Logical reasoning and programming, lab session 3

(October 9, 2023)

3.1 Decide the satisfiability of

$$\{\{\overline{p}, r, s, t\}, \{\overline{r}, s, t\}, \{\overline{p}, r, \overline{s}\}, \{p, q\}, \{p, \overline{q}\}, \{\overline{p}, \overline{t}\}, \{\overline{r}, \overline{s}, t\}\}$$

by DPLL. Use the order of branching: p, q, r, $\dots$.

3.2 Given the formula $\{\{p_1, p_2\}, \{\overline{p_1}, p_3\}, \{p_2, \overline{p_3}\}, \{\overline{p_2}, \overline{p_4}\}, \{\overline{p_3}, p_4\}\}$, what clause will a CDCL solver learn first if it begins by deciding that p_1 is true?

3.3 Assume that our version of CDCL fails to produce a satisfiable valuation for φ and hence φ is unsatisfiable. How would you produce a resolution proof of this fact from the run of CDCL?

3.4 Decide the satisfiability of

$$\begin{aligned}\{\{p_1, p_{13}\}, \{\overline{p_1}, \overline{p_2}, p_{14}\}, \{p_3, p_{15}\}, \{p_4, p_{16}\},\\ \{\overline{p_3}, \overline{p_5}, p_6\}, \{\overline{p_5}, \overline{p_7}\}, \{\overline{p_6}, p_7, p_8\}, \{\overline{p_4}, \overline{p_8}, \overline{p_9}\}, \{\overline{p_1}, p_9, \overline{p_{10}}\},\\ \{p_9, p_{11}, \overline{p_{14}}\}, \{p_{10}, \overline{p_{11}}, p_{12}\}, \{\overline{p_2}, \overline{p_{11}}, \overline{p_{12}}\}\}\end{aligned}$$

by CDCL using the first UIP. Do not use pure literal elimination, but check what happens if you do. Use the order of branching: p_1, p_2, p_3, $\dots$.

3.5 How many symmetries does your formulation of PHP_n^{n+1} have?

3.6 We can define the lexicographic order on two bit vectors $x_1, \dots, x_n$ and $y_1, \dots, y_n$, denoted $x_1 \dots x_n \leq_{lex} y_1 \dots y_n$, as follows

$$\bigwedge_{i=1}^{n} ((\overline{x_i} \vee y_i \vee \overline{a_{i-1}}) \wedge (\overline{x_i} \vee a_i \vee \overline{a_{i-1}}) \wedge (y_i \vee a_i \vee \overline{a_{i-1}})),$$

where $\overline{a_0}$ is always false, using new auxiliary variables $a_0, a_1, \dots, a_{n-1}, a_n$.

(a) What is the purpose of auxiliary variables?

Hint: When is it necessary to satisfy $x_i \leq y_i$?

(b) Why is $\overline{a_0}$ always false and hence useless?

(c) Why can we replace $(\overline{x_n} \vee y_n \vee \overline{a_{n-1}}) \wedge (\overline{x_n} \vee a_n \vee \overline{a_{n-1}}) \wedge (y_n \vee a_n \vee \overline{a_{n-1}})$ just by $(\overline{x_n} \vee y_n \vee \overline{a_{n-1}})$? Hence we need only $3n-2$ clauses and $n-1$ auxiliary variables (a_n is also useless).

(d) How does the meaning of the formula change if you replace $(\overline{x_n} \vee y_n \vee \overline{a_{n-1}})$ by $(\overline{x_n} \vee \overline{a_{n-1}}) \wedge (y_n \vee \overline{a_{n-1}})$?

3.7 How can we exploit the lexicographic order to decrease the number of symmetries in PHP_n^{n+1}?

Hint: Order hole-occupancy or pigeon-occupancy vectors.

3.8 A very nice symmetry breaker for PHP_n^{n+1} is based on columnwise symmetry, namely we can add the following clauses

$$p_{i(i+1)} \vee \overline{p_{ij}}$$

for $1 \leq i < j \leq n$, where p_{kl} means that pigeon k is in hole l, for $1 \leq k \leq n+1$ and $1 \leq l \leq n$. Why?

3.9 Try PicoSAT/pycosat and PySAT on PHP_n^{n+1} with various symmetry breakers.

3.10 Symmetry breaking and PHP_n^{n+1}, for further details see Knuth's TAOCP on satisfiability or slides Symmetry in SAT: an overview.

3.11 Try BreakID.