Web-based Autonomous Systems
— Exercise 2 —

Stephan Nef

Web-based Autonomous Systems, Course 8,864

March 5, 2024

Contents

1. Task 3

1.1. Anomaly in Domain Problem Solving

The anomaly refers to a situation where direct actions taken to achieve subgoals disrupt the path to the final goal. Specifically, in planning domains like the one described, actions performed to reach a certain state can inadvertently undo previously achieved states, making it challenging to reach the overall goal efficiently. In our Task 3 the problem is not solved in the most efficient way because the most efficient would be "storage1" directly instead over "room1" first.

1.2. Circumstances of the Anomaly

The Sussman Anomaly typically occurs in classical STRIPS (Stanford Research Institute Problem Solver) planning when:

- **Interdependent Subgoals:** Our plan involves (multiple) subgoals that are interdependent. Achieving one subgoal may negatively affect another.
- **Non-Linear Plans:** A linear sequence of actions is insufficient to achieve the goal without undoing previous progress. Necessitating a non-linear approach that considers the complex interplay between actions.

  In our Task 3 the anomaly occurs because each subgoal of the final goal is solved individually and the goals contradict each other in some state.

1.3. Problem and Domain Susceptibility

The given problem and domain are susceptible to the Sussman Anomaly due to:

- **Interdependent Actions:** The actions such as moving between spaces, acquiring the hoover, and cleaning the room, are interdependent. Moving to obtain the hoover after reaching the room initially appears to be a step away from the goal of cleaning the room.
- **Sequential Dependencies:** The necessity to very first move to the "room1", then to the storage to get the hoover, and back to the room for cleaning demonstrates sequential dependencies that are not straightforwardly linear without considering the overarching goal context.

  To achieve the first goal the solver must go to "room1" first (sequential dependencies). For the second goal it needs to pick up the hoover from "storage1" first.

However, the solver faces a dilemma: to accomplish one goal it might have to undo a previously achieved goal (like leaving "room1") to achieve the other goal (like obtaining the hoover). This situation necessitates prioritizing the goals because the solver cannot pursue both goals simultaneously.

1.4. Anomaly Non-Observance in Task 2 Planner

The behavior is not observable with the planner implementation from Task 2 with A* due to:

- **Heuristic-Based Search:** In Task 2 the solution with A* employs heuristics to efficiently navigate through the search space and therefore effectively address complex dependencies and anomalies by evaluating the total cost (path cost plus heuristic estimate) to prioritize paths that seem most promising towards achieving the goal.
- **Heuristic Guidance:** A* uses heuristic information to guide its search and prioritise nodes that are estimated to be closer to the goal. That involves balancing between the cost to reach the node and the heuristic estimate of the cost from the node to the goal allowing it to navigate through the search space even if it means diverging (temporarily) from intermediate objectives.