

Artificial Intelligence

Programming Assignment 2

Name: Sai Subhash Yalamadala

ID: 1002031729

Problem 1 Pick and Place Task

For the given problem I have modified the PDDL code which is given in the lecture slides and allowed to compute 5balls and 10 balls accordingly. I have generated the report plan for both and provided the screenshots of the same below.

Plan for 5 Balls:

(pick ball5 rooma right)
(move rooma roomb)
(drop ball5 roomb right)
(move roomb rooma)
(pick ball4 rooma right)
(move rooma roomb)
(drop ball4 roomb right)
(move roomb rooma)
(pick ball3 rooma right)
(move rooma roomb)
(drop ball3 roomb right)
(move roomb rooma)
(pick ball2 rooma right)
(move rooma roomb)
(drop ball2 roomb right)
(move roomb rooma)
(pick ball1 rooma right)
(move rooma roomb)
(drop ball1 roomb right)

Screenshot:

The screenshot shows the PDDL Editor web application. The browser address bar displays "editor.planning.domains/#". The application interface includes a menu bar with "File", "Session", "Import", "Solve", "Plugins", and "Help". The main workspace is divided into three sections:

- Left Panel:** A file explorer showing "domaina.pddl", "problem-ball5.pddl", "Error (1)", and "Plan (2)".
- Center Panel:** A list of actions for the found plan, including "(pick ball5 rooma right)", "(move rooma roomb)", "(drop ball5 roomb right)", "(move roomb rooma)", "(pick ball4 rooma right)", "(move rooma roomb)", "(drop ball4 roomb right)", "(move roomb rooma)", "(pick ball3 rooma right)", "(move rooma roomb)", "(drop ball3 roomb right)", "(move roomb rooma)", "(pick ball2 rooma right)", "(move rooma roomb)", "(drop ball2 roomb right)", "(move roomb rooma)", "(pick ball1 rooma right)", "(move rooma roomb)", and "(drop ball1 roomb right)".
- Right Panel:** A code editor showing the PDDL code for the found plan, titled "Found Plan (output)".

```
(:action pick
:parameters (ball15 rooma right)
:precondition
  (and
    (ball ball15)
    (room rooma)
    (gripper right)
    (at ball15 rooma)
    (at-robbey rooma)
    (free right)
  )
:effect
  (and
    (carry ball15 right)
    (not
      (at ball15 rooma)
    )
    (not
      (free right)
    )
  )
)
```

Plan for 10 Balls:

(pick ball10 rooma right)
(move rooma roomb)
(drop ball10 roomb right)
(move roomb rooma)
(pick ball9 rooma right)
(move rooma roomb)
(drop ball9 roomb right)
(move roomb rooma)
(pick ball8 rooma right)
(move rooma roomb)
(drop ball8 roomb right)
(move roomb rooma)
(pick ball7 rooma right)
(move rooma roomb)
(drop ball7 roomb right)
(move roomb rooma)
(pick ball6 rooma right)
(move rooma roomb)
(drop ball6 roomb right)
(move roomb rooma)
(pick ball5 rooma right)
(move rooma roomb)
(drop ball5 roomb right)
(move roomb rooma)
(pick ball4 rooma right)
(move rooma roomb)
(drop ball4 roomb right)
(move roomb rooma)
(pick ball3 rooma right)
(move rooma roomb)
(drop ball3 roomb right)
(move roomb rooma)
(pick ball2 rooma right)
(move rooma roomb)
(drop ball2 roomb right)
(move roomb rooma)
(pick ball1 rooma right)
(move rooma roomb)

```
(drop ball1 roomb right)
```

Output:

[illegible]

Problem 2 Package Delivery Task

For the given problem I have modified the provided PDDL code and added up a third driver driver3, third truck truck3, fourth location s3 and specified the goal accordingly.

Plan:

```
(board-truck driver3 truck2 s1)
(drive-truck truck2 s1 s0 driver3)
(walk driver1 s2 p1-2)
(walk driver1 p1-2 s1)
(load-truck package1 truck3 s1)
(walk driver2 s2 p2-0)
(walk driver2 p2-0 s0)
(board-truck driver1 truck3 s1)
(drive-truck truck3 s1 s2 driver1)
(unload-truck package1 truck3 s2)
(disembark-truck driver1 truck3 s2)
(walk driver1 s2 p1-2)
(walk driver1 p1-2 s1)
```

Output:

The screenshot shows the PDDL Editor interface. On the left, a sidebar lists files: domain-driverlog.pddl, problem-driverlog.pddl, Plan (1), and Plan (2). The main area is titled "Found Plan (output)" and displays a list of actions in a table. To the right of the table, the corresponding PDDL code for the first action is shown.

Found Plan (output)
(board-truck driver3 truck2 s1)
(drive-truck truck2 s1 s0 driver3)
(walk driver1 s2 p1-2)
(walk driver1 p1-2 s1)
(load-truck package1 truck3 s1)
(walk driver2 s2 p2-0)
(walk driver2 p2-0 s0)
(board-truck driver1 truck3 s1)
(drive-truck truck3 s1 s2 driver1)
(unload-truck package1 truck3 s2)
(disembark-truck driver1 truck3 s2)
(walk driver1 s2 p1-2)
(walk driver1 p1-2 s1)

```
(:action board-truck
:parameters (driver3 truck2 s1)
:precondition
  (and
    (at truck2 s1)
    (at driver3 s1)
    (empty truck2)
  )
:effect
  (and
    (not
      (at driver3 s1)
    )
    (driving driver3 truck2)
    (not
      (empty truck2)
    )
  )
)
```