

## Task 2

---

Git Url to exercise fork: <https://github.com/Scraylex/WAS-exercise-2>

## Task 3 Findings

---

### 1) What is the anomaly that occurs when solving the problem in the domain?

The planner takes an unnecessary step in moving explicitly to the storage1 to get the Hoover before cleaning the room. This extra step is taken because of the chosen graph search algorithm.

### 2) Under what circumstances does the anomaly occur?

The Sussman anomaly occurs in planning problems that involve a set of interdependent actions that must be performed in a specific order to achieve a particular goal state. Every state and action is interpreted individually as edges (actions) and vertices (states) in a graph. This is to say that it is possible that an automated plan might take unnecessary state transitions before reaching the end goal or even getting stuck in a loop.

### 3) What specifically in the problem and the domain make it susceptible?

This problem occurs because of the backward search used in STRIPS-like planners. It is possible to take a path which ultimately satisfies the given initial state but may take suboptimal steps according to the search algorithm used in the graph traversal.

Specifically at the point after getting the Hoover it is possible to either go directly back into the wanted initial state of being in the reception or going back into room. After that it is possible to either go into the storage or to the reception from the room.

According to how the traversal algorithm functions this could recurse indefinitely.

### 4) Why is the behavior not observable with your planner implementation from Task 2?

Because the A\* algorithm is a deterministic pathfinding algorithm i.e. in its heuristic it will always search for the most optimal traversal path across the graph.

This behaviour is in fact observable with using the Enforce Hillclimb Search algorithm using the Python planner. Sometimes it will take 4 steps sometimes 5. Probably because of the heuristic metric associated.