



Sebastian Hack
Christoph Weidenbach

January 13, 2009

Tutorials for “Advanced C”
Exercise sheet 10

Exercise 10.1: (5 P)

Implement a timing module with the interface of generating clocks, starting clocks, stopping clocks, and getting clock values (in `<seconds>.<hundredth seconds>`). You are allowed to use further libraries but try to stay as portable with your code as possible.

Integrate the module into your SAT solver and use clocks for spent time for input, unit propagation, and overall. Add to your output before the report of the result the lines

```
SAT Timing
Input:      <seconds>.<hundredth seconds>
Unit Prop:  <seconds>.<hundredth seconds>
Overall:    <seconds>.<hundredth seconds>
```

Exercise 10.2: (10 P)

Implement the rule Merging Replacement Resolution from the lecture and apply it to the input clause set of your SAT algorithm. Just apply it to the input clause set, not during search. Add a command line option to turn the rule off/on and another one that documents the rule, i.e., the clauses it is successfully applied to and the result. Calling your program without arguments should reveal the option names.

Exercise 10.3: (0 + 10 P)

Profile your SAT algorithm, the most recent version you submitted before January 13, 2009 and reimplement crucial parts such that it becomes at least two times faster on the known input examples.

Submit your solution of Exercise 10.1 until the lecture on January, 20 and the solutions to Exercise 10.2 and Exercise 10.3 until the lecture on January, 27.

Note: Joint solutions are not permitted.