Research Review

In this research review, I survey several developments in languages and algorithms in planning and search field.

STRIPS: Stanford Research Institute Problem Solver

STRIPS is the first major planning system. It is an automated planner developed by Fikes and Nilsson in 1971, and modeled on a state-space search system, the General Problem Solver (Newell and Simon, 1961). It is composed of an initial state, a specification of goal states, and a set of actions. Actions have preconditions and postcondition (or effects) and can change states accordingly. The planning system uses actions to change/search state-space such that it moves from the initial state to the goal state.

The STRIPS system's major influence is actually its representation language, and inspired other attempts, such as the Planning Domain Definition Language

PDDL: the Planning Domain Definition Language

Developed by McDermott and his colleagues (1998), the PDDL extended the language used by STRIPS. It is a computer-parsable, standardized syntax for representing planning problems. It separated the model of the planning in two major parts: 1) domain description and 2) problem description. The domain description specifies predicates and actions. The problem description specifies objects, initial state, and goal specification.

PDDL has been used as the standard language for the International Planning Competition since 1998. During the years, it has developed and incorporated more features, such as numeric fluents, which model resources that have multiple levels, object-fluents, which extends the range of functions beyond numbers, and other features. The most recent version is now PDDL 3.1.

PDDL is important mainly because it formalize the planning problem with a standard language, and provides other programs or algorithms an abstract interface to tackle the planning problem.

POP: Partial-order Planning

Partial-order planning is a set of algorithms that "search in plan space and use least commitment, when possible"" (Veloso, 2001). In contrast to simple forward and regression planners, whose

search enforces a total ordering on actions, the partial-order planning usually leave actions unordered, or provide temporal constraints to show to actions are ordered, but not immediately before one or the other. This partial-order leads to a **non-linear** planning strategy to solve the problem. It starts with a *start* action that leads to the initial, and gradually add new temporal action constraints `<a0, P, a1>` that specifies, `a0` should happen before `a1` and `P` should be achieved by `a0` and `a1` should be `P` 's result. Gradually add in these temporal constrains can construct a graph structure, and an action that deletes `P` should come before `a0` or after `a1`. The planner search until no conditions exists before the goal state.

The Partial-order planning dominated for 20 years of research, and is proved to be complete. It also helps to prove the completeness and intractability (NP-hardness and undecidability) of various planning problems (Norvig and Russell, 2009).

New Planners: FF and Fast Downard

FF (Fast-Forward) (Hoffman, 2001) and Fast Downward (Helmert, 2006) algorithms are forward direction state-space search algorithms to solve planning problems.

FF has an outstanding performance at the 2nd International Planning Competition (IPC) and 3rd IPC. It is a state-space planner with heuristics. It simplifies a task at hand by ignoring the negative effects of actions and finds an explicit solution for this simplification by using a GRAPHPLAN algorithm. The solution is treated as a goal-distance estimation and guides a heuristic search among state-space.

Fast Downward algorithm is the winner of 2004 planning competition. It translate the propositional PDDL representation into an alternative multi-valued representation to make several implicit constraints explicit. Using this new representation, Fast Downward constructs a causal graph heuristic to estimate goal-distance estimation (Helmert 2006).

These two algorithms represents new state-space search algorithms that performs better than POP, and recently there are lots of investigations along this line.