IPALAMA: Planner Abstract

Daniel Gnad, Álvaro Torralba
Saarland University
Saarland Informatics Campus
Saarbrücken, Germany
{gnad,torralba}@cs.uni-saarland.de

Martín Domínguez, Carlos Areces, Facundo Bustos
Universidad Nacional de Córdoba
Córdoba, Argentina
{mardom75, carlos.areces, facundojosebustos}@gmail.com

Abstract

We describe the IPALAMA planner that participated in the Sparkle Planning Challenge 2019. IPALAMA, as the name suggests, heavily builds on top of the LAMA planner, which won the International Planning Competitions 2008 and 2011. It is implemented in the Fast Downward planning system. IPALAMA runs the first iteration of LAMA, but instead of fully grounding the PDDL description of a given planning task, it only partially grounds the task. This avoids the potentially exponential blow-up of the grounding process, leading to a significant speed-up in domains where a compact partial grounding is enough to solve a task. However, it renders the planning process incomplete. We fix this issue by incrementally grounding more, if a previous iteration of the planner fails, resulting in a complete overall approach.

Planner Description

In classical planning, (almost?) all planners compute a grounded representation of a given planning task prior to attempting to actually solve the task. Planning tasks are typically described in the *lifted* Planning Domain Definition Language (PDDL) (McDermott et al. 1998), which is very compact. To solve a task at hand, planners translate the PDDL model into a *grounded* representation, usually either STRIPS (Fikes and Nilsson 1971) or FDR (finite domain representation) (Bäckström and Nebel 1995; Helmert 2006). The drawback of fully grounding the input task is that the grounded representation can be exponentially larger than the lifted description.

In IPALAMA, we try to address this weakness by only partially grounding the input task (Gnad et al. 2019).[1] We do so by guiding the grounding with a novelty-based heuristic that gives preference to actions that achieve new ground facts. This heuristic is based on the notion of novelty for heuristic-search planners (Lipovetzky and Geffner 2012), somewhat adapted to the setting of grounding. For technical details, we refer the reader to Gnad et al. (2019). We are running their round-robin novelty priority queue.

Our implementation builds on a version of Fast Downward (Helmert 2006) (FD) from 2018. We only adapted the translator component, which, given the PDDL input, produces an FDR representation of the task. For the search part, we stick to FD's version of the first iteration of the LAMA-11 planner (Richter, Westphal, and Helmert 2011).

Incrementally **PA**rtially grounding **LAMA** – IPALAMA for short – differs from its ancestor only by how the translator builds the FDR representation. It starts by partially grounding the task until the goal becomes relaxed reachable, or until at least 5000 actions have been grounded. For the following search on the partial grounding, we allow $60s$. In case the search fails, we incrementally increase the minimum number of actions to be grounded by doubling the number from the last iteration. We expect this exponential growth to lead to fewer iterations when the "goal is relaxed reachable" condition is not a good indicator of whether the partial grounding is solvable. In such domains, we will quickly get to the full grounding, basically falling back to a standard LAMA-11. In domains where a small partial grounding is sufficient, however, we expect IPALAMA to perform significantly better than the original.

[1] We remark that our approach is sound, i. e., a solution to the partially grounded task is always a solution to the lifted input task.

References

Bäckström, C., and Nebel, B. 1995. Complexity results for SAS$^+$ planning. *Computational Intelligence* 11(4):625–655.

Fikes, R. E., and Nilsson, N. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2:189–208.

Gnad, D.; Torralba, Á.; Domínguez, M.; Areces, C.; and Bustos, F. 2019. Learning how to ground a plan – partial grounding in classical planning. In Hentenryck, P. V., and Zhou, Z.-H., eds., *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI'19)*. AAAI Press.

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

Lipovetzky, N., and Geffner, H. 2012. Width and serialization of classical planning problems. In Raedt, L. D., ed., *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI'12)*, 540–545. Montpellier, France: IOS Press.

McDermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. *The PDDL Planning Domain Definition Language*. The AIPS-98 Planning Competition Comitee.

Richter, S.; Westphal, M.; and Helmert, M. 2011. LAMA 2008 and 2011 (planner abstract). In *IPC 2011 planner abstracts*, 50–54.