Recent Advancements in AI Planning - Review

While the rapid rate of advancement in AI would imply that 40 years is far from recent, it would be irresponsible to leave the **STRIPS** representation model out of a discussion of the most impactful and exciting developments in the planning domain. The representation method is used in the vast majority of planning problems to create a solvable logic model out of real world problems. Originally designed as a general-purpose problem solver for robot tasks(1), STRIPS creates a set of well-formed formulas in first order logic representing the real-world state space, basic problem operators, their effects on the state space, and a goal condition to be programmatically achieved through planning. While the search algorithm itself was not necessarily the lasting gem in the STRIPS approach, the representation model was a crucial piece of the puzzle in solving planning problems, and went on to be a necessary stepping stone for many further advancements in the domain, including the next two mentioned in this review.

Our first example of a planning method in post-STRIPS world, Blum and Furst's **Graphplan** algorithm can be considered one of the most significant advancements that was critical in the formation of further planning architectures. Graphplan relates primarily to problems within a STRIPS representation domain and relies on the usage of graph expansion and solution extraction in alternation. Over the course of a single iteration, the graph of proposition and action nodes will be expanded by one layer or time-step and then the model will attempt solution extraction, it will then be decided whether a goal can be reached at that time-step. If no goal can be reached, the algorithm will iterate again and begin to expand the graph for the next time-step. The process of solution extraction utilizes a backward chaining search of the current expanded graph to guarantee that any possible solution in the graph will be found before further expansion (if necessary) is done. The results of this algorithm have proven it complete and more efficient than previous planning models, leading to several further optimizations building on this model(3).

Whereas Graphplan has proven to be an excellent "special purpose theorem prover" for solving planning problems, modern advancements in propositional satisfiability methods have shown promise for compilation to a **SAT**-based problem model to lead to even more efficient solving of STRIPS represented planning problems(2)(4). SAT-based models hold many similarities to Graphplan including iterative search and the use of propositional logic, SAT models differ by inputting a traditional planning problem and compiling the problem into a pure propositional logic formula. The model will then use a simplifier to reduce the formula to a more solvable problem in conjunctive normal form to be passed to a SAT-based solver that will search for a satisfying assignment to the formula. If the solver manages to find a satisfying assignment, this implies a feasible solution to the original problem and the assignment will be sent to a decoder to output a possible solution plan for the original input problem. When this returned solution is insufficient, the process will begin again with a longer search path length to find a new plan.

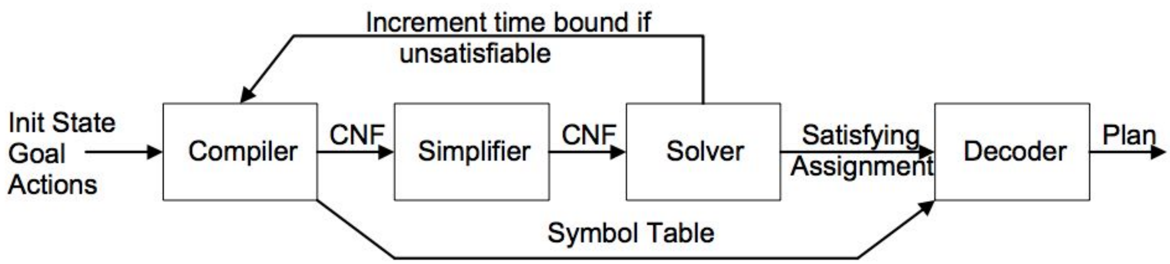

Figure 12: Architecture of a typical SAT-based planning system.

(1)

1) Richard E. Fikes Nils J. Nilsson "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving'" 1 Sep. 1971
http://ai.stanford.edu/~nilsson/OnlinePubs-Nils/PublishedPapers/strips.pdf

2) Weld, Daniel S. "Recent Advances in AI Planning." 8 Oct. 1998
https://homes.cs.washington.edu/~weld/papers/pi2.pdf

3) Avrim L. Blum, Merrick L. Furst "Fast planning through planning graph analysis" Feb. 1997 http://www.sciencedirect.com/science/article/pii/S0004370296000471

4) H. Kautz and B. Selman "Pushing the envelope: Planning, propositional logic, and stochastic search" 1996 https://www.aaai.org/Papers/AAAI/1996/AAAI96-177.pdf