

CSCI 561 – Foundations of Artificial Intelligence

Instructor: Dr. K. Narayanaswamy

Assignment 4 – Prolog

Due: May 3rd 2013 11:59:59pm

The purpose of this assignment is to give you experience in dealing with GNU Prolog/gprolog systems and working with practical systems that support Horn knowledge bases and inference by using unification and modus ponens/backward chaining.

Problem: It is now time to register for 2013 Summer courses. Since Operating Systems is quite a popular course, all allocated spaces are taken up and there are six USC students (Jacob, Noah, Ethan, Olivia, Emma, and Chloe) in the waiting list. Each student in the waiting list has a different nationality. They each like a certain kind of fruit and each has a particular hobby. No two of them have the same hobby or like the same fruit. Each student prefers to buy fruit from some fixed seller on Amazon, and there are 6 sellers, who are Jones, Moore, Harris, Maria, Sharon and Lisa. No two of the students buy fruit from the same seller.

Clues:

1. Emma comes from Cuba and buys fruit from seller Maria.
2. The student, who likes to eat Coconut, also likes to play Badminton.
3. Jacob likes to eat Blackberry, and in the waiting list he is in front of the student who buys fruit from seller Harris.
4. The student who comes from Jamaica is the last one in the waiting list.
5. Olivia likes to run the Marathon in her spare time, and she comes from Belgium.
6. The student, who buys fruit from seller Jones, comes from Australia.
7. The student, who has a hobby of Shooting, buys fruit from seller Harris.
8. Noah has a hobby of Swimming.
9. The student, who loves to eat Mango and buys them from seller Lisa, never plays Badminton or Volleyball in his or her spare time.
10. Ethan has a hobby of Diving and likes to eat Cherry.
11. The student who comes from Morocco likes to eat Grapes.
12. Chloe and Jacob are next to each other in the waiting list (i.e. they are neighbors in the list).
13. Chloe is also next to the student who likes to eat Coconut.
14. Jacob buys fruit from seller Moore.
15. The student, who has a hobby of Shooting, is behind the student who comes from Chile in the waiting list.

16. Noah likes to eat Pineapple.
17. The student who likes to eat Mango is behind the student who comes from Australia in the waiting list.
18. Both Emma and Jacob are not at the front of the waiting list.

Part I:

Questions:

- a) How many solutions are consistent with the clues?
- b) Who is the second student in the waiting list?

Programming:

1. Please construct a prolog file students1.pl to represent the knowledge base in questions (a) and (b).
2. Please use GNU prolog or gprolog to solve the above riddle and to find the answers to the questions above. Write out your query sentences and their outputs in the Readme.txt file.

Part II:

Changes:

1. Replace Clue 13 with 'Jacob and Olivia are not neighbors in the waiting list';
2. Add Clue 19: Olivia stands between Emma and Ethan.

Questions:

- c) What is the order of the students in the waiting list?
- d) Which Amazon seller does Noah buy fruit from?

Programming:

1. Please construct a prolog file students2.pl to represent the knowledge base in questions (c) and (d).
2. Please use GNU prolog or gprolog to solve the above riddle and to find the answers to the questions above. Write out your query sentences and their outputs in the Readme.txt file.

Hints:

There are many ways to solve this problem. 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 "member" function.

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 this problem

more easily.

Lists are useful in this assignment for two reasons:

- a) The configuration of students in the problem is best represented as a List because the problem mentions the order of the students in the waiting list.
- b) We also prefer to think of each student as a list: [Name, Hobby, Nationality, Fruit, Seller], for example.

So, the list of the students can be represented as a list of lists, containing 5 elements. Where you know the facts, you can fill them in. For example, if Jacob has a hobby of Swimming, you can represent that fact as:

[Jacob, Swimming, _, _, _]

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. Or if Noah likes to eat Coconut, you can represent that as:

[Noah, _, _, Coconut, _]

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 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).

Grading Policy: (Total Points: 100)

Program Correctness (95 pts):

1. Correct representation of each clue: 3 pts*20 = 60 pts
2. Correct query sentence and output for Question a) & b): 15 pts
3. Correct query sentence and output for Question c) & d): 20 pts

We will test your program by loading the file you provide into gprolog on aludra.usc.edu. You must make sure that the file you submit works on that interpreter. Otherwise, the maximum credit you get will be 30% for program correctness.

Readme.txt (5 pts):

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. Please include your name, student ID and email address on the top.
3. Your query sentences and solutions for Questions a), b), c) and d).
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.

Submission: Your program files will all be submitted via blackboard. You MUST follow the guidelines below. Failure to do so will incur a -25 point penalty.

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 2 files for this assignment: **students.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.

Plagiarism on Programming Assignments: Discussion of concepts, design issues, and programming ideas with colleagues and fellow students is proper and an integral part of learning. However, everyone should do their own coding. Code will be automatically compared and any suspected cases of plagiarism will be investigated and pursued thoroughly. If you happen to reuse snippets of code from the Internet or other sources, please document the sources and nature of what was used in detail to avoid potential problems. If you have any questions or doubts regarding the proper modes of collaboration or using external reference sources, please check with the Professor or TA before you submit your assignment.