Shane Harmon Udacity AIND project 3

Research Review

To many, the term Artificial Intelligence will bring to mind a a computer playing chess, while the best chess (or GO) playing programs may not be strict state solvers,they were likely built from algorithms that try to find a goal state given a current state or an ensemble of algorithms. There have been many advances in the field of AI, but state planning still seems to be synonymous with general AI. Three advancements worth noting in the planning domain of AI are the advent of STRIPS, Planning Graphs, and the addition of Heuristic search to planning graphs. Each of these improved on it’s former and results in efficient domain independent planning.

STRIPS(Stanford Research Institute Problem Solver) was a major advance in the problem solving domain. It is a problem solver that uses predicate calculus, similar to first order logic, to define world states with preconditions and actions(1). STRIPS improved upon the earlier work of Green, by only using theorem proving for finding the validity of a the current state given applicable conditions and preconditions instead of trying to prove theorem validity in relation to the entire set of world models(1). The basic element of a STRIPS model is and Operator which corresponds to and action routine(1), each state is evaluated based on conditions and preconditions and actions that will potentially lead to a goal state are added to the current state to create a new state. Then the current state is evaluated. The output of STRIPS is a list of actions that will lead to a desired goal state, given that there is a possible list of actions that will reach to goal state.

A major improvement in problem solving was the introduction of Planning Graph Analysis, which works in STRIPS domains but loads operators into a Graph structure. This allows for relationships between nodes to be searched and thus an improvement in planning efficiency because less unnecessary expansions need to occur during the search for a plan. The Planning graph has two kinds of nodes and three kinds of edges(2). The proposition nodes represent a level where conditions and preconditions are evaluated for validity, and then possible actions are explored to see which current preconditions and conditions of adjacent layers will have an effect on each other, such as being mutually exclusive, once mutual exclusion is discovered, the edge connection the nodes may be removed from the graph(2). Action nodes represent a level in the graph in which actions are taken. The graph is traversed in this fashion, alternating between evaluation preconditions, and then possible actions, until a goal state is reached.

One improvement on Planning Graphs and STRIPS is the addition of heuristic search which allows the search to be conducted with an algorithms such as A* rather than simply Breadth First or Depth First searches. The upside of this is that it is guaranteed to find an optimal solution, the down side being that it may end up being slower or more resource intensive. The heuristic or cost function is found by relaxing the planning problem by ignoring delete lists, which the planning graph does, thus it doesn’t ignore any operators. Each level of the graph is expanded and each action is given a cost, the heuristic function adds the action costs in each node which informs the planner on the least expensive routes to nodes that could possible lead to a goal state, and the least expensive nodes are expanded first (3).

Though domain independent planning may not share the popular focus of machine and deep learning it is still very relevant to AI and is still being improved upon, and will play a large role in General AI in the future.

Fikes, R.E and Nilsson, N. J (1971) STRIPS: A New Approach to the. Application of .Theorem Proving
Blum, A. and Furst, M.(1997), Fast planning through planning graph analysis.
Blai, B and Geffner, H.(1998) HSP: Heuristic search planner