

February 2, 2023

## 1 Task 1

Imagine a single-floor office building with a fire alarm system that is controlled by a simple reflex agent. The system has smoke detectors and temperature sensors placed throughout the building to detect any signs of fire. The agent's rules are as follows: 1. if smoke is detected, the alarm will sound and the sprinkler system will activate to put out the fire; 2. if a high temperature is detected, the alarm will sound and the fire department will be called. 3. If neither smoke nor high temperature are detected, the system remains in its normal state with the alarm off and the sprinkler system deactivated. The goal of the agent is to keep the building and its occupants safe by quickly and efficiently responding to any signs of fire. Write a program to develop a simple reflex Agent.

```
[1]: # Temperature > 100 'the alarm will sound and the fire department will be
    ↳called.'
# if is_smoke = 1 the alarm will sound and the sprinkler system will activate
    ↳to put out the fire;
# if high_temperature and smoke do both
# if no smoke and high_temp the system remains in its normal state with the
    ↳alarm off and the sprinkler system deactivated.

percept = ['smoke_and_temperature_detector']
states = ['smoke', 'temperature']
rules = ['sound the alarm and activate the sprinkler system to put out the
    ↳fire', 'sound the alarm and call the fire department', 'keep the system in
    ↳normal state with the alarm off and the sprinkler system deactivated']

def getState(is_smoke, temp):
    if (is_smoke and temp > 100):
        return states
    elif (is_smoke and temp < 100):
        return states[0]
    elif (not(is_smoke) and temp > 100):
        return states[1]
    else:
        return []

def getRules(state_value):
```

```

if (len(state_value) == 2):
    return rules[0],rules[1]

elif (state_value == 'smoke'):
    return rules[0]

elif (state_value == 'temperature'):
    return rules[1]

elif (len(state_value) == 0):
    return rules[2]

def simpleReflexAgent(visual_input_one, visual_input_two):
    return getRules(getState(visual_input_one, visual_input_two))

test = 0
while(test < 5):
    visual_input_one = bool(int(input('Is there is smoke in the room [0/1]')))
    visual_input_two = int(input('Enter the Temperature of the building'))
    rule = simpleReflexAgent(visual_input_one, visual_input_two)
    if (len(rule) == 2):
        print(rule[0], "and", rule[1])
    else:
        print(rule)
    test += 1

```

Is there is smoke in the room [0/1] 0

Enter the Temperature of the building 55

keep the system in normal state with the alarm off and the sprinkler system deactivated

Is there is smoke in the room [0/1] 1

Enter the Temperature of the building 99

sound the alarm and activate the sprinkler system to put out the fire

Is there is smoke in the room [0/1] 1

Enter the Temperature of the building 105

sound the alarm and activate the sprinkler system to put out the fire and sound the alarm and call the fire department

Is there is smoke in the room [0/1] 0

Enter the Temperature of the building 105

sound the alarm and call the fire department

Is there is smoke in the room [0/1] 1

Enter the Temperature of the building 55

sound the alarm and activate the sprinkler system to put out the fire

## 2 Task 2

An Automatic Watering System is set up in a greenhouse. The system has sensors that detect the moisture level in the soil, and a control unit that operates the watering system. The task of the simple reflex agent program is to control the watering system based on the moisture level of the soil. Percepts: Moisture level sensor States: Dry soil, Moist soil, Wet soil Rules: 1. If the moisture level sensor detects dry soil, the agent activates the watering system to water the plants. 2. If the moisture level sensor detects moist soil, the agent keeps the watering system off to avoid over watering the plants. 3. If the moisture level sensor detects wet soil, the agent deactivates the watering system to prevent waterlogging.

```
[1]: # 80 to 90 (dry_soil), 90 to 100 (moist_soil), 100 to 110 (wet_soil)
percept = ['moisture_level_sensor']
states = ["dry_soil", "moist_soil", "wet_soil"]
rules = ['activates the watering system to water the plants.', 'keeps the_
↳watering system off to avoid over watering the plants.', 'deactivates the_
↳watering system to prevent waterlogging.']

def getState(percept_value):
    if (percept_value >= 80 and percept_value <= 90):
        return states[0]
    elif(percept_value >= 90 and percept_value <= 100):
        return states[1]
    elif (percept_value >= 100 and percept_value <= 110):
        return states[2]

def getRules(state_value):
    if (state_value == 'dry_soil'):
        return rules[0]
    elif (state_value == 'moist_soil'):
        return rules[1]
    elif (state_value == 'wet_soil'):
        return rules[2]

def simpleReflexAgent(percep):
    return getRules(getState(percep))

test = 0
while test < 5:
    visual_input = int(input('Enter moist level'))
    rule = simpleReflexAgent(visual_input)
    print(rule)
    test += 1
```

Enter moist level 90

activates the watering system to water the plants.

Enter moist level 91

keeps the watering system off to avoid over watering the plants.

Enter moist level 101

deactivates the watering system to prevent waterlogging.

Enter moist level 81

activates the watering system to water the plants.

Enter moist level 109

deactivates the watering system to prevent waterlogging.