MS-Unsat and SimulationDominance: Merge-and-Shrink and Dominance Pruning for Proving Unsolvability

Álvaro Torralba, Jörg Hoffmann, Peter Kissmann
Saarland University
Saarbrücken, Germany
{torralba,hoffmann}@cs.uni-saarland.de, kissmann@googlemail.com

Abstract

This paper describes three different planners that participated in the 2016 unsolvability International Planning Competition (IPC). They use the Merge-and-Shrink (M&S) framework in different ways. MS-unsat tailors M&S to derive perfect unsolvability abstractions, proving unsolvability without any search. MS-unsat-irr uses the same approach with irrelevance pruning techniques to eliminate transitions and operators from the planning task. SimulationDominance performs a search using simulation-based dominance and irrelevance pruning, making use of M&S heuristics and h^{max} as dead-end detectors.

Introduction

Abstractions map the state space of the problem into a smaller abstract state space. They are commonly used to derive admissible heuristics for cost-optimal planning, by using the optimal distance in the abstract state space as an admissible estimation for the original problem. Abstraction techniques are very promising for proving unsolvability since proving that any abstraction is unsolvable is a sufficient condition for proving unsolvability (Bäckström *et al.* 2013). The question is how to design suitable abstractions for the problem at hand.

Merge-and-shrink (M&S) is a framework for deriving abstractions in a flexible way. It was originally devised for model-checking (Dräger *et al.* 2006; 2009) and later adapted to planning (Helmert *et al.* 2007; 2014; Sievers *et al.* 2014). The behavior of M&S is determined by the *shrinking* and *merging* strategies. Some shrinking strategies are *safe*, meaning that they preserve plan-existence so that the resulting abstraction is solvable if and only if the original problem is (Hoffmann *et al.* 2014). If non-safe shrinking is used, the resulting abstractions can be used as dead-end detector heuristics in a A^* search.

Another further use of M&S was to derive a set of transition systems in order to compute a dominance relation (Torralba and Hoffmann 2015). This dominance relation can be used for dominance pruning during search, eliminating states such that another "at least as good" state is known. Also, this dominance relation can be used for irrelevance pruning, removing transitions during the M&S process or even planning actions while preserving at least one optimal plan (Torralba and Kissmann 2015).

In this paper we present three different planners. MS-unsat employs M&S with safe shrinking to prove unsolvability without any search on the original state space. MS-unsat-irr uses the same strategy as MS-unsat, plus irrelevance pruning. The SimulationDominance planner uses search with simulation-based dominance and irrelevance pruning, h^{max}, and a set of M&S heuristics. The core ideas of these planners were introduced in previous work (Hoffmann *et al.* 2014; Torralba and Hoffmann 2015; Torralba and Kissmann 2015). This paper provides a general overview of the related literature and describes the configuration we chose for the planners.

Merge-and-Shrink

Merge-and-shrink is a framework to construct abstraction functions (Helmert *et al.* 2007; 2014). M&S works with a set of transition systems, initialized with the *atomic abstractions*, i.e. projections onto single state variables. Then, it interleaves *merging steps*, in which two transition systems are replaced by their synchronized product, with *shrinking steps*, which apply abstraction to keep the size of the transitions systems at bay. The algorithm stops when only one transition system remains and this is guaranteed to be an abstraction of the original problem. The algorithm depends on two strategies. The shrinking strategy selects how to apply abstraction to reduce the size of the transition systems. The merging strategy selects which two transition systems to merge at every step.

Shrinking strategies

Shrinking strategies decide which states to aggregate in order to reduce the size of the transition systems. The most popular shrinking strategy is bisimulation (Nissim *et al.* 2011), which computes the coarsest goal-preserving bisimulation relation and aggregates states that are bisimilar. An important property of bisimulation is that, if only bisimulation shrinking is applied at every step, the resulting transition system is a bisimulation of the original planning task. Since bisimulation preserves goal-distance, the resulting heuristic will be perfect and cost-optimal planning can be decided without any search. Exact label reduction aggregates some labels while preserving the structure of the state space, increasing the shrinking achieved by bisimulation while preserving its useful properties.

