Comp 333 Project #5 (35 pts) **Problem 3 REVISED**

Due: May 11

**GENERAL DIRECTIONS:** In this project you will write and compile and run a set of Prolog programs. Use SWI-Prolog Your source code file must be named project5.pl.

1. Write recursive Prolog predicates to determine if a list of atoms satisfies the following regular expressions or grammar rules. You may write helper predicates. Do no use any of the “grammar rule capabilities” of Prolog. Write your own rules using predicates we have discussed in class. Note that append(L1,L2,L3) will be a very helpful predicate to use in some cases.
   1. **a+** listOfa(L) returns true only if L is a list of one or more a’s
   2. **b** + listOfb(L) returns true only if L is a list of one or more b’s
   3. **c+** listOfc(L) returns true only of L is a list of one or more c’s
   4. **a+b+** listOfab(L) returns true only if L is a list of a’s followed by b’s
   5. **a+b+c+** listOfabc (L) returns true only if L matches the pattern **a+b+c+**
   6. **bba\*bbb** regExpr1(L) returns true only if L matches the pattern **bba\*bbb**
   7. **(aa |cc)+** regExpr2(L) returns true only if L matches the pattern **(aa | cc)+**
   8. **a\*b\*c\***  regExpr3(L) return true only if L matches pattern **a\*b\*c\***
   9. **anban** gRule1(L) returns true only if L is of the form anban for n >=0

Test each of your predicates on good and bad lists. For example listOfabc([a,a,b,c,c,c]) should return true while listOfabc([a,a,c,c,c]) should return false.

1. Write a Prolog program to solve the 6 by 6 Sudoku puzzle below. Do not use a solution downloaded from the Internet or elsewhere. Write your own. It should be designed along these lines: Label the squares X1, X2, …, X36 as discussed in class. The X1,…, X36 represent the 36 squares in the Sudoku puzzle (row by row, left to right). Set the known values of the X’s and then generate needed permutations and test for compliance with problem constraints. Your query should be

sudoku (X1,X2,X3,X4,X5,X6,X7,X8,X9,X,X11,X12,X13,X14, X15,X16,X17,X18,X19,X20,X21,X22, X23,X24,X25,X26,X27,X28,X29,X30,X31,X32,X33,X34,X35,X36).

The result will be values for X1, … , X36 that solve the problem. Your program should run in less than 2 minutes. If it takes a “long time” try to speed it by checking more constraints earlier in the program. You should be able to get your program to run is less than a minute.

1. Once Problem 2 is working, create another predicate called sudokuSolution as follows

sudokuSolution(L) :-

L =[X1,X2,X3,X4,X5,X6,X7,X8,X9,X10,X11,X12,X13,X14,X15,X16,X17,X18,X19,X20,X21,X22,X23,X24,X25,X26,X27,

X28,X29,X30,X31,X32,X33,X34,X35,X36],

sudoku(X1,X2,X3,X4,X5,X6,X7,X8,X9,X10,X11,X12,X13,X14,X15,X16,X17,X18,X19,X20,X21,X22,X23,X24,

X25,X26,X27,X28,X29,X30,X31,X32,X33,X34,X35,X36).

In the interpreter use

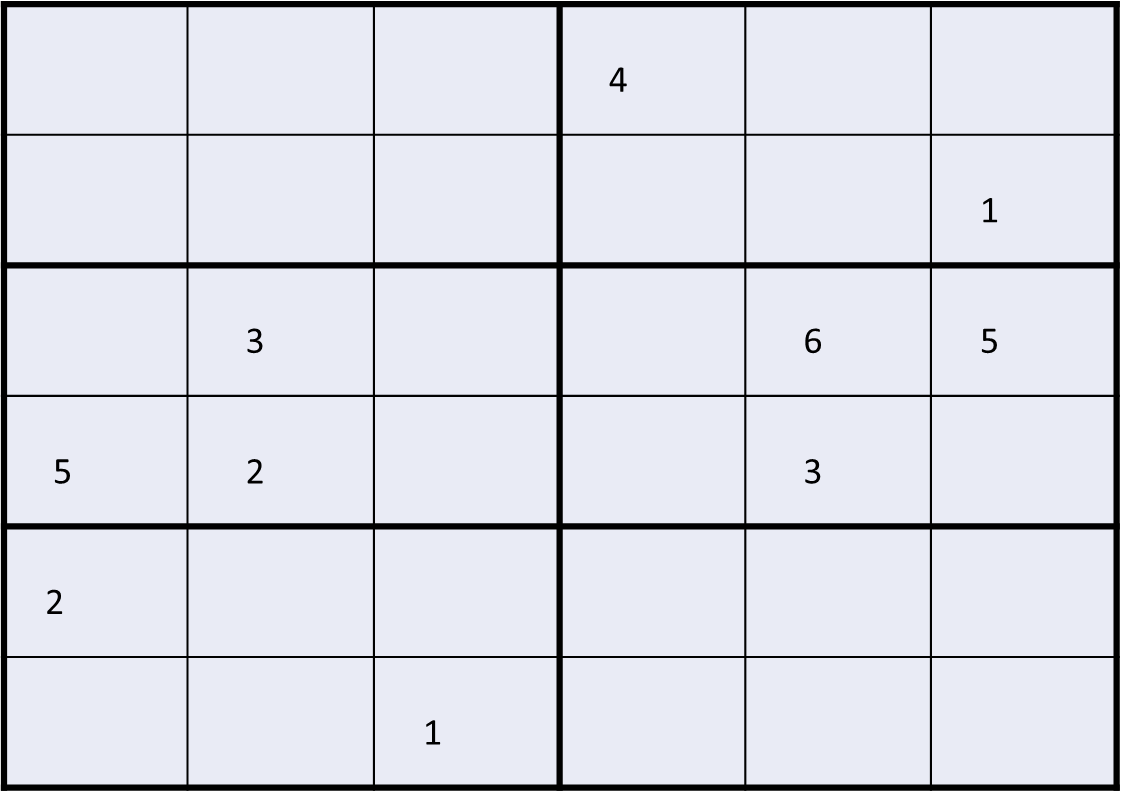
?-sudokuSolution(L); true.

If the L is only partially printed when the interpreter comes back, type w <enter> to get all of L printed.

Sudoku Puzzle

Complete the grid so each row and column and 2-by-3 box contains every digit from 1 to 6.

To solve problem using Prolog: Label the grid squares X1,X2,X3,X4,… X36 row by row, FROM TOP TO BOTTON AND LEFT TO RIGHT. For example the value 4 is in X4 and the value 6 is in X17.



**Testcases for Sudoku Problem.**

1. Solve given sudoku problem . Test using

?- sudoku (X1,X2,X3,X4,X5,X6,X7,X8,X9,X,X11,X12,X13,X14, X15,X16,X17,X18,X19,X20,X21,X22, X23,X24,X25,X26,X27,X28,X29,X30,X31,X32,X33,X34,X35,X36).

How many seconds or minutes did Prolog take to find the solution?

1. Solve given sudoku problem . Test using ?-sudokuSolution(L);true. The answers should be in the list L.

How many seconds or minutes did Prolog take to find the solution?

**Electronic Copy Turn in: (Due 5 pm May 11)**

Upload a single source file called project5.pl with

* source code to solve problems 1 -3
* test results for testcases for the sudoku problem. Copy and paste the interpreter window that shows your test runs and results. Comment out using /\* \*/ notation.
* Also include in source code a diagram of the 6 -by-6 array that shows the numeric solution to the given sudoku problem. Comment out.
* Your source file should contain as a comment your name, course, date and Project #.