Institut für Informatik
der Universität München
Dr. J. Johannsen
Leah Neukirchen

WS 2018/2019

November 15, 2018

Exercises on **SAT Solving**

Problem set 2

Practical Exercise 1: Your own DPLL-Solver.

Your task is to implement a SAT solver based on the DPLL-Algorithm.

You can use any general-purpose programming language for this task. You may hand in your solution in groups of one to three people. Everyone in the group is expected to be able to explain all the code! Larger groups are expected to do more bonus exercises.

This exercise consists of four parts. The first part is neccessary to pass the problem sheet. The other parts will get you a better grade.

1. Compulsory part: Implement a program that can read a SAT instance as DIMACS CNF format (You'll need to support `p cnf`-lines and ignore `c`-lines), solve it using the DPLL approach (with Unit Propagation), and output the solution in DIMACS format (generate `s`- and `v`-lines).

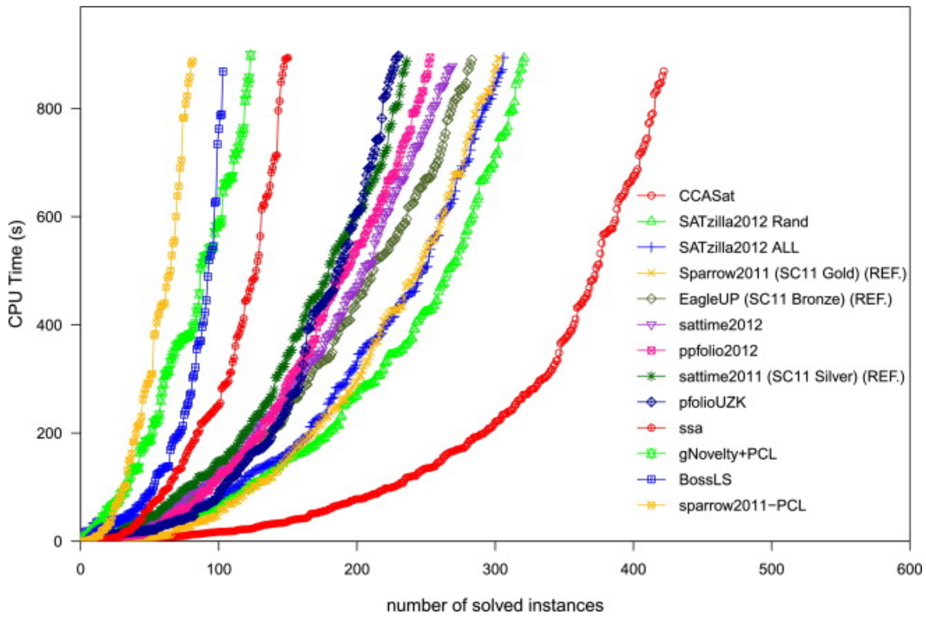
   We include a set of basic tests (in `test/`) that this solver needs to pass correctly.

   Benchmark your solver using the included SATlib benchmarks. Given one minute of CPU time for each problem, how many problems is your solver able to solve? (There are some hard problems included, you are not supposed to solve all of them in a short time.)

2. Bonus part: Implement Pure Literal Elimination.

3. Bonus part: Generate a “cactus plot” of your solver for the given SATlib benchmarks: First, measure the run time of the solver for each problem (use a sensible timeout). Sort these numbers. Then plot the number of solved instances on the x-axis, and the running total of the CPU time on the y-axis.

   Example from the SAT 2012 competition:

4. Bonus part: Implement various selection heuristics (such as DLIS, DLCS, MOM, Jeroslow-Wang) and compare them (e.g. by overlaying cactus plots).

Hand in as groups of one to three people on UniWorX until Thursday, December 20, 2018, 2pm.