Mini PS

● Graded

Student

Chetan Hiremath

Total Points

19 / 20 pts

Question 1

(no title) **5** / 5 pts

✔ **– 0 pts** Correct

Question 2

(no title) **14** / 15 pts

2.1 **(no title)** **Resolved** **8** / 9 pts

✔ **– 1 pt** Missing On(Shakey,Floor) for Go, Push, and/or ClimbUp

Regrade Request **Submitted on: Mar 06**

I have actually added On(Shakey, Floor) for ClimbUp. Will you regrade this question?

you are missing for Go and rubric is and/or

Reviewed on: Mar 06

2.2 **(no title)** **3** / 3 pts

✔ **– 0 pts** Correct

2.3 **(no title)** **3** / 3 pts

✔ **– 0 pts** Correct

Q1
5 Points

Given the action schemas from Figure 10.1 [reproduced below], list all the applicable concrete instances of `Fly(p, from, to)` in the state described by

`At(P1, JFK) ^ At(P2, SFO) ^ Plane(P1) ^ Plane(P2) ^ Airport(JFK) ^ Airport(SFO)`

$$
\begin{aligned}
&Init(At(C_1, SFO) \land At(C_2, JFK) \land At(P_1, SFO) \land At(P_2, JFK) \\
&\quad \land Cargo(C_1) \land Cargo(C_2) \land Plane(P_1) \land Plane(P_2) \\
&\quad \land Airport(JFK) \land Airport(SFO)) \\
&Goal(At(C_1, JFK) \land At(C_2, SFO)) \\
&Action(Load(c, p, a), \\
&\quad \text{PRECOND: } At(c, a) \land At(p, a) \land Cargo(c) \land Plane(p) \land Airport(a) \\
&\quad \text{EFFECT: } \neg At(c, a) \land In(c, p)) \\
&Action(Unload(c, p, a), \\
&\quad \text{PRECOND: } In(c, p) \land At(p, a) \land Cargo(c) \land Plane(p) \land Airport(a) \\
&\quad \text{EFFECT: } At(c, a) \land \neg In(c, p)) \\
&Action(Fly(p, from, to), \\
&\quad \text{PRECOND: } At(p, from) \land Plane(p) \land Airport(from) \land Airport(to) \\
&\quad \text{EFFECT: } \neg At(p, from) \land At(p, to))
\end{aligned}
$$

Figure 10.1 A PDDL description of an air cargo transportation planning problem.

Enter your answer in the box below. For example, if the state had been `At(Planey-McPlane, MCI) ^ Airport(MCI)` then there is only one concrete `Fly` action applicable: `Fly(Planey-McPlane, MCI, MCI)`

The applicable concrete instance of Fly(p, from, to) in the state is described by At(P1, JFK) ^ At(P2, SFO) ^ Plane(P1) ^ Plane(P2) ^ Airport(JFK) ^ Airport(SFO). Here are the applicable concrete instances of Fly(p, from, to):

1. Fly(P1, JFK, SFO)

2. Fly(P1, JFK, JFK)

3. Fly(P2, SFO, JFK)

4. Fly(P2, SFO, SFO)

Q2
15 Points

The original STRIPS planner was designed to control Shakey the robot. The figure below shows a version of Shakey's world consisting of four rooms lined up along a corridor, where each room has a door and a light switch.

