Practical SAT Solving (ST 2024) | Assignment 3

Markus Iser, Dominik Schreiber, Tomáš Balyo

Algorithm Engineering (KIT) | 2024-06-04 – 2024-06-18

1 CDCL (7 Points)

Simulate the CDCL algorithm by hand on the formula F. Draw the implication graph for each conflict and learn the 1-UIP clause. Select branching literals in the order $x_1, x_2, x_3, \ldots$

$$\begin{aligned} F = \big\{ &\{x_1, x_{13}\}, \{\overline{x}_1, \overline{x}_2, x_{14}\}, \{x_3, x_{15}\}, \{x_4, x_{16}\}, \\ &\{\overline{x}_5, \overline{x}_3, x_6\}, \{\overline{x}_5, \overline{x}_7\}, \{\overline{x}_6, x_7, x_8\}, \{\overline{x}_4, \overline{x}_8, \overline{x}_9\}, \\ &\{\overline{x}_1, x_9, \overline{x}_{10}\}, \{x_9, x_{11}, \overline{x}_{14}\}, \{x_{10}, \overline{x}_{11}, x_{12}\}, \{\overline{x}_2, \overline{x}_{11}, \overline{x}_{12}\} \big\} \end{aligned}$$

2 Variable Elimination (3 Points)

For a gate encoding E with output x in a formula $F = E \cup R$, we simplified the resolvents $(E_x \cup R_x) \otimes (E_{\overline{x}} \cup R_{\overline{x}})$ by $S := (E_x \otimes R_{\overline{x}}) \cup (R_x \otimes E_{\overline{x}})$, dropping both $R_x \otimes R_{\overline{x}}$ and $E_x \otimes E_{\overline{x}}$. Show that the clauses in $R_x \otimes R_{\overline{x}}$ can be derived from S by resolution. You can assume that E encodes a binary AND gate.

3 Variable Elimination (2 × 2 Points)

Let the formula S with gate encodings E_1 and E_2 be given. Apply variable elimination for gates for variables a and r. Give the clause sets after each elimination step. Try the following two strategies.

1. Eliminate variable a first, and then r if possible.
2. Eliminate variable r first, and then a if possible.

$$S = \big\{ \underbrace{\{\neg x, \neg y, a\}, \{x, \neg a\}, \{y, \neg a\}}_{E_1}, \underbrace{\{\neg a, r\}, \{\neg z, r\}, \{a, z, \neg r\}}_{E_2}, \{a, z, r\}, \{\neg a, \neg r\} \big\}$$

4 Blocked Clauses (3 × 3 Points)

If Blocked Clause Elimination (BCE) reduces a formula F to the empty formula then F is called a blocked set. Prove the following statements.

1. Any formula F can be partitioned into two blocked sets S and L such that $F = S \cup L$. Design a linear algorithm that produces L and S from F.
2. Blocked sets are not closed unter resolution. If F is a blocked set then $F \cup C_1 \otimes C_2$, where $C_1, C_2 \in F$ may not be a blocked set anymore.
3. Blocked sets are not closed unter partially assigning variables. If F is a blocked set then $F_{x=v}$ (the result of assigning v to x and subsequent simplification) may not be a blocked set anymore.

5 Hidoku Competition (12 Points)

Hidoku a.k.a Hidato a.k.a Number Snake is a logic puzzle where the goal is to fill a grid with consecutive numbers that connect horizontally, vertically, or diagonally. The grid is rectangular and some of the cells are pre-filled.

1			5
	7		
			14
		16	

Unsolved Hidoku

1	3	4	5
2	7	6	13
8	11	12	14
9	10	16	15

Solved Hidoku

1			5
2			14
		16	

Unsolvable Hidoku

The input is a string, which represents a Hidoku puzzle. The first two numbers are the width and heigth of the grid followed by the values separated by commas, rows are separated by semicolons, 0 represents an empty cell. For the example above it looks as follows.

$$4, 4 : 1, 0, 0, 5; 0, 7, 0, 0; 0, 0, 0, 14; 0, 0, 16, 0$$

The output is a string, which represents the solution of the given Hidoku puzzle. A Hidoku puzzle may be unsatisfiable, in that case the output is `sol:UNSAT`, otherwise the solution is given in the same format as the input. For the example above it looks as follows.

$$sol : 1, 3, 4, 5; 2, 7, 6, 13; 8, 11, 12