Carl Hinkle
AIND - December 2017 Cohort
Project 3
Research Review - Planning and Searching

Planning and searching have always been central to the study of Artificial Intelligence. In this brief report, I discuss three important developments in this field of study, which are drawn from the Biographical and Historical Notes section of AIAMA - Chapter 10.

The first planning system I will discuss is the Strips system of Fikes and Nilsson, 1971[1]. This system was used in the planning system of the Shakey robot at Stanford Research Institute. STRIPS requires a well defined concept of the world and allowed actions. This requirement drove the development of both the Action Description Language (ADL) and the Problem Domain Definition Language (PDDL), which is computer parsable and has a standardized syntax. Using a suitable representation language, objects, preconditions, actions, and effects are defined (this is what we did in the first part of this week's project, in PDDL). STRIPS then makes use of these tokens to transform a given initial state (world model) into the goal state, if possible. The primary application that the authors of STRIPS were interested in was robot motion and problem solving, such as determining how to move a pile of blocks from one place to another.

The TWEAK planner[2], which is a non-linear, domain-independent, contstraint-posting planner which avoids some of the limitations of previous non-linear planners. In particular, it defines the "algebra of plan transformations" with a degree of rigor that had not been done before. Chapman felt that the time had come to move past the "scruffy" formulations of older systems and to provide a "neat" foundation on which further "scruffy" advances could be based. To this end, he provides a mathematical proof of the planning algorithm, which is something that all previous, Ad-hoc, planning systems lacked. Using this system, he solves the Susan anomaly problem (block stacking, as described in the AIAMA text).

The final algorithm that will be discussed is the SatPlan-2006[3], which makes use of the STRIPS subset of PDDL in order to provide solutions with "minimal parallel length". It provides a guarantee that the number of time steps in the solution is as small as possible. There are two key distinctions between the 2006 version of SatPlan and the previous version. First, it encodes a subset of inferred mutexes to binary clauses (which wasn't done at all in the 2004 version due to the size of the graph it can generate and memory constraints). Second, it encodes *both* actions and fluent into boolean variables, instead of only actions. SatPlan is a planning graph based algorithm, which it generates up to level *k*, which is the first level that contains all of the goal literals.

---

[1] http://ai.stanford.edu/~nilsson/OnlinePubs-Nils/PublishedPapers/strips.pdf

[2] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.323.3914&rep=rep1&type=pdf

[3] http://www.cs.rochester.edu/users/faculty/kautz/papers/kautz-satplan06.pdf