Plan Recognition – Dissertation Abstract

Kristýna Pantůčková

Supervisor: Roman Barták
Charles University, Faculty of Mathematics and Physics
Ke Karlovu 3, 121 16 Praha 2, Czech Republic
pantuckova@ktiml.mff.cuni.cz

Abstract

The topic of the dissertation is plan recognition. Plan recognition is the task of recognizing the goal of an agent based on the observed actions. The aim of the current research is to develop an efficient approach to plan recognition in hierarchical task networks (HTN). We intend to improve the performance of existing parsing-based approach by heuristics based on landmarks.

Introduction

Plan recognition is relevant to many fields of artificial intelligence. For instance, plan recognition is related to behaviour recognition, which can be used to recognize suspicious behaviour in public space (Niu et al. 2004). In the field of computer security, plan recognition can be used to predict cybernetic attacks (Li et al. 2020). Other applications include multi-agent systems (Kaminka, Pynadath, and Tambe 2002), or artificial intelligence in computer games (Ha et al. 2011).

We focus on plan recognition in hierarchical task networks (HTN), which allow to express a natural hierarchy of tasks. A domain model of HTN planning consists of a set of abstract tasks, actions and methods. Abstract tasks can be decomposed into subtasks via methods. The aim of hierarchical planning is to decompose the given goal task into a sequence of actions (indecomposable tasks). In hierarchical plan recognition, we intend to find the goal task whose decomposition covers all of the observed actions. In contrast to plan verification, we do not expect that the sequence of observed actions given on input is a complete plan; the goal task will be decomposed into a sequence of actions which contains the set of observed actions as a subset.

Currently there appear to be only two approaches to recognition of hierarchical plans. The first of these approaches is based on compilation to HTN planning (Höller et al. 2018). The second approach (Barták, Maillard, and Cardoso 2020) was inspired by parsing of grammars. As the approach of (Barták, Maillard, and Cardoso 2020) performs worse than the approach of (Höller et al. 2018) on instances with a high number of missing (unobserved) actions, the aim of our current research is to improve the performance of the approach of (Barták, Maillard, and Cardoso 2020) by using landmarks. Our algorithm is based on composing tasks from subtasks until a goal task is found, and we intend to use method landmarks to guide the search. A fact landmark of a decomposition method m is a fact that must be true at some point in all plans created by decomposing the root task of m via m; a task landmark of m is an abstract task or an action which must be contained in all such plans.

Background on HTN plan recognition

Hierarchical planning focuses on planning problems where goals (tasks) can be hierarchically decomposed into subgoals (subtasks). Indecomposable (primitive) tasks are called actions. A planning problem can be described by a hierarchical task network (HTN).

An HTN is described by a pair $w = (T, C)$, where T is a set of tasks and C is a set of constraints over tasks. There are four types of constraints: $t_1 \prec t_2$ is a precedence constraint over tasks t_1 and t_2, $before(T', p)$ indicates that the proposition p must be true in the state before executing tasks in the set of tasks T', $after(T', p)$ indicates that p must be true in the state after executing tasks in T' and $between(T', T'', p)$ indicates that p must be true in all states between the sets of tasks T' and T''. Task decomposition is described by methods, where a method $m = (t, w)$ decomposes a task t to a hierarchical task network w.

A planning problem can be defined as $P = (F, C, A, M, s_0, w_0)$, where F is a set of fluents describing states, C is a set of compound (decomposable) tasks, A is a set of actions (primitive tasks), M is a set of decomposition methods, s_0 is an initial state and w_0 is the initial task network which represents the goal. Actions in A are defined by preconditions and positive and negative effects. Precondition of an action a is a proposition that must be true in order to execute a, positive effect is a proposition that will be true after executing a and negative effect is a proposition that will be false after executing a. The task of a planner is to decompose the tasks in the initial network to primitive tasks. If $w = (T, C)$ is a task network obtained from w_0 using methods from M, all abstract tasks in w are decomposed, and $\pi =< a_1, ..., a_k >$ are all actions in w, where the ordering of actions in π corresponds to the ordering of nodes in w and a_1 is executable in the state s_0, then π is a solution to the HTN planning problem P.

