CSCI561 – Introduction to Artificial Intelligence
Instructor: Dr. K. Narayanaswamy

Assignment 4 – Due: - 11/07/2011 11:59:59pm

Problem4a
This is a world-famous logic problem (**Einstein's Riddle**). It is said to be written by Albert Einstein early during the 20th century, although there is no concrete proof of this. He said that 98% of the world's population would not be able to solve it. Are you one of the 2%?

- There are 5 houses that are each a different color.
- There is a person of a different nationality in each house.
- The 5 owners drink a certain drink. They each smoke a certain brand of cigarettes and also have a certain pet. No owner has the same pet, smokes the same brand of cigarettes nor drinks the same drink.
- The question is. "Who has the fish?"

Clues:
1. The British man lives in the red house.
2. The Swedish man has a dog for a pet.
3. The Danish man drinks tea.
4. The green house is to the left of the white house.
5. The owner of the green house drinks coffee.
6. The person that smokes Pall Mall has a bird.
7. The owner of the yellow house smokes Dunhill.
8. The person that lives in the middle house drinks milk.
9. The Norwegian lives in the first house.
10. The person that smokes Marlboro lives next to the one that has a cat.
11. The person that has a horse lives next to the one that smokes Dunhill.
12. The one that smokes Winfield drinks beer.
13. The German smokes Rothmans.
14. The Norwegian lives next to a blue house.
15. The person that smokes Marlboro has a neighbor that drinks water.

Questions:
1. Please construct a prolog file 5houses.pl to represent the knowledge base.
2. Please use GNU prolog to solve this Einstein's Riddle, and writes out your query sentences and their outputs in the Readme.txt file.

Problem 4b
Seven students from Springfield Elementary were on a fieldtrip to a river near town. They decided to row down the river, and each student was given their own seat on the boat, all in a single row. Unfortunately, after lunch, they all forgot their places and have all gotten out of their proper seats. As the teacher's assistant, you must help put them back in their correct seats, but this is all you remember.
Students: Bart, Lisa, Martin, Ralph, Milhouse, Jimbo, and Nelson

Clues:
1. Nelson doesn't like the other students, so he was on one of the ends.
2. Milhouse was somewhere to the left of Bart.
3. Jimbo was in the third seat from the right.
4. The only student between Martin and Ralph was Milhouse.
5. Lisa was directly to the left of Ralph.

What is the seating order of the students on the boat?

Questions:
1. Please construct a prolog file springfield.pl to represent the knowledge base.
2. Please use GNU prolog to solve the Springfield riddle, and writes out your query sentences and their outputs in the Readme.txt file.

Hints:

There are many ways to solve these problems. One of the elegant solutions is to formulate the problem using the Prolog list primitives. This involves the ability to combine elements into Lists and to pull them apart using Prolog's built-in | operator and Prolog's built in "member" function for lists.

The class notes and tutorials on the web site show how to use these operators. We have described the relevance of Prolog list utility to solving these problems more easily.

Lists are useful in this assignment because the configurations of the 5houses and Springfield problems are best represented as a Prolog list, since the problem mentions sequences or positions of objects.

- Where you know the facts, you can fill them in. For example, if Nelson sat in the middle seat, you could represent that fact as: [_, _, _, Nelson, _, _, _] where "_" character is used as a wild card that matches any value in the slot where it appears, rather than introducing a variable that you do not use.

- If Babe Ruth was in the third spot from the right, you could represent that as: [_, _, _, Babe Ruth, _, _] and so on.

Once you fill out all the information using the above representation, Prolog will help you answer the other questions through backward chaining! So, basically Prolog will do the heavy lifting, if you represent the “facts” of the problem correctly.

You might find the following helper concepts to be useful as well. They are used to define "next_to" and "left" and "right" -- which are used in the problem statement. You can use "iright" for that.

```
next_to(X, Y, List):- iright(X, Y, List).
next_to(X, Y, List):- iright(Y, X, List).
iright(L, R, [L | [R | _]]).
iright(L, R, [_ | Rest]) :- iright(L, R, Rest).
```

You can build many more helper functions as you need them in the same style. The 5houses problem can be represented as a list of lists, which Prolog is able to do pretty easily.

a) The configuration of houses in the problem is best represented as a List because the problem mentions sequences of houses.
b) We also prefer to think of each house as a list:
[nationality, pet, cigarette, drink, house color] for example.

So, the list of the houses can be represented as a list of lists, containing 5 elements. Where you know the facts, you can fill them in. For example, if the Norwegian owned the horse, you can represent that fact as: [Norwegian, horse, _, _, _] Remember that the "_" character is used as a wild card that matches any value in the slot where it appears, rather than introducing a variable that you do not use. Or if the British person lived in the white house, you can represent that as: [British, _, _, _, white] and so on.

Once you fill out all the information using the above representation, you must ask the proper query, and Prolog will help you answer the query through backward chaining!

Grading Policy: (Total Points: 100)

Problem4a [10 + 50 Points]
1) Correct query sentence (to be provided via the Readme.txt file described below) **[10 Points]**
2) Correct representation of the knowledge base **[50 Points]**

Problem4b [10 + 30 Points]
1) Correct query sentence (to be provided via the Readme.txt file described below) **[10 Points]**
2) Correct representation of the knowledge base **[30 Points]**

Readme.txt:
1. A brief description of the program structure, and any other issues you think that will be helpful for the grader to understand your program.
2. Instruction on how to compile and execute your code. Submit a makefile if necessary.
3. Please include your name, student ID and email address on the top.
4. You must submit a program in order to get any credit for the Readme.txt. In short, if you submit ONLY a Readme.txt file you will get 0.

We will test your programs by loading the file you provide into gprolog on aludra.usc.edu. Then, we will use the query sentence that you provide in the readme.txt file to test your program. You must make sure that the file you submit works on that interpreter. Otherwise, the maximum credit you get will be 20% for program correctness.

Submission: Your program files will all be submitted via blackboard. You MUST follow the guidelines below.

1) Compress and zip ALL your homework files (this includes the Readme.txt and all source files you wish to include) into one .zip file. There are 3 files for this assignment: springfield.pl, nemo.pl and Readme.txt. Note that only .zip file extensions are allowed. Other compression extensions such as .tar, .rar, or .7z will NOT be accepted.
2) Name your zip file as follows: firstname_lastname.zip. For example John_Smith.zip would be a correct file name.
3) To submit your assignment, simply select the appropriate assignment link from the Assignments subsection of the course blackboard website. Upload your zip file and click submit.

Please make sure ALL source files are included in your zip file when submitted. Errors in submission will be assessed –25 points. A program that does not compile as submitted will be given 0 points. Only your FINAL submission will be graded.

For policies on late submissions, please see the Syllabus from the course home page. These policies will be enforced with no exceptions.