**Virtual Initio Programming: Exercises with Inheritance**

**AIM:** This exercise sheet provides additional exercises using inheritance.

**Exercise 1:** Create a class LineFollower() that is a sub-class of InitioAgent. This class should follow the line in **line\_following.xml** world when the method run\_agent() is called. The robot should move forward if its line sensors are either side of the line and turn left or right, as appropriate, if one of the line sensors detects the line. The agent will need to start on the line to work properly.

**Exercise 2:** Create a class ProximityActivatedAgent() that is a sub-class of InitioAgent. This agent should

1. add a belief started when the an obstacle is detected on the left (then wait for a couple of seconds for the obstacle to be removed).
2. The agent should add a belief stopping when an obstacle is detected on the left *and* the agent believes started, and then it should drop the belief started.
3. Lastly the agent should stop the Initio and call the done() method in the InitioAgent class, if it believes stopping.

**Exercise 3:** Adapt your agent from Exercise 1, so that it is a sub-class of ProximityActivatedAgent.

**Exercise 4:** Create a class WallFollower() that is a sub-class of ProximityActivatedAgent. This class should follow a wall when the method run\_agent() is called. The robot should move forward if the agent detects a wall on its right, turn left if it detects and obstacle to the front and a wall to its right, and turn right if it doesn’t detect a wall to its right. The agent will need to start next to a wall to work properly. The agent should also stop if a black surface is detected. You can test this agent in **house.xml** world or **zigzag.xml** world.

**Hint:** You will need to use beliefs about distance (from the ultrasonic sensor) for this and this will be difficult since those beliefs return a number from the belief base dictionary, not true or false, so they can’t be used with agent.B (which only works with beliefs that return true or false). Instead you can define the following function:

def b\_obstacle\_centre(self):

if (self.beliefbase['distance'] < 30):

return True

return False

This can be used in conditions and with agent.AND, agent.OR and agent.NOT. For instance:

wall\_in\_front = self.AND(agent.B('started'), self.b\_obstacle\_centre)

self.add\_condition\_rule(wall\_in\_front, left)



University of Liverpool, 2020

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).