Selected Topics on Planning and Search

Kan-Hua Lee

I. Planning Domain Definition Language

Planning domain definition language (PDDL) is a lanauge for describing classcial AI planning problem[1]. It extends the capability of Stanford Research Institute Problem Solver (STRIPS) representation and Action Description Language (ADL). PDDL has been used as the standard language for International Planning Competition since 1998. Similar to STRIPS, a problem defined by PDDL is composed of a initial state, a goal state, and a set of actions. Each action has its preconditions and effects. However, unlike STRIPS, PDDL can have negative literals. PDDL can also take conditional effects. In the latest version of PDDL, it also introduced object-fluents, that is, the return types of function can not only be numerical, but also custom defined object type.

II. Partial-Order Plan

Partial-Order plan contains a set of actions and a set of contraints that some actions must occur befor the others. Unlike total order plans, the partial-order plan searches the space of plans rather than the space of states. The search starts by adding actions into an empty plan, and examine whether there is conflicts caused by this action. If a conflict exsits, add another action to correct this flaw. Also, we add the least possible actions to fix the conflicts. This is also known as least commitment planning[2], [3]. One of the most important algorithms for solving partially-ordered planning problem is UCPOP[3]. This is more efficient than foward or regression search of conventional totally-ordered plan problems and therefore dominated the research of AI planning for some time, until the invention of Graphplan.

III. Graphplan

Graphplan emerges as a much more efficient way to extract the solution of a classical planning problem[4]. Graphplan represents a planning problem described by STRIPS into a graph, where the states and actions are connected alternately. After that, the nodes in the same layer that conflict each other is set to be ”mutually exclusive”. More detailed descriptions can be found in [4], [5], [6]. It turns out that this arrangment can greatly reduce the search space and is much more computationally efficient that other algorithms of such as UCPOP.

The efficiency of brought by Graphplan unpacks many applications of boolean satisfiability (SAT) problems which took too much time to solve by using other algorithms. One of these applications is space fault protection and configuration management[5].

References

[1] M. D, M. Ghallab, A. Howe, C. Knoblock, and A. Ram, “PDDL-the planning domain definition language,” 1998.
[2] D. Weld, “An introduction to least commitment planning,” *AI magazine*, 1994.
[3] J. Penberthy and D. Weld, “UCPOP: a sound, complete, partial order planner for ADL.” *Kr*, 1992.
[4] A. Blum and M. Furst, “Fast planning through planning graph analysis,” *Artificial intelligence*, 1997.
[5] D. Weld, “Recent advances in AI planning,” *AI magazine*, 1999.
[6] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2010.