**Planning Problem – Research Review**

Rob Straker

**STRIPS**

Richard Fikes and Nils Nilsson developed the first major planning system, drawing on knowledge from a variety of domains, including state-space search, theorem proving, control theory, robotics and scheduling.[[1]](#endnote-1) Described as a ‘problem solving program’ by the authors, STRIPS (Stanford Research Institute Problem Solver) was initially written in LISP to run on a PDP-10 mainframe computer. Like other world model problem solvers, it was meant to search a space to find the world model in which a particular goal was achieved. It represented the initial world model as a collection of first-order calculus formulas, and attempted to find a sequence of operators which would transform the initial world into the world in which the goal is achieved.[[2]](#endnote-2)

In their retrospective, Fikes and Nilsson describe STRIPS as “often cited as providing a seminal framework for attacking the ‘classical planning problem’ ”, but acknowledge that the key technical contribution of STRIPs may have been its representation of actions (operators) and their effects.[[3]](#endnote-3) In fact, Stuart Russell and Peter Norvig make the case that this representation language has been far more influential than its framework for classical planning problems.i

**Linear Planning**

Another approach to planning was developed in the early 1970s, and involved completely ordered sequences of actions. Problems were solved by decomposing the problem into subplans, solving each of these for subgoals, and stringing the subplans together. i

Sacerdoti not only named this approach to planning as ‘Linear Planning’,[[4]](#endnote-4) but he also observed that plans are not bounded by the limitations of linearity, despite the popular notion that they consist of linear sequences of actions. He illustrated this with an example of 3 blocks and a table, which he described as an ‘anomalous situation’, and which others have called the Sussman anomalyi. Linear Planning programs fail to produce an optimal solution in this case, whereas breaking the goal into subgoals and partial ordering of activities into subplans does produce an optimal solution. Sacerdoti was a pioneer in constructing such partially ordered plans, then called ‘task networks’.i For this reason, Russell and Norvig describe Linear Planning as an ‘incomplete planner’, which leads to the important notion that any complete planner must support the ‘interleaving’ of actions from different subplans.i

**Binary Decision Diagrams**

Clarke and Grumberg were proponents of representing plans as ‘binary decision diagrams’, which are compact data structures that can be used to represent a Boolean function, and which were widely investigated within the domain of hardware verification.i

Binary Decision Diagrams (BDD), also known as branching programs, can be constructed by applying two rules to an ordinary binary decision tree:

1. Merge any isomorphic subgraphs
2. Eliminate any node whose children are isomorphic

By representing transition relations and sets of states with Binary Decision Diagrams, Clarke, Grumberg and Long found that concurrent systems with more than 10120 states could be searched, whereas previous techniques could handle only a few thousand states.[[5]](#endnote-5)

**Summary**

The innovations represented by STRIPS, Linear Planning and Binary Decision Diagrams emphasize the importance of representations and language to the domain of planning and search. Furthermore, they demonstrate the incredible value of bringing in developments from other domains to the field of AI planning and search.

1. Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach, 3/e* (New York: Prentice Hall, 2011), Kindle Edition, Chapter 10, Bibliographical and Historical Notes. [↑](#endnote-ref-1)
2. Richard E. Fikes and Nils J. Nilsson, “ STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving” (paper presented at the 2nd IJCAI, Imperial College, London, England, September 1-3, 1971). [↑](#endnote-ref-2)
3. Richard E. Fikes and Nils J. Nilsson, “STRIPS, a retrospective,” *Artificial Intelligence* 59 (1993): 227-232. [↑](#endnote-ref-3)
4. Earl D. Sacerdoti, “The Nonlinear Nature of Plans,” *JJ-CAI*-75 (1975): 206-214. [↑](#endnote-ref-4)
5. E. Clarke, O. Grumberg, and D. Long, “Verification Tools for Finite-State Concurrent Systems,” *Lecture Notes in Computer Science 803* (1994): 269-290. [↑](#endnote-ref-5)