



Sebastian Hack
Christoph Weidenbach

November 18, 2008

Tutorials for “Advanced C”
Exercise sheet 5

Exercise 5.1: (5 P)

Finish the SAT algorithm from exercise sheet 4, i.e., in sum you receive now 15+2 points. Out of the 15 points we assign 7 points for a correct working program (tested on small examples) and 8 points for following the lecture code development discipline: documentation, format of the solution, makefile tasks, modularization etc.

Exercise 5.2: (5 P)

Implement a normalization algorithm following the marking approach of the lecture. In case the input files contains “holes”, i.e., unused literal identifiers, then the normalization algorithm produces a cnf without holes starting with 1 as the first identifier. For example, the clauses

$$\begin{array}{l} 4 -8 3 0 \\ 7 -3 2 0 \end{array} \text{ can be transformed into the equisatisfiable clause set } \begin{array}{l} 1 -2 3 0 \\ 4 -3 5 0 \end{array}$$

Your program should be named `cnf2cnf` and accept a cnf input file and output file. It reads the input file, normalizes the clauses and writes the clauses to the output file in DIMACS cnf format. No need to transfer comment lines. In sum:

`cnf2cnf <input cnf file> <output cnf file >`

Submit your solution until the lecture on November 25 using the SVN repository. Please recall the guidelines from the lectures when preparing the solution.