

POPL2 Assignment

Logic programming

Prolog Assignments

The overall objective of this assignment is for you to gain some hands-on experience with problem solving using Prolog, using simple facts and rules, recursion, and database handling capabilities of the language.

Prolog programming with swi-prolog (<https://www.swi-prolog.org/features.html>). All implementations should be based on swi-prolog

SWI-Prolog is a [free](#) implementation of the [programming language Prolog](#), commonly used for teaching and [semantic web](#) applications.

You are expected to write your own code on the assignment. You may engage in general discussions with your classmates regarding the assignment, but specific details of a solution, including the solution itself, must always be your own work. Incidents which violate this will be taken seriously.

Code for all programming assignments should be **well documented**. A working program with no comments will **receive only partial credit**. Documentation entails writing a description of each predicate as well as comments throughout the code to explain the program logic. Provide a README file with each program explaining how to run the program, input details, expected output etc. Submit a zip file containing all the programs.

Assignment 1 : Map coloring

- **Map Coloring Problem** It's known that only 4 colors are needed to paint any map so that no two neighboring states have the same color.
- Write a Prolog program that receives a map and a list of 4 colors and produces a colored map.
- The map is represented by a list of states, each of which is a state name and a list of neighboring states.
- For example, part of india map is represented as follows

```
map('india', [KL: [TN,KA], TN: [KL, KA,AP], KA:[KL,TN, AP, ], ....]).
```

Assignment 2 : Language processing

- Write a program that understands simple English sentences having the following forms:
- ___ is a ___ . (e.g. George is a man)
- A ___ is a ___ . (e.g. A man is a person)
- Is ___ a ___ ? (e.g. Is George a person)
- The program should be able to recognize the type of sentence. First two types of sentences should be processed and add the fact to the knowledge base. Third type of sentence should be used to query the knowledge base.

Assignment 3 : Eight Queens problem

- The problem here is to place eight queens on the empty chessboard in such a way that no queen attacks any other queen.
- Represent the position by a list of eight items, each of them corresponding to one queen : $[1/4, 2/2, 3/7, 4/3, 5/6, 6/8, 7/5, 8/1]$.
- Find all solutions

$S = [1/4, 2/2, 3/7, 4/3, 5/6, 6/8, 7/5, 8/1];$

$S = [1/5, 2/2, 3/4, 4/7, 5/3, 6/8, 7/6, 8/1];$

$S = [1/3, 2/5, 3/2, 4/8, 5/6, 6/4, 7/7, 8/1];$

...

- Provide prolog programs based on
 - Depth first search strategy
 - Breadth first search strategy