However, when only plan existence matters, one can further shrink the transition systems while keeping a perfect heuristic such that the abstraction is solvable if and only if the original problem is. Hoffmann *et al.* (2014) introduced safe shrinking strategies based on the concept of *own-labels*, i.e. labels that only affect a single transition system and have no preconditions or effects on the rest. *Own-path* shrinking aggregates all abstract states in a cycle of own-labeled transitions. Intuitively, since those transitions can be performed with no preconditions or effects on the rest of the problem those abstract states are interchangeable and can be aggregated. Moreover, if all goal variables have been merged in a transition system, states with an own-labeled path to a goal state can be aggregated since they are always solvable. Own-path and bisimulation shrinking are *safe* shrinking strategies, so if no other shrinking is used, the resulting heuristic is the unsolvability-perfect heuristic so that it can decide whether the problem is solvable without any search.

If the size of the abstraction is still too large, other approximations can be used, such as greedy bisimulation (Nissim *et al.* 2011) or K-catching bisimulation (Katz *et al.* 2012). We use the approximate bisimulation strategy introduced by Nissim *et al.*, in which they set a maximum limit for the abstraction size.

Merge strategies

Merge strategies can be classified into *linear* and *non-linear* merge strategies. Linear merge strategies are characterized by a variable ordering, merging an atomic abstraction at every iteration of the algorithm. The first merge strategies were linear merge strategies based on causal graph (Knoblock 1994; Helmert *et al.* 2007). Hoffmann *et al.* (2014) made an empirical study of 81 different linear merge strategies for proving unsolvability, based on the following criteria:

- **Tr**, **TrOwn**, **TrGoal**, **TrOwnGoal**: Maximize number of transitions whose labels are relevant for both transition systems. If *own* is activated, ignore transitions that are not own-labeled. If *goal* is activated considers only transitions going into a goal state.
- **CG**, **CGRoot**, and **CGLeaf:** Prefer variables with an outgoing causal graph arc to an already selected variable. If there are several such variables prefer the one ordered before (CGRoot) or behind (CGLeaf) in the strongly connected components of the causal graph. It may use the complete causal graph (Com) or only pre-eff edges.
- **LevelRoot** and **LevelLeaf:** Derived from FD's full linear order (Helmert 2006). LevelRoot prefers variables "closest to be causal graph roots", and LevelLeaf prefers variables "closest to be causal graph leaves".
- **Goal:** Prefer goal variables over non-goal variables.

Sievers *et al.* (2014) reformulated the M&S framework and generalized label reduction to work with non-linear merge strategies. They also introduced in planning the DFP non-linear merge strategy, originally used in the context of model-checking (Dräger *et al.* 2006). Other relevant non-linear merge strategy is MIASM (Fan *et al.* 2014). A recent analysis of linear and non-linear merging strategies was made by Sievers *et al.* (2016).

Simulation-Based Dominance Pruning

Dominance pruning techniques aim to avoid the exploration of some parts of the state space, if they are proven to be worse than others (Hall *et al.* 2013). This is formalized in terms of a relation on the state space of the planning task, $\preceq$, such that $s \preceq t$ implies that t is "at least as close to the goal" as s. Our approach is based on the well-known notion of simulation relations (Milner 1971; Gentilini *et al.* 2003). A relation $\preceq$ is a simulation if for any two states s, t such that $s \preceq t$ and any transition $s \xrightarrow{l} s'$, exists another transition $t \xrightarrow{l} t'$ such that $s' \preceq t'$. The coarsest goal-respecting simulation relation can be computed in polynomial time on the size of the state space, though this is still exponential in the size of the planning task.

