Research Review for Logic and Planning

Tharathorn (Joy) Rimchala

Udacity Artificial Intelligence Nanodegree
Project 3: Logic and Planning

Summary

Automated planning is a branch of AI that seeks to find strategies or action sequences that achieve a goal given an initial state of the world. Classical planning algorithms require three inputs: the initial of the world, the goal state, and a set of actions that the executor is able to perform. This article provides brief summaries of three important advances in standardizing the encoding languages for classical planning problems: *STRIPS*, *ADL*, *PDDL* such that they can be solved by planning algorithms.

Stanford Research Institute Problem Solvers: STRIPS

STRIPS (Stanford Research Institute Problem Solvers) is the first major action language for specifying planning graph in finite domain. *STRIPS* was originally developed by Fikes and Nilsson[1] as part of the Shakey Robot Project at SRI international. The main contributions of *STRIPS* are the compact representation of actions and the support for the divide and conquer strategies to be used in automated planning. In the original version of *STRIPS*, state variables have the domain 0,1 (FALSE, TRUE), and an action consists of sets of state variables, the *PRE-CONDITION*, the *ADD* clause list, and the *DELETE* clause list. A state is a goal state if the goals are satisfied. An *applicable* in a state if all the variables in the *PRECONDITION*s are met

and the result of taking an action a state is the set union of predicates in the ADD list and the set substraction of *DELETE* list. The original *STRIPS* representation using first-order logic faces many technical difficulties. While *STRIPS* was first invent as part of a planning algorithm[1], the action description language has a far more lasting impact on the field of AI. Many later variations the *STRIPS representations* use only propositional logic. Although efficient reasoning is possible using *STRIPS* language, *STRIPS* is not suitable for modeling actions in many real world applications. This inadequacy motivated the development of the *ADL* and *PDDL*.

Action Description Language: ADL

The *Action Description Language* or *ADL* is first proposed by Edward Pednault[3,4] with the goal of making *STRIPS representation* applicable to encode real world problems. Pednault introduces several extensions to *STRIPS* action schema that allows *ADL* to be able to encode more realistic problems. First, while *STRIPS* only allows positive literals, *ADL* supports both positive and negative literals. In *STRIPS*, the goals are conjunctions, while in *ADL*, goals may involve conjunctions and disjunctions. More importantly, STRIPS operates on the *Closed World Assumption (CWA)* – i.e. unmentioned literals are *false*, while *ADL* operates under the *Open World Assumption (OWA)* in which the truth values of unmentioned literals are unknown. First, the effects of an action are allowed to be conditional. Lastly, while in *STRIPS*, effects are only allow to be conjunctions, *ADL* effects can be conditional. The extended syntax provided by *ADL* not only provides convenience with which a domain can be encoded, but can also reduce the size of the domain descriptions needed.

Planning Domain Definition Language: PDDL

Planning Domain Definition Language was first defined in 1998 to standardize encoding language for automated planning for the International Planning Competition possible[6] (IPC). In-

spired by *STRIPS* and *ADL*, McDermott the inventor of *PDDL*, decompose the planning problem into two parts: domain description and problem description. In the original version of *PDDL*, the *domain description* consists of requirements, object-type hierarchy, predicates, and set of actions specified by preconditions and effects. The problem description consists of list of all possible objects, initial conditions, and goal-states. *PDDL* has been continually evolved overtime with each IPC competition. There are many descendents of the original *PDDL* that are still in used today.

References and Notes

[1] Richard E. Fikes, Nils J. Nilsson (1971). "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving". *Artificial Intelligence*. 2 (3-4): 189-208. doi:10.1016/0004-3702(71)90010-5.

[2] Tom Bylander (1994). "The Computational Complexity of Propositional STRIPS Planning". Artificial Intelligence. 69 (1-2): 165-204. doi:10.1016/0004-3702(94)90081-7.

[3] Edwing P.D. Pednault. (1987) Formulating multi-agent dynamic-world problems in the classical planning framework. *Reasoning about actions and plans* (47-82). Morgan Kaufmann, San Mateo, CA.

[4] Edwing P.D. Pednault. (1994) ADL and the State-Transition Model of Action *J Logic Computation* 4 (5): 467-512. doi: 10.1093/4.5.467.

[5] Edwin P.D. Pednault. ADL. (1989) Exploring the Middle Ground Between STRIPS and the Situation Calculus. *Proceedings of the first international conference on Principles of knowledge representation and reasoning*, 324-332.

[6] McDermott, Drew; Ghallab, Malik; Howe, Adele; Knoblock, Craig; Ram, Ashwin; Veloso, Manuela; Weld, Daniel; Wilkins, David (1998) "PDDL - The Planning Domain Definition Language" *Tenical Report CVC TR-98-003/DCS TR-1165*.