AI planning languages

Planning or more precisely: **automated planning and scheduling** is one of the major fields of AI (among the others like: Machine Learning, Natural Language Processing, Computer Vision and more). Planning focuses on realization of strategies or action sequences executed by:

- **Intelligent agents**—the autonomous entities (software of hardware) being able to observe the world through different types of sensors and perform actions based on those observations.
- **Autonomous robots**—physical intelligent agents which deliver goods (factory robots), keep our house clean (intelligent vacuum cleaners) or discover outer worlds in space missions.
- **Unmanned vehicles**—autonomous cars, drones or robotic spacecrafts.

To accomplish given tasks, these systems need to have input data containing descriptions of **initial states** of the world, desired **goals** and **actions**. And the role of planning systems is to find sequences of actions which lead from initial state to given goal.

STRIPS

STRIPS is an action language which was a part of the first major planning system with the same name.

Shakey, the robot

Originally STRIPS was a name for the planning component in software used in Shakey, the robot developed at the Stanford Research Institute (SRI), which was the first machine to be able to reason about its own actions.

STRIPS, classical planning language

But what is the most interesting, representational language used by STRIPS planner has much bigger impact on field of AI than its algorithms and is the base for the most of languages used to describe planning problems.

STRIPS as a classical planning language is composed from states, goals and set of actions:

- **State** is a conjunction of positive literals which cannot contain variables and invoke functions.
- **Goal**, similarly to the state, is conjunction of positive and ground (no variables and no functions) literals.

- **Actions** (also called operators) include preconditions and postconditions. Both represented as a conjunction of function-free literals. Preconditions describe the state of world required to perform action, while postconditions describe state of the world after action is executed.

ADL, PDDL—further developments in representational languages

STRIPS language was a good starting point for planning problems representation but there was room for improvements. **ADL** (Action Description Language) is one of STRIPS extensions which removed some of its constraints to handle more realistic problems. Unlike STRIPS, ADL doesn't assume that unmentioned literals are false, but rather unknown, what is better known as the Open World Assumption. It also supports negative literals, quantified variables in goals (e.g. ∃x At (P1, x) ∧ At(P2, x)), conditional effects and disjunctions in goals (all not allowed in STRIPS).

STRIPS and ADL were inspiration for another extension of representational languages—**PDDL** (Planning Domain Definition Language). It was an attempt to standardise planning languages what made International Planning Competition (IPC) series possible. In other words PDDL contains STRIPS, ADL and much more other representational languages.

References, further readings

- Shakey the robot, Technical note 323, http://www.cs.uml.edu/~holly/91.549/readings/629.pdf

- Russell, S, and P Norvig. **Artificial Intelligence: A Modern Approach,**Chapter 10: Classical Planning

- https://machinelearnings.co/historical-intro-to-ai-planning-languages-92ce9321b538

- Shakey in Computer History Museum: http://www.computerhistory.org/revolution/artificial-intelligence-robotics/13/289