Table Of Contents

STRIPS 1
GRAPHPLAN 1
HEURISTICS SEARCH PLANNER 1
WORKS CITED 2

STRIPS

STRIPS (**ST**anford **R**esearch **I**nstitute **P**roblem **S**olver) is a problem-solving program and a formal-language developed in 1971 for the purposes of AI Planning. STRIPS language is used today as the base for the most languages for expressing automated planning problem (action languages).

The problem space for STRIPS is defined by the initial world model, the set of available operators and their effects on the world model, and the goal statement [1, p. 191]

GraphPlan

Graphplan is an algorithm for automated planning developed by Avrim Blum and Merrick Furst in 1995. Graphplan takes as input a planning problem expressed in STRIPS and produces, if one is possible, a sequence of operations for reaching a goal state.

Planning Graph is a directed, leveled graph with two kinds of nodes and three kinds of edges. The levels alternate between proposition levels containing proposition nodes (each labeled with some proposition) and action levels containing action nodes (each labeled with some action). The first level of a Planning Graph is a proposition level and consists of one node for each proposition in the Initial Conditions. The levels in a Planning Graph, from earliest to latest are: propositions true at time 1, possible actions at time 1, propositions possibly true at time 2, possible actions at time 2, propositions possibly true at time 3, and so forth [2, p.4]

An integral part of Planning-Graph Analysis is noticing and propagating certain mutual exclusion relations among nodes. [2, p. 5]

GraphPlan alternates between two phases: graph expansion and solution extraction. The graph expansion phase extends a planning graph forward in time until it has achieved a necessary (but insufficient) condition for plan existence. The solution extraction phase then performs a backward chaining search on the graph looking for a plan that solves the problem if no solution is found the cycle repeats by further expanding the planning graph [3, p.3]

In many cases orders of magnitude faster than previous systems such as SNLP, Prodigy or UCPOP. [3, p.3]

Heuristics Search Planner

In the AIPS98 Planning Contest, the HSP planner showed that heuristic search planners can be competitive with state-of-the-art Graphplan and SAT planners. Heuristic search planners like

HSP transform planning problems into problems of heuristic search by automatically extracting heuristics from Strips encodings. They differ from specialized problem solvers such as those developed for the 24-Puzzle and Rubik's Cube in that they use a general declarative language for stating problems and a general mechanism for extracting heuristics from these representations. [4, p.5]

HSPr is a modification of HSP which addresses the need of re-computation of the heuristics from scratch in every new state (this bottleneck takes more than 80% of the total time and makes the node generation rate very low). [4, p.18]

The operation of the regression planner HSPr consists of two phases. In the first, a forward propagation is used to estimate the costs of all atoms from the initial state s_0 , and in the second, a regression search is performed using those measures. These two phases are in correspondence with the two operation phases in Graphplan where a plan graph is built forward in a first phase, and is searched backward for plans in the second. The two planners are also related in the use of mutexes, and idea that HSPr borrows from Graphplan. For the rest, HSPr and Graphplan look quite different. However, Graphplan can also be understood as an heuristic search planner with a precise heuristic function and search algorithm. [4, p. 28]

Works Cited

- [1] STRIPS, A new approach to the Application of Theorem Proving to Problem Solving. Richard E. Fikes, Nils J. Nillson. 1971. http://ai.stanford.edu/~nilsson/OnlinePubs-Nils/PublishedPapers/strips.pdf
- [2] Fast Planning Through Planning Graph Analysis. Avrim L. Blum, Merrick L. Furst. 1997. http://www.cs.cmu.edu/~avrim/Papers/graphplan.ps
- [3] Recent Advances in AI Planning. Daniel S. Weld. 1998. ftp://ftp.cs.washington.edu/pub/ai/pi2.ps
- [4] Planning as heuristic search. Blai Bonet, Héctor Geffner. 2000. http://www.cs.toronto.edu/~sheila/2542/s14/A1/bonetgeffner-heusearch-aij01.pdf