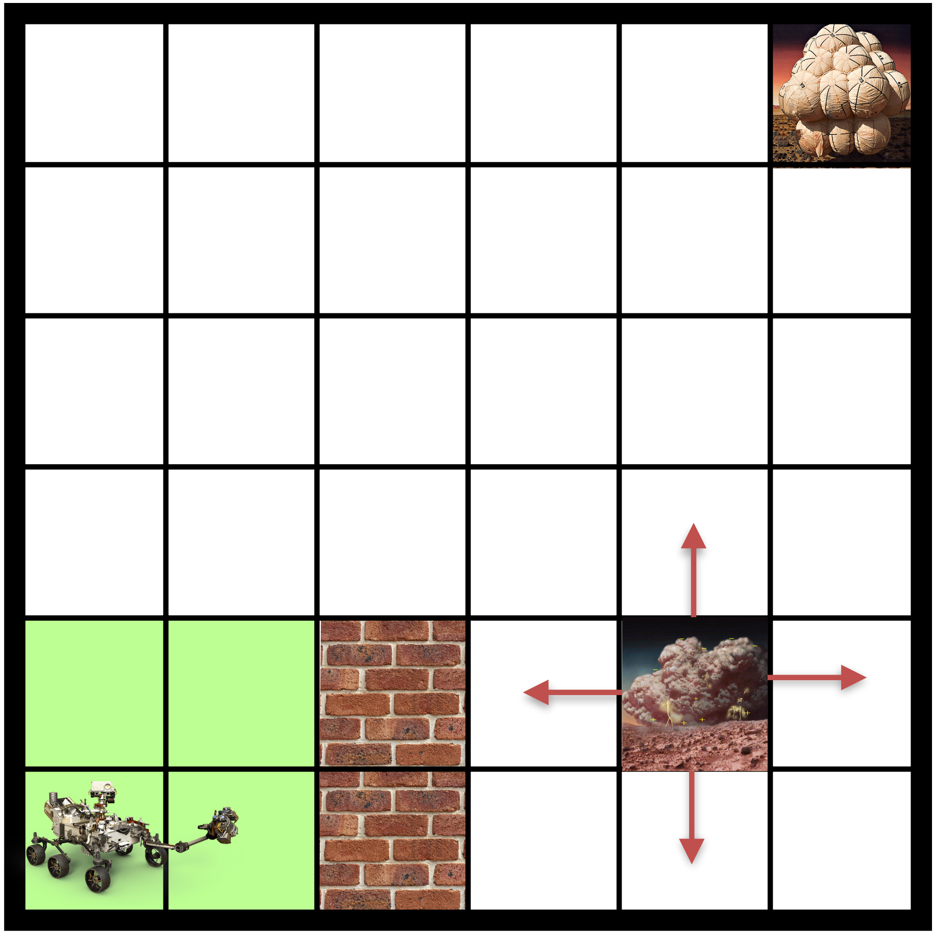


Underlying (family of) POMDPs

Idea: Human-in-the-loop Synthesis for POMDPs

Turn scenario into an arcade game

Collect data of human playing



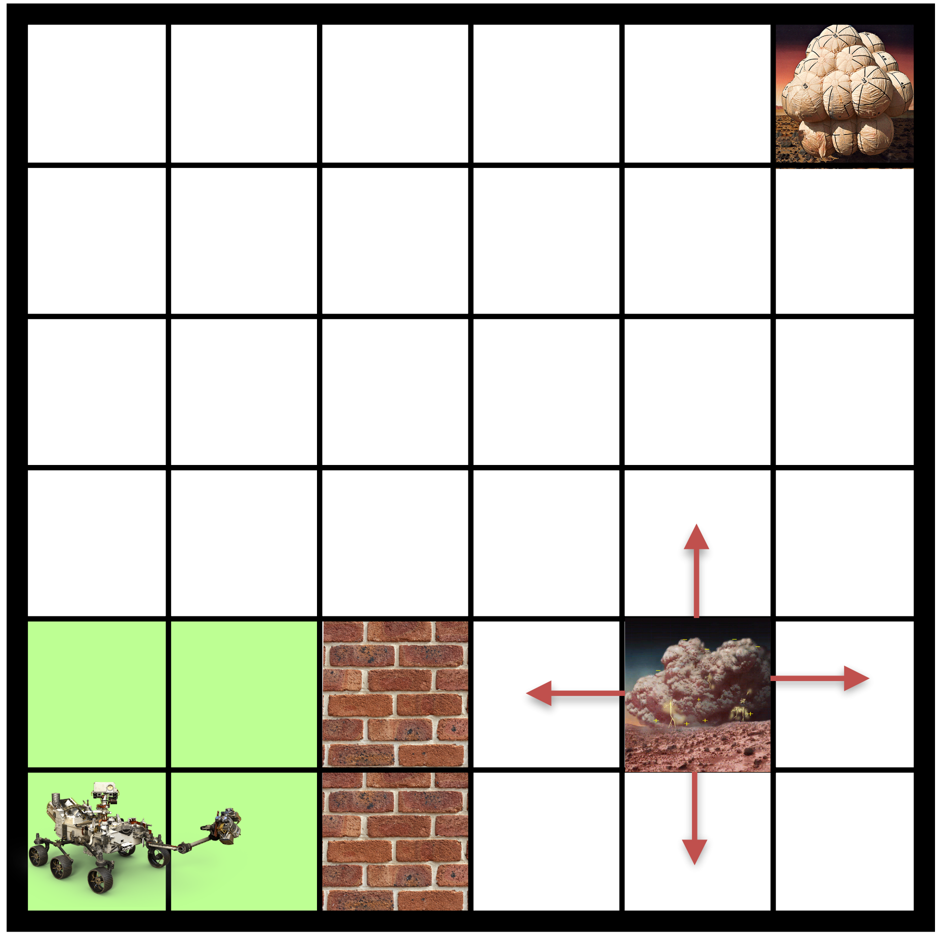
Underlying (family of) POMDPs

Idea: Human-in-the-loop Synthesis for POMDPs

