Research Review

This review focuses on some of the important historical developments in the field of AI planning and search.

Stanford Research Institute Problem Solver (STRIPS)

STRIPS was the first major planning system and developed by Richard Fikes and Nils Nilsson in 1971. The goal of STRIPS is to find a sequence of operators that transforms an initial world model into one where some goal condition is satisfied. STRIPS uses a collection of first-order predicate calculus formulas and means-ends analysis to derive the goal-satisfying model. [1] STRIPS's main impact came from the representation language rather than the algorithm itself. It became the basis for representation planning language and what is known today as "classical" language is similar to what was used for STRIPS. [2]

Problem Domain Description Language (PDDL)

Inspired by STRIPS and Action Description Language (ADL), Drew McDermott and his team introduced the Problem Domain Description Language (PDDL) in 1981.[3] PDDL was developed to standardize the syntax for representing planning problems. PDDL is computer-parsable and has the ability to express objects in a domain, parameters in actions, preconditions and more through a type structure. Since 1998, PDDL has become the standard language for the International Planning Competition.

WARPLAN

In order to be classified as a complete planner, there must be an "interleaving of actions from different subplans within a single sequence". [2] Goal-regression planning solves this problem by reordering steps in a completely ordered plan in order to avoid subgoal conflicts. David Warren employs this technique in WARPLAN to use chronological backtracking. WARPLAN is famous for being the first planner to be written in Prolog and for only being 100 lines of code.

References

[1] http://ai.stanford.edu/~nilsson/OnlinePubs-Nils/PublishedPapers/strips.pdf

[2] Stuart J. Russell, Peter Norvig (2010), Artificial Intelligence: A Modern Approach (3rd Edition).

[3] https://en.wikipedia.org/wiki/Planning_Domain_Definition_Language