Developments in the field of AI: Planning and Search

Kai Schoos

August 24, 2017

1 STRIPS

As Shakey, the robot that was used at the STI, needed a way to come up with plans to solve the problems he was given, his makers needed to come up with a way to represent these plans. Fikes and Nilsson [Fik93] write in their paper how they developed the syntax for STRIPS. STRIPS is at the same time the name of the language that was used to define the problems for Shakey and the planner, that came up with plans in STRIPS-language (STRIPS April 21, 2017 In *Wikipedia*. Retrieved August 22, 2017, from https://en.wikipedia.org/wiki/STRIPS). The language was used to define a classical planning problem: Finding a plan to a problem in a known, static, single agent environment where all actions are deterministic. It can be considered as one of the earliest versions of PDDL, which was not introduced until 1998 [RN10]. STRIPS was, like all linear planners, not complete and could not optimally solve the *Sussman Anomaly*. [RN10]

2 Replanning on failure

Although Shakey was able to recover from problems along the execution of his plan, the act of replanning was not thoroughly thought out, according to Philip J. Hayes:

> [STRIPS] dealt with replanning after execution failures, tried to save work by making arbitrary subsequences of operations from the original plan available to the replanning process as primitive operations.

[Hay75]

As a reaction, Hayes develops a system that recovers from failure by adding information about the process. He writes:

> [The presented system] facilitates reconstruction of detailed plans after failure in execution [...] by explicitly recording in the representation of a particular plan the structure of the process which produced that plan.

[Hay75]

The plan produced by his system has the form of a tree, where the layers represent higher levels of abstraction. The leaf nodes constitute the plan in its most detailed from. The planning process is saved in a graph that is referenced by two-way pointers by the tree nodes. Whenever the execution of a step in the plan is rendered impossible, the algorithm finds what Hayes calls the *resumption point* in the decision graph in order to easily resume the planning process from a critical node. Sadly the paper does not present any metrics so it is hard to say anything about the success of this method.

3 Partial-order planning

In order to solve problems like the *Sussman Anomaly*, planning researchers had to move away from the idea that subgoals could always be solved one after another without interaction. This is where partial-order planning, also known as least-commitment planning came in. According to Earl Sacerdoti, who first wrote about partial-order planning:

> By avoiding premature commitments to a particular order for achieving subgoals, a problem-solving system using [partial-order plans] can deal with problems that are otherwise very difficult to solve.

[DS75] In his paper, Sacerdoti explains how he implements least-commitment planning in his NOAH planner, which is able to solve the Sussman Anomaly successfully.

References

[DS75] Earl D. Sacerdoti. The nonlinear nature of plans. pages 206–214, 01 1975.

[Fik93] Richard Fikes. Strips, a retrospective. 59:227–232, 02 1993.

[Hay75] Philip J. Hayes. A representation for robot plans. In *Proceedings of the 4th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'75, pages 181–188, San Francisco, CA, USA, 1975. Morgan Kaufmann Publishers Inc.

[RN10] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 3 edition, 2010.