Tibor Zahorecz
Udacity Artificial Intelligence Nanodegree
History of AI Planning Systems

STRIPS

STRIPS (Stanford Research Institute Problem Solver) is an automated planner developed by Richard Fikes and Nils Nilsson in 1971 at SRI International.[1] In SRL AI Laboratory they focused their energies on a single experimental project in which a mobile robot was being developed that could navigate and push objects around in a multi-room environment. In particular, it provided the context and motivation for development of the STRIPS. STRIPS: A New Approach to the Application of .Theorem Proving to Problem Solving.[2] For searching through the space of world models, STRIPS uses a GPS-like means-end analysis strategy. (Instead, we have adopted the GPS strategy of extracting "differences" between the present world model and the goal and of identifying operators that are "relevant" to reducing these differences). This flexible search strategy embodied in STRIPS combines many of the advantages of both forward search (from the initial model toward the goal) and backward search (from the goal toward the initial model). STRIPS has a heuristic mechanism to select nodes with uncompleted successors to work on next. For this purpose we use an evaluation function that takes into account such factors as the number of remaining goals on the goal list, the number and types of predicates in the remaining goal formulas, and the complexity of the difference attached to the node. STRIPS was limited in both the scope of planning issues it is designed and the complexity of problems it could solve.

ADL, PDDL[3],[4]

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. It also supports negative literals, quantified variables in goals, conditional effects and disjunctions in goals (all not allowed in STRIPS). PDDL is an attempt to standardize Artificial Intelligence (AI) planning languages. Originally PDDL was first developed by Drew McDermott and his colleagues in 1998. PDDL has a long history and innovation path from PDDL 1.2 – PDDL 3.1. Successors/variants/extensions of PDDL: PDDL+, NDDL, MAPL, OPT… [5]

Graphplan [6]

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. Given a problem statement, Graphplan explicitly constructs and annotates a compact structure called a Planning Graph, in which a plan is a kind of "flow" of truth-values through the graph. Graphplan was created by ***Avrim Blum*** and ***Merrick Furst***, with subsequent extensions and improvements made by many researchers at many different institutions around the world. Graphplan' home directory.

[1] Wikipedia: https://en.wikipedia.org/wiki/STRIPS

[2] Richard E. Fikes, Nils J. Nilsson (Winter 1971). "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving http://ai.stanford.edu/~nilsson/OnlinePubs-Nils/PublishedPapers/strips.pdf

[3] Historical intro to AI planning languages https://machinelearnings.co/historical-intro-to-ai-planning-languages-92ce9321b538

[4] PDDL – The Planning Domain Definition Language http://homepages.inf.ed.ac.uk/mfourman/tools/propplan/pddl.pdf

[5] PDDL Wikipedia https://en.wikipedia.org/wiki/Planning_Domain_Definition_Language

[6] Graphplan Wikipedia https://en.wikipedia.org/wiki/Graphplan