The TOAD System for Totally Ordered HTN Planning in the 2023 IPC

Daniel Höller

Saarland University, Saarland Informatics Campus,
Saarbrücken, Germany
hoeller@cs.uni-saarland.de

Abstract

The TOAD system is a translation-based planning system for totally ordered HTN planning. It translates a given HTN planning problem into a classical planning problem. To overcome the differences in expressiveness, it does not bound the problem like other translation-based systems, but approximates the problem instead by modifying the decomposition hierarchy such that the set of solutions increases. Then we encode it as classical planning problem and solve it using classical planners. To ensure that only solutions for the original HTN problem are returned, we apply HTN plan verification.

Introduction

The TOAD (Totally Ordered HTN Approximation using DFA) system (Höller 2021) is a translation-based planning system for totally ordered HTN planning. It translates a given HTN planning problem into a classical problem and uses classical planners to solve it. While translation-based systems from the literature *bound* the HTN problem to overcome the differences in expressiveness (Alford, Kuter, and Nau 2009; Alford et al. 2016; Behnke et al. 2022), TOAD over-approximates the set of solutions. I.e., all solutions to the HTN planning problem are also solutions for the classical problem, but the latter might have more.

The approach is inspired by our results on the expressiveness of planning formalisms (Höller et al. 2014, 2016). The set of solutions to a totally ordered HTN planning problem can be seen as the intersection of two languages: a context-free language describing which action sequences can result from the decomposition process, and a regular language describing which action sequences are applicable and lead to a goal state in the transition system induced by the non-hierarchical part of the HTN problem (actions/state). TOAD uses techniques from the field of formal languages (Nederhof 2000a,b) to create a finite automaton (FA) accepting the words of the context-free language (which might require approximation), which is then combined with the non-hierarchical part of HTN problem.

The TOAD System

We use the preprocessing of the PANDA framework for hierarchical planning (Höller et al. 2021), i.e., HDDL as input language (Höller et al. 2020) and the PANDA grounder to ground the model (Behnke et al. 2020).

Figure 1 illustrates the overall TOAD system, which is described in the following.

Analysis. First, TOAD analyzes whether the HTN problem can be translated exactly or approximation is needed. This is done based on a criterion from formal languages called *self-embedding* (Chomsky 1959), which is checked on the decomposition rules (i.e., the methods). We first construct the decomposition graph, i.e., a graph with the tasks of the problem as nodes in which two nodes c_a and c_b are connected by a directed edge (c_a, c_b) when there is a method decomposing c_a into a task sequence including c_b. We compute the strongly connected components (sccs) of this graph. A problem is self-embedding if there is a scc N_i such that

- there is a method $(c_a, \alpha c_b \beta)$, $c_a, c_b \in N_i$ and $\alpha \neq \varepsilon$ and
- there is a method $(c_a, \alpha c_b \beta)$, $c_a, c_b \in N_i$ and $\beta \neq \varepsilon$.

When a problem is *not* self-embedding, this is a sufficient criterion that it describes a regular language, which for us means that approximation is not needed.

Approximation. When approximation is needed, the grammar rules (methods) are modified such that the set of solutions increases. We use an approximation introduced by Nederhof (2000a; 2000b).

Consider a grammar $G = (\mathcal{C}, \mathcal{A}, M, c_I)$ with the nonterminal and terminal symbols $\mathcal{C} = \{A\}$ and $\mathcal{A} = \{a, b\}$, the production rules $M = \{(A, b), (A, aAa)\}$, and the start symbol $c_I = A$. It describes the context-free language $\{a^n \; b \; a^n \mid n \geq 0\}$. The approximation disconnects the part left and right of the b, resulting in a grammar generating the language $\{a^n \; b \; a^m \mid n, m \geq 0\}$, which is regular.

Based on the method by Nederhof (2000a; 2000b) we construct a finite automaton (FA) accepting action sequences derivable via the (maybe modified) hierarchy.

Classical Encoding. Based on the FA and the non-hierarchical part of the HTN planning problem (actions/state) we build a classical planning problem.

Solving. We use the Fast Downward (FD) planning system (Helmert 2006) to solve the resulting problems. We use a multi-fringe configuration similar to the classical LAMA system (Richter and Westphal 2010) with two fringes. One

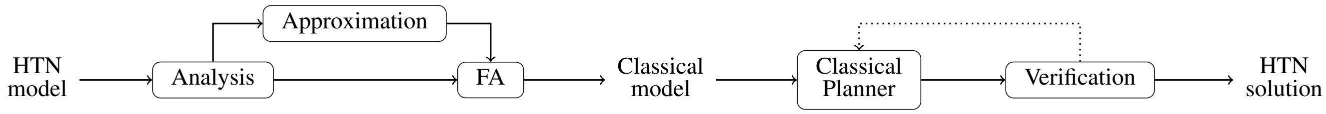

Figure 1: Schema of the TOAD system (based on Figure 1 by Höller, 2021).

uses the FF heuristic (Hoffmann and Nebel 2001) and also its helpful actions, and one uses a precomputed heuristic returning the distance from the current state in the FA to the nearest goal state in the FA as heuristic value.