Turn scenario into an arcade game

Collect data of human playing

From data, infer a strategy



Underlying (family of) POMDPs

Applying strategy yields DTMC, efficient verification

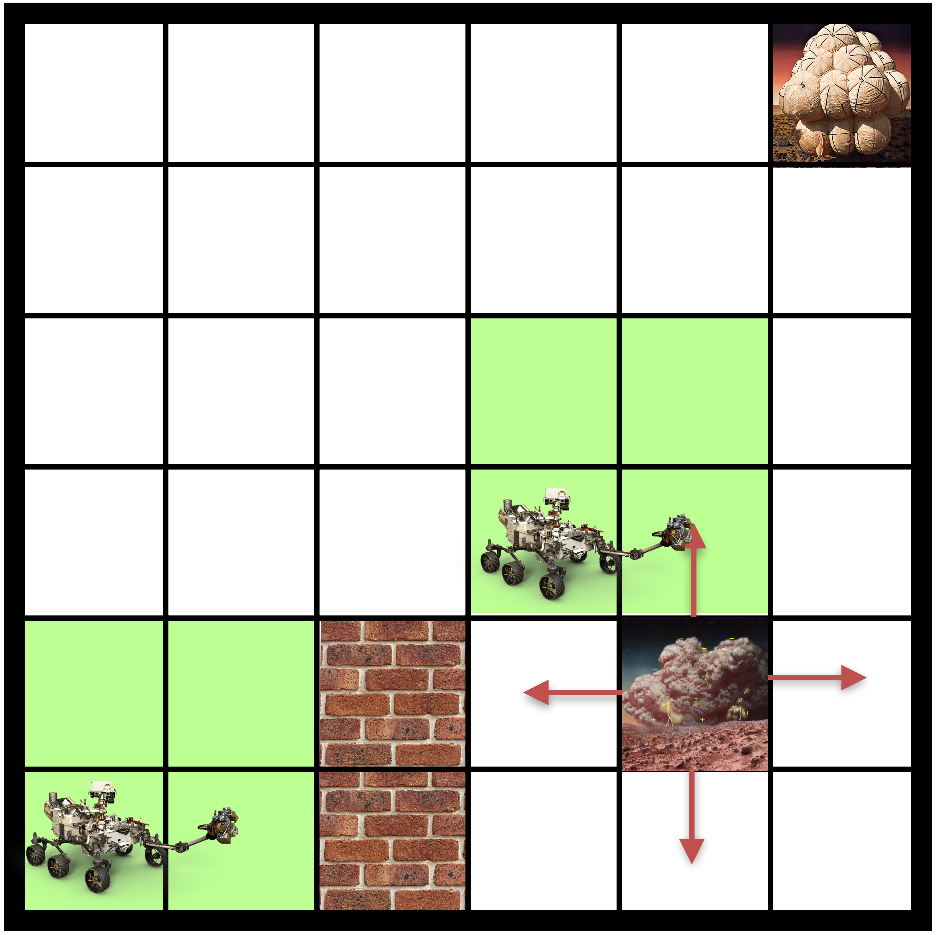
Idea: Human-in-the-loop Synthesis for POMDPs