In order to compute a relation in polynomial time we follow a compositional approach in which the dominance relation is derived from simulation relations computed on a partition of the planning task (Torralba and Hoffmann 2015). A partition of the planning task is a set of transition systems, $\Theta_1, \dots, \Theta_k$ such that their synchronized product equals the state space of the planning task. In order to derive such partition, we use the M&S algorithm with bisimulation shrinking, changing the stopping condition by forbidding any merge that would exceed a maximum limit on the number of transitions. The coarsest goal-respecting simulations for each Θ_i, $\preceq_i$, can then be combined to define a dominance relation on the state space of the planning task, $s \preceq t$ iff $\alpha_i(s) \preceq_i \alpha_i(t)$ for all $i \in 1, \dots, k$. However, the number of problems in which non-trivial simulation relations exist are limited because the transition $t \xrightarrow{l} t'$ has to use exactly the same label as $s \xrightarrow{l} s'$.

To overcome this limitation, we introduce a relation between the labels of the transition systems. A label l' dominates l in a transition system Θ_i iff for any transition $s \xrightarrow{l} s'$ exists another $s \xrightarrow{l'} s''$ such that $s' \preceq s''$. Then, a label-dominance simulation computes the simulation relation of all transition systems $\preceq_1, \dots, \preceq_k$ simultaneously, allowing $t \xrightarrow{l'} t'$ to simulate $s \xrightarrow{l} s'$ if $s' \preceq t'$ and l' dominates l on all other transition systems. Moreover, a *noop* action with no preconditions and effects is introduced in order to capture the notion of "doing nothing". Label-dominance simulation with *noop* actions finds coarser relations that are able to achieve pruning in many different benchmark domains.

Once a dominance relation has been computed, in order to perform dominance pruning during search, we keep a Binary Decision Diagram (Bryant 1986) that represents the set of all states dominated by any expanded state. Anytime a state is generated, it is pruned if it is contained in such set. To avoid unnecessary overhead, we disable dominance pruning if no state has been pruned after *1000* expansions.

Irrelevance Pruning

Irrelevance pruning removes actions from the planning task while preserving at least one (optimal) solution. Label-dominance simulation relations can be used to detect such irrelevant transitions. Subsumed transition pruning (Torralba and Kissmann 2015) eliminates transitions $s \xrightarrow{l} t$ from the M&S transition systems if there exists another transition from s, $s \xrightarrow{l'} t'$ that simulates it, i.e. $t \preceq t'$ and l' dominates l in all other transition systems. Removing such transitions might cause some parts of the abstract state space to become unreachable, leading to additional pruning and simplification of the M&S transition systems. If all transitions corresponding to a planning action are removed, the action can be completely removed from the planning task while still preserving plan existence.

Subsumed transition pruning can be interleaved with label reduction and bisimulation shrinking but not with other shrinking strategies such as own-path shrinking. Even though both subsumed transition pruning and own-path shrinking preserve solvability (so their combination does as well) the resulting abstraction cannot safely be used to detect dead-ends on the original state space. Also, applying label reduction is not always beneficial for subsumed transition pruning so we follow three different steps, where M is a parameter that controls how large the transition systems are:

1. M&S with subsumed transition pruning and a limit of M transitions. Without label-reduction or any shrinking.
2. M&S with subsumed transition pruning, label-reduction and bisimulation shrinking. Limit of M transitions.
3. M&S with label-reduction, and own-path + bisimulation shrinking.

If dominance pruning is used, the label-dominance simulation relation is computed after the second step.

IPC Configuration

We implemented the new merge and shrinking strategies on top of the Fast Downward Planning System (Helmert 2006) (version from July 16th, 2014). All our planners use h^2 forward and backward relevance analysis in order to eliminate operators and simplify the planning task prior to the search (Alcázar and Torralba 2015).

All runs of M&S use the exact label reduction by Sievers *et al.* (Sievers *et al.* 2014), interrupting it if it takes more than 60 seconds. To avoid overhead, if there are more than 200 labels, label-dominance is computed only with respect to the *noop* action.

