Research review

Planning as satisfiability

Planning as satisfiability is constraint-based approach for automated planning system proposed by Kautz and Selman (1992). Main idea is that problem is a set of axioms and any model of axioms corresponds to a valid plan. Goals, initial state, actions - all of them are axioms. Comparing to the deduction approach, where planning problem is theorem, that needs to be proven it gives power to specify and check more constraints. For example, "not" constraints, that is hard to prove that they are true. Also, "it is easy to specify conditions in any intermediate state of the world, not just the initial and goal states"[1]. This approach was used to develop original SATPLAN (1992 - 1996) and it triggered development of many SAT based planners. Henry Kautz continued his work and made improved version of SATPLAN called BLACKBOX. BLACKBOX transforms STRIPS-based to SAT-based problem and then solves it[2]. It combines SATPLAN and GRAPHPLAN and uses many different satisfiability engines. After BLACKBOX there was also SATPLAN 2004 that win International Planning Competition in 2004 and SATPLAN 2006. Of course, planning as satisfiability has disadvantages, for example, poor scalability. All different problems of this approach was highlighted by Jussi Rintanen is his paper 'Engineering Efficient Planners with SAT'[3] He not only explain main problems of this approach but also implement improved version called The Madagascar planner[4]. MpC and M are 2 versions of this planner that took 2nd and 3rd places in 2014 planning competition("agile" track). So, this approach evolving and is still used in modern AI.

STRIPS

STRIPS is an acronym for Stanford Research Institute Problem Solver. STRIPS program was developed in 1971 to control robot that could move and push objects. Models for robot world was more complex, than for example, models for solving puzzles. To describe complex robot's world was used set of well-formed formulas[5]. Operator was described using precondition, delete list and add list. So, new formal language was created to make planner work. This language become popular and was named by the project it first appears - STRIPS. STRIPS instance described with 4 items: conditions, operators, initial state and goal. So, each planning program, that uses STRIPS language should return set of operators that leads from initial state to goal. Action description should include precondition and effects (clauses that will be added and

---

[1] H. Kautz and B. Selman, Planning as satisfiability, Proceedings of the Tenth European Conference on Artificial Intelligence (ECAI'92), John Wiley, 1992
[2] BLACKBOX - http://www.cs.rochester.edu/users/faculty/kautz/SATPLAN/index.htm
[3] J. Rintanen, Engineering Efficient Planners with SAT, ECAI'12 Proceedings of the 20th European Conference on Artificial Intelligence, Montpellier, France, August 27 - 31, 2012
[4] J. Rintanen. Madagascar, an overview of the techniques in the Madagascar (M, Mp, MpC) planners for the 2014 planning competition, 2014. - https://users.ics.aalto.fi/rintanen/papers/Rintanen14IPC.pdf
[5] Richard E. Fikes Nils J. Nilsson STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. Presented at the 2nd IJCAI, Imperial College, London, England, September 1-3, 1971.

removed). STRIPS language has some disadvantages: it doesn't support equality, allow only positive literals and effects could not be conditional. Nevertheless it was very popular. That is why one of main features of BLACKBOX is transformation from STRIPS-base to SAT-based problem. Many modern action languages such as PDDL and ADL is based on STRIPS. So, STRIPS played significant role in development of expressing AI planning problems.

Planning Domain Definition Language (PDDL)

One of important event that boost research and development in AI planning is International Planning Competition (IPC)[6]. There was one important obstacle to arrange first IPC: although STRIPS was widely used, it has many disadvantages, so there was no common, standardized action language that could describe input for competition. That is why Drew McDermott and other members of IPC Committee developed and described PDDL[7]. First version supports STRIPS-actions, conditional effects, universal quantification, specification of safety constraints and hierarchical actions, domain axioms. PDDL changed rapidly. In 2002 2.1 version arrived. It was capable to describe numeric and temporal properties[8]. In 2005 appeared PDDL 3.0 that allows to set up strong and soft constraints on plan trajectories and strong and soft goals[9]. In 2008 appears PDDL 3.1 that add object fluents, that "map a tuple of objects to an object of the problem"[10]. History of PDDL shows how rapidly AI planning changed through this years. There is many specific successors of PDDL: Multi-Agent Planning Language, Ontology with Polymorphic Types, Probabilistic PDDL. It shows how important it was PDDL in AI planning world.

---

[6] International Planning Competition - http://icaps-conference.org/index.php/Main/Competitions
[7] Planning Domain Denition Language. Version 1.2, Malik Ghallab, Adele Howe, Craig Knoblock, Drew McDermott,Yale Center for Computational Vision and Control, October, 1998
[8] pddl2.1 : An Extension to pddl for Expressing Temporal Planning Domains, Maria Fox and Derek Long, Department of Computer and Information Sciences University of Strathclyde, Glasgow, UK, 2003.
[9] Plan Constraints and Preferences in PDDL3, Alfonso Gerevini+ and Derek Long, Technical Report, Department of Electronics for Automation, University of Brescia, Italy, August 2005
[10] Changes in PDDL 3.1- http://icaps-conference.org/ipc2008/deterministic/PddlExtension.html