Stephen Chambers
smx227
October 8th, 2015

Assignment 5 Writeup

1. What is the state space for this problem?

2^n

n = number of variables

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

My WALK-SAT implementation can solve *all* of the test files provided quickly (including aim50.cnf). However, it is slower than the DLL algorithm for large scale problems where the number of variables is far less than the number of clauses. My conjecture is that I used an array list of array lists to keep track of the formula, and iterated through this structure continuously when finding all unsatisfied clauses, positive flips, flip values, etc. Therefore, I don't think my implmentation of WALK-SAT is scalable. Theoretically, the DLL algorithm should be slower than WALK-SAT, since it has to try every possible combination of variables. I am including this information because you mentioned in class that the purpose of the writeup is to make sure I understand what the code is doing. This is me trying to show you that I recognize that WALK-SAT should be faster, and this is my best guess to why it is not.

3. When there are around 3.3 clauses per variable, how does the larget formula DLL can solve compare to the largest for your WALK-SAT?

Algorithm	Hardest 3.3 Problem
DLL	273 variables, 901 clauses
WALK-SAT	250 variables, 825 clauses

4. Repeat the process for 4.4 clauses per variable. Explain your findings.

Algorithm	Hardest 3.3 Problem
DLL	156 variables, 670 clauses
WALK-SAT	75 variables, 344 clauses

As I said above, I believe my WALK-SAT implmentation is slower due to the amount of times I iterate over my formula. Theoretically, WALK-SAT should be much faster, but with the caveat that it may not ever find a solution. Additionally, if DLL returns unsatisfiable, you *know* that formula can never be satisfied. With WALK-SAT, that is not the case.

5. What suggestions do you have for improving this assignment in the future?

I have no suggestions for this assignment.

DLL Algorithm Transcript:

```
-bash-4.3$ ./satvalidator ./run.sh < tiny.cnf
Executing solver...
Picked up JAVA_TOOL_OPTIONS: -Xmx256m
Execution time: 0.14426088333129883 seconds
Retrieving assignments...
Parsing assignments...
2 branching nodes explored.

Satisfiable!
Validating assignments...
Valid assignment!
-bash-4.3$ ./satvalidator ./run.sh < unit.cnf
Executing solver...
Picked up JAVA_TOOL_OPTIONS: -Xmx256m
Execution time: 0.14534258842468262 seconds
Retrieving assignments...
Parsing assignments...
4 branching nodes explored.

Satisfiable!
Validating assignments...
Valid assignment!
-bash-4.3$ ./satvalidator ./run.sh < small.cnf
Executing solver...
Picked up JAVA_TOOL_OPTIONS: -Xmx256m
Execution time: 0.1273646354675293 seconds
Retrieving assignments...
Parsing assignments...
10 branching nodes explored.

Not satisfiable!
-bash-4.3$ ./satvalidator ./run.sh < quinn.cnf
Executing solver...
Picked up JAVA_TOOL_OPTIONS: -Xmx256m
Execution time: 0.14046335220336914 seconds
Retrieving assignments...
Parsing assignments...
16 branching nodes explored.

Satisfiable!
Validating assignments...
Valid assignment!
-bash-4.3$ ./satvalidator ./run.sh < aim50.cnf
Executing solver...
Picked up JAVA_TOOL_OPTIONS: -Xmx256m
Execution time: 0.15361261367797852 seconds
Retrieving assignments...
Parsing assignments...
104 branching nodes explored.

Satisfiable!
Validating assignments...
Valid assignment!
```

WALK-SAT Algorithm Transcript:

```
-bash-4.3$ ./satvalidator ./run.sh -w < tiny.cnf
Executing solver...
Picked up JAVA_TOOL_OPTIONS: -Xmx256m
Execution time: 0.13983821868896484 seconds
Retrieving assignments...
Parsing assignments...
0 total tries.

0 total flips.

Satisfiable!
Validating assignments...
Valid assignment!
-bash-4.3$ ./satvalidator ./run.sh -w < unit.cnf
Executing solver...
Picked up JAVA_TOOL_OPTIONS: -Xmx256m
Execution time: 0.14858293533325195 seconds
Retrieving assignments...
Parsing assignments...
0 total tries.

4 total flips.

Satisfiable!
Validating assignments...
Valid assignment!
-bash-4.3$ ./satvalidator ./run.sh -w < small.cnf
Executing solver...
Picked up JAVA_TOOL_OPTIONS: -Xmx256m
Execution time: 30.165209770202637 seconds
Retrieving assignments...
Parsing assignments...
6593 total tries.

65930000 total flips.

No solution found! (timeout)
-bash-4.3$ ./satvalidator ./run.sh -w < quinn.cnf
Executing solver...
Picked up JAVA_TOOL_OPTIONS: -Xmx256m
Execution time: 0.14139509201049805 seconds
Retrieving assignments...
Parsing assignments...
0 total tries.

20 total flips.

Satisfiable!
Validating assignments...
Valid assignment!
```

```
-bash-4.3$ ./satvalidator ./run.sh -w < aim50.cnf
Executing solver...
Picked up JAVA_TOOL_OPTIONS: -Xmx256m
Execution time: 0.3236422538757324 seconds
Retrieving assignments...
Parsing assignments...
4 total tries.

44125 total flips.

Satisfiable!
Validating assignments...
Valid assignment!
```