Aaron Zhang

730084254

Comp 560

19 February, 2020

A1: Constraint Satisfaction Report

My assignment consists of five files: BackTracker.py, BackTrackerMain.py, LocalSearch.py, LocalSearchMain.py, and State.py.

The two main files (BackTrackerMain.py & LocalSearchMain.py) are used to read the input files and to create a new "search object" to color the states. I found it easier and cleaner to implement the two by using two separate files. Sorry for the hassle.

The two files (BackTracker.py & LocalSearch.py) run the search. They take in the read input from their corresponding "mains" and run the algorithm. An output is printed directly to the terminal.

My State.py file simply contains the State Object code.

For my backtracker, I used the recursive method. Each call picks a new uncolored state, selects and verifies a color selection, and continues down the path. If the path were to be backtracked, then the data would be updated as needed. Additionally, I implemented forward checking by pruning adjacent states to help with efficiency. I also added a final check to test my output for accuracy. If it runs into errors, the program will print an error message.

For my local search, I implemented a simple hill climbing algorithm. I started by assigning every state a random color. Then I picked a random start location and traversed the map until all states fell in line with the requirements. I used a while loop to maintain the traversal. Every loop checks for completion. The program stops if the run time exceeds one minute. Otherwise it prints an output to the terminal when a solution has been achieved. Thus, solution accuracy has already been implemented within the inherent structure of the search.

Lastly, I worked individually on this project, so all work that has been submitted was done by only me.