Krystle Bulalakaw

301134933

CMPT310   
artificial intelligence survey

Assignment 2 Report

# Outline of Procedure

This Java program solves instances of map colouring problems.

The program is made of the following components:

* Assignment.java – Holds a list of variables and their assignments, and checks if those assignments satisfies some constraints
* BacktrackSearch.java – A specific solution strategy using Recursive Backtracking Search
* Constraint.java – An interface for limitations over assignments for variables.
  + NotEqualConstraint.java – An implementation of the above, specifically for the constraint that the value of variable 1 must not be equal to the value of variable 2.
* CSP.java – An abstract representation of CSPs with helper methods such as adding variables, adding constraints, getting domain, etc.
  + MapColouringCSP.java – An implementation of the above, specifically using the strategy of Backtracking Search to solve an instance of a map colouring problem.
* Domain.java – Represents a domain of values for which a variable can take on.
* Variable.java – Represents some object which can be assigned a value in a nonempty domain.

First, the program reads input data passed from command line arguments in order to initialize parameters for the problem to be solved. It assumes the data is given in such a format that was used in “Asst2.data.txt”. The vertices will be initialized along with their adjacencies. The domain and constraints will also be set. The parameters (num. vertices, num. colours, and domain are printed). Then the problem-solving attempt will begin.

As a generalized algorithm, the backtracking search will pick some starting point (unassigned variable). While the problem is still unsolved, it will iterate through each value in the domain values, make an assignment of that value to the variable.

If this assignment satisfies the inequality constraint, it will make a recursive call for the rest of the problem. If the result is not null, it will return the assignment. Else, it will undo the current assignment and return null.

Once out of the loop, if there has been no assignment, the search failed and will report in a print statement. Else, the solution is printed in the format “vertex = assigned colour” for each vertex.

I believe this algorithm will work as long as a solution exists. It closely follows the pseudocode outlined in the course textbook, with some extra abstraction and object oriented-design to add clarity to my solution. I also tested by hand to check that the output solution was correct, and also if no solution could be found.