AI Planning and Search Research Review

In this paper, I will discuss three major developments in the field of AI Planning and Search. STRIPS was the first major planning language and the basis for most of the field of planning today [1]. ABSTRIPS was built on top of STRIPS and attempted to add abstractions to the planning process [2]. Lastly, I will discuss Planning Domain Definition Language, which was an attempt to standardize AI planning languages and built upon the work in STRIPS and other planning languages [3].

One of the first major developments in AI Planning was the development of STRIPS (STanford Research Institute Problem Solver) [1]. STRIPS was a problem solver that “attempts to find a sequence of operators in a space of world models to transform a given initial world model into a model in which a given goal formula can be proven to be true” [1]. It used well-formed formulas (wffs)[1] in order to represent a world model. Using these formulas, you could describe a robot’s initial state, goal states, and actions with preconditions and postconditions.

Following STRIPS, a system referred to as ABSTRIPS was the “first system to automate the construction of abstraction hierarchies for planning” [2]. ABSTRIPS builds upon the STRIPS planning system, differing in that it “would first construct an abstraction hierarchy for a problem space and then use the abstractions for hierarchical planning” [2]. In order to solve for a given plan, a number indicating relative difficulty was assigned to each operator’s precondition. It first solves for the preconditions with the highest difficulty level and then proceeds down until it has solved for those at the lowest levels. Unfortunately, ABSTRIPS would actually degrade performance in problems where preconditions were not independent, as it operated under the assumption that they would be independent [2].

The final development I researched was PDDL (Planning Domain Definition Language) [3]. PDDL was also inspired by STRIPS and attempts to standardize AI planning languages. It supports: basic STRIPS-style actions, conditional effects, universal quantification over dynamic universes, domain axioms over stratified theories, specification of safety constraints, specification of hierarchical actions composed of subactions and subgoals, management of multiple problems in multiple domains using differing subsets of language features [3]. There have been several iterations of PDDL since its introduction, with the current version being 3.1 [4]

Planning languages allow just about any world problem to be defined and actions specified in order to reach a specific goal state. It will be interesting to see what new developments and languages are created in the future and what types of problems and limitations they will overcome.

References:

1. Richard E. Fikes and Nils J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence, 2:189-208, 1971. http://ai.stanford.edu/~nilsson/OnlinePubs-Nils/PublishedPapers/strips.pdf
2. Craig A. Knoblock. An Analysis of ABSTRIPS. ISI/RR-92-292. March, 1992. http://www.dtic.mil/dtic/tr/fulltext/u2/a269528.pdf
3. Drew McDermot; Malik Ghallab; Adele Howe; Craig Knoblock; Ashwin Ram; Manuela Veloso; Daniel Weld; David Wilkins. PDDL | The Planning Domain Definition Language. October, 1998. http://icaps-conference.org/ipc2008/deterministic/data/mcdermott-et-al-tr-1998.pdf
4. Héctor Geffner. Functional Strips: a more flexible language for planning and problem solving. In Jack Minker (editor), Logic-based Artificial Intelligence, p. 188-209, Kluwer, 2000. http://icaps-conference.org/ipc2008/deterministic/data/geffner-lbai-2000.pdf