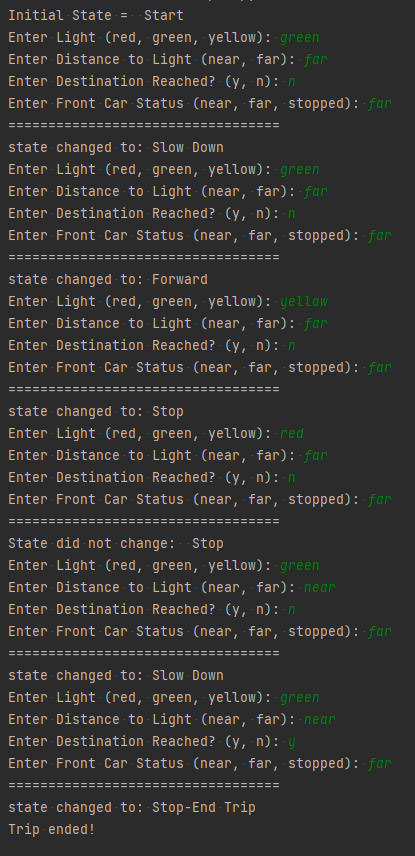
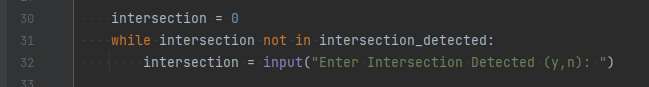
Lab 10

Using a FSM to program a Self-Driving Car Agent



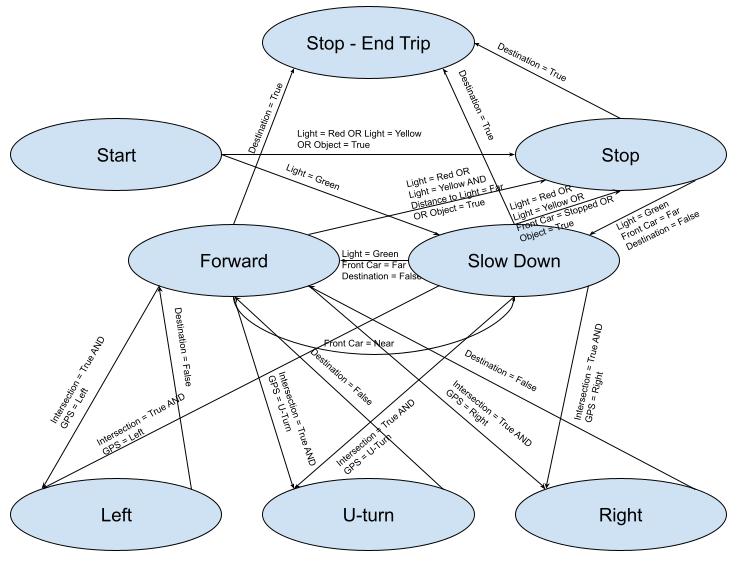
1. You will need to add sensors that allows the agent to decide whether the intersection was reached or not. You will also need to consider the GPS command as an input. The commands are for making the turns. List new inputs and their possible values.
   1. intersection\_detected = {'y': 0, 'n': 0}
   2. y,n



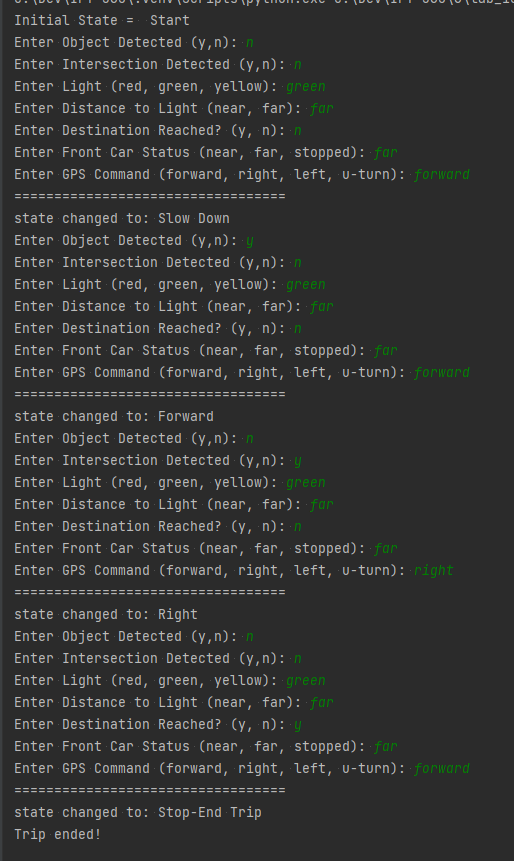
* 1. gps\_commands = {'forward': 0, 'right': 0, 'left': 0, 'u-turn': 0}
  2. forward, right, left, u-turn



1. List the new set of actions that the agent can take.
   1. "Right",
   2. "Left",
   3. "U-turn"
   4. states = {"Start", "Stop", "Slow Down", "Forward", "Stop-End Trip", "Right", "Left", "U-turn"}
2. Modify the FSM diagram to account for these actions, percepts and show the corresponding transition model.