Verification. To return only valid solutions for the original HTN problem, HTN plan verification is applied as last step. We use the verification system introduced by Höller et al. (2022) in combination with the PANDA progression planner (Höller et al. 2018, 2020).

We modified FD to verify a solution before returning it. When it is not valid, search is continued.

Discussion

While the basic approach is sound and complete, the combination with a graph search like used by FD leads to an incomplete overall system. This is caused by the fact that such a system does not (eventually) return *every* solution to the underlying classical problem. Whenever a particular state in the search space needs to be visited twice before a solution for the HTN problem is found, TOAD will fail. However, at least on the benchmark set of the last (i.e., 2020) IPC, this seems not to be an issue. To the contrary, in this set most problems can be translated without using the approximation.

References

Alford, R.; Behnke, G.; Höller, D.; Bercher, P.; Biundo, S.; and Aha, D. 2016. Bound to Plan: Exploiting Classical Heuristics via Automatic Translations of Tail-Recursive HTN Problems. In *Proceedings of the 26th International Conference on Automated Planning and Scheduling (ICAPS)*, 20–28. AAAI Press.

Alford, R.; Kuter, U.; and Nau, D. S. 2009. Translating HTNs to PDDL: A Small Amount of Domain Knowledge Can Go a Long Way. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, 1629–1634.

Behnke, G.; Höller, D.; Schmid, A.; Bercher, P.; and Biundo, S. 2020. On Succinct Groundings of HTN Planning Problems. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, 9775–9784. AAAI Press.

Behnke, G.; Pollitt, F.; Höller, D.; Bercher, P.; and Alford, R. 2022. Making Translations to Classical Planning Competitive with Other HTN Planners. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, 9687–9697. AAAI Press.

Chomsky, N. 1959. On Certain Formal Properties of Grammars. *Information and Control*, 2(2): 137–167.

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

Hoffmann, J.; and Nebel, B. 2001. The FF Planning System: Fast Plan Generation Through Heuristic Search. *Journal of Artificial Intelligence Research*, 14: 253–302.

Höller, D. 2021. Translating Totally Ordered HTN Planning Problems to Classical Planning Problems Using Regular Approximation of Context-Free Languages. In *Proceedings of the 31st International Conference on Automated Planning and Scheduling (ICAPS)*, 159–167. AAAI Press.

Höller, D.; Behnke, G.; Bercher, P.; and Biundo, S. 2014. Language Classification of Hierarchical Planning Problems. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI)*, 447–452. IOS Press.

Höller, D.; Behnke, G.; Bercher, P.; and Biundo, S. 2016. Assessing the Expressivity of Planning Formalisms through the Comparison to Formal Languages. In *Proceedings of the 26th International Conference on Automated Planning and Scheduling (ICAPS)*, 158–165. AAAI Press.

Höller, D.; Behnke, G.; Bercher, P.; and Biundo, S. 2021. The PANDA Framework for Hierarchical Planning. *Künstliche Intelligenz*, 30(1): 11–20.

Höller, D.; Behnke, G.; Bercher, P.; Biundo, S.; Fiorino, H.; Pellier, D.; and Alford, R. 2020. HDDL: An Extension to PDDL for Expressing Hierarchical Planning Problems. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, 9883–9891. AAAI Press.

Höller, D.; Bercher, P.; Behnke, G.; and Biundo, S. 2018. A Generic Method to Guide HTN Progression Search with Classical Heuristics. In *Proceedings of the 28th International Conference on Automated Planning and Scheduling (ICAPS)*, 114–122. AAAI Press.

Höller, D.; Bercher, P.; Behnke, G.; and Biundo, S. 2020. HTN Planning as Heuristic Progression Search. *Journal of Artificial Intelligence Research (JAIR)*, 67: 835–880.

Höller, D.; Wichlacz, J.; Bercher, P.; and Behnke, G. 2022. Compiling HTN Plan Verification Problems into HTN Planning Problems. In *Proceedings of the 32nd International Conference on Automated Planning and Scheduling (ICAPS)*, 145–150. AAAI Press.

Nederhof, M.-J. 2000a. Practical experiments with regular approximation of context-free languages. *Computational Linguistics*, 26(1): 17–44.

Nederhof, M.-J. 2000b. Regular approximation of CFLs: A grammatical view. In *Advances in Probabilistic and other Parsing Technologies*, chapter 12, 221–241. Kluwer Academic Publishers.

Richter, S.; and Westphal, M. 2010. The LAMA Planner: Guiding Cost-Based Anytime Planning with Landmarks. *Journal of Artificial Intelligence Research (JAIR)*, 39: 127–177.