Turn scenario into an arcade game

Collect data of human playing

From data, infer a strategy

Put human in critical situations



Underlying (family of) POMDPs

Applying strategy yields DTMC, efficient verification

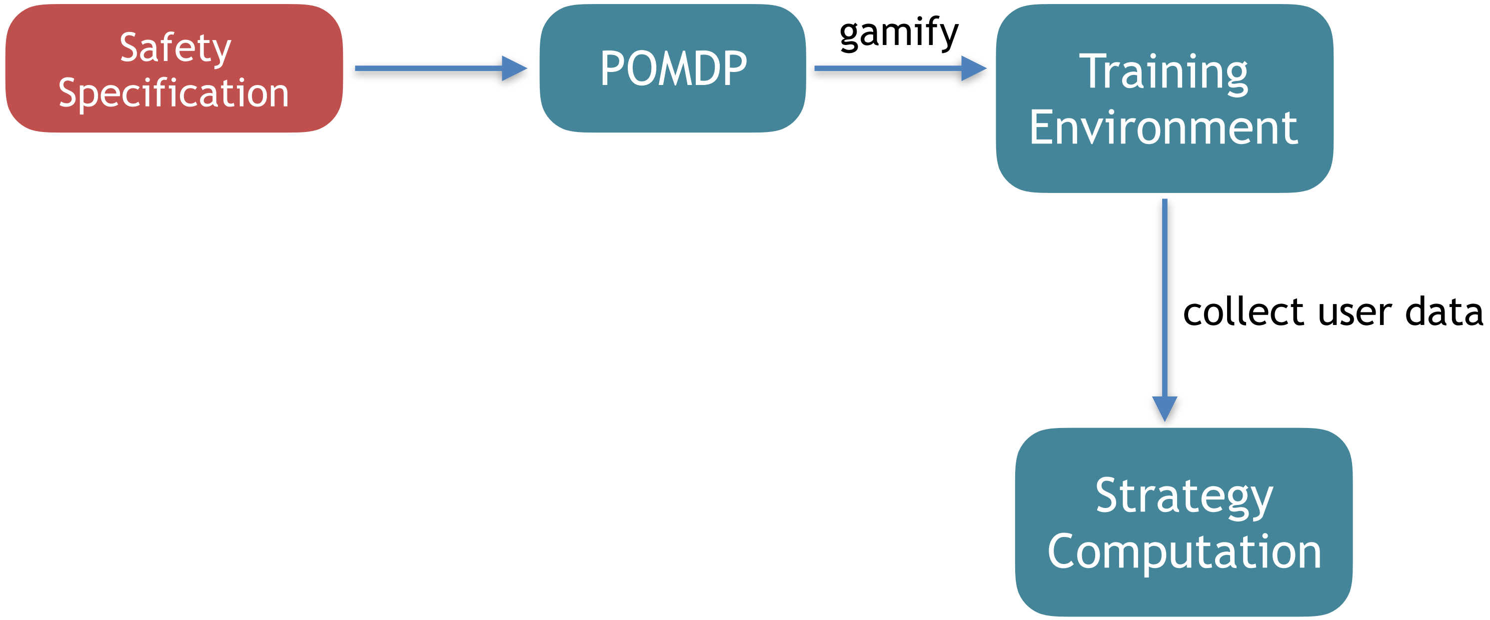
Counterexamples point to critical parts

Story: HiL Synthesis for POMDPs

