Strategy Synthesis in POMDPs via Game-Based Abstractions

Leonore Winterer*, Sebastian Junges†, Ralf Wimmer*‡, Nils Jansen§,
Ufuk Topcu¶, Joost-Pieter Katoen† and Bernd Becker*

*Albert-Ludwigs-Universität Freiburg, Freiburg im Breisgau, Germany
{winterel | wimmer | becker}@informatik.uni-freiburg.de
†RWTH Aachen University, Aachen, Germany
{sebastian.junges | katoen}@informatik.rwth-aachen.de
‡Concept Engineering GmbH, Freiburg im Breisgau, Germany
http://www.concept.de
§Radboud University, Nijmegen, The Netherlands
n.jansen@science.ru.nl
¶The University of Texas at Austin, Austin, Texas, USA
utopcu@utexas.edu

***Abstract*—Partially observable Markov decision processes (POMDPs) are a natural model for scenarios where one has to deal with incomplete knowledge and random events. Applications include, but are not limited to, robotics and motion planning. However, many relevant properties of POMDPs are either undecidable or very expensive to compute in terms of both runtime and memory consumption. In our work, we develop a game-based abstraction method that is able to deliver safe bounds and tight approximations for important sub-classes of such properties. We discuss the theoretical implications and showcase the applicability of our results on a broad spectrum of benchmarks.**

I. Challenge

In offline motion planning, we aim to find a *strategy* for an agent that ensures certain desired behavior, even in the presence of dynamical obstacles and uncertainties [1]. If random elements – like uncertainty in the outcome of an action or in the movement of dynamic obstacles – need to be taken into account, the natural model for such scenarios are *Markov decision processes* (MDPs). MDPs are non-deterministic models which allow the agent to perform actions under full knowledge of the current state of the agent its surrounding environment. In many applications, though, full knowledge cannot be assumed, and we have to deal with *partial observability* [2]. For such scenarios, MDPs are generalized to *partially observable MDPs* (POMDPs). In a POMDP, the agent does not know the exact state of the environment, but only an observation that can be shared between multiple states. Additional information about the likelihood of being in a certain state can be gained by tracking the observations over time. This likelihood is called the belief state. Using an update function mapping a belief state and an action as well as the newly obtained observation to a new belief state, one can construct a (typically infinite) MDP, commonly known as the *belief MDP*.

While model checking and strategy synthesis for MDPs are, in general, well-manageable problems, POMDPs are much harder to handle and, due to the potentially infinite belief space, many problems are actually undecidable [3]. Our aim is to apply *abstraction* and *abstraction refinement* techniques to POMDPs in order to get good and safe approximative results for different types of properties.

II. Approach

As a case study, we work with a scenario featuring a controllable agent. Within a certain area, the agent needs to traverse a room while avoiding both static obstacles and randomly moving opponents. The area is modeled as a grid, the static obstacles as grid cells that may not be entered. Our assumption for this scenario is that the agent always knows its own position, but the positions of an opponent is only known if its distance from the agent is below a given threshold and if the opponent is not hidden behind a static obstacle. We assume that the opponents move probabilistically. This directly leads to a POMDP model for our case study. For simplification purposes, we only deal with one opponent, although our approach supports an arbitrary number of opponents. We assume the observation function of our POMDPs to be deterministic, but more general POMDPs can easily be simplified to this case.

The goal is to find a strategy which maximizes the probability to navigate through the grid from an initial to a target location without collision. For a grid size of $n \times n$ cells and one opponent, the number of states in the POMDP is in $O(n^4)$, i. e., the state space grows rapidly with increasing grid size. In order to handle non-trivial grids, we propose an approach using *game-based abstraction* [4].

Intuitively, we lump together all states that induce the same observation; for each position of the agent, we can distinguish between all states in which the opponent's position is known, but states in which the position is unknown are merged into one *far away* state [5]. In order to get a safe approximation

TABLE I: Comparing the POMDP solution (PRISM-pomdp) with the PG abstraction solution (PRISM-games).

	POMDP solution					PG solution					Lifting	MDP
Grid size	States	Choices	Result	Model Time	Sol. Time	States	Choices	Result	Model Time	Sol. Time	Result	Result
3×3	299	515	**0.8323**	0.063	**0.26**	396	639	**0.8323**	0.075	**0.040**	0.8323	0.8323
4×4	983	1778	**0.9556**	0.099	**1.81**	1344	2192	**0.9556**	0.098	**0.078**	0.9556	0.9556
5×5	2835	5207	**0.9882**	0.144	**175.94**	6016	10448	**0.9740**	0.193	**0.452**	0.9825	0.9882
5×6	4390	8126	**0.9945**	0.228	**4215.06**	7986	14199	**0.9785**	0.220	**0.534**	0.9893	0.9945
6×6	6705	20086	**?**	0.377	**– MO –**	10544	19150	**0.9830**	0.267	**1.414**	0.9933	0.9970
8×8	24893	47413	**?**	1.735	**– MO –**	23128	43790	**0.9897**	0.470	**6.349**	0.9992	0.9998
10×10	66297	127829	**?**	9.086	**– MO –**	40464	78054	**0.9914**	0.921	**12.652**	0.9999	0.9999
20×20	– Time out during model construction –					199144	395774	**0.9921**	9.498	**127.356**	0.9999	0.9999
30×30	– Time out during model construction –					477824	957494	**0.9921**	40.929	**489.369**	– MO –	0.9999
40×40	– Time out during model construction –					876504	1763214	**0.9921**	135.551	**1726.489**	– MO –	0.9999
50×50	– Time out during model construction –					1395184	2812934	**0.9921**	355.732	**3963.281**	– MO –	– MO –