An HTN plan recognition problem is defined as $R = (F, C, A, M, s_0, O, G)$, where $O =< o_1, ..., o_k >$ is an observed plan prefix. The aim of plan recognition is to decide whether there is a goal $g \in G$ and a sequence of actions $< o_{k+1}, ..., o_n >$ such that $< o_1, ..., o_n >$ is a valid plan for the goal g applicable in s_0.

Related work

(Höller et al. 2018) developed an HTN plan recognition algorithm inspired by the "plan recognition as planning" approach of Ramírez and Geffner (Ramírez and Geffner 2009), who leveraged compilation to planning to recognize classical sequential plans. This compilation-based hierarchical plan recognition algorithm requires only one run of a hierarchical planner in a modified hierarchical task network. For an instance of an HTN plan recognition problem, (Höller et al. 2018) define a new goal, which can be decomposed into the initial network of one of the candidate goals, and introduce new constraints to ensure that the resulting plan will contain all observed actions.

(Barták, Maillard, and Cardoso 2020) proposed a different approach, which was inspired by parsing of grammars. Their algorithm firstly tries to find a goal task whose decomposition tree can cover all observations. If the task is not found, the algorithm guesses missing observations by adding all possible actions after the observed action sequence. Plan length is iteratively extended until a goal task is found. In this paper, the authors extended their older algorithm for HTN plan verification (Barták, Maillard, and Cardoso 2018), which described decomposition rules of an HTN planning domain by rewriting rules of attribute grammars.

(Barták, Maillard, and Cardoso 2020) also presented an empirical comparison of the two HTN plan recognition approaches. In contrast to (Höller et al. 2018), their algorithm does not require the initial state to be specified as part of the input as it can be computed during plan recognition. According to the empirical comparison presented in (Barták, Maillard, and Cardoso 2020), the parsing-based algorithm (Barták, Maillard, and Cardoso 2020) was faster than the compilation-based algorithm (Höller et al. 2018) on problem instances with only few missing observations. However, the authors of (Barták, Maillard, and Cardoso 2020) go on to observe that as the number of missing observations grows, the solving time of the parsing-based algorithm grows exponentially, while the compilation-based algorithm (Höller et al. 2018) performs significantly better.

Other hierarchical plan recognition approaches work with models weaker than HTN. For instance, there are approaches based on manipulations with tree-based structures, parsing or rewriting of strings (e.g. (Avrahami-Zilberbrand and Kaminka 2005), (Mirsky, Gal, and Shieber 2017), or (Geib, Maraist, and Goldman 2008)).

Currently, we aim to develop a landmark-based HTN plan recognition approach. Landmarks have already been used for classical plan recognition. The algorithm of (Pereira, Oren, and Meneguzzi 2017) finds the most likely goal by observing the landmarks that were achieved in the plan and comparing them with known landmarks of candidate goals. In comparison to an older approach based on compilation to planning (Ramírez and Geffner 2009), this landmark-based approach is significantly faster with a similar accuracy. (Pereira, Oren, and Meneguzzi 2017) proposed two heuristics for comparing candidate goals based on achieved landmarks. The basic heuristic computes the proportion of all landmarks and achieved landmarks, while the second heuristic, which leads to a better efficiency, takes into account "uniqueness" of landmarks among all goals.

(Vered et al. 2018) used landmarks combined with compilation to planning to develop an algorithm for on-line classical plan recognition, where time efficiency is crucial. Nevertheless, the authors utilize landmarks differently than (Pereira, Oren, and Meneguzzi 2017). After each new observation arriving in an on-line setting, they recompute optimal plans consistent with the observations. Compilation to planning is used to compute the probability distribution of candidate goals; landmarks are used to rule out improbable goals, for which this expensive computation is not necessary.

