Counterexample-Guided Abstraction Refinement for POND Planning

Jonas Thiem and **Robert Mattmüller** and **Manuela Ortlieb**
University of Freiburg, Germany
{thiemj,mattmuel,ortlieb}@informatik.uni-freiburg.de

Abstract

Counterexample-guided abstraction refinement (CEGAR) allows to gradually refine a problem until the required detail for a solution is reached. We propose the use of CEGAR to demonstrate unsolvability of partially observable nondeterministic planning tasks while avoiding search through the entire state space.

Partially observable tasks are ubiquitous in planning and robotics (Oliehoek 2010, p. 3). Sometimes, it is important to show unsolvability of such a task fast. Examples include algorithms minimizing necessary sensors, where unsolvability proofs are needed to show that a certain sensor cannot be left out (Mattmüller, Ortlieb, and Wacker 2014), and algorithms for strong and strong cyclic planning (Cimatti et al. 2003) that first try to find a strong plan, and if strong plan non-existence has been established, resort to finding a strong cyclic plan instead. The Counterexample-Guided Abstraction Refinement (CEGAR) technique originating from model checking (Clarke et al. 2003) can be used to speed up unsolvability proofs and has recently been used for classical planning (Seipp 2012), and, in a setting closely related to ours, in the context of games with incomplete information (Dimitrova and Finkbeiner 2008). CEGAR works as follows: It starts with a small initial abstraction of the planning task and searches for an abstract plan. If no such plan exists, it makes use of the fact that abstractions induce over-approximations of reachability and concludes that no concrete plan can exist, either. Otherwise, CEGAR tries to concretize the abstract plan found. Either, the solution is concretizable. Then CEGAR terminates. Otherwise, the solution is *spurious* and the abstraction must be refined. In our setting, instead of requiring abstractions where every single transition is preserved, preserving goal reachability is sufficient. A central question is how to define abstractions (guaranteeing over-approximations). A straightforward way for POND planning is to define an abstract belief state $\mathcal{B}$ as a set of concrete belief states B, where each such B consists of the set of world states s considered possible in B. Then the abstract initial state, goal states, and transitions can be defined easily. E.g., an action precondition φ is satisfied in $\mathcal{B}$ iff it is satisfied in *some* concrete belief state B represented by $\mathcal{B}$ (to ensure over-approximation), and φ is satisfied in B iff it is satisfied in *all* states $s \in B$ (to account for the uncertainty of the belief B). Unfortunately, representing such a set of sets $\mathcal{B}$ compactly is hard. On the other hand, representing a set B of states s compactly is, although exponential in the worst case, often feasible using binary decision diagrams (BDDs) (Bryant 1986). Therefore, in this work we approximate abstract belief states $\mathcal{B}$ by BDD-encoded sets of world states B. Furthermore, as abstractions we use simple projections to *patterns* P, i.e., sets of variables (Culberson and Schaeffer 1996). This raises several questions: (a) How to define and compute an (approximate) abstraction to a pattern P efficiently, (b) how to ensure that goal reachability is preserved, and (c) how to refine an abstraction if necessary. For (a), we use a simple *syntactic* projection to P similar to the one used for PDB heuristics in classical planning. However, when we use sets B of world states as abstract states and thus let the layers "belief" and "abstraction" collapse into one, we introduce an error that violates the over-approximation. We amend this as follows: We only allow variables in P that can always and unconditionally be observed, or that are known initially and can never become unknown. In addition, we forbid observations of variables outside of the pattern. This guarantees that we only ever produce singleton abstract belief states. Since we forbid some observations, we have no longer an over-approximation, but it can be proven that the goal reachability, possibly along longer paths, is preserved with the chosen restrictions. Regarding (c), we refine an abstraction by collecting all actions in the abstract policy whose precondition is violated in the concrete task and add the violated precondition variables to the refined pattern. If this violates the restriction of patterns, we immediately move to a pattern consisting of all variables in the planning task, i.e., to the identity abstraction. We implemented this variant of the CEGAR algorithm on top of the MYND planner (Mattmüller et al. 2010). For the benchmarked unsolvable problems, CEGAR leads to an increase of 20 to 50 percent in successfully handled unsolvable problems. As a downside, solvable problems suffer a slowdown which gradually widened in our benchmarks. For future work, we plan to investigate the performance of CEGAR as part of the two motivating scenarios, sensor minimization and strong/strong-cyclic planning. We also plan to investigate alternative abstractions such as the doubly exponential one mentioned above.

Acknowledgments

This work was partly supported by the DFG as part of the SFB/TR 14 AVACS.

References

Bryant, R. E. 1986. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers* 35(8):677–691.

Cimatti, A.; Pistore, M.; Roveri, M.; and Traverso, P. 2003. Weak, strong, and strong cyclic planning via symbolic model checking. *Artificial Intelligence* 147(1–2):35–84.

Clarke, E.; Grumberg, O.; Jha, S.; Lu, Y.; and Veith, H. 2003. Counterexample-guided abstraction refinement for symbolic model checking. *Journal of the ACM (JACM)* 50(5):752–794.

Culberson, J. C., and Schaeffer, J. 1996. Searching with pattern databases. In *Advances in Artificial Intelligence*, 402–416. Springer-Verlag.

Dimitrova, R., and Finkbeiner, B. 2008. Abstraction refinement for games with incomplete information. In *FSTTCS*, 175–186. Citeseer.

Mattmüller, R.; Ortlieb, M.; Helmert, M.; and Bercher, P. 2010. Pattern database heuristics for fully observable nondeterministic planning. In Brafman, R.; Geffner, H.; Hoffmann, J.; and Kautz, H., eds., *Proceedings of the 20th International Conference on Automated Planning and Scheduling (ICAPS 2010)*, 105–112.

Mattmüller, R.; Ortlieb, M.; and Wacker, E. 2014. Minimizing necessary observations for nondeterministic planning. In *KI 2014: Advances in Artificial Intelligence*. Springer. 309–320.

Oliehoek, F. 2010. *Value-based planning for teams of agents in stochastic partially observable environments*. Amsterdam University Press.

Seipp, J. 2012. *Counterexample-guided abstraction refinement for classical planning*. Master's thesis, Albert-Ludwigs-Universität Freiburg.