Research Review - Planning

Overview

This aticle introduces five major developments in the field of AI planing research and how they had an impact on AI field as a whole.

1. STRIPS (1971)

STRIPS (Stanford Research Institute Problem Solver) is an automated planner developed by Richard Fikes and Nils Nilsson in 1971 at SRI International(Fikes and Nilsson, 1971). The model attempts to find a sequence of operators in a space of world models to transform the initial world model into a model in which the goal state exists. It attempts to model the world as a set of first-order predicate formulas and is designed to work with models consisting of a large number of formulas.
In the STRIPS formulation, we assume that there exists a set of applicable operators which transform the world model into some other world model. The task of the problem solver is to find a sequence of operators which transform the given initial problem into one that satisfies the goal conditions. Operators are the basic elements from which a solution is built. Each operator corresponds to an action routine whose execution causes the agent to take certain actions. In STRIPS, the process of theorem proving and searching are separated through a space of world models.

2. ADL(1987)

ADL (Action Description Language) is one of STRIPS extensions which removed some of its constraints to handle more realistic problems. Edwin Pednault proposed this language in 1987(Edwin Pednault, 1987). Unlike STRIPS, ADL doesn't assume that unmentioned literals are false, but rather unknown, what is better known as the Open World Assumption. It also supports negative literals, quantified variables in goals, conditional effects and disjunctions in goals (all not allowed in STRIPS).

3. Planning Graphs (1997)

In 1997, Avrium Blum and Merrick Furst at Carnegie Mellon developed a new approach to planing in STRIPS-like domains(Avrium Blum, 1997). It involved constructing and analyzing a brand new object called a Planning Graph. They developed a routine called GraphPlan which obtains the solution to the planning problem using a Planning Graph construct. The idea is that rather than greedily searching, we first create a Planning Graph object. The Planning Graph is useful because it inherently encodes useful constraints explicitly, thereby reducing the search overhead in the future. Planning Graphs can be constructed in polynomial time and have polynomial size. On the other hand, the state space search is exponential and is much more work to build. Planning graphs are not only based on domain information, but also the goals and initial conditions of the problem and an explicit notion of time.
Planning Graphs have similar features to dynamic programming problem solvers. The GraphPlan algorithm uses a planning graph to guide its search for a plan. The algorithm guarantees that the shortest plan will be found (similar to BFS).

4. PDDL(1998)

The Planning Domain Definition Language (PDDL) is an attempt to standardize Artificial Intelligence (AI) planning languages. It was first developed by Drew McDermott and his colleagues in 1998 (inspired by STRIPS and ADL among others) mainly to make the 1998/2000 International Planning Competition (IPC) possible, and then evolved with each competition. It was an attempt to standardise planning languages what made International Planning Competition (IPC) series possible. In other words PDDL contains STRIPS, ADL and much more other representational languages.

5. Heuristic Search Planner (HSP) (1998)

HSP is based on the idea of heuristic search. A heuristic search provides an estimate of the distance to the goal. In domain independent planning, heuristics need to be derived from the representation of actions and goals. A common way to derive a heuristic function is to solve a relaxed version of the problem. The main issue is that often the relaxed problem heuristic computation is NP-hard.
The HSP algorithm instead estimates the optimal value of the relaxed problem. The algorithm transforms the problem into a heuristic search by automatically extracting heuristics from the STRIPS encodings.

Reference

- Russell, S, and P Norvig. Artificial Intelligence: A Modern Approach, Chapter 10: Classical Planning
- Historical intro to AI planning languages, https://machinelearnings.co/historical-intro-to-ai-planning-languages-92ce9321b538
- AI Planning Historical Developments, https://towardsdatascience.com/ai-planning-historical-developments-edcd9f24c991
- STRIPS, https://en.wikipedia.org/wiki/STRIPS
- Action description language, https://en.wikipedia.org/wiki/Action_description_language
- Graphplan, https://en.wikipedia.org/wiki/Graphplan
- Planning Domain Definition Language, https://en.wikipedia.org/wiki/Planning_Domain_Definition_Language