Artificial Intelligence

Exercise sheet 4
Planning & Software Agents

1 Planning

> *Note: This is an individual task, you should answer it with your own words.*

The following exercises have been adopted from the book "Artificial Intelligence: A Modern Approach" [1]

1.1 Flight Problem (7 points)

Given the axioms from Figure 1.1, find all applicable concrete **instances** of **Fly(p,from,to)** in the state described by:

$$At(P_1, JFK) \wedge At(P_2, SFO) \wedge Plane(P_1) \wedge Plane(P_2) \wedge Airport(JFK) \wedge Airport(SFO)?$$

1.2 Shakey (14 points)

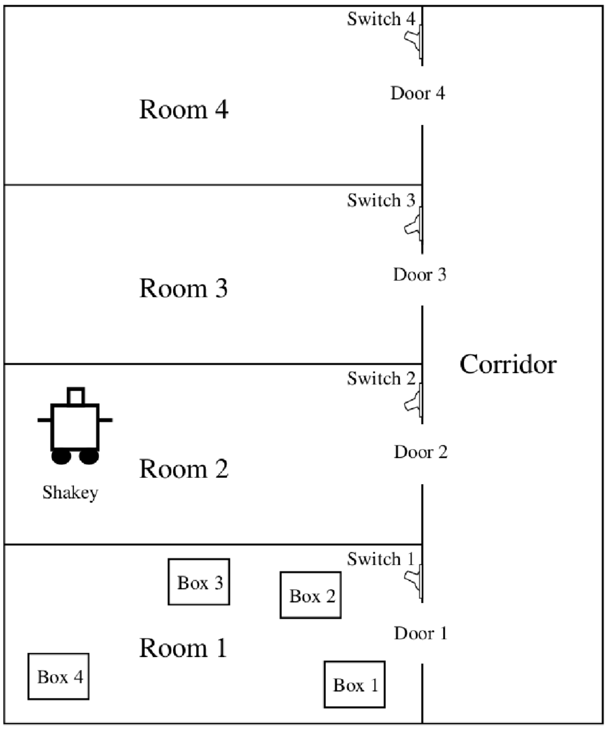
The original STRIPS program was designed to control Shakey the robot. Figure 1.2 shows a version of Shakey's world consisting of four rooms lined up along a corridor, where each room has a door and a light switch.

[1] Chapter 11 (Planning), Stuart Russell and Peter Norvig, 1995

Figure 1.1: A STRIPS Problem

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

Figure 1.2: A Shakey's World



The actions in Shakey's world include moving from place to place, pushing moveable objects (such as boxes), climbing onto and down from rigid objects (such as boxes), and turning light switches on and off. The robot itself was never dexterous enough to climb on a box or toggle a switch, but the STRIPS planner was capable of finding and printing out plans that were beyond the robot's abilities. Shakey's six actions are the following:

1. **Go(x,y)**, which requires that Shakey be at `x` and that `x` and `y` are locations in the same room. By convention a door between two rooms is in both of them.
2. Push a box `b` from location `x` to location `y` within the same room: **Push(b,x,y)**. We will need the predicate **Box** and constants for the box.
3. Climb onto a box: **ClimbUp(b)**; climb down from a box: **ClimbDown(b)**. We will need the predicate **On** and the constant **Floor**.
4. Turn a light switch on: **TurnOn(s)**; turn it off: **TurnOff(s)**. To turn a light on or off, Shakey must be on top of a box at the light switch's location.

Describe Shakey's six **actions** and the **initial state** from Figure 1.2 in STRIPS notation. Construct a **plan** for Shakey to get `Box-3` into `Room-3`. You have to include the complete map of the world in the initial state.

2 Software Agents

> *Note: This is a group task, you may solve it in groups up to the size of two people, everybody must however be able to explain all parts of the implementation*

The following exercise is adopted from the book "Artificial Intelligence: A Modern Approach" [2].

2.1 Simulator (10 points)

Implement a performance-measuring environment simulator for the vacuum-cleaner world. This world can be described as follows:

- **Perceptions.** Each vacuum-cleaner agent gets a three-element percept vector on each turn:
    1. A touch sensor, should be 1 if the machine has bumped into something and 0 otherwise
    2. A photosensor under the machine, which emits 1 if there is dirt there and 0 otherwise
    3. An infrared sensor, which emits 1 when the agent is in its home location and 0 otherwise

[2] Chapter 2 (Intelligent Agent), Stuart Russell and Peter Norvig, 1995

- **Actions**. There are five actions available:
  1. Go forward
  2. Turn right by 90^o
  3. Turn left by 90^o
  4. Suck up dirt
  5. Turn off
- **Goals**. The goal for each agent is to **clean up** and **go home**. To be precise, the performance measure will be:
  - Earn 100 points for each piece of dirt vacuumed up
  - Minus 1 point for each action taken
  - Minus 1000 points if it is not in the home location when it turns itself off
- **Environment**. The environment consists of a **grid of squares**:
  - Some squares contain **obstacles** (walls and furniture) and other squares are **open space**
  - Some of the open squares contain **dirt**
  - Each **go forward** action moves one square unless there is an obstacle in that square, in which case the agent stays where it is, but the touch sensor goes on
  - A **suck up dirt** action always cleans up the dirt
  - A **turn off** command ends the simulation

Create a vacuum-cleaner world consisting of a at least 5 x 5 grid of squares, at least 8 squares contain obstacles and at most 2 squares will contain dirt. Start the vacuum-agent from upper-left corner facing right, run the simulation and record its performance.

Repeat the simulation for at least 3 different worlds and compare the performances. Draw the environments you created with a drawing tool and upload with your source code.