Khayyam Zubair

AIND Project 3: Research Review

Historical Developments in AI Planning

This paper is a short overview of three major milestones in the development of Automated Planning and Scheduling.

STRIPS:

The Stanford Research Institute Planner (STRIPS) was the first major planning language devised in 1971 by Richard Fikes and Nils Nilsson . The system was developed by discovering the application of theorem proving in problem solving . The system was designed to be used in robot research at SRI[1].

STRIPS uses first order predicate calculus formulas to prove whether a goal state could be achieved given an initial model of the world. A model of the world is described using well-formed formulas (wffs). These assert true and false information about the state of the world. The state of a world could be altered by an operator given that it satisfies its preconditions. Finally, the problem space has a goal condition wff, if this condition is satisfied the goal state is achieved. STRIPS proves a problem can be solved if there is a sequence of operations that leads an initial state to be altered into a goal state[2]

The syntax of the language served as the skeleton for future development in planning systems.

PDDL:

The Problem Domain Description Language (PDDL) is a widely used planning language that evolved STRIPS and the Action Description Language (ADL). The language was introduced as a means to standardize planning language by Drew McDermott in 1998. It was made the official language of the inaugural International Planning Competition held that year and remains the standard language since. [1]

PDDL describes a problem in a similar spirit as STRIPS. Each problem is encoded with an initial state which is described by predicates which are true at the start of a problem and a goal state which are predicates that are true at the end of a problem. PDDL also lets you define actions which are functions that transform the state of the world. Each function must be defined with paramaters, preconditions, and effects. The language was written to be computer parable[3]

Graph Plan:

Graphplan is an algorithm that solves planning problems. It takes in a problem described in STRIPS as input and outputs a sequence of operations to achieve the goal state if possible[1]

Graph Plan encodes the problem into a planning graph. It was designed to reduce the complexity of a state space search. In a state space graph, all possible world states of the problem are nodes. An edge between two states indicates that it is possible to reach one state from another through an action. Whereas in a Graphplan Planning graph nodes are actions and atomic facts arranged into alternate levels. An edge between a fact and an action indicate that the fact is a precondition to that action. Whereas an edge from an action to fact makes it true or false. The first level encodes the initial state, with the initially true atomic facts. The algorithm iteratively extends the planning graph until it achieves a goal state. It makes use of pruning to look for states that have already been achieved[4]

References:

1. Stuart Russell, Peter Norvig: Artificial Intelligence, New Jersey, 2003
2. Richard E. Fikes , Nils J. NHsson: STRIPS: A New Approach to the Application of .Theorem Proving to Problem Solving, Stanford,1971
3. http://www.cogsys.wiai.uni-bamberg.de/teaching/ws0405/s_planning/slides/Introduction_AI_Planning_addon.pdf
4. https://www.cs.cmu.edu/~avrim/Papers/graphplan.pdf