AI Planning and Search Research Review

To represent planning problems, Artificial Intelligence planning languages are leverages to describe environment's conditions which then lead to desired goals by generating chain of actions based on these conditions. This review covers three main planning languages: STIPS, ADL/PDDL and Graphplan/LPG.

STRIPS – The base

Planning emerged as a specific sub-field with the original work of Fikes and Nilsson on the Stanford Research Institute Problem Solver (STRIPS). The STRIPS project introduced a simple syntax for defining action schemas, in terms of the preconditions, add effects and delete effects of the action. It is the base for the most of languages used to describe planning problems. STRIPS consist of states, goals and set of actions:

1. **State** is a conjunction of positive literals which cannot contain variables and invoke functions.
2. **Goal**, similarly to the state, is conjunction of positive and ground (no variables and no functions) literals.
3. **Actions** (also called operators) include preconditions (states required to perform action) and postconditions (states after an action is executed). Both represented as a conjunction of function-free literals.

ADL, PDDL—Extension/Improvements of STIPS

STRIPS language was a good starting point for planning problems representation but there was room for improvements. Taking inspiration from STRIPS, the Action Description Language (ADL) relaxed the restrictions of STRIPS. ADL doesn't assume that unmentioned literals are false, but rather unknown, what is better known as the Open World Assumption. It also supports negative literals, quantified variables in goals (e.g. $\exists$ x At (P1, x) $\wedge$ At(P2, x)), conditional effects and disjunctions in goals (all not allowed in STRIPS).

A computer-parsable standardized syntax was introduced for representing planning problems called, the Planning Domain Definition Language (PDDL) was introduced in the late 90s and is still in use today. PDDL was inspired by both STRIPS and ADL and made International Planning Competition (IPC) series possible.

GRAPHPLAN, LPG – Graph based and local search enhanced agents

Graphplan is a general-purpose planner for STRIPS-style domains, based on ideas used in graph algorithms. Given a problem statement, Graphplan explicitly constructs and annotates a compact structure called a Planning Graph, in which a plan is a kind of "flow" of truth-values through the graph. This graph has the property that useful information for constraining search can quickly be propagated through the graph as it is being built. Graphplan then exploits this information in the search for a plan. Graphplan was created by Avrim Blum and Merrick Furst, with subsequent extensions and improvements made by many researchers at many different institutions around the world.

One successful example is LPG, a fast planner using local search for solving planning graphs. LPG replaces the Graphplan search with a far more efficient and more powerful local search

technique. An advantage of LPG is that its heuristics exploit the structure of the planning graph. In addition, LPG can handle action costs to produce good quality plans by minimizing the objective function that takes into account the number of parallel steps and the overall plan duration. Experimental results show that for a set of well-known benchmark domains LPG is significantly faster than existing Graphplan-style planners.