CSC 322 (Winter/Spring 2019) – Project One

Task Four Report

Michael Windels – V00854091
Chengyang He – V00791092
Mathieu Trottier – V00872174

Testing Results

Task four involves comparing the execution time of a brute force validity checker with the execution time of a minisat-based validity checker (essentially `vcheck1`). The code which compares the execution times can be found in `vcheck4.py`, and can be run with `make vcheck4`. In this report, three formulae were used to compare the validity checkers under different circumstances.

The first formula was (A1 v ~A1), which has few variables and few connectives. The second formula was (A1 v ~A1) & (A1 v ~A1) & … & (A1 v ~A1) with twenty total conjoined clauses, which has few variables but many connectives. The third and final formula was (A1 v ~A1) & (A2 v ~A2) & … & (A20 v ~A20), which has many variables and many connectives.

Each of the three formulae are valid. Validity is important if we want to find the worst-case running time of the brute force checker. If a formula is not valid, the brute force checker may terminate prematurely, which could skew timing results.

The following results were obtained by running `vcheck4` on the three formulae (and averaging the running times over five executions per formula). All of the times are in milliseconds or seconds.

	Brute Force Validity Checker	***Minisat-Based Validity Checker***
(A1 v ~A1)	*0.1 ms*	*15.4 ms*
(A1 v ~A1) & … & (A1 v ~A1)	*0.4 ms*	*22.7 ms*
(A1 v ~A1) & … & (A20 v ~A20)	*67.0 s*	*28.8 ms*

Evidently, when the number of variables in the formula is low, the brute force validity checker outperforms the minisat-based checker. Even when the number of connectives increases, the brute force checker still outperforms the minisat-based checker. However, when the number of variables increases (even just to twenty), the brute force checker takes a massive performance hit. This is because the brute-force checker must check that all two to the twenty possible permutations of the variables satisfy the formula.

Execution time does not directly correlate with performance. However, since our later timing results show differences in orders of magnitude, we can conclude that the minisat-based validity checker easily outperforms the brute force validity checker as the number of variables in the formula increases.