A review of selected pieces of work in the field of AI planning and search
By Rohit Patel

There is a vast body of work on the structured approaches to solving planning problems, dating back to Newwell and Simon, 1961. A modern planning problem solver typically consists of three elements: 1) A planning system or language, 2) A search algorithm, and 3) A heuristic function to drive the search. We briefly summarize three important pieces of work highlighting a key development in each aspect of a modern problem solver.

STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving **(Fikes and Nilsson, 1971)**
This paper introduces the first major planning system, called STRIPS, for problem solving that combines the formal theorem-proving methods, such as in Green, 1969, with a General Purpose Solver like control structure (Newwell and Simon, 1961). Theorem-proving methods are used only within a given world model to answer questions about it concerning which operators are applicable and whether or not goals are satisfied, whereas searching is performed using a GPS-like means-end analysis strategy. 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. The search is performed by a GPS strategy whereby a measure of difference between the current state and the goal is considered, and operators are identified that reduce the difference. Once such an operator is found, the current state is updated with the effect of the goal, and the recursive cycle is restarted with the new difference and set of operators. Fikes and Nilsson also refer to the use of heuristic function for operator (or node) selection as a potential extension of the model for future research.

Planning for Conjunctive Goals **(Chapman, 1987)**
Partial order planning is an approach to planning that leaves the decision about ordering of actions in a plan as open as possible, and specifies a relative order only when necessary. Chapman builds on the NOAH planner (Sacerdoti, 1975, 1977) to propose an algorithm TWEAK and proves it's completeness. TWEAK is a constraint-posting partial order planner, in that it incrementally defines the constraints a solution must fit, until a goal is achieved. TWEAK uses a dependency directed backtracking. In essence, progress toward each goal is built independently by satisfying the preconditions, and ordering constraints are added when necessary.

Fast Planning Through Planning Graph Analysis **(Blum and Furst, 1997)**
This paper introduces planning graphs for STRIPS-like domains, and describes a new planner, Graphplan, that uses search on the planning graph to extract a solution.Graphplan always returns a shortest possible partial order plan, or states that no plan exists. The authors compare Graphplan with two popular planners at the time, Prodigy, and UCPOP with encouraging results. The authors surmise that the four key factors accounting for Graphplan's efficiency are mutual exclusion (where pairwise mutual exclusions account for most of the important constraints), consideration of parallel plans (where valid parallel plans are relatively short compared to fully ordered plans), memoizing (where graphplan can prove certain goals unattainable in a certain number of steps from initial condition using the planning graph), and low-level costs (where the use of planning graph eliminates the need for recurrent instantiations during search). Much of the novelty of the algorithm comes from the fact that the planning graphs level-off relatively quickly, thus reducing the search space to a manageable level.