NAME: Meith Navlakha
SAP: 60004190068
AI Practical

EXPERIMENT 10

AIM: Identify, analyze, implement a planning problem/Rule based Expert System in a real world scenario.

THEORY:

Partial-order planning is an approach to automated planning that maintains a partial ordering between actions and only commits ordering between actions when forced to that is, ordering of actions is partial. Also, this planning doesn't specify which action will come out first when two actions are processed. By contrast, total-order planning maintains a total ordering between all actions at every stage of planning. Given a problem in which some sequence of actions is required in order to achieve a goal, a partial-order plan specifies all actions that need to be taken, but specifies an ordering between actions only where necessary.

A partial-order plan or partial plan is a plan which specifies all actions that need to be taken, but only specifies the order between actions when necessary. It is the result of a partial-order planner. A partial-order plan consists of four components:

- A set of actions (also known as operators).
- A partial order for the actions. It specifies the conditions about the order of some actions.
- A set of causal links. It specifies which actions meet which preconditions of other actions. Alternatively, a set of bindings between the variables in actions.
- A set of open preconditions. It specifies which preconditions are not fulfilled by any action in the partial-order plan.

In order to keep the possible orders of the actions as open as possible, the set of order conditions and causal links must be as small as possible.

A plan is a solution if the set of open preconditions is empty.

A linearization of a partial order plan is a total order plan derived from the particular partial order plan; in other words, both order plans consist of the same actions, with the order in the linearization being a linear extension of the partial order in the original partial order plan.

NAME: Meith Navlakha
SAP: 60004190068
AI Practical

PERFORMANCE:

Chosen Problem: Vacuum Cleaner

Problem Description:

Goal (Room1Clean ^ Room2Clean)

Init ()

Action: Room1Clean

- PRECOND: ScanRoom1
- EFFECT: Room1Cleaned

Action: ScanRoom1

- PRECOND: None
- EFFECT: Room1Scanned

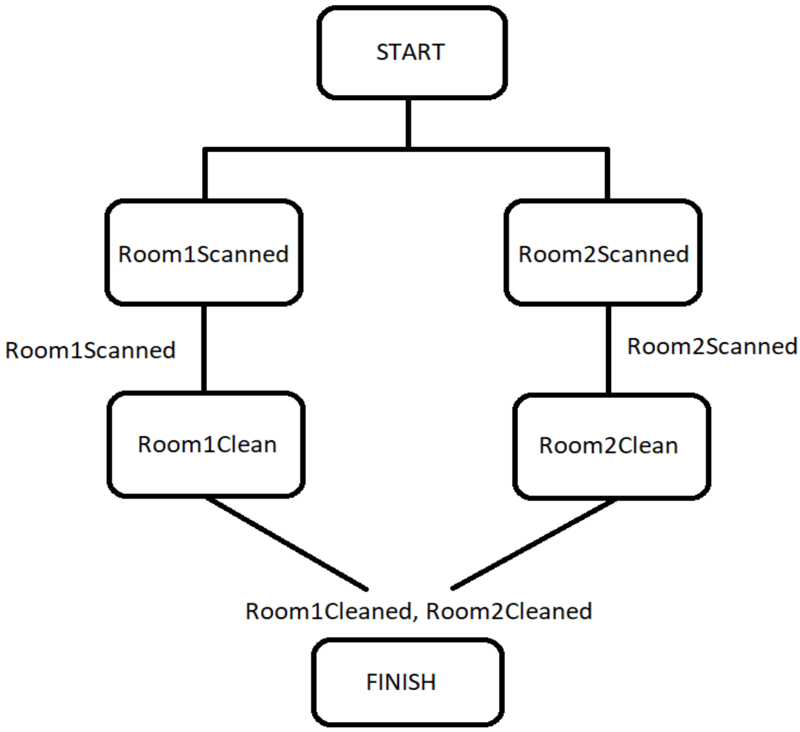
Action: Room2Clean

- PRECOND: ScanRoom2
- EFFECT: Room2Cleaned

Action: ScanRoom2

- PRECOND: None
- EFFECT: Room2Scanned

Partial Order Plan:



NAME: Meith Navlakha
SAP: 60004190068

Components in Final Plan:

Actions: {ScanRoom, SuckGarbage, Left, Right, Start, Finish}

Orderings: {ScanRoom < SuckGarbage}

Open preconditions: {}

Links:

Room1 --- Room1Dirty ---> SuckGarbage

Room1 --- Room1Clean ---> Finish

Room2 --- Room2Dirty ---> SuckGarbage

Room2 --- Room2Clean ---> Finish

CONCLUSION:

In this experiment, I have successfully implemented Partial Order Planning with an example of an automatic Vacuum Cleaner. The tasks of the cleaner are to check whether the room is dirty or clean, and clean if it is dirty. The task is completed when both rooms are clean. One drawback of this planning system is that it requires a lot more computational power for each node because the algorithm for partial order planning is more complex than others. Partial order planning has huge importance and usage in artificial intelligence applications.