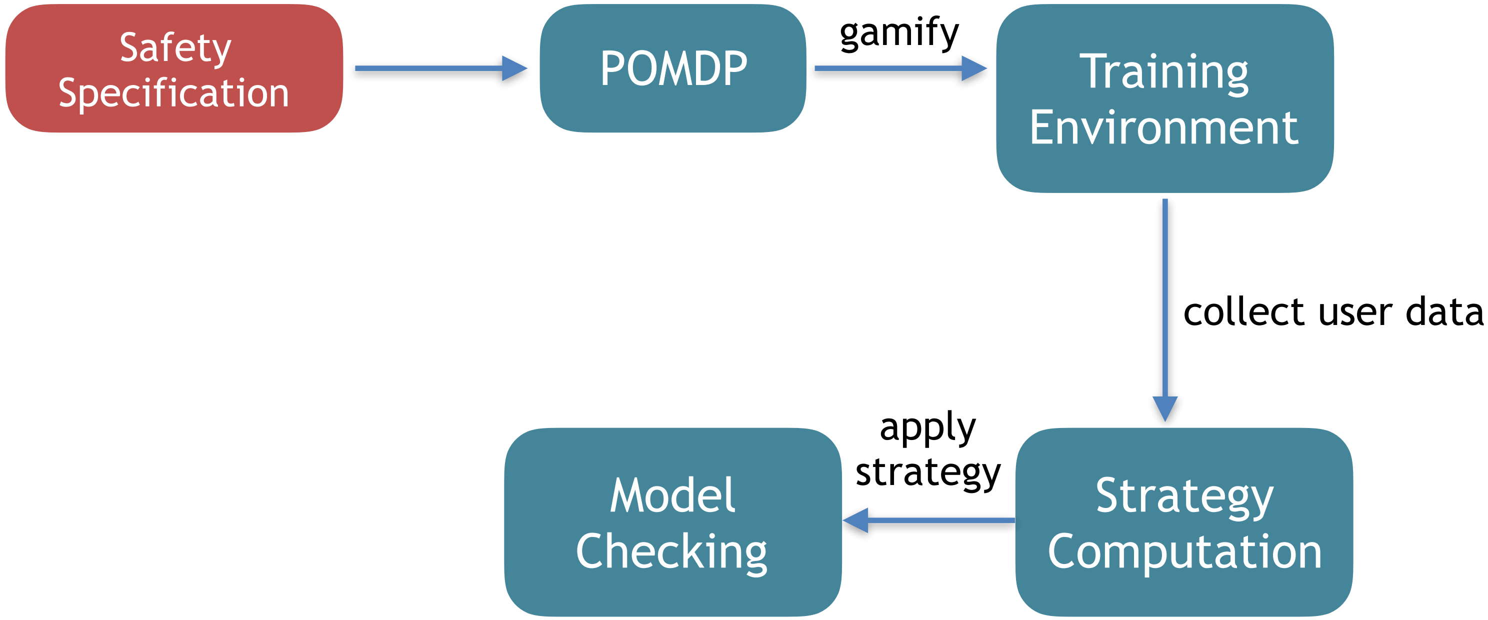
Safety Specification → POMDP

Story: HiL Synthesis for POMDPs

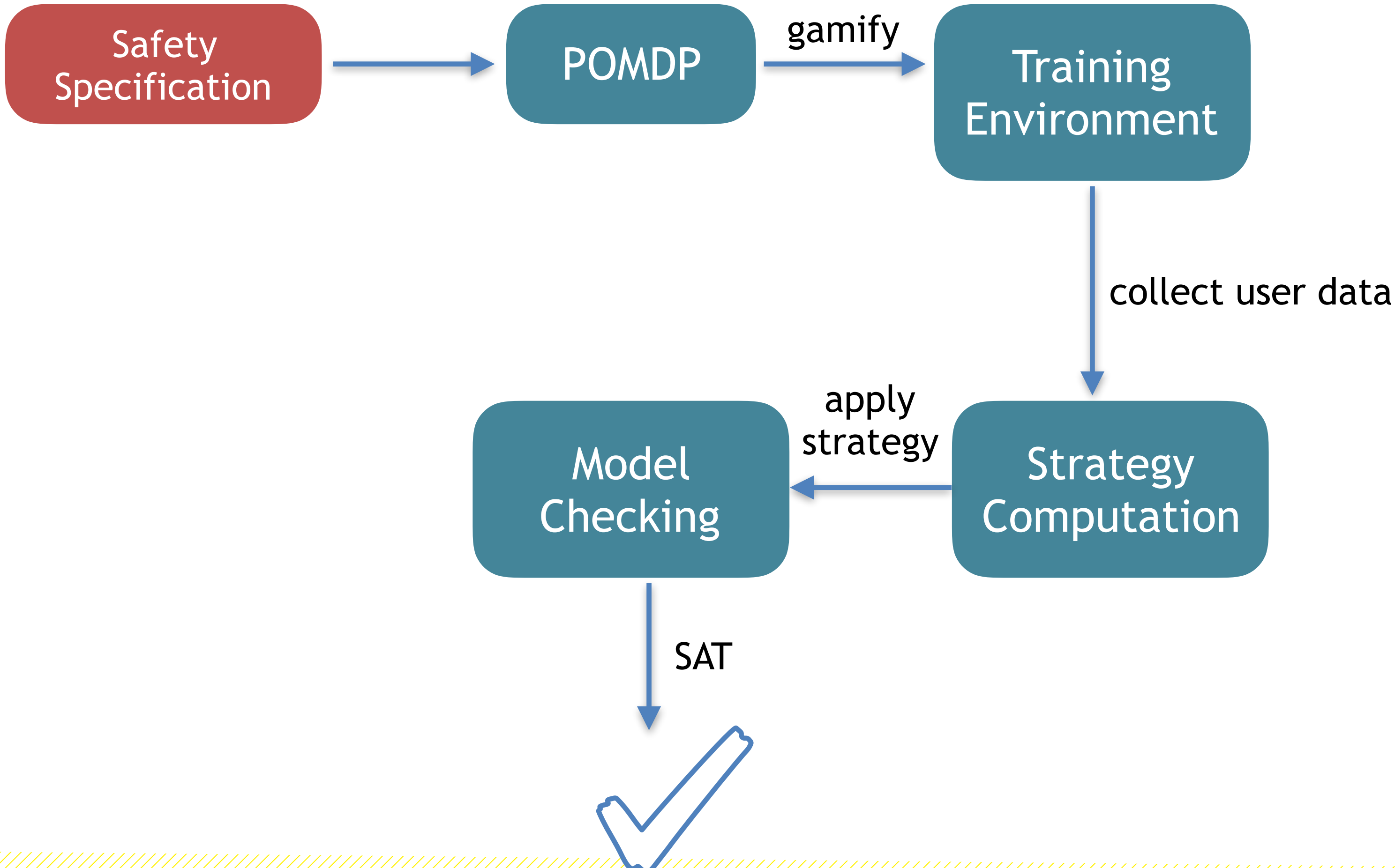
Story: HiL Synthesis for POMDPs



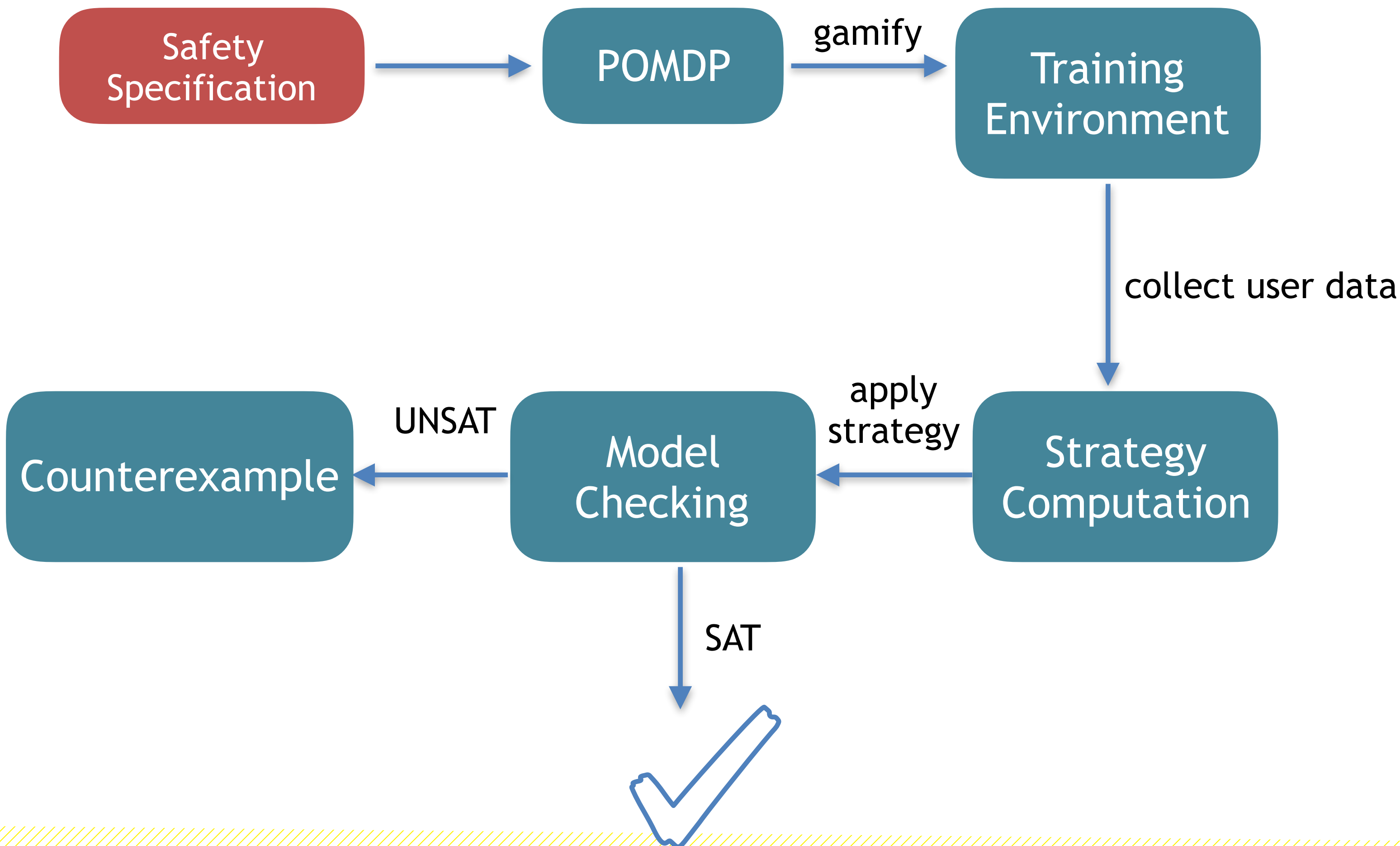
Story: HiL Synthesis for POMDPs



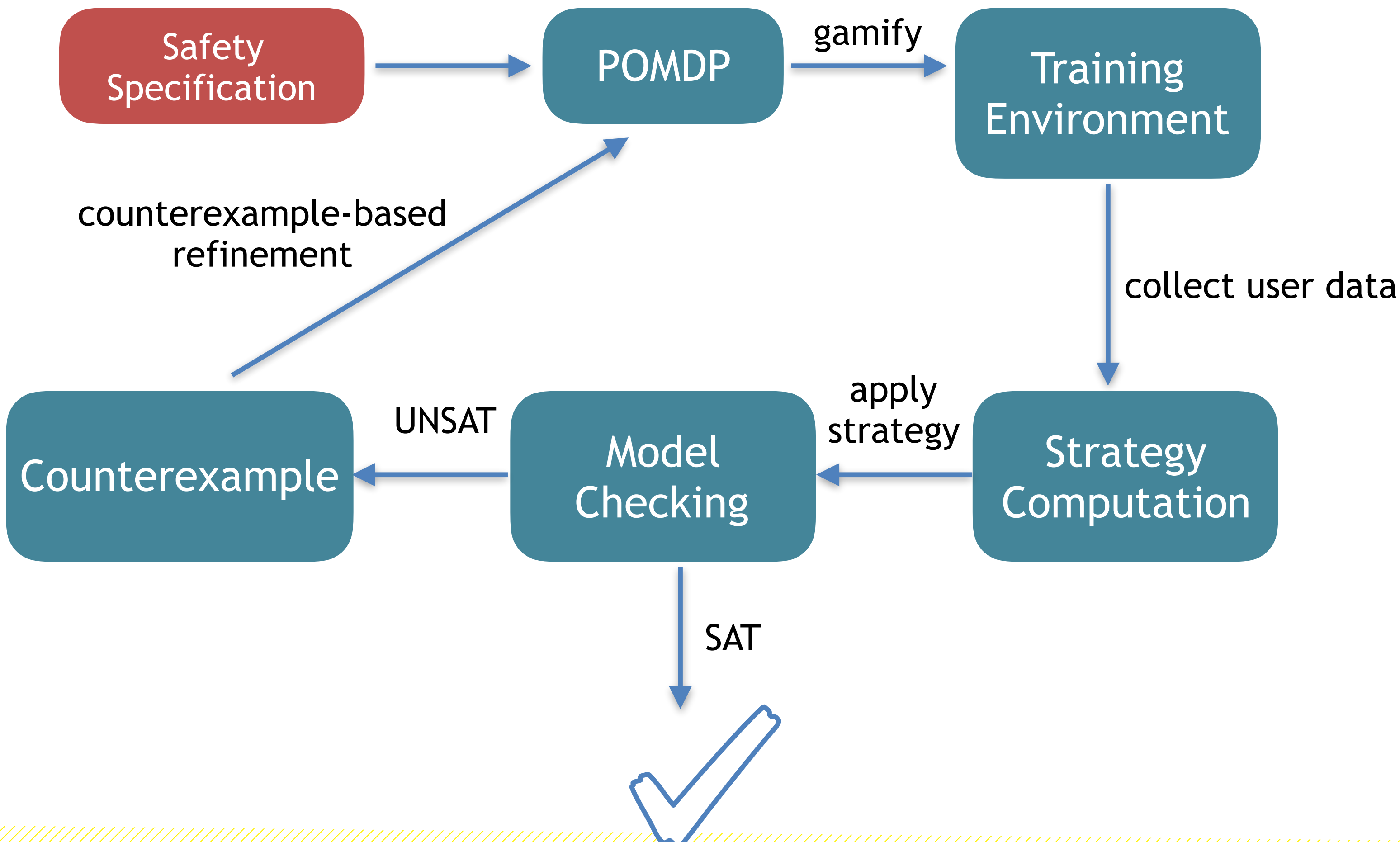
Story: HiL Synthesis for POMDPs



