Planning Research Review

Using automated systems to provide solutions for planning problems requires that the problem be formulated in some symbolic language that allows the solution space to be searched and reasoned about. This short overview details three important historical systems in planning representation languages.

STRIPS [1]

In 1971, Richard Fikes and Nils Nilsson developed STRIPS [2] (Stanford Research Institute Problem Solver) as an automated planner. Later, the same name was given to the language that describes the input to the planning system. Most planning languages in use today are based on STRIPS but have made improvements. STRIPS had a lot of features we recognize in todays planning languages such as initial state, goal states and actions with pre and post conditions. STRIPS is a closed world [3] in that anything not known to be true is considered false. This is a restriction which limited the expressiveness of the language.

ADL [4]

In 1987, Edwin Pednault developed ADL as an improvement over STRIPS. He realized that an improvement could be made if the effects of operators were conditional. More improvements were made in B and C versions. In response to STRIPS limitation of a closed world, ADL is an open world. This means that in that everything not mentioned in the initial state is considered unknown as opposed to false in the closed world of STRIPS. STRIPS also had only positive literals and conjunctions, which ADL remedied with the allowance of negative literals and disjunctions.

PDDL [5]

In 1998, Drew McDermott and colleagues saw the need to have a standard planning language to drive progress of basic research. Previously, researchers would use formalisms that were oriented towards specific applications, which was not conducive to shared research. This facilitated the 1998/2000 International Planning Competition(IPC). The language evolved as more IPC competitions were held. Here is a list of some of the features added.

2.1: Numeric fluents allowed, to model non-binary objects.
2.2: Derived predicates, A is reachable from B and B is reachable from C, then A is reachable from C.
3.0: State-Trajectory Constrains
3.1: Object Fluents. Fluents could now be objects, in addition to numeric values

---

1 https://en.wikipedia.org/wiki/STRIPS
2 http://ai.stanford.edu/~nilsson/OnlinePubs-Nils/PublishedPapers/strips.pdf
3 https://en.wikipedia.org/wiki/Closed-world_assumption
4 https://en.wikipedia.org/wiki/Action_description_language
5 https://en.wikipedia.org/wiki/Planning_Domain_Definition_Language