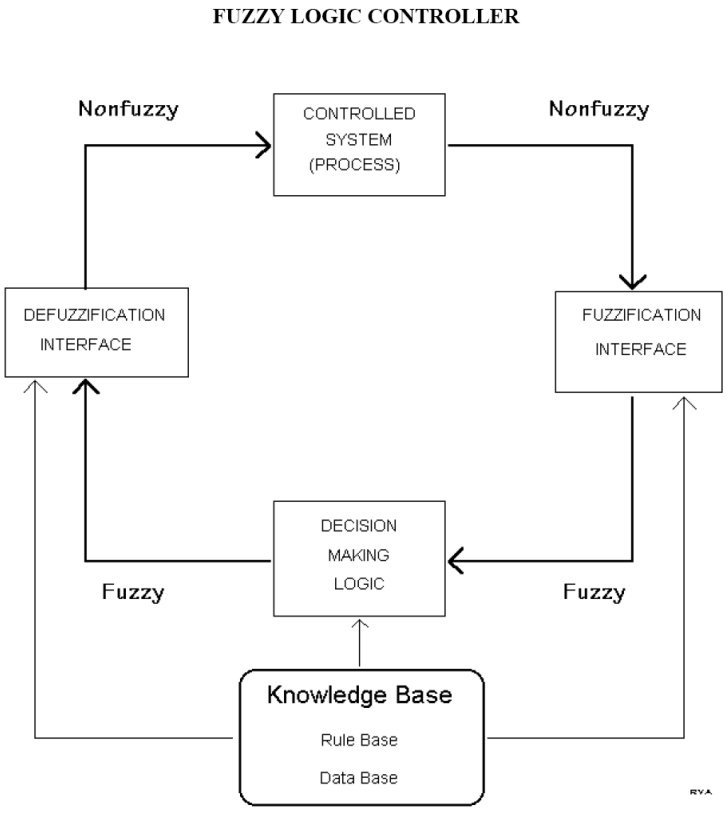
AML-Assignment#2
(Fuzzy Logic Control (PyFCSimbot))

After you have an experience in designing a reactive control for a very simple simulated robot last assignment. In this assignment, you will have to apply the idea of fuzzy logic control for mobile robot programming. The task and the robot are the same as the previous one. The proper steps for designing this fuzzy logic system are listed as follows…

1) From the task, select what the input and output signals are.
2) Design the fuzzy set for each signal
3) Define all membership functions for all input signals
4) Design a set of common rules toward solving the problem
5) Implement decision making process by the process of fuzzification and defuzzification
6) Try with the problem (executing the robot)
7) Tune and adjust the above designed parameters until you satisfy with the results.

Use the examples and exercise to guide you way to program it, it should not be too difficult. If it is hard for you, come and see me or TA before next due date.



You can do whatever you want; however, this is an example which shows some parts of the final version. At least, you should get an idea how fuzzy control is implemented.

```
class FuzzyRobot(Robot):

    def __init__(self):
        super(FuzzyRobot, self).__init__()
        self.pos = START_POINT

    def update(self):
        ''' Update method which will be called each frame
        '''
        self.ir_values = self.distance()
        self.target = self.smell()

        # initial list of rules
        rules = list()
        turns = list()
        moves = list()

        # rule 0
        rules.append(self.front_far() * self.left_far() * self.right_far())
        turns.append(0)
        moves.append(10)

        # rule 1
        rules.append(self.smell_left() * self.front_far() * self.left_far() * self.right_far())
        turns.append(-80)
        moves.append(5)

        # rule 2
        rules.append(self.smell_right() * self.front_far() * self.left_far() * self.right_far())
        turns.append(100)
        moves.append(5)

        ans_turn = 0.0
        ans_move = 0.0
        for r, t, m in zip(rules, turns, moves):
            ans_turn += t * r
            ans_move += m * r

        self.turn(ans_turn)
        self.move(ans_move)

    def front_far(self):
        irfront = self.ir_values[0]
        if irfront <= 10:
            return 0.0
        elif irfront >= 40:
            return 1.0
        else:
            return (irfront-10.0) / 30.0

    def front_near(self):
        return 1 - self.front_far()

    def left_far(self):
        irleft = self.ir_values[6]
        if irleft <= 10:
            return 0.0
        elif irleft >= 30:
            return 1.0
        else:
            return (irleft-10.0) / 20.0

    def left_near(self):
        return 1 - self.left_far()
```

```
    def right_far(self):
        irright = self.ir_values[2]
        if irright <= 10:
            return 0.0
        elif irright >= 30:
            return 1.0
        else:
            return (irright-10.0) / 20.0

    def right_near(self):
        return 1 - self.right_far()

    def smell_right(self):
        target = self.smell()
        if target >= 90:
            return 1.0
        elif target <= 0:
            return 0.0
        else:
            return target / 90.0

    def smell_center(self):
        target = abs(self.smell())
        if target >= 45:
            return 1.0
        elif target <= 0:
            return 0.0
        else:
            return target / 45.0

    def smell_left(self):
        target = self.smell()
        if target <= -90:
            return 1.0
        elif target >= 0:
            return 0.0
        else:
            return -target / 90.0

if __name__ == '__main__':
    app = PySimbotApp(FuzzyRobot, ROBOT_NUM, mapPath=MAP_FILE, interval=TIME_INTERVAL,
maxtick=MAX_TICK)
    app.run()
```

Note:

You also need to download PyFCSimbot.rar from the class's FB group and then use it to start your work, which is on the "assignment2.py" file.