Planning Graph, GraphPlan and SATPlan. Three ideas that shaped AI planning.

In this review paper we summarize the concept of a planning graph, graphplan, and SAT based systems for solving problems in artificial intelligence (AI) planning. We further highlight the relationships between these ideas and their importance to the field of AI planning.

A classical planning problem begins with a state of the world, a goal state which is to be attained and a set of possible actions that can be executed to get to the goal. The idea of a planning graph was introduced in [2] and for so called STRIPS planning problem facilitates the extraction of an optimal plan if it exists. The planning graph is a leveled graph that is built iteratively. The levels alternate between propositions and actions in the planning problem. The zeroth level consists of atomic propositions that are true in the initial state, the first level consists of actions for which some proposition is the zeroth level is a precondition, and the next level consists of the propositions which are a result of the applying the actions in the first level to the propositions in the zeroth level. The graph is built continually in this way until there are two consecutive proposition levels with the same set of propositions. There is also a do nothing (noop) action for every proposition that does nothing. An important concept is that of mutual exclusivity, basically two actions in the same level are mutually exclusive if no actual plan can contain both of them, while two propositions are mutually exclusive if there are no sequences of actions that can make them both true. Note that this concept is based on the planning graph construct itself and not an absolute logical property of the planning problem. The last step in constructing a planning graph is to identify at every level and every node, mutually exclusive nodes at the same level.

From the planning graph, one can extract a solution if it exists by backtracking, using the graphplan algorithm also introduced in [2]. The high-level description of the basic algorithm is the following. Starting with a Planning Graph that only has a single proposition level containing the Initial Conditions, Graphplan runs in stages. In stage i Graphplan takes the Planning Graph from stage $i-1$, extends it one time step (the next action level and the following proposition level), and then searches the extended Planning Graph for a valid plan of length i. Graphplan search either finds a valid plan (in which case it halts) or else determines that the goals are not all achievable by time i (in which case it goes on to the next stage). Thus, in each iteration through this Extend/Search loop, the algorithm either discovers a plan or else proves that no plan having that many time steps or fewer is possible [2].

Closely related to graphplan is the SATplan approach to classical planning [1]. In this approach one maps a planning problem to a SAT problem. One either shows that the planning problem is solvable by proving that the SAT problem is satisfiable or vice-versa. If there is a solution, in the constructive approach [1], one shows that a solution to the planning problem can be extracted in polynomial time from any truth assignment of the SAT problem. The key to success in this approach is to choose an efficient encoding so that the fastest systematic and stochastic SAT algorithms can be applied to solve the SAT problem. A planning graph can be converted automatically into CNF notation for solution with SAT solvers by constructing propositional formulas stating the following: (1) The (fully specified) initial state holds at level zero, and the goal holds at the highest leve. (2) Conflicting actions are mutually exclusive. (3) Actions imply their preconditions. (4) Each fact at a positive even level implies the disjunction of all actions at the previous level that have this fact in their effects (including a maintenance action if it exists) [1]. The BLACKBOX system [3] used this GRAPHPLAN-based encoding to provide a very fast planner.

The planning graph is a very useful idea, for example, as explained above it leads directly to graphplan which is closely related to SAT plan. Furthermore as we have seen in class the planning graph can be used to construct heuristics which can be used in other tradition graph search techniques. The graphplan and SATplan approaches to classical planning enabled the solution to planning problems that would otherwise be very difficult and expensive to solve. For example, the BLACKBOX planner required only 6 minutes to find a 105-action logistics plan in a world with 10^{16} possible states [3].

References

[1] Computational Challenges in Propositional Reasoning and Search. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, 5054. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

[2] Blum, A., and Furst, M. 1997. Fast Planningthrough Planning Graph Analysis. Journal of Artificial Intelligence 90(12): 281300.

[3] BLACKBOX: A New Approach to the Application of Theorem Proving to Problem Solving. In AIPS98 Workshop on Planning as Combinatorial Search, 5860. Pittsburgh, Penn.:Carnegie Mellon University