Artificial Intelligence Planning Research Review

The part of this project I found most fascinating is the representation of complex problems in reasonably straight forward notation, that when combined with situational calculus can be enumerated to every possible action and state to reach a specific set of goal conditions. Artificial Intelligence (AI) arrived at this utility through a number of developments, three of which will be discussed here. These three important developments in AI planning are the Stanford Research Institute Problem Solver (STRIPS), Action Description Language (ADL), and Planning Domain Definition Language (PDDL).

STRIPS was developed by Richard Fikes and Nils NIllson in 1971 for the purpose of automated planning. STRIPS was designed as an actual planner but it is the language which represents its most significant contribution to AI. The STRIPS language became the basis for most automated planning languages currently in use. This includes both ADL and PDDL. A strips instance consists of a familiar scheme. That is, an initial state, specified goal states, and a set of actions. For each action in the action set preconditions must be established prior to the action and postconditions must be established after the action. A critical restriction of STRIPS, which limits its utility, is the assumption of perfect knowledge of the initial state. That is, any condition not in the initial state is deemed False [1]. This is not always the case for planning problems and in those cases this assumption could prevent valid actions from being explored.

ADL was developed by Penault in 1986 with its primary use case being for planning the actions of robots. Pednault improved upon STRIPS by allowing operator effects to be conditional. This augmented the overall goal of planning languages which is to generate a plan by defining a goal, available actions, and the conditions under which those actions can take place. Key to advancing AI planning over STRIPS is that ADL is open world (STRIPS is not open world). Instead of assuming everything not defined in the initial condition is false, ADL assumes it is unknown. Open world and the allowance of negative literals and disjunctions makes ADL more flexible for planning applications in that it allows for more flexibility in the execution of actions towards goals. [2]

PDDL, inspired by STRIPS and ADL, was developed by McDermott *et al.* in 1998 when they recognized the need for a more standardized AI planning language for use in the International Planning Competition (IPC). The main contribution of PDDL is that it has enabled researchers and competitors to share, re-use, and compare their work. This represents a significant advance in AI planning and has made many other advances in AI possible. As a sign of how useful and robust PDDL is, it continues to evolve through competition [3]. This is the language used in a large portion of this project, in combination with situational calculus. It allows us to define a goal, actions, conditions, and mutex, to enumerate over fluents to develop plans for reaching the goal.

References

1. https://en.wikipedia.org/wiki/STRIPS
2. https://en.wikipedia.org/wiki/Action_description_language
3. https://en.wikipedia.org/wiki/Planning_Domain_Definition_Language