AIND-Planning Search

Research Review

Cheng Wang

Planning is to come up with a sequence of actions to achieve a goal.[1] In order to efficiently develop planning algorithms, the representation language of the planning problem is important and should be able to describe a variety of problems and handle a large number of facts and relations. This review will describe three representation languages used in planning: STRIPS, ADL, and PDDL.

STRIPS

The STanford Research Institute Problem Solver (STRIPS) is designed by Richard Fikes and Nils Nilsson in 1971 as a problem solver that searches a space of world models to find one satisfied the given goal.[2] At that time, 'frame problem' prevent many solvers to solve nontrivial problems.[2] STRIPS provides a seminal framework for attacking the classical planning problem.[3] The problem space is defined by the initial state, a set of available operators (actions), and the goal state in STRIPS. Each operator is defined by an operator description consisting of the effects of the operator and the applicable conditions (or precondition). All of them are represented by a set of well-formed formulas (wffs) of the first-order predicate calculus.[2] The representation framework (language) of STRIPS has been really influential for automatic planning research and is adapted to many variations.[3]

ADL

The Action Description Language (ADL) is introduced by Edwin Pednault in 1986 to demonstrate how to formulate some multiagent, dynamic-world planning into single-agent, static-world problems.[4] It can be considered as an improvement of STRIPS by combining the STRIPS language with the semantics and expressive power of the situation calculus. It relaxed some restrictions in STRIPS. For example, only positive literals in states are allowed in STRIPS, while both positive and negative literals are allowed in ADL. It has the same facilities as STRIPS language and is also more expressive than STRIPS with support of variable with types. Therefore, it can encode more realistic problems.[1]

PDDL

The Planning Domain Definition Language (PDDL) was created by Drew McDermott and his colleagues in 1988 as the standard language for International Planning Competition (IPC).[1,5] PDDL1.2 has the roughly expressiveness of ADL for propositions and roughly the expressiveness of UMCP for actions. It has been improved over year for IPC and current official version is PDDL 3.1. Moreover, there are many successors, variants and extensions of PDDL. The standardized representation language PDDL allows direct comparison of systems and approaches and facilitates the development of new methods for planning.

Reference

[1] Russell, S.J.; and Norving, P. *Artificial Intelligence: A Modern Approach* 3rd ed. (2009).

[2] Fikes, R.E.; and Nilsson, N.J. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* **2**, 189-208 (1971).

[3] Fikes, R.E.; and Nilsson, N.J. STRIPS, a retrospective. *Artificial Intelligence* **59**, 227-232 (1993).

[4] Pednault, E.P.D. Formulating multiagent, dynamic-world problems in the classical planning framework. In *Reasoning about Actions and Plans: Proc. 1986 Workshop*. pp. 47-82 (1986).

[5] Web: https://en.wikipedia.org/wiki/Planning_Domain_Definition_Language