Homework 4

COS470/570 Fall2024

DUE: Nov 8th 2024

Goal

Use Prolog to escape the spooky mansion.

Instructions

I have provided the beginnings of a Prolog solution with the basic facts stated to get you started. Your job is to provide the code that will cause the player to collect the three things they need to escape the mansion: the key, the code, and the message.

The player starts in the kitchen. They must collect the three things and then move to the gate. When they possess all three things and are located at the gate, the game is won.

The catch is, your solution must use knowledge of the map to find paths between the rooms with the different items. It cannot hard-code a series of moves given the current map. It must search through the edges.

The message, code and key can be collected in any order.

You do not have to discover the location of the things. This information is provided in the facts included in the .pl file. In other words, the player knows where the things are. Your job is to move there and take them.

There are two predicates provided with alter the known facts: move and take.

The *move/1* predicate will cause the player's location to change to the room argument. But this will only happen if there is a connection between the rooms. If the move can happen, all location facts are retracted and a new one is asserted for the new room.

The *take/1* predicate will cause the player to take a thing, provided the player is in the same location as the thing. For example, if the player is in the lab and issues `take(message)`, the has/1 fact will be set such that has(message) is true. At the same time, the thing's location will be set to a fake location called, nowhere. This is to simulate that the thing is not in the location and prevent the player from taking it multiple times.

Outline of the Task

The problem has several steps.

- Find a path to a desired location.
- Use the found path to issue a series of `move` predicates.
- Take the thing at the destination.
- Repeat until all things are collected.
- Find the path to the gate.
- Move there.

I strongly suggest that you spend a few hours practicing and reading about Prolog before embarking on the task itself. Consider some of the sub-problems required by the homework and try to program outside the confines of the larger problem. Get comfortable with the syntax and the swipl interpreter. Don't try to rush this at the last minute. You have a lot of time. Spend some of it absorbing the language.

One thing to note: the move and take commands alter facts in the knowledge base. This means they change what is true about the world model. When you rerun your code or reload your file, previously changed facts are not retracted automatically. For example, if you have a partial solution that moves the player from the kitchen to the hall and you reload the file, the code I have provided will set location(kitchen) to true, but the previous fact, location(hall), will still be true. You should write a predicate that resets the game. This can be done using a combination of `retract`, `retractall` and `assertz`.

When grading I will issue two commands: - play(). - win().

The play/0 will initiate your solution. The win/0 will tell me if you've succeeded.

If you have any questions, please ask. Here is sample output from my code. The file I used was called *main.pl*. The command to load your knowledge base will depend on the name of your file. NOTE: Your code does not have to output each step of the solution like mine does. But it might help to debug where things are failing.

```
?- [main].
true.

?- play().
Starting in: kitchen
Moving to: hall
Moving to: tower
Moving to: lab
Found message
Moving to: tower
Moving to: diningroom
Moving to: kitchen
Moving to: hall
Moving to: basement
Moving to: crypt
Moving to: dungeon
Found code
Moving to: crypt
Moving to: basement
Moving to: hall
Moving to: garden
Moving to: church
Found key
Moving to: garden
Moving to: graveyard
Moving to: crypt
Moving to: basement
Moving to: hall
Moving to: gate
true .

?- win().
You are free of the spooky mansion.
true.
```

Figure 1: run.png