Historical developments in the field of AI planning and search.

The problem planning and search are one of the fields in AI that concerns the realization to perform decision making to find sequences of actions lead from the initial state to achieve the goal for a given tasks and conditions. Many developments contributed to growth impact on the field of AI as a whole. In these following are listed some of them.

STRIPS (Stanford Research Institute Problem Solver)

The automated planning algorithm developed by Richard Fikes and Nils Nilsson in 1971 at SRI International.[2] STRIPS can then search all possible states, starting from the initial one, executing various actions, until it reaches the goal, Typically STRIPS, this search is done backward from the goal state rather than forward from the current world state; the first is faster, but the second is more flexible when goals more complex. The impact of STRIPS in the AI field was later used to refer to the formal language of the inputs to this planner. This language is the base for most of the languages for expressing automated planning problem instances in use today.

PDDL (Planning Domain Definition Language)

The PDDL was introduced as a computer-parsable, standardized syntax for representing planning problems and has been used as the standard language for the International Planning Competition since 1998. The first was developed by Drew McDermott and his colleagues in 1998 (inspired by STRIPS and ADL among others)[3] There have been several extensions; the most recent version, PDDL 3.0, includes plan constraints and preferences (Gerevini and Long, 2005)[4]

Graphplan

The Graphplan is an algorithm for automated planning developed by Avrim Blum and Merrick Furst in 1995[5] The Graphplan approach is novel because it first formulates the problem in terms of a planning graph and then takes advantage of the features planning graphs to reduce the amount of search needed to find the solution of the state space. Graphplan is partial order planner functions, that its plans only make total order commitments at the level underneath which all actions may be parallelized given some initial conditions. at this high level, Graphplan guarantees that the shortest existing plan will be found. Graphplan is proven to be sound complete: any plan the algorithm finds is a legal plan, and if there exists a legal plan, then Graphplan will find one. Given a constructed planning graph, the algorithm performs backward-chaining to search for a plan. Also, given a problem with no solution, graphplan will eventually detect cyclic behavior and terminate

References

1. Stuart J. Russell, Peter Norvig (2010), Artificial Intelligence: A Modern Approach (3rd Edition).
2. Fikes, Richard E, and Nils J. Nilsson (1971). STRIPS: A New Approach to the Application of
   Theorem Proving to Problem Solving.
3. McDermott, Drew; Ghallab, Malik; Howe, Adele; Knoblock, Craig; Ram, Ashwin; Veloso, Manuela; Weld, Daniel; Wilkins, David (1998). PDDL---The Planning Domain Definition Language. Technical Report CVC TR98003/DCS TR1165. New Haven, CT: Yale Center for Computational Vision and Control.
4. Gerevini, A. Long, D. (2005). Plan Constraints and Preferences in PDDL3. Technical Report R. T. 2005-08-47. Dipartimento di Elettronica per l'Automazione, Università degli Studi di Brescia
5. A. Blum and M. Furst (1997). Fast planning through planning graph analysis. Artificial intelligence. 90:281-300.