History of Planning

Automated planning and scheduling is one of the major fields of AI. The task of coming up with a sequence of actions that will achieve a goal is called planning. Automated planning and scheduling, sometimes denoted as simply planning is a branch of artificial Intelligence that concerns the realization of strategies or action sequences, typically for execution by intelligent agents, autonomous robots and unmanned vehicles. Unlike classical control and classification problems, the solutions are complex and must be discovered and optimized in multidimensional space. Planning is also related to decision theory.

AI planning arose from investigations into state-space search, theorem proving, and control theory and from the practical needs of robotics, scheduling, and other domains. Shakey, developed at the Stanford Research Institute (SRI) from 1966 to 1972, was the first mobile robot to reason about its actions. Although not a practical tool, it led to advances in AI techniques, including visual analysis, route finding, and object manipulation. Researchers developed a sophisticated software search algorithm called "A*" that would also work for more complex environments. Today A* is used in applications such as understanding written text, figuring out driving directions, and playing computer games.

A planning system called STRIPS ("Stanford Research Institute Problem Solver") reasoned about complicated goals. This language is the base for most of the languages for expressing automated planning problem instances in use today; such languages are commonly known as action languages. The sense of a planning language is to represent certain conditions in the environment and, based on these, automatically generate a chain of actions which lead to a desired goal. A goal is a certain partially specified condition. Before an action can be executed its preconditions must be fulfilled; after the execution the action yields effects, by which the environment changes. The environment is described by means of certain predicates, which are either fulfilled or not.

STRIPS language was a good starting point for planning problems representation but there was room for improvements. ADL (Action Description Language) is one of STRIPS extensions which removed some of its constraints to handle more realistic problems. Unlike STRIPS, ADL doesn't assume that unmentioned literals are false, but rather unknown, what is better known as the Open World Assumption. STRIPS only positive literals and conjunctions are permitted; ADL allows negative literals and disjunctions as well. STRIPS and ADL were inspiration for another extension of representational languages — PDDL. It was an attempt to standardize planning languages. The adoption of a common formalism for describing planning domains fosters far greater reuse of research and allows more direct comparison of systems and approaches, and therefore supports faster progress in the field

Planning combines the two major areas of AI: search and logic. That is, a planner can be seen either as a program that searches for a solution or as one that (constructively) proves the existence of a solution. The cross-fertilization of ideas from the two areas has led to both improvements in performance amounting to several orders of magnitude in the last decade and an increased use of planners in industrial applications. Planners such as GRAPHPLAN, SATPLAN, and BLACKBOX have moved the field of planning forward, both by raising the level of performance of planning systems and by clarifying the representational and combinatorial issues involved

References

- Artificial Intelligence: A Modern Approach, Russell, S, and P Norvig.
- http://www.computerhistory.org/revolution/artificial-intelligence-robotics/13/289
- https://en.wikipedia.org/wiki/Automated_planning_and_scheduling