Story: HiL Synthesis for POMDPs

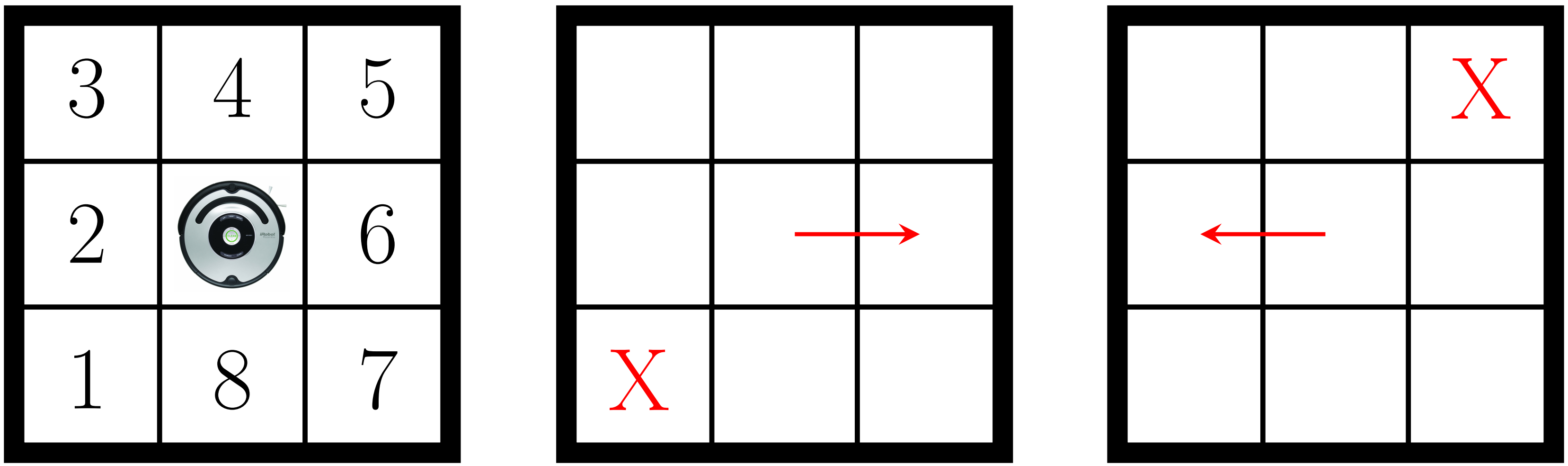


Story: HiL Synthesis for POMDPs



Data Augmentation

- Strategy is trained on randomly generated environments
- Training set needs samples until further environments wouldn't likely change the strategy



- To reduce training set, similar observations are handled similar

