Exercises for the Lecture
Practical SAT Solving

Summer Semester 2016

Assignment 4

Institute for Theoretical Informatics, Karlsruhe Institute of Technology (KIT) 16.06.2016

Tomas Balyo, Carsten Sinz Due 30.06.2016

Exercise 1 (Variable Elimination) [3] In Lecture 8, we presented a variable elimination procedure based on gates, where, for a CNF-representation of a gate G in a formula F with $F = G \mathbin{\dot\cup} R$, we replaced $S = (G_x \cup R_x) \cup (G_{\overline{x}} \cup R_{\overline{x}})$ by $S' = (G_x \otimes R_{\overline{x}}) \cup (R_x \otimes G_{\overline{x}})$, dropping both $R_x \otimes R_{\overline{x}}$ and $G_x \otimes G_{\overline{x}}$ from S'.

Show that the clauses in $R_x \otimes R_{\overline{x}}$ can be derived by resolution from S', and thus do not have to be included in S'. You may assume that G is a binary AND-gate.

Exercise 2 (Variable Elimination) [4] Given

$$S = \{\underbrace{\{\neg x, \neg y, a\}, \{x, \neg a\}, \{y, \neg a\}}_{G_1}, \underbrace{\{\neg a, r\}, \{\neg z, r\}, \{a, z, \neg r\}}_{G_2}, \{a, z, r\}, \{\neg a, \neg r\}\} \ .$$

Apply elimination by clause distribution for gates (as presented in the lecture) to S.

1. Eliminate gate G_1 (variable a) first, then gate G_2 (variable r), if possible.

2. Eliminate gate G_2 (variable r) first, then gate G_1 (variable a), if possible.

Give the clause sets after each elimination step.

Exercise 3 (Pigeon Hole Problem, Extended Resolution) [15 points] Read the paper *A short proof of the pigeon hole principle using extended resolution* by Stephen Cook (available in the university network under `http://dx.doi.org/10.1145/1008335.1008338`).

On page 2 of this paper, it is claimed that the clause set S_{n-1}, containing clauses

$$\begin{aligned} Q_{i,1} \vee \cdots \vee Q_{i,n-2}, &\quad 1 \leq i \leq n-1 \\ \neg Q_{i,k} \vee \neg Q_{j,k}, &\quad 1 \leq i < j \leq n-1, 1 \leq k \leq n-2 \end{aligned}$$

can be derived by resolution from S_n and $(1)-(4)$. (See the paper for definitions of S_n and $(1)-(4)$.)

Give explicit resolution derivations for the clauses in S_{n-1}.

Exercise 4 (Preprocessing Challenge) [8+(10) points] Implement a formula preprocessor that does failed literal probing and blocked clause elimination (until fixpoint, i.e., no more clauses can be eliminated and no more unit clauses can be derived by literal probing. Given a CNF formula F in the DIMACS format (`http://www.satcompetition.org/2004/format-solvers2004.html`) output F after preprocessing also in the DIMACS format. For a working preprocessor you get 8 points. The author of the fastest implementation receives a bonus of 10 points. For testing you can use the test instances from the Unit Proagation Challenge.