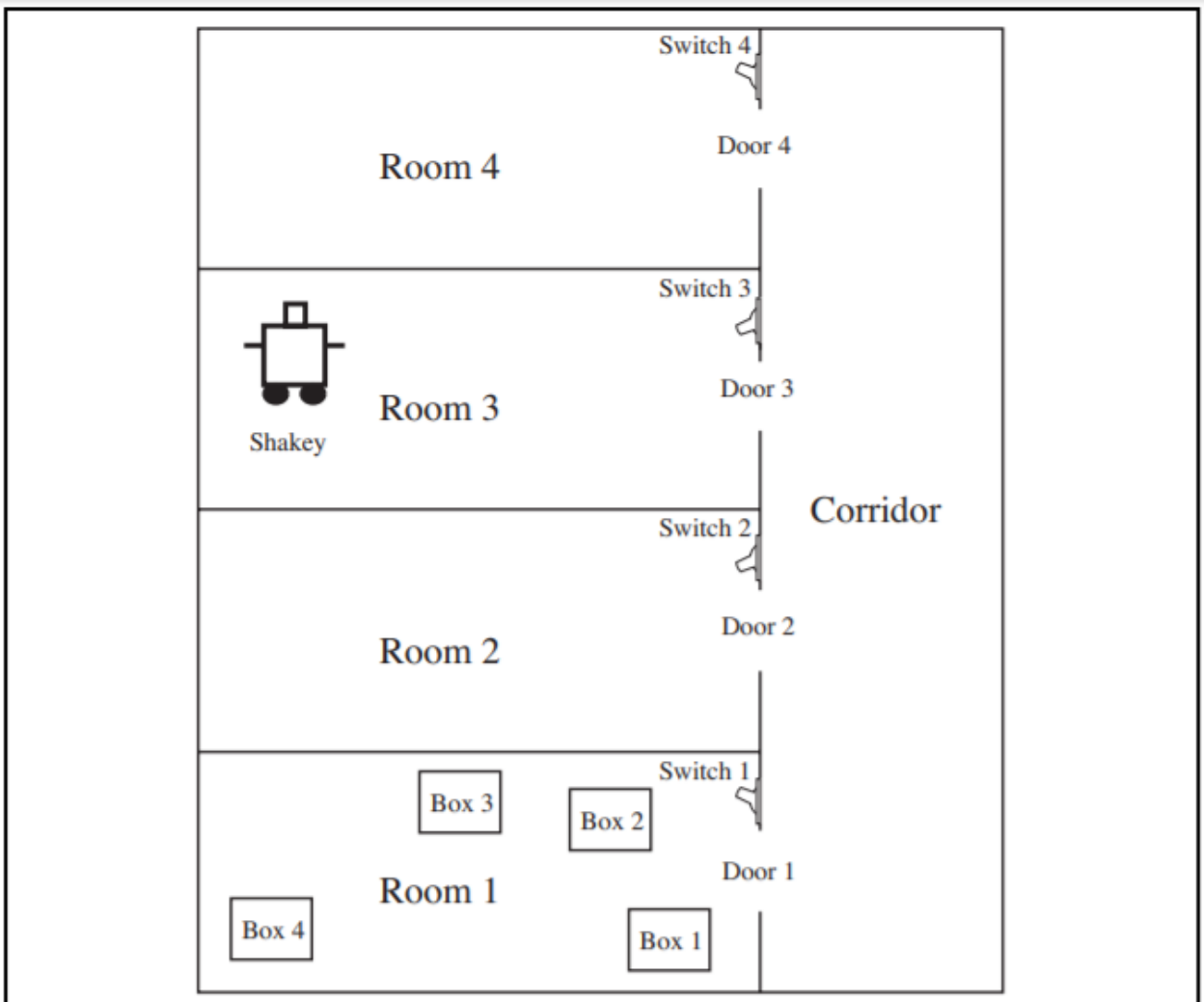

Figure 10.14 Shakey's world. Shakey can move between landmarks within a room, can pass through the door between rooms, can climb climbable objects and push pushable objects, and can flip light switches.

The actions in Shakey's world include moving from place to place, pushing movable objects (such as boxes), climbing onto and down from rigid objects (such as boxes), and turning light switches on and off. The robot itself could not climb on a box or toggle a switch, but the planner was capable of printing out plans that were beyond the robot's abilities. Shakey's six actions are the following:

`Go(x, y, r)` which requires Shakey be `At x` and the `x` and `y` are locations `In` the same room. By convention a door between two rooms is in both of them.

Push a box `b` from location `x` to location `y` within the same room: `Push(b, x, y, r)`. You will need the predicate `Box` and constants for the boxes.
Climb onto a box from position `x`: `ClimbUp(x, b)`; climb down from a box to position `x`: `ClimbDown(b, x)`. We will need the predicate `On` and the constant `Floor`.
Turn a light switch on or off: `TurnOn(s,b)`; `TurnOff(s,b)`. To turn on or off, Shakey must be on top of a box at the light switch's location.