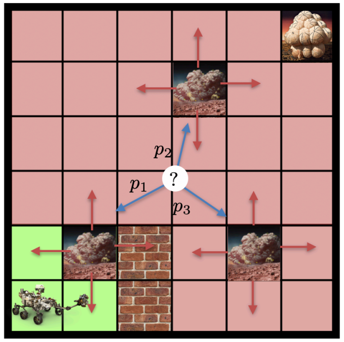
Experiments

Iteration	$\Pr(\neg B \cup G)$	Expected Cost ($\mathrm{EC}_{=?}[C]$)
0	0.225	13.57
1	0.503	9.110
2	0.592	7.154
3	0.610	6.055
4	0.636	5.923
Optimal	– n. a. –	5

	HiL Synth		PRISM-POMDP		PBVI	
grid	states	time (s)	states	time (s)	states	time (s)
3×3	277	43.74	303	2.20	81	3.86
4×4	990	121.74	987	4.64	256	2431.05
5×5	2459	174.90	2523	213.53	625	– MO –
6×6	5437	313.50	5743	– MO –	1296	– MO –
10×10	44794	1668.30	54783	– MO –	– MO –	– MO –
11×11	– MO –	– MO –	81663	– MO –	– MO –	– MO –

Conclusion and Future Work

1. Game-based abstraction
2. Finite-memory controllers
3. Recurrent neural networks
4. Fun: Humans in the loop



- Several directions to compute provably correct finite-memory policies for POMDPs
- Work on the intersection of AI, Machine Learning, and Formal Methods
- Future work: A toolbox!
- Call for collaboration:
  - Extract finite-state controllers from recurrent neural networks
  - Automata learning for stochastic systems
  - Whatever is fun!