Stephen Chambers
smx227
September 30, 2015

Assignment 4 Writeup

1. What is the size of the state space for this problem?

n^m

where:
n = number of vertices
m = number of colors

2. Describe any implmentation choices you made that you felt were important. Mention anything else we should know when evaluating your program.

I kept track of my domains with a reference counter array. If I removed a color from the domain, I would increase the counter corresponding to that color. When backtracking, I would decrease the counter corresponding to that color.

If I got a node 'for free', i.e. the domain was one color, I did not count it as a branching node. I also counted before validating whether or not I could use that color.

All of my algorithms were recursive.

3. What's the average speedup you get for fc over dfs? For mcv over fc?
I will be using the queen_8_12 example with 12 colors to determine the speedup.

Algorithm	Nodes Expanded	Percent Speedup
DFS	15798874	
FC	721044	~95.44%
MCV	233	~99.99%

4. What suggesitons do you have for improving this assignment in the future?
Be more precise on what exactly a branched node is. Professor Ruml in class said to not count nodes that you got "for free", but the reference solution counts those nodes.

Small transcript:

-bash-4.3$ ./color-validator ./run.sh **dfs** 12 < queen8_12.col.txt
Executing planner...
Picked up JAVA_TOOL_OPTIONS: -Xmx256m
Execution time: 0.726861953735 seconds
Parsing plan...
15798874 branching nodes explored.

Validating plan...
Valid coloring!

-bash-4.3$ ./color-validator ./run.sh **fc** 12 < queen8_12.col.txt
Executing planner...
Picked up JAVA_TOOL_OPTIONS: -Xmx256m
Execution time: 1.46826004982 seconds
Parsing plan...
721251 branching nodes explored.

Validating plan...
Valid coloring!

-bash-4.3$ ./color-validator ./run.sh **mcv** 12 < queen8_12.col.txt
Executing planner...
Picked up JAVA_TOOL_OPTIONS: -Xmx256m
Execution time: 0.432862043381 seconds
Parsing plan...
233 branching nodes explored.

Validating plan...
Valid coloring!