CSC 3310 Concepts of Programming Languages

Fall 2018

Assignment Name	Prolog Programming Assignment
Due Date	October 17th, 2018
Delivery Method	Through Canvas Only
Points	10% of final grade

Motivation

Artificial Intelligence has gotten more attention in recent years. There are more AI tools available to the public than ever before: Alexa, Siri, Ericka, etc. not to mention services like Google Cloud Machine Learning, Deep Learning on AWS, IBM Cognitive, Microsoft Azure Machine Learning, etc. One of the early attempts to deal with AI was the development of "AI specific" languages. This is the case of Prolog, a declarative language that stores facts and rules (and other constructs as well), and has an interface where the user can query and make questions that Prolog's engine will answer inferring from the facts and rules that where stored.

Another motivation for this assignment is to expose students to a completely different paradigm of programming. Up to this moment, most of the students have been working and coding with imperative and object oriented languages, and a declarative language should pose a new challenge, as it requires a different way to think.

Description

Write a Prolog program that stores information about geometric objects. You must construct rules for geometric objects. The objects to construct are:

- 2D Point
- Triangle

The program must be able to answer the following questions:

English	Prolog
Is the line defined by two points vertical?	`vertical(point2d(x,y), point2d(x,y))`
Is the line defined by two points horizontal?	`horizontal(point2d(x,y), point2d(x,y))`
Do the given three points define a line?	`line(point2d(x,y), point2d(x,y), point2d(x,y))`
Do the given three points define a triangle?	`triangle(point2d(x,y), point2d(x,y), point2d(x,y))`
Is the triangle isosceles?	`isosceles(point2d(x,y), point2d(x,y), point2d(x,y))`
Is the triangle equilateral?	`equilateral(point2d(x,y), point2d(x,y), point2d(x,y))`
Is the triangle right?	`right(point2d(x,y), point2d(x,y), point2d(x,y))`
Is the triangle scalene?	`scalene(point2d(x,y), point2d(x,y), point2d(x,y))`
Is the triangle acute?	`acute(point2d(x,y), point2d(x,y), point2d(x,y))`
Is the triangle obtuse?	`obtuse(point2d(x,y), point2d(x,y), point2d(x,y))`

Assignment Requirements

- Good programming practices
    - Meaningful atoms and rules naming
- This is an strictly individual assignment
- A query file will be provided for you to test your program, this test (and additional ones) will be used for grading.
- Use the rules names given in the table above.

Delivery Method

- Files to be uploaded
    - `threepoints.pl` [You MUST name your program this, <u>failure will result in zero grade</u>]
- Uploaded in Canvas

Assessment and Grading

Assessment will consider the following factors in the grading of the project:

- Adherence to instructions
- Correct function of the program
- No runtime errors and no warnings
- Late deliveries will have a zero mark
- Plagiarism will have a double zero mark (in addition to losing 10% of your final grade, you will lose an additional 10% of their final grade), besides there will be a report filed in the students' academic record.
- Each program will be loaded in Prolog and tested to check if the functions are in working order.
- The programs will be "automatically" run using a bash shell script, it is important that you follow the instructions, so the script runs smoothly.

Extra Challenge

- Use Prolog more advance features in this project: Lists, IO and Structures. You will need to send the test queries to try your code while grading. (up to 3 extra points)
- Use points in 3D. You will need to send the test queries to try your code while grading. (up to 5 extra points)
- To take advantage of the extra credit, you must comply with the given tests first.

Instructions for Testing

A file named `test.pl` is provided. You will need to append this file to the end of your code, and then run Prolog from the command line: `swipl -q -f full.pl -t main > your-output.txt`

The command tells prolog to run **q**uietly using the **f**ile `full.pl` and to execute **t**arget `main`. The query main will then run each of the queries. Once the process is finished, there will be a file named `your-output.txt` you can then compare this file with another file that is provided: `output.txt` to check if your results are correct. Before turning in your assignment, you must delete the test appended lines in your code.

Steps:

1. Append the file test.pl at the end of your code
2. Run the following command:
   `swipl -q -f full.pl -t main > your-output.txt`
3. Compare the file your-ouput.txt with the provided output.txt, you may use diff to compare the files.
4. If your output matches, then you are ready to turn in your work
5. Remove the lines appended in step 1
6. Turn in `threepoints.pl`