Current research

Previously we focused on survey of work related to the topic of the thesis – classical and hierarchical plan recognition. Currently we aim to develop an efficient algorithm for HTN plan recognition. Our approach is based on the algorithm of (Barták, Maillard, and Cardoso 2020). Instead of systematically generating all possible plans and composing abstract tasks from the available subtasks, we try to guess suitable abstract tasks which can be decomposed into some of the available tasks. We generate partial plans, which consist of actions, abstract tasks whose decomposition covers some of the observed actions, and extra tasks ordered after the observation sequence, which were generated by the abstract tasks. However, the implementation has not been finished yet. The idea of the algorithm is shown in Figure 1.

The procedure is described in Algorithm 1. The set S contains all generated partial plans. A partial plan contains the sequence of the observed actions and some abstract tasks that decompose into tasks in the partial plan. Additionally, abstract tasks may add some extra tasks which are not mapped to tasks in the plan. These tasks are ordered after the observed actions. The root task of a partial plan is a goal task if all observed actions are covered (all observed actions are contained in decomposition trees of some abstract tasks from the plan) and the new tasks after the plan can be ordered and decomposed to create a valid plan. For the validation of the latter condition, we need to call an HTN planner to decompose the new abstract tasks.

For a partial plan P, app_P denotes the set of methods that are applicable to P. A method m is applicable to a plan P if one of the potential first subtasks of m (one of the tasks that can be ordered as the first subtask in decomposition) is contained in the set of available (uncovered) tasks in P. Application of m to P is mapping of some of the subtasks of m to some of the available tasks in P. For each possible application, there will be one new partial plan. Each new partial plan will add the root task of m into the set of its available tasks. Some of the tasks available in P will become unavailable and m may add some new tasks which will not be mapped to tasks in P.

We intend to select the pair (P, m) based on a heuristic value. The heuristic function will depend on the proportion of landmarks of m achieved in P and the total number of

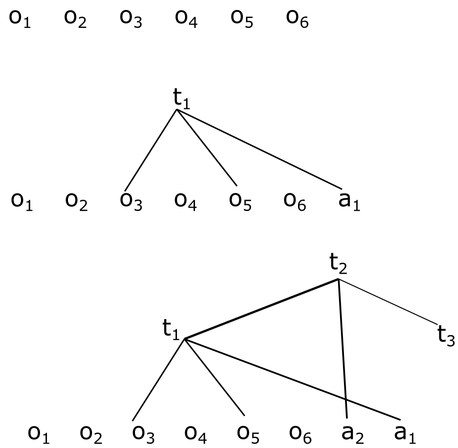

Figure 1: This figure describes the idea of our algorithm. In the first picture, there is an initial partial plan containing only the observed actions $o_1, ...o_6$. The second partial plan is the result of application of a decomposition method with the root task t_1. This method covers observations o_3 and o_5 and adds a new action a_1 after the plan. Application of the next method with the root task t_2 creates a partial plan with a new action a_2 and a new abstract task t_3. At this point, the order of a_1, a_2 and t_3 is not decided and t_3 is not decomposed; we try to resolve these problems only after a potential goal task covering all observations is found.

landmarks of m. For extracting landmarks of methods, we use the algorithm proposed by (Höller and Bercher 2021). In our settings, we expect a set of possible goal tasks as part of the input. For these tasks, we create AND/OR graphs and find landmarks of tasks, methods, actions and facts. Nevertheless, we will need only method landmarks in our heuristic function.

After creating a new partial plan, we use the procedure described in (Barták, Maillard, and Cardoso 2020) to check if all conditions of m are satisfied in the new plan. Our algorithm is clearly sound as if it returns a goal task t, t can be decomposed such that all observed actions are covered and the resulting plan is valid. However, the algorithm is not complete. We may move towards completeness for example by interleaving heuristic selections and random selections of plan-method pairs. Moreover, the solution will not be optimal (with respect to plan length). Quality of solutions may be improved by introducing a more complex heuristic function, which could depend for instance on length of a plan, number of uncovered actions, and number of new abstract tasks.

