Constructive Logic (15-317), Fall 2016
Assignment 7: Practicing Prolog Programming

Contact: Giselle Reis (`giselle@cmu.edu`)

Due Tuesday, November 1, 2016, 1:30pm

This assignment is due at the beginning of class on the above date and must be submitted electronically via autolab. Submit your homework as a **tar** archive containing the files:

- `g4ip.pl`
- `houses.pl`
- `coloring.pl`
- `ternary.kyt`.

After submitting via autolab, please check the submission's contents to ensure it contains what you expect. No points can be given to a submission that isn't there.

1 Implementing a theorem prover (one more time)

Since you are now experts in implementing **G4ip**, it is time to try that using a different language.

Task 1 (15pts)**.** Implement a theorem prover in **G4ip** in Prolog. You must define predicates `prove(A)` and `refute(A)` for proving and refuting a formula, and use the predefined logical operators (see accompanying `g4ip.pl` file). This means that, given a valid *ground* formula `a`, the query `prove(a)` should succeed (with *true* or *yes*). Analogously, `refute(b)` should succeed for an invalid formula `b`. HINT: you can use negation as failure to implement `refute`.

2 Neighborhood

In an arbitrary street in the world, there are five consecutive houses of different colors. Each house is inhabited by a person of different nationality (British, Swedish, Danish, Norwegian, German), drinking a different drink (tea, coffee, beer, water, milk), having a different pet (dog, bird, cat horse, fish) and practicing a different sport (football, running, swimming, cycling, rugby). You know that:

- The British lives in the red house.
- The Swedish has a dog for a pet.
- The Danish drinks tea.
- The green house is on the left of the white house.
- The person living in the green house drinks coffee.
- The person who cycles has a bird.
- The person living in the yellow house swims.
- The person in the middle house drinks milk.
- The Norwegian lives in the first house.
- The football player lives next to the one that has cats.
- The person that has a horse lives next to the one that swims.
- The person who runs drinks beer.
- The German plays rugby.
- The Norwegian lives next to the blue house.
- The football player lives next to the one that drinks water.
- Someone has a fish.

Task 2 (10pts)**.** Implement a program in Prolog that finds the solution to this puzzle. You should define a predicate `houses/1` with one parameter for the solution of the puzzle (i.e. the query `houses(S)` binds `S` to the solution). Submit a file named `houses.pl` for this task.

3 Coloring maps

An interesting problem in graph theory is called graph coloring. Given a graph, a coloring for this graph consists of coloring each vertex such that no two adjacent vertices have the same color, and the minimum number of colors should be used. The four-color theorem states that any planar graph[1] can be colored using at most four colors. The theorem was proved in 1976 using a computer program, and has caused much controversy (is a computer proof really a proof?). Since then, it was simplified and proved (again using computers) at least twice (in 1997 and 2005).

As a consequence of this theorem, any map can be colored with at most four colors such that no adjacent regions have the same color. This is because every map can be represented by a planar graph. Take Australia's map in Figure 1.

Its corresponding planar graph consists of one vertex for each region and an edge between two regions if they are adjacent. The color of the vertex in a

[1]A graph that can be drawn on 2D with no crossing edges.

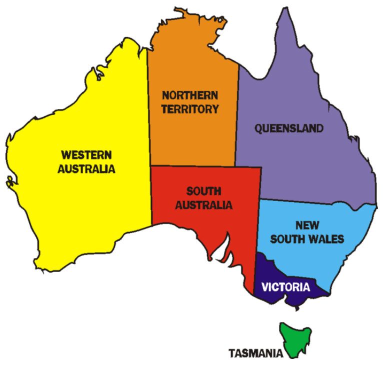

Figure 1: Australia (more colorful than necessary)

graph coloring will be the color of the region in the map (observe that this map is using more colors than necessary, although this might make it more visually appealing).

Task 3 (10pts)**.** In this question, you will need to implement graph coloring. Your job is to write a predicate `color_graph(nodes, edges, colors)` that associates with the graph (*nodes*, *edges*) all of the valid 4-colorings of the graph. We will only test your program with planar graphs.

Submit a file named `coloring.pl` for this task.

1. You should define a `color/1` predicate with four colors.
2. Assume there are predicates `node/1` and `edge/2`. You will need to come up with some of these for testing purposes and you are encouraged to share your tests with the rest of the class on Piazza.
3. In `color_graph/3`, the first parameter is a list of `node/1` terms and the second parameter is a list of `edge/2` terms.
4. Implement `color_graph(nodes, edges, colors)`, where `colors` is a list of pairs `(a,c)`, where `a` is a node and `c` is a color.

4 More quantifiers in KeYmaera

Task 4 (5pts)**.** Prove the following theorem (in file `ternary.kyx`) in KeYmaeraI and submit your solution in a file named `ternary.kyt`.

$$(\forall x.\forall y.\exists a.(p(a) \supset p(x) \wedge p(y)))$$
$$\supset (\forall u \forall v \forall w \exists b (p(b) \supset p(u) \wedge p(v) \wedge p(w)))$$