LTP: Lifted Tree Path

Gaspard Quenard, Damien Pellier, Humbert Fiorino
Univ. Grenoble Alpes - LIG
F-38000 Grenoble, France

Abstract

In this paper, we present our planner named LTP which stand for Lifted Tree Path aimed at solving Totally Ordered Hierarchical Task Network (TOHTN) problems. Our planner is based on the Satisfiability (SAT) planning paradigm and builds upon the concepts of the Lilotane planner (Schreiber [2021]), which has scored 2nd in the last IPC in the HTN Total Order track.

Introduction

Satisfiability (SAT) planning is a widely-used planning paradigm that employs Boolean satisfiability solvers to find solutions for planning problems (Kautz et al. [1992, 2006]). SAT solvers are efficient tools for solving propositional logic problems. The main challenge in SAT planning lies in identifying and formulating the appropriate set of rules and constraints that effectively encode a given planning problem into SAT clauses. Once the planning problem is encoded into SAT clauses, the solution process relies on the underlying SAT solver to efficiently search for satisfying assignments.

Several SAT planners have been developed to encode TOHTN problems (Behnke et al. [2018], Schreiber et al. [2019], Schreiber [2021]). These planners utilize a structure referred to as a hierarchical tree to represent the problem hierarchy up to a certain depth. This hierarchical tree is subsequently used to encode the set of relevant SAT clauses.

The difference between previous approaches and LTP (Lifted Task Planning) is that the latter does not directly encode the entire hierarchy of the problem into propositional logic. Instead, it selectively extracts only the primitives from the hierarchical tree that may appear in valid plans and encodes them into propositional logic. It focuses solely on the actions of the plan rather than the full hierarchy. Therefore, LTP does not utilize boolean variables to encode tasks or methods during the SAT clause encoding process.

Hierarchical Tree

LTP utilizes the same hierarchical tree structure as the lilotane and TreeRex planners.

The hierarchical tree can be described as a sequence of hierarchical layers, where each layer is an array of positions, each containing a set of elements. These elements can be facts, reductions, or actions. The layers are computed incrementally, starting with an initial layer (L0) that includes the initial reduction. Subsequently, each layer is defined by including all operations that match a subtask of some operation from the previous layer.

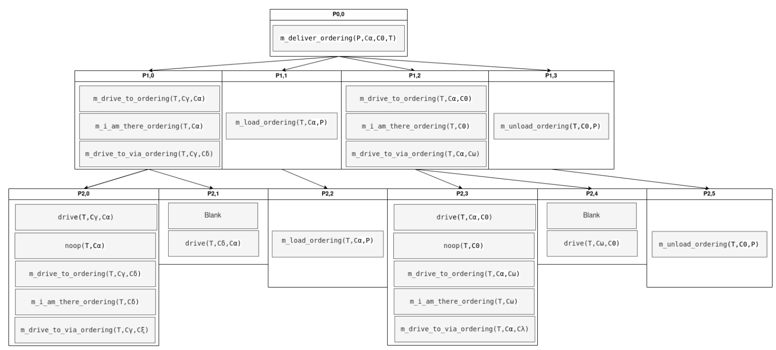

Figure 1: Example a hierarchial tree containing 3 hierarchical layers for a problem of the domain Transport as defined in the Lilotane paper. The first subtask of the method m_deliver_ordering can be accomplish by the three methods reported in the position P1,0)

Figure 1 illustrates an example of a hierarchical tree containing three layers for a problem in the Transport domain, as defined in the Lilotane paper. In this example, Lilotane encodes the entire decomposition tree into SAT clauses. However, LTP differs by keeping only the last layer of the decomposition tree. From this layer, it encodes only the actions that may be part of a solution plan. The ordered constraints between these actions can be inferred from the hierarchical tree, and the method's preconditions can be encoded to the relevant actions in their first subtask.

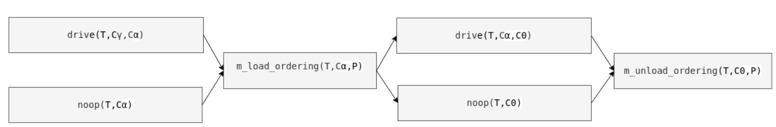

Figure 2: Space of reseach encoded by LTP into SAT clauses

Instantiation

The general planning procedure of LTP is similar to the other SAT planners for TOHTN problems:

1. Initialize the first layer (l0) of the hierarchial tree following the problem description.
2. Construct the next layer (l+1) of the hierarchical tree on the basis of the layer l.
3. Use the current hierarchical tree to encode the SAT clauses.
4. Launch the solver. If no solution is found, goto 2

References

Dominik Schreiber. Lilotane: A lifted sat-based approach to hierarchical planning. *Journal of artificial intelligence research*, 70:1117–1181, 2021.

Henry A Kautz, Bart Selman, et al. Planning as satisfiability. In *ECAI*, volume 92, pages 359–363. Citeseer, 1992.

Henry Kautz, Bart Selman, and Joerg Hoffmann. Satplan: Planning as satisfiability. In *5th international planning competition*, volume 20, page 156, 2006.

Gregor Behnke, Daniel Höller, and Susanne Biundo. totsat-totally-ordered hierarchical planning through sat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Dominik Schreiber, Damien Pellier, Humbert Fiorino, et al. Tree-rex: Sat-based tree exploration for efficient and high-quality htn planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, pages 382–390, 2019.