AI Planning & Search: Major developments through time

Krishna Kumar

In this paper, we will be reviewing the major developments in the field of AI planning search, and the impacts they've had on the field of AI. We will start with STRIPS, the planning language, and will be continuing onward with planning graphs and heuristic-search planners.

STRIPS [1966-1972]: Coming in from the development of a robot system nicknamed "Shakey" at SRI Internationals AI Intelligence. STRIPS was an automated planner / language schema developed by Richard Fikes and Nils Nilsson in 1971.

STRIPS used initial states, goal states and a set of actions/operators, with each action having its own preconditions and postconditions. Modelling the world as a set of first-order predicate formulas, a STRIPS plan is a sequence of actions in order from the initial state, such that successive results from these actions satisfy the goal conditions. It left a lasting impact on how actions are represented, variants of the feature are present in almost all planning systems released since.

Planning Graphs [1997]: Developed by Avrium Blum and Merrick Furst at Carnegie Mellon in 1997. It involves constructing and analysing a graph network from the problems defined in STRIPS-like domains, it is based-on goals and initial conditions, has parallelly executable action pipelining with an explicit notion of time, mostly insensitive to goal ordering and guarantees finding the shortest path in time, should it exist*.

The utility of the planning graph lies in its ability to reduce the search overhead by constructing meaningful abstractions which can then represent the propagation of underlying constraints imposed in the problem. Since planning graphs are constructed in advance, the costs of propagating to a newer level for the search are lower, as the graph creation costs are provably in polynomial time, this makes them more useful for hard problems with smaller number of objects

Its efficiency is dependent on having a good selection of mutexes capturing all important constraints in the problem, and/or by allowing parallelizable actions.

Heuristic Search Planners [2000]: Coming in from the advances made in GRAPHPLAN that showed how useful reachability heuristics could be, Bonet and Geffner in their paper "*Planning as Heuristic Search*" showed that simpler reachability heuristics that assumed action preconditions were independent managed to best all other existing algorithms in a large set of problems. The planner computes costs from the ground up at every state, and therefore needs attention to runtime considerations, in ways such as improving mutexes, and/or moving backwards from the goal state.

References:

- Russel, S. J., & Peter, Norvig. (2018). ARTIFICIAL INTELLIGENCE: A Modern Approach. Chapter 11.
- Richard, E. F., & Nils J. N. (1971). "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving". Artificial Intelligence. 2 (3–4): 189–208
- [*] Blum, A., & Furst, M. (1997). Fast planning through planning graph analysis. Artificial intelligence. 90:281-300.
- Blai, Bonet. & Héctor, Geffner. (2001) Planning as heuristic search, Artificial Intelligence, Volume 129, Issues 1–2, Pages 5-33.