Technische Universität München
Institut für Informatik
Prof. Dr. Helmut Veith
Dipl.-Inf. Stefan Kugele

SS 2007
Übungsblatt 3
Mittwoch, 23. Mai 2007

Übung zur Vorlesung Model Checking

This exercise focuses on Boolean Satisfiability (SAT), a classical search problem in computer science. Stephen A. Cook and Leonid Levin showed, that this problem is **NP**-complete, i.e., great many differently structured and at the same time practically relevant search problems can be translated to SAT in a strictly formal sense by so-called *reductions*. Bounded Model Checking (BMC) is just one domain in which SAT-based techniques could be applied very successfully.

SAT-Solving

Use `minisat` as a state-of-the-art SAT solver to solve the following exercises. `minisat` can be obtained from:

`http://www.cs.chalmers.se/Cs/Research/FormalMethods/MiniSat/MiniSat.html`

Any other SAT-solver which is able to read and write DIMACS compatible input and output format can be used as well.

(a) Repeat the DIMACS format for CNF encoding.

(b) Repeat the Tseitin transformation.

(c) Translate the formula into a CNF formula in DIMACS format: $(a \vee b) \Rightarrow (c \wedge d)$

(d) Sudoku

	6	8				2		
			5			7		
	2							
				2	6			
1							4	
5								
4			3					5
	7			8				
							1	

Figure 1: Sudoku puzzle.

A SUDOKU puzzle is represented by a 9x9 grid, which comprises nine 3x3 sub-grids. Some of the entries in the grid are filled with numbers from 1 to 9, whereas other entries are left blank. Such a puzzle is solved by assigning numbers from 1 to 9 to the blank entries such that every row, every column, and every 3x3 sub-grid contains each of the nine possible numbers. Figure 1 shows an partially filled grid.

i) Develop a suitable reduction from SUDOKU to SAT.

ii) Implement an application, script, etc. in a language of your choice which encodes the given puzzle into a CNF formula in DIMACS format.

iii) Use `minisat` to solve the puzzle.