MS-unsat and MS-unsat-irr

We submit two different configurations MS-unsat and MS-unsat-irr. MS-unsat uses the best configuration reported by Hoffmann *et al.* (2014), using CGRoot-Goal-LevelLeaf merge and own-label shrinking.

MS-unsat-irr uses two runs of M&S with irrelevance pruning. In the first one, it uses the DFP non-linear merge strategy with irrelevance pruning with a limit of $M = 50\,000$ transitions and 300 seconds. If the task has not been proven unsolvable by the first run, irrelevant operators are removed from the problem. Afterwards, it performs another M&S run using CGRoot-Goal-LevelLeaf, and subsumed transition pruning up to a limit of 50 000 transitions.

SimulationDominance

The SimulationDominance planner performs a search using dominance and irrelevance pruning, and the h^{max} heuristic (Bonet and Geffner 2001) and M&S abstractions as dead-end detectors.

The dominance pruning relation is derived using the DFP-merge strategy with a limit of 100 000 transitions. Then, it uses M&S to generate a list of M&S abstractions, that are used during the search to detect dead-ends. All M&S runs use subsumption pruning up to $M = 100\,000$ transitions and set a limit of 500000 abstract states for bisimulation on the third step. Multiple linear merge strategies are used in a sequential fashion: TrOwnGoal-CGComLeaf-Goal, Tr, TrOwnGoal, Tr, TrOwn, CG-Goal, CGLeaf-Goal, CGRoot-Goal, CGComLeaf-Goal, TrOwnGoal-CGComLeaf-Goal. All these strategies are run twice, using the LevelLeaf and random tie-breaking, respectively. Each run of M&S may take up to 300 seconds and the overall abstraction generation may take up to 1400 seconds, after which the search starts.

Conclusions

In this paper, we have introduced three different papers that participated in the 2016 edition of the unsolvability IPC: MS-unsat, MS-unsat-irr, and SimulationDominance. MS-unsat and MS-unsat-irr make use of M&S with a safe shrinking strategy that allows to prove unsolvability without searching the original state space. SimulationDominance uses M&S to construct a set dead-end detection heuristics as well as a label-dominance simulation relation used for dominance and irrelevance pruning.

Acknowledgments We'd like to thank the Fast Downward development team for sharing the latest version of their Fast Downward Planning System and, in particular, to Silvan Sievers, Martin Wehrle, and Malte Helmert for their work on M&S (Sievers *et al.* 2014). This work was partially supported by the German Research Foundation (DFG), under grant HO 2169/5-1, "Critically Constrained Planning via Partial Delete Relaxation".

References

Vidal Alcázar and Álvaro Torralba. A reminder about the importance of computing and exploiting invariants in planning. In Ronen Brafman, Carmel Domshlak, Patrik Haslum, and Shlomo Zilberstein, editors, *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS'15)*. AAAI Press, 2015.

Christer Bäckström, Peter Jonsson, and Simon Ståhlberg. Fast detection of unsolvable planning instances using local consistency. In Malte Helmert and Gabriele Röger, editors, *Proceedings of the 6th Annual Symposium on Combinatorial Search (SOCS'13)*, pages 29–37. AAAI Press, 2013.

Blai Bonet and Héctor Geffner. Planning as heuristic search. *Artificial Intelligence*, 129(1–2):5–33, 2001.

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

Klaus Dräger, Bernd Finkbeiner, and Andreas Podelski. Directed model checking with distance-preserving abstractions. In Antti Valmari, editor, *Proceedings of the 13th International SPIN Workshop (SPIN 2006)*, volume 3925 of *Lecture Notes in Computer Science*, pages 19–34. Springer-Verlag, 2006.

Klaus Dräger, Bernd Finkbeiner, and Andreas Podelski. Directed model checking with distance-preserving abstractions. *International Journal on Software Tools for Technology Transfer*, 11(1):27–37, 2009.

