Historical Developments of Planning

In this report, we will outline the history of planning as a component of AI research by considering three influential developments.

The first of these is **Shakey the Robot** [cite], the first robot to incorporate AI. Shakey was developed by the Stanford Research Institute (SRI) and took 8 years to build from start to finish (1966 - 1972). This was the first time that multiple cutting edge technologies in the fields of television (computer vision), robotics and computer science were combined into one "automaton". Shakey was designed using the *STRIPS* problem solver [1]. *STRIPS*, as a solver boils down to using already defined strategies such a General Problem Solving (GPS), but with a very specific way of defining the problem. This way of defining the problem, was adopted as a new planning *language*.

The STRIPS language sets some rules for defining the planning project. To adhere to these rules, each problem needs to have:

- *1) An initial state*
- *2) A set of possible actions, each with their own pre- and post conditions*, and
- *3) A goal state.*

Thus, it is clear that Part 1 of this project (`my_cargo_planner`) adheres to the rules of and thus was written in STRIPS and that Problem Domain Description Language (PPDL) is basically STRIPS that is computer parseable and with a standardized syntax.

More than 20 years later, a significant improvement on STRIPS came in the form of the *Graphplan* planner [2]. *Graphplan* introduced the concept of a *planning graph*, such as the one implemented in Part 2 of this project. The input to the *Graphplan* planner is a problem defined in the STRIPS language, form which it constructs a planning graph. The solution provided will always be the shortest possible partial order plan, or else *Graphplan* will state that no valid plan exists.
Graphplan was a distinct deviation from the major trend of the time - partial order planning (POP). It proved to be a significant improvement on state of the art POP based systems, and soon after it's introduction various other *planning graph* based planners emerged [3,4]

In the early 2000's however, Nguyen and Kambhampati aimed to revitalize the dwindling interest in POP, which was seen as poorly scalable after the introduction of *planning graphs*. They introduced a new planner (aptly) name *RePOP* [5], which improved on the *planning graph* based solvers by adding the missing efficiency and flexibility when solving parallel planning domains. *RePOP* is based on *UCPOP* [6], a successful POP solver, but drastically improved in

terms of efficiency by, amongst other things presenting new ways of adapting distance based heuristics. *RePOP* was a significant overhaul of how POP solvers were designed at the time and definitely a noteworthy contribution to the field of planning and AI.

References

- [1] R.E Fikes and N.J Nilsson: STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence 2 (1971), 189--208*
- [2] A.L Blum and M. L Furst: Fast Planning Through Planning Graph Analysis. *Artificial Intelligence, 90:281–300, 1997*
- [3] Fox, M. S. and Long, D. (1998). The automatic in- ference of state invariants in TIM. *JAIR, 9, 367–421*.
- [4] Koehler, J., Nebel, B., Hoffmann, J., and Dimopou- los, Y. (1997). Extending planning graphs to an ADL subset. *In ECP-97, pp. 273–285.*
- [5] Nguyen, X. and Kambhampati, S. (2001). Reviving partial order planning. *In IJCAI-01, pp. 459–466.*
- [6] D. Weld. An introduction to least commitment planning. *AI magazine, 1994.*