SAARLAND UNIVERSITY
COMPUTER SCIENCE

Sebastian Hack
Christoph Weidenbach

November 11, 2008

Tutorials for "Advanced C"
Exercise sheet 4

Exercise 4.1: *(10+2 P)*
Implement the SAT algorithm introduced in the first lecture on top of the parser. Stick to the programming conventions introduced in the lecture.

The program shall output "unsatisfiable" if the clause set is unsatisfiable and "satisfiable" if the clause set is satisfiable. In case of a satisfiable clause set the program shall also output the satisfying assignement, i.e., the literals satisfying the cnf (blank separated list of integers according to the input). For example, if the input contains exactly the two clauses "2 3 0" and "-1 3 0" a correct output could be "satisfiable 3 -1".

You receive 2 Bonus points for making your parser foolproof against all kinds of input files. The expected behaviour of the parser is to finish with an error message if the input file does not match the input format given in the lecture.

Submit your solution until the lecture on November 18 using the SVN repository. Please recall:

- create a subdirectory for each programming exercise sheet with name ex<sheetnumbertwodigits>, e.g., ex02
- in the ex<sheetnumbertwodigits> directory create for each prgramming exercise the respective exercise directory with name exercise<numberonedigit>, e.g., exercise1
- your solution should be in the exercise<numberonedigit> directory as a gziped tar archive file exercise<numberonedigit>.tgz
- the tutors will checkout your SVN directory Tuesday after the lecture and pick the tgz archive. They will unpack the archive in a fresh directory and then run "make". This should produce the binary "SAT" that is then run on test examples.
- keep your space tidy

Note: Joint solutions are not permitted.