Research Review on AI Planning

Taufiq Dhanani

Planning, which can be loosely defined as finding a sequence of actions to reach a goal state, is a major part of the field of Artificial Intelegence. What's interesting about planing, is that it's roots go back beyond even the 1950s with Markovian Decision Process, MDP (Bellman, R. (1957). While the intiial thought processes were more theoretical based, there were three key developments that allowed domain specific problems be represneted in computer models. The first is the STanford Research Institute Problem Solver or STRIPS (1971 Files and Nilsson). The task of creating a language for planning problems needed to be expressive enough to be able to solve general problems, however, there had to be strict constriains so that algorithms could efficiently operate over them. Figure 1 depicts how the language is able to be used as a general purpose problem solver for robotic tasks.

Although STRIPS was very ground breaking, the restrictions imposed on it made it less practical to use for some real problem spaces. For example, the language operated in a closed world assumption (all unknowns are false) and all literals had to be positive. The restrictions of STRIPS, led to the creation of The Action Description Language or ADL (Pednault, 1986). The ADL action representation, opened the gate for J. Scott Penberthy and Daniel S. Weld to to come up with UCPOP, the sound, complete, partial order planner (1992).

Inspired by ADL, The Planing Domain Defiition Language (PDDL) was intrudced by at the Artificial Intelligence Planning System 1998 Competion (AIPS-98) (Ghallab, et. all 1998). According to the manual, the language has "the expressiveness of Pednault's ADL for propositions." The reason why PDDL is groundbreaking, is because it was the first attempt to standarize artificial intellegence planning langages. PDDL has many succossors, and the lates version of PDDL3.1 and PDDL+ are still widely used today. The PDDL langauge is easily parsable by computers which makes it a truely groundbreaking innovation.

Operators

goto1(*m*): Robot goes to coordinate location *m*.

Preconditions:

(ONFLOOR) $\wedge$ $(\exists x)$[INROOM(ROBOT,*x*) $\wedge$ LOCINROOM(*m*,*x*)]

Delete list: ATROBOT($),NEXTTO(ROBOT,$)

Add list: ATROBOT(*m*)

goto2(*m*): Robot goes next to item *m*.

Preconditions:

(ONFLOOR) $\wedge$ {$(\exists x)$[INROOM(ROBOT,*x*) $\wedge$ INROOM(*m*,*x*)] $\vee$ $(\exists x)(\exists y)$ [INROOM(ROBOT,*x*) $\wedge$ CONNECTS(*m*,*x*,*y*)]}

Delete list: ATROBOT($),NEXTTO(ROBOT,$)

Add list: NEXTTO(ROBOT,*m*)

pushto(*m*,*n*): robot pushes object *m* next to item *n*

Precondition:

PUSHABLE(*m*) $\wedge$ ONFLOOR $\wedge$ NEXTTO(ROBOT,*m*) $\wedge$ {$(\exists x)$[INROOM(*m*,*x*) $\wedge$ INROOM(*n*,*x*)] $\vee$ $(\exists x, \exists y)$[INROOM(*m*,*x*) $\wedge$ CONNECTS(*n*,*x*,*y*)]}

Delete list: AT ROBOT ($) NEXTTO (ROBOT $) NEXTTO ($,*m*)
AT (*m*$) NEXTTO (*m*$)

Add list: NEXTTO(*m*,*n*)
NEXTTO(*n*,*m*)
NEXTTO(ROBOT,*m*)

Artificial Intelligence **2** (1971), 189–208

Figure 1: An example of the STRIPS language in the paper by Fikes and Nilsson (1971)