Gaojian Fan, Martin Müller, and Robert Holte. Non-linear merging strategies for merge-and-shrink based on variable interactions. In Stefan Edelkamp and Roman Bartak, editors, *Proceedings of the 7th Annual Symposium on Combinatorial Search (SOCS'14)*. AAAI Press, 2014.

Raffaella Gentilini, Carla Piazza, and Alberto Policriti. From bisimulation to simulation: Coarsest partition problems. *Journal of Automated Reasoning*, 31(1):73–103, 2003.

David Hall, Alon Cohen, David Burkett, and Dan Klein. Faster optimal planning with partial-order pruning. In Daniel Borrajo, Simone Fratini, Subbarao Kambhampati, and Angelo Oddi, editors, *Proceedings of the 23rd International Conference on Automated Planning and Scheduling (ICAPS'13)*, Rome, Italy, 2013. AAAI Press.

Malte Helmert, Patrik Haslum, and Jörg Hoffmann. Flexible abstraction heuristics for optimal sequential planning. In Mark Boddy, Maria Fox, and Sylvie Thiebaux, editors, *Proceedings of the 17th International Conference on Automated Planning and Scheduling (ICAPS'07)*, pages 176–183, Providence, Rhode Island, USA, 2007. Morgan Kaufmann.

Malte Helmert, Patrik Haslum, Jörg Hoffmann, and Raz Nissim. Merge & shrink abstraction: A method for generating lower bounds in factored state spaces. *Journal of the Association for Computing Machinery*, 61(3), 2014.

Malte Helmert. The Fast Downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006.

Jörg Hoffmann, Peter Kissmann, and Álvaro Torralba. "Distance"? Who Cares? Tailoring merge-and-shrink heuristics to detect unsolvability. In Thorsten Schaub, editor, *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI'14)*, Prague, Czech Republic, August 2014. IOS Press.

Michael Katz, Jörg Hoffmann, and Malte Helmert. How to relax a bisimulation? In Blai Bonet, Lee McCluskey, José Reinaldo Silva, and Brian Williams, editors, *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS'12)*, pages 101–109. AAAI Press, 2012.

Craig Knoblock. Automatically generating abstractions for planning. *Artificial Intelligence*, 68(2):243–302, 1994.

Robin Milner. An algebraic definition of simulation between programs. In *Proceedings of the 2nd International Joint Conference on Artificial Intelligence (IJCAI'71)*, pages 481–489, London, UK, September 1971. William Kaufmann.

Raz Nissim, Jörg Hoffmann, and Malte Helmert. Computing perfect heuristics in polynomial time: On bisimulation and merge-and-shrink abstraction in optimal planning. In Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, pages 1983–1990. AAAI Press/IJCAI, 2011.

Silvan Sievers, Martin Wehrle, and Malte Helmert. Generalized label reduction for merge-and-shrink heuristics. In Carla E. Brodley and Peter Stone, editors, *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'14)*, pages 2358–2366, Austin, Texas, USA, January 2014. AAAI Press.

Silvan Sievers, Martin Wehrle, and Malte Helmert. An analysis of merge strategies for merge-and-shrink heuristics. In Amanda Coles, Andrew Coles, Stefan Edelkamp, Daniele Magazzeni, and Scott Sanner, editors, *Proceedings of the 26th International Conference on Automated Planning and Scheduling (ICAPS'16)*. AAAI Press, 2016.

Álvaro Torralba and Jörg Hoffmann. Simulation-based admissible dominance pruning. In Qiang Yang, editor, *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15)*, pages 1689–1695. AAAI Press/IJCAI, 2015.

Álvaro Torralba and Peter Kissmann. Focusing on what really matters: Irrelevance pruning in merge-and-shrink. In Levi Lelis and Roni Stern, editors, *Proceedings of the 8th Annual Symposium on Combinatorial Search (SOCS'15)*, pages 122–130. AAAI Press, 2015.