Research Review

In this paper we'll be reviewing the following three seminal works in the field of Artificial Intelligence Planning:

- STRIPS
- Graphplan
- Satplan

STRIPS (Stanford Research Institute Problem Solver)

Developed in 1971, the STRIPS[1] system made a very important contribution to Planning research by introducing the Strips Assumption as a way to avoid the complexity of the frame problem for the purposes of planning within the constraints of situation calculus. The assumption is that the only changes that arise on application of an action to a situation are those that are explicitly mentioned as positive effects of the action. Planning problems are fundamentally dynamic in structure. It is therefore natural to interpret collections of action schemas as defining the transitions in a parameterised automaton and a plan as the transitions traversed by an accepting trace through the instantiated automaton, which is in stark contrast to the static view imposed by situation calculus. Both the dynamic and static views have influenced the design of algorithms for planning, although the dynamic view has dominated approaches taken to representation of planning problems.
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.[2]

Graphplan

Graphplan[3] 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. It searches for a plan in two stages: The first stage is the construction of a data structure, the plan graph, that efficiently represents information about what the agent could possibly achieve by executing actions from the initial state. The second stage searches backwards from the goals for a substructure within the plan graph that represents a subset of actions that will actually achieve the goals.
The algorithm iteratively extends the planning graph, proving that there are no solutions of length l-1 before looking for plans of length l by backward chaining: supposing the goals are

true, Graphplan looks for the actions and previous states from which the goals can be reached, pruning as many of them as possible thanks to incompatibility information.[4]

Satplan (Planning as Satisfiability)

Satplan[5] is a method for automated planning that converts the planning problem instance into an instance of the Boolean satisfiability problem, which is then solved using a method for establishing satisfiability such as the DPLL algorithm or WalkSAT. Given a problem instance in planning, with a given initial state, a given set of actions, a goal, and a horizon length, a formula is generated so that the formula is satisfiable if and only if there is a plan with the given horizon length. This is similar to simulation of Turing machines with the satisfiability problem in the proof of Cook's theorem. A plan can be found by testing the satisfiability of the formulas for different horizon lengths.

Satplan has the huge advantage that it can flexibly deal with conditions that need to be satisfied at any state at any point.

---

[1] Richard E. Fikes, Nils J. Nilsson (Winter 1971). "**STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving**"

[2] https://en.wikipedia.org/wiki/STRIPS

[3] A. Blum and M. Furst (1997). **Fast planning through planning graph analysis**

[4] https://en.wikipedia.org/wiki/Graphplan

[5] H. A. Kautz and B. Selman (1992). Planning as satisfiability. In Proceedings of the Tenth European Conference on Artificial Intelligence (ECAI'92)