Implement a planning search

Research review

AIND Nanodegree

Michail Kovanis

In this literature review in the field of AI planning and search I will summarize three main historical developments. First the STanford Research Institute Problem Solver (Fikes & Nilson 1971, Artificial Intelligence), second the Planning Domain Definition Language (Ghallab et al. 1998), and third the Graphplan algorithm (Blum & Furst 1997, Artificial Intelligence).

STRIPS

STRIPS can be described by its problem space and search strategy. The problem space is defined by three main elements; an initial world model, a set of operators, and a goal condition. All three of them represented in well-formed formulas (wff). The initial world model represents the starting state of all agents and objects along with any exclusion statements (e.g. different objects or agents cannot be at the same time in the same place). The set of operators contains all possible actions that different agents can perform. These actions have certain preconditions that is necessary to be present in the current world model in order to happen and certain effects (types of wffs, which will be added and/or removed to/from the world model after an action). The goal condition is a set of wffs, which when they appear in the current world model, the planning search is terminated. Finally, the approach that STRIPS follows to solve a problem is to form a search tree and then use the General Problem Solving strategy to identify the actions that produce differences between the world model and the goal state (after an action is applied) and then to apply them.

PDDL

The Planning Domain Definition Language is a standardized formalism, in which one can represent every problem in the domain of AI planning and search. It is designed to express world models, actions, preconditions and effects of actions, and goal states. Moreover, in cases in which the problem requires the extension of the language, no specific guidelines are given for that and therefore one may extend it in the way that it is most appropriate to the problem. An example of the language can be seen below (Veloso 2002):

```
(define (problem strips-gripper2)
   (:domain gripper-strips)
   (:objects rooma roomb ball1 ball2 left right)
   (:init (room rooma)
         (room roomb)
         (ball ball1
         (ball ball2)
         (gripper left)
         (gripper right)
```

```
        (at-robby rooma)
        (free left)
        (free right)
        (at ball1 rooma)
        (at ball2 rooma))
    (:goal (at ball1 roomb)))
```

Graphplan

The Graphplan applies in STRIPS-like domains and is based on the Planning Graph Analysis paradigm. This planner, instead of trying to build a complete search tree, builds a Planning Graph at each step of the search in order to guide it. Planning Graphs have the advantage of expanding significantly less nodes than a complete search tree, they are of polynomial size, and they are built in polynomial time. A Planning Graph is directed and levelled. It includes proposition levels (containing proposition nodes), which alternate with action levels (containing action nodes). Every action node of level i is connected to its preconditions at proposition level i and to its Add-Effects and Delete-Effects states at proposition level i+1. Moreover, a Planning Graph includes mutual exclusion (mutex) relations between nodes. Mutex relations between actions and between propositions exist when no valid plan could include both actions or make both states true. Graphplan starts from the initial state of the problem. The algorithm extends a level i by applying all possible actions and constructing all possible states resulting from these actions at level i+1. Then, it searches between the new states to find whether the goal was reached or not. If not it continues the expansion. Finally, in the case that the algorithm starts producing the same propositional levels it is terminated.

References

1. Fikes RE, Nilsson NJ. STRIPS: A new approach to the application of theorem proving to problem solving. Artificial intelligence. 1971 Dec 1;2(3-4):189-208.
2. McDermott D, Ghallab M, Howe A, Knoblock C, Ram A, Veloso M, Weld D, Wilkins D. PDDL-the planning domain definition language. URL: http://homepages.inf.ed.ac.uk/mfourman/tools/propplan/pddl.pdf
3. Veloso M. PDDL by Example. Carnegie Mellon University. URL: http://www.cs.toronto.edu/~sheila/384/w11/Assignments/A3/veloso-PDDL_by_Example.pdf
4. Blum AL, Furst ML. Fast planning through planning graph analysis. Artificial intelligence. 1997 Feb 28;90(1):281-300.