ECE 479/579 Spring 2023
Final Project

Due: May 5th, 2023 11:59pm via D2L

Documents to be submitted:

1. **Project report**
2. **Code (in C/C++, Python, Matlab)**

You are to design an AI expert system FOODIE (Food Intelligence Electrified) to run a network of food delivery robots in a locale (e.g., a university campus). In what follows, very broad requirements are given with specific goals. Your task is to refine them as you see fit and most appropriate to make such as a system as automated, efficient, and smart. **You have a "creative license" to propose an interesting solution.**

My initial assumptions are:

1. The locale is a known bounded terrain. More specifically, it has pathways for the robots to move about. Robots may encounter obstacles along pathways. Such obstacles may not be known a priori.
2. There is a finite number of movable robots, each equipped with HW/SW and "intelligence" that can be as complex as you will desire for it to be. In addition, robots have compartments, and a robotic arm that allows them to pick and bag items, and place bags in the compartments.
3. Robots will depart from and return to a food warehouse (FW) in which they:
    a. Park and recharge.
    b. Receive orders for food items.
    c. Bag the items.
    d. May deliver more than one order to more than one location and then return to the FW.
    e. Load up the items into their compartment.
4. Once a robot has configured and loaded the order, it departs to the destination specified in the order.

Propose the design of FOODIE (and implement some of its aspects as required below) with the following goals:

A Optimize the routes that the robots take based on the orders received.

For this goal, select or propose an algorithm that minimizes delivery time and minimizes energy consumption by robots. Consider potential for replanning the route as robots may encounter small or intractable obstacles (for instance, a pathway closed to traffic due to construction).

Please remember that you have more than one robot and several of them can deliver several orders concurrently.

For this task, implement your solution and demonstrate it on a small simulation.

B Propose an expert module to bag (FOODIE_BAGGER) the food items into a robot's compartment.

FOODIE will have a rule-based system that will decide where each item from the order will go. Specifically, a) the bag large items step will load large items first (for instance, big bottles of water), b) the bag medium items step will load medium items next, and c) bag small items step will be next. Assume that frozen items need freezer bags. Other items need paper bags of equal size but clearly capable of only accommodating a finite number of items, depending on the items' sizes.

A rule might have the following form:

R: if step is bag-medium items
there is a medium item.
the current bag contains < 10 items.
then add this item to the bag.

Your rules should make sure that fragile items do not get crushed in a bag.

Run a simulation of this rule-based system for a small order, for example, 2 x 1-gallon bottles of water, pint of ice cream, granola box, loaf of bread. (Make up your own order as input). A sample trace of your simulation might look as follows:

```
Rule Ri says:   Bag lager items.
Rule Rk says:   Put 1 gallon water bottle in bag_1.
Rule Rm says:   Put watermelon in bag_1.
Rule Rx says:   Start a new bag.
Rule Ry says:   Put ice cream in a freezer bag.
…
…
```

C Propose an expert system module (FOODIE_Springs_to_Action – Foodie_SPA) to save you when guests show up unexpectedly at your house.

Design a backward chaining system and a small attendant rule base for selecting the right beverage for your guests, that needs to be added to the order. Imagine that in general, the beverages can be water, juice, wine, beer, liquor, with specific brands for each type for example Corona beer, or carrot juice (*if guest is a health nut and guest has allergies to citrus, then choose carrot juice*).

Foodie_SPA will take your hypothesis (for instance, "*shall I serve carrot juice*?"), backward chain through the rule base and if needed, will ask you for facts that you know to be true (for example, *guest is not well liked, entre is chicken, it's New Year's Eve*).

Implement such a system (no more than 15 rules) and show its operation on your own example. A trace of execution might look as follows.

```
Trying to establish CHOOSE Polish Vodka using rule Rx
Trying to establish LIQUOR is indicated using Rm
Rule Rm fails to establish LIQUOR is indicated.
….
….
CHOOSE Dos Equis is True.
```

IMPORTANT: FOR IMPLEMENTATION, YOU HAVE A CHOICE TO EITHER IMPLEMENT FOODIE_BAGGER OR FOODIE_SPA. HOWEVER, YOU HAVE TO DEVELOP RULE BASES FOR BOTH AND INCLUDE THEM IN YOUR REPORT.

D Show how the robot would use its arm to load the bags into its compartment.

Use STRIPS basic rules or modified rules to describe the process and what actions should be in place in case the order has to be modified by last minute change (i.e., unexpected guests arrive). You need not to encode this part.

Report

Your report need not be long and elaborate. In it, I expect you to state your own assumptions in additions to mine, the algorithms and methods selected, rules bases with rules, and STRIPS rules for section D. In the appendix, please attach sample runs for sections (A and B) or (A and C) as you have a choice of implementing either B or C.