Q2.1
9 Points

(a) Write PDDL [Planning Domain Definition Language] sentences for Shakey's six actions:

Here are the PDDL sentences of the 6 actions of Shakey:

1. Action(Go(x, y, r),
Precond: At(Shakey, x) ^ In(x, r) ^ In(y, r)
Delete: At(Shakey, x)
Add: At(Shakey, y)
Effect: ¬At(Shakey, x) ^ At(Shakey, y))

2. Action(Push(b, x, y, r),
Precond: Box(b) ^ In(Shakey, r) ^ At(Shakey, x) ^ In(b, r) ^ At(b, x) ^ Pushable(b)
Delete: At(Shakey, x) ^ At(b, x)
Add: At(Shakey, y) ^ At(b, y)
Effect: ¬At(Shakey, x) ^ ¬At(b, x) ^ At(Shakey, y) ^ At(b, y))

3. Action(ClimbUp(x, b),
Precond: On(b, Floor) ^ ¬On(Shakey, b) ^ On(Shakey, Floor) ^ Box(b) ^ At(b, x) ^ At(Shakey, x)
Delete: On(Shakey, Floor)
Add: On(Shakey, b)
Effect: ¬On(Shakey, Floor) ^ On(Shakey, b))

4. Action(ClimbDown(b, x),
Precond: On(Shakey, b) ^ On(b, Floor) ^ Box(b) ^ At(b, x) ^ ¬On(Shakey, Floor)
Delete: On(Shakey, b)
Add: On(Shakey, Floor)
Effect: ¬On(Shakey, b) ^ On(Shakey, Floor))

5. Action(TurnOn(s, b),
Precond: On(Shakey, b) ^ At(b, s) ^ ¬TurnedOn(s) ^ Box(b)
Delete: N/A
Add: TurnedOn(s)
Effect: TurnedOn(s))

6. Action(TurnOff(s, b),
Precond: On(Shakey, b) ^ At(b, s) ^ TurnedOn(s) ^ Box(b)

Delete: TurnedOn(s)
Add: N/A
Effect: ¬TurnedOn(s))

Q2.2
3 Points

(b) Write PDDL sentences for Shakey's initial state [as depicted in the figure above]

Here are the PDDL sentences of the initial state of Shakey:

Init(In(Door1, Room1) ^ In(Door1, Corridor) ^ In(Door2, Room2) ^ In(Door2, Corridor) ^ In(Door3, Room3) ^ In(Door3, Corridor) ^ In(Door4, Room4) ^ In(Door4, Corridor) ^ In(Switch1, Room1) ^ In(Switch2, Room2) ^ In(Switch3, Room3) ^ In(Switch4, Room4) ^ Box(Box1) ^ Box(Box2) ^ Box(Box3) ^ Box(Box4) ^ At(Box1, Box1InitLoc) ^ At(Box2, Box2InitLoc) ^ At(Box3, Box3InitLoc) ^ At(Box4, Box4InitLoc) ^ In(Box1InitLoc, Room1) ^ In(Box2InitLoc, Room1) ^ In(Box3InitLoc, Room1) ^ In(Box4InitLoc, Room1) ^ In(Shakey, Room3) ^ At(Shakey, ShakeyInitLoc))

Therefore, this initial state defines the world of Shakey.

Q2.3
3 Points

(c) Construct a plan for Shakey to get `Box2` into `Room2`. Enter your plan as a series of PDDL sentences.

Shakey needs to get Box2 into Room2 by pushing Box2 from Room1 to Room2. So, here is my plan as a series of PDDL sentences:

Goal(In(Box2, Room2) ^ At(Box2, Box2NewLoc))

Plan(Go(ShakeyInitLoc, Door3, Room3), Go(Door3, Door1, Corridor), Go(Door1, Box2InitLoc, Room1), Push(Box2, Box2InitLoc, Door1, Room1), Push(Box2, Door1, Door2, Corridor), Push(Box2, Door2, Box2NewLoc, Room2))

Therefore, this plan will allow Shakey to push Box2 from Room1 to Room2 after Shakey goes from Room3 to Room1. So, it will achieve the goal.