Future directions

Currently, we are working on implementation of our approach. We will compare our approach with the existing algorithms for HTN plan recognition ((Barták, Maillard, and Cardoso 2020) and (Höller et al. 2018)). Based on the results, we will focus on the heuristic function to improve the performance of the algorithm. In the future, we plan to deal with missing or incorrect observation, as our current approach requires a complete and correct plan prefix as an input.

Algorithm 1: Landmark-based HTN plan recognition

Input: a sequence of observed actions
Output: a corresponding goal task
Variables: S – a set of partial plans, app_P for each partial plan P – a set of all methods applicable to P

```
P_0 = initial partial plan containing observed actions
S = {P_0}
while true do
    P = argmax_{P∈S} max{h(m, P) | m ∈ app_P}
    m = argmax_m {h(m, P) | m ∈ app_P}
    for all possible applications of m to P do
        P_1 = apply m to P
        if P_1 is consistent with all conditions then
            if P_1 covers all observations and a valid plan can
            be generated from P_1 then
                return the root task of P_1
            else
                add P_1 to S
            end if
        end if
    end for
end while
```

Acknowledgements

Research is supported by the Charles University, project GA UK number 156121.

References

Avrahami-Zilberbrand, D.; and Kaminka, G. A. 2005. Fast and Complete Symbolic Plan Recognition. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 653–658.

Barták, R.; Maillard, A.; and Cardoso, R. 2018. Validation of hierarchical plans via parsing of attribute grammars. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 28.

Barták, R.; Maillard, A.; and Cardoso, R. C. 2020. Parsing-based Approaches for Verification and Recognition of Hierarchical Plans. In *Plan, activity and intent recognition workshop at the thirty-fourth AAAI Conference on Artificial Intelligence*.

Geib, C. W.; Maraist, J.; and Goldman, R. P. 2008. A New Probabilistic Plan Recognition Algorithm Based on String Rewriting. In *Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling*, 91–98.

Ha, E.; Rowe, J.; Mott, B.; and Lester, J. 2011. Goal recognition with Markov logic networks for player-adaptive games. In *Proceedings of the seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 6.

Höller, D.; Behnke, G.; Bercher, P.; and Biundo, S. 2018. Plan and goal recognition as HTN planning. In *2018 IEEE*

Thirtieth International Conference on Tools with Artificial Intelligence, 466–473.

Höller, D.; and Bercher, P. 2021. Landmark Generation in HTN Planning. In *Proceedings of the thirty-fifth AAAI Conference on Artificial Intelligence (AAAI)*, 11826–11834.

Kaminka, G. A.; Pynadath, D. V.; and Tambe, M. 2002. Monitoring teams by overhearing: A multi-agent plan-recognition approach. *Journal of Artificial Intelligence Research*, 17: 83–135.

Li, T.; Liu, Y.; Liu, Y.; Xiao, Y.; and Nguyen, N. A. 2020. Attack plan recognition using hidden Markov and probabilistic inference. *Computers & Security*, 97: 101974.

Mirsky, R.; Gal, Y.; and Shieber, S. M. 2017. CRADLE: an online plan recognition algorithm for exploratory domains. *ACM Transactions on Intelligent Systems and Technology*, 8(3): 1–22.

Niu, W.; Long, J.; Han, D.; and Wang, Y.-F. 2004. Human activity detection and recognition for video surveillance. In *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME)*, volume 1, 719–722.

Pereira, R. F.; Oren, N.; and Meneguzzi, F. 2017. Landmark-based heuristics for goal recognition. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 3622–3628.

Ramírez, M.; and Geffner, H. 2009. Plan recognition as planning. In *Proceedings of the twenty-first International Joint Conference on Artifical Intelligence*, 1778–1783.

Vered, M.; Pereira, R. F.; Kaminka, G.; and Meneguzzi, F. R. 2018. Towards online goal recognition combining goal mirroring and landmarks. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, 2112–2114.