## Modified code results



# -\*- coding: utf-8 -\*-

"""

Created on Sun Sep 25 11:30:50 2022

@author: aelbadra

"""

# Define All States (also represents actions):

states = {"Start", "Stop", "Slow Down", "Forward", "Stop-End Trip", "Right", "Left", "U-turn"}

# Define Different Input Percepts and their values

light = {'red': 0, 'green': 0, 'yellow': 0}

dist\_to\_light = {'near': 0, 'far': 0}

destination\_reached = {'y': 0, 'n': 0}

front\_car = {'near': 0, 'far': 0, 'stopped': 0}

object\_detected = {'y': 0, 'n': 0}

intersection\_detected = {'y': 0, 'n': 0}

gps\_commands = {'forward': 0, 'right': 0, 'left': 0, 'u-turn': 0}

# set initial state

state = 'Start'

print('Initial State = ', state)

# start driving!

while True:

o = 0

while o not in object\_detected:

o = input("Enter Object Detected (y,n): ")

intersection = 0

while intersection not in intersection\_detected:

intersection = input("Enter Intersection Detected (y,n): ")

# read input percept:

l = 0; dtl = 0; dst = 0; fc = 0; gps = 0

while l not in light:

l = input("Enter Light (red, green, yellow): ")

while dtl not in dist\_to\_light:

dtl = input("Enter Distance to Light (near, far): ")

while dst not in destination\_reached:

dst = input("Enter Destination Reached? (y, n): ")

while fc not in front\_car:

fc = input("Enter Front Car Status (near, far, stopped): ")

while gps not in gps\_commands:

gps = input("Enter GPS Command (forward, right, left, u-turn): ")

# ========================

print("==================================")

if state == 'Start':

if l == 'red' or l == 'yellow' or o == 'y':

state = 'Stop'

print("state changed to:", state)

continue

elif l == 'green':

state = 'Slow Down'

print("state changed to:", state)

continue

else:

print('State did not change: ', state)

# ========================

elif state == 'Stop':

if dst == 'y':

state = 'Stop-End Trip'

print("state changed to:", state)

print('Trip ended!')

break

elif (l == 'green' and fc == 'far' and dst == 'n') or o == 'y':

state = 'Slow Down'

print("state changed to:", state)

continue

else:

print('State did not change: ', state)

# ========================

elif state == 'Slow Down':

if dst == 'y':

state = 'Stop-End Trip'

print("state changed to:", state)

print('Trip ended!')

break # end the trip!

elif intersection == 'y':

state = gps.capitalize()

print("state changed to:", state)

continue

elif l == 'green' and fc == 'far' and dst == 'n':

state = 'Forward'

print("state changed to:", state)

continue

elif l == 'red' or l == 'yellow' or fc == 'stopped' or o == 'y':

state = 'Stop'

print("state changed to:", state)

continue

else:

print('State did not change: ', state)

# ========================

elif state == 'Forward':

if dst == 'y':

state = 'Stop-End Trip'

print("state changed to:", state)

print('Trip ended!')

break # end the trip!

elif (l == 'red') or (l == 'yellow' and dtl == 'far') or o == 'y':

state = 'Stop'

print("state changed to:", state)

continue

elif fc == 'near':

state = 'Slow Down'

print("state changed to:", state)

continue

elif intersection == 'y':

state = gps.capitalize()

print("state changed to:", state)

continue

else:

print('State did not change: ', state)

# ========================

elif state == 'Right' or state == 'Left' or state == 'U-turn':

if dst == 'y':

state = 'Stop-End Trip'

print("state changed to:", state)

print('Trip ended!')

break # end the trip!

else:

state = 'Forward'

print("state changed to:", state)

# ========================

elif state == 'Stop-End Trip':

print('Destination Reached! Trip Ended!')

break

# ========================

else:

print("ERROR: Unknown State!!!")

break

# charrel5

# 4/23/2023

