Research Review

This research review looks at three developments in Artificial Intelligence planning and search:

- STRIPS: A New Approach to Application of Theorem Proving to Problem Solving [1]
- PDDL - The Planning Domain Definition Language [2]
- Planning as Heuristic Search: New Results [3]

In [1], Fikes and Nilsson described a problem-solving program called STRIPS (STanford Research Institute Problem Solver). Fikes and Nilsson used a set of well-formed formulas (wffs) of the first-order predicate calculus to present a “world model”. Fikes and Nilsson assumed that there was a set of applicable operators that each could transform the world model to another world model. STRIPS is to discover a combination of operators which transform the initial world model to another that achieves a given goal: such as re-arranging objects and navigation. Apart from world model, the problem space in [1] is also defined by the set of available operators and the goal statement. An operator description defines an operator and includes two parts: the conditions under which the operator is applicable and the operator’s effects. The conditions can also be called precondition and stated as a wff schema. The effects can be defined by a list of wffs, some of which must be added to the world model and some are no longer true and should be deleted. The search strategy in [1] is to extract the differences between the world models (after applying an operator) and the goal, and identify relevant operators which can reduce the differences. Fikes and Nilsson presented the implementation of STRIPS and an example problem solved in [1].

STRIPS is “the first major planning system” and the way that it describes problem space and the corresponding representation language influenced AI planning and search field [4]. After STRIPS, the Action Description Language [5] was developed. It relaxes some restrictions from STRIPS and is possible to encode more realistic problem. Ghallab et al. introduced a computer-parsable language with standardized syntax in [2]: Problem Domain Description Language (PDDL). PDDL expresses a problem domain by presenting the existence of predicates, possible actions, the structure of compound actions, and action effects. PDDL does not provide annotations, such as advice about carrying out certain compound actions under certain circumstances, but allows planner to extend PDDL notation in different ways to enable annotations. It supports different syntactics features such as basic STRIPS-style actions, conditional effects, universal quantification over dynamic universes, domain axioms over stratified theories, specification of safety constraints, specification of hierarchical actions composed of subactions and subgoals. With the abundance of features, PDDL is factored into subsets of features which are called requirements. Problem domain should declare which requirements it assumes. Therefore, a planner can compare the requirements it handles and the domain assumes and skip over definitions connected to the requirements that it cannot handle. Examples and definitions of features are presented in [2]. PDDL has been used as the standard language for the International Planning Competition since 1998 [4]. Researchers did not stop working on PDDL and its most recent version is 3.0 [6].

Partial-order planning, which searches through the space of plans and solves independent subproblems, dominated research in 1980s and 90s. However, its disadvantage is that it lacks an explicit representation of states in the state-transition model. State-space planning with excellent heuristics reappeared in 1990s. Heuristic Search Planner (HSP) and its derivatives [3] were the first to present practical state-space search solution for large planning problems [4]. From the initial state, HSP searches forward, carries out actions based on the heuristics and reaches the goal. HSP can solve more problems than other planners such as Graphplan and Satisfiability but it takes a long time to run and may produce longer plans. In [3], Bonet et al. analyzed HSP and pointed out that the bottleneck was the computation of heuristic for every new state. Based on the same ideas and heuristic as HSP, the authors presented a new planner, HSP-R, which searched backward from the goal and computed heuristic once. It takes less time and can produce better plans. However, HSP-R is not always better than HSP as it cannot solve all problems in block-world as HSP can. Based on the strengths and weaknesses of both planners, Bonet et al. suggested to combine them together and produce a bi-directional search planner which could perform better.

[1] R. E. Fikes, N. J. Nilsson, “STRIPS: A New Approach to Application of Theorem Proving to Problem Solving”, AIJ, 2(3-4), pp. 189-208
[2] M. Ghallab, A. Howe, C. A. Knoblock, D. McDermott, A. Ram, M. Veloso, D. Weld, D. Wilkins, “PDDL - The Planning Domain Definition Language”, Tech. rep. DCS TR-1165, Yale Center for Computational Vision and Control
[3] B. Bonet, H. Geffner, “Planning as Heuristic Search: New Results”, ECP-99, pp. 360-372
[4] S. Russell, P. Norvig, “Classical Planning”, Artificial Intelligence A Modern Approach (third edition), Pearson, 2009, pp. 366-400
[5] E. P. D. Pednault, “Formulating Multiagent, Dynamic-World Problems in the Classical Planning Framework”, Reasoning about Actions and Plans: Proc. 1985 Workshop, pp. 47-82
[6] A. Gerevini, D. Long, “Plan Constraints and Preferences in PDDL3”, Tech. rep., Dept. of Electronics for Automation, University of Brescia, Italy