

Logical reasoning and programming, lab session 3

(October 3, 2022)

3.1 Check the details of two watched literals, for example, these and these slides illustrate the data structure.

3.2 Decide the satisfiability of

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

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

3.3 Given the formula $\{\{p_1, p_2\}, \{\bar{p}_1, p_3\}, \{p_2, \bar{p}_3\}, \{\bar{p}_2, \bar{p}_4\}, \{\bar{p}_3, p_4\}\}$, what clause will a CDCL solver learn first if it begins by deciding that p_1 is true?

3.4 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?

3.5 Decide the satisfiability of

$$\begin{aligned} &\{\{p_1, \bar{p}_{13}\}, \{\bar{p}_1, \bar{p}_2, p_{14}\}, \{p_3, p_{15}\}, \{p_4, p_{16}\}, \\ &\quad \{\bar{p}_3, \bar{p}_5, p_6\}, \{\bar{p}_5, \bar{p}_7\}, \{\bar{p}_6, p_7, p_8\}, \{\bar{p}_4, \bar{p}_8, \bar{p}_9\}, \{\bar{p}_1, p_9, \bar{p}_{10}\}, \\ &\quad \{p_9, p_{11}, \bar{p}_{14}\}, \{p_{10}, \bar{p}_{11}, p_{12}\}, \{\bar{p}_2, \bar{p}_{11}, \bar{p}_{12}\}\} \end{aligned}$$

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 .

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

3.7 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

$$\bigwedge_{i=1}^n ((\bar{x}_i \vee y_i \vee \bar{a}_{i-1}) \wedge (\bar{x}_i \vee a_i \vee \bar{a}_{i-1}) \wedge (y_i \vee a_i \vee \bar{a}_{i-1})),$$

where \bar{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 \bar{a}_0 always false and hence useless?

(c) Why can we replace $(\bar{x}_n \vee y_n \vee \bar{a}_{n-1}) \wedge (\bar{x}_n \vee a_n \vee \bar{a}_{n-1}) \wedge (y_n \vee a_n \vee \bar{a}_{n-1})$ just by $(\bar{x}_n \vee y_n \vee \bar{a}_{n-1})$? 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 $(\bar{x}_n \vee y_n \vee \bar{a}_{n-1})$ by $(\bar{x}_n \vee \bar{a}_{n-1}) \wedge (y_n \vee \bar{a}_{n-1})$?

3.8 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.

- 3.9** 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?

- 3.10** Try PicoSAT/pycosat and PySAT on PHP_n^{n+1} with various symmetry breakers.
- 3.11** Symmetry breaking and PHP_n^{n+1} (cont'd). For details see Knuth's TAOCP on satisfiability or slides Symmetry in SAT: an overview.
- 3.12** Try BreakID.