of the behavior of the opponent, for all of these lumped states we add a non-deterministic choice over the potential positions of the opponent. We formalize this as a *2-player probabilistic game* [4] (PG), in which one player controls the actions of the agent, and the other controls the non-determinism added by the abstraction. Both players can optimize according to different goals. The abstraction player can create a worst-case scenario to *over-approximate* the realistic behavior, thus ensuring that the obtained bounds are safe and the resulting strategy cannot perform worse when mapped back to the original scenario.

III. Soundness

We show that any strategy computed with our abstraction that guarantees a certain level of safety can be mapped to a strategy for the original POMDP guarantiing at least the same level of safety. In particular, we establish a simulation relation between paths in the probabilistic game and paths in the POMDP. Intuitively, each path in the POMDP can be reproduced in the probabilistic game if the second player resolves the non-determinism in a certain way. Game-based model checking assumes the non-determinism to be resolved in the worst way possible, so it will provide a lower bound on the level of safety achievable in the actual POMDP. For full proof see [5].

IV. Results

We analyzed the game-based models using the PRISM-games model checker and compared the obtained results with the state-of-the-art POMDP model checker PRISM-pomdp [6], showing that we can handle grids that are considerably larger than what PRISM-pomdp can handle, while still getting schedulers that induce values which are close to optimal. Table I shows a few of our experiments for verifying a reach-avoid property on a grid without obstacles. The *result* colums show the probability (computed by the respective method) to reach a goal state without a collision. As one can see, the abstraction approach is faster by orders of magnitude than solving the POMDP directly, and the game model also is much smaller for large grids while still getting very good approximations for the actual probabilities. The strategies induce even better values when they are mapped back to the original POMDP.

TABLE II: Results for the PG for differently sized models with and without refinement.

		PG			Run times		
	Grid	States	Choices	Result	Create	Model	Solve
w/o ref.	4×40	50880	93734	**0.9228**	0.01	1.6	**37**
	4×60	77560	143254	**0.8923**	0.01	3.1	**41**
	4×80	104240	192774	**0.8628**	0.01	5.4	**128**
	4×100	130920	242294	**0.8343**	0.02	8.6	**101**
with ref.	4×40	68316	131858	**0.9733**	0.01	2.46	**102**
	4×60	104516	202338	**0.9733**	0.01	4.94	**324**
	4×80	140716	272818	**0.9733**	0.01	8.45	**697**
	4×100	176916	343298	**0.9733**	0.02	12.10	**1332**

V. Completeness

While being provably sound, our approach is still targeting an undecidable problem and as such not complete in the sense that in general no strategy with maximum probability for success can be deduced. In particular for cases with few paths to the goal location, the gap between the obtained bounds and the actual maximum can become large. For those cases, we define a scheme to refine the abstraction by encoding one or several steps of history into the current state, which leads to larger games and accordingly longer computation times, but also to better results. Table II showcases an implementation of this *one-step history refinement*. We use a benchmark representing a long, narrow tunnel, in which the agent has to pass the opponent once, but, due to the abstraction, can actually run into the it repeatedly if the abstraction-player has the opponent re-appear in front of the agent. With longer tunnels, the probability to safely arrive in a goal state diminishes. Adding a refinement which remembers the last known position of the opponent and thus restricting the non-deterministic movement keeps the probability constant for arbitrary length.

VI. Conclusion

We developed a game-based abstraction technique to synthesize strategies for a class of POMDPs. This class encompasses typical grid-based motion planning problems under restricted observability of the environment. For these scenarios, we efficiently compute strategies that allow the agent to maneuver the grid in order to reach a given goal state while at the same time avoiding collisions with faster moving obstacles. Experiments show that our approach can handle state spaces up to three orders of magnitude larger than general-purpose state-of-the-art POMDP solvers in less time, while at the same time using fewer states to represent the same grid sizes.

REFERENCES

[1] R. A. Howard, *Dynamic Programming and Markov Processes*, 1st ed. The MIT Press, 1960.

[2] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1, pp. 99–134, 1998.

[3] K. Chatterjee, M. Chmelík, and M. Tracol, "What is decidable about partially observable Markov decision processes with ω-regular objectives," *Journal of Computer and System Sciences*, vol. 82, no. 5, pp. 878–911, 2016.

[4] M. Kattenbelt and M. Huth, "Verification and refutation of probabilistic specifications via games," in *Proc. of FSTTCS*, ser. LIPIcs, vol. 4. Schloss Dagstuhl, 2009, pp. 251–262.

[5] Removed for blind review, "Journal version of the extended abstract," 2019.

[6] G. Norman, D. Parker, and X. Zou, "Verification and control of partially observable probabilistic systems," *Real-Time Systems*, vol. 53, no. 3, pp. 354–402, 2017.