CS3360: Design and Implementation of Programming Languages

Programming Project #4
Due: December 6th at 11:59pm via email to your TA

Notes:

This project is to be done ***individually***.

The program must fulfill the requirements, regardless of the interface. This means that the quality of the interface will not be graded, as long as it clearly shows that the requirements are fulfilled.

The grading of this project will be as follows: ***40% for the program*** you have to implement, ***55% for the report*** (*see below for details*), and ***5% for your progress report*** at mid-point, in person to your TA.

Goals of the project: Develop your understanding of the logic programming through the practice of Prolog. Understand how to design more efficient Prolog programs by appropriately designing predicates.

To be done:

1. Write a Prolog program that*, given a partially filled Sudoku board, fills it to solution or concludes that there is no solution.*

Requirement: Make use of Prolog's cut symbol and everywhere it appears, explain why you put it there and what it is expected to do, how it is expected to help.

2. In a supporting document:

1) Describe what your program does and how to use it (including a readme file);

2) Explain each of the predicates you implemented and justify each of their implementations;

3) Justify why your program runs efficiently (refer to the size and shape of the search tree).

4) Reflect on the main differences between the OOP, Functional Programming, Logic Programming. In particular, focus on the variables of your Prolog program and explain their specific features as opposed to variables you used in the other programming projects.

Note: *a template of the report will be provided on piazza.*

Deliverables:

- Prolog program
- A readme file

- A report

Note: Make sure to follow best programming practices (including proper indentation, commenting, documenting, ...) as well as the report template (provided on piazza by your TA).