**Virtual Pi2Go Programming: Using Logic in BDI Rule Conditions**

**AIM:** After completing this worksheet you should be able to use logical expressions in the conditions for BDI rules.

**You Need:** To complete this worksheet you need to have a virtual Pi2Go simulator (see WS1), understand how to control the robot’s sensors and motors (WS3 & WS4). You should be able to run Python programs from files (WS5), understand Python’s time module (WS6), be able to use variables (WS12), strings (WS13), and functions (WS15) in Python programming. You should understand about objects (WS27) and cognitive agents (WS28 & WS29).

**If the simulator isn’t already running: Start the Simulator, Select the Pi2go Simulation and default\_world.xml, then start IDLE (open a *new IDLE window* if you have used IDLE to start the simulator).**

**BDI Rule Conditions:** You are accustomed when using if … then … else in normal Python programs to be able to use logic in the conditions – for instance using **and** or **not**. You can’t use the regular Python expressions for this in the conditions for BDI rules because the conditions are functions not expressions but the bdi.pi2goagent library provides substitutes for these which you can use instead.

|  |  |
| --- | --- |
| Function | Example |
| NOT | agent.NOT(agent.B(‘obstacle\_centre’)) |
| AND | agent.AND(agent.B(‘obstacle\_right’), agent.B(‘obstacle\_left’)) |
| OR | agent.OR(agent.B(‘obstacle\_right’), agent.B(‘obstacle\_left’)) |

Consider the following condition functions:

cond = agent.AND(agent.B('switch\_pressed'), agent.NOT(agent.B('obstacle\_centre')))

**Question 1:** When does the function **cond** return true?

cond = agent.AND(agent.B('switch\_pressed'), agent.NOT(agent.B('started')))

**Question 2:** When does the function **cond** return true? Remember that as well as getting beliefs from sensors agent programs can add beliefs to the belief base.

cond = agent.AND(agent.B('switch\_pressed'), agent.B('started'))

**Question 3:** When does the function **cond** return true?

cond = agent.B('started')

**Question 4:** When does the function **cond** return true?

Consider the following program.

import bdi.pi2goagent as cognitive

import time

agent = cognitive.Pi2GoAgent()

def start\_agent():

 agent.add\_belief('started')

 time.sleep(5)

 return

def stop\_agent():

 agent.drop\_belief('started')

 agent.add\_belief('stopping')

 time.sleep(5)

 return

def forward():

 agent.robot.forward(10)

 return

def stop\_rule():

 agent.robot.stop()

 agent.done()

 agent.drop\_belief('stopping')

 return

start = agent.AND(agent.B('switch\_pressed'), agent.NOT(agent.B('started')))

stop = agent.AND(agent.B('switch\_pressed'), agent.B('started'))

agent.add\_condition\_rule(start, start\_agent)

agent.add\_condition\_rule(stop, stop\_agent)

agent.add\_condition\_rule(agent.B('started'), forward)

agent.add\_condition\_rule(agent.B('stopping'), stop\_rule)

agent.run\_agent()

**Question 5:** What will happen when it is run?

**Exercise 1:** Modify the program so that it starts when the switch is briefly pressed and stops when it is pressed again. When it has started it moves forward when it does not believe there is an obstacle in front of it and turns when it does believe there is an obstacle in front of it.



 University of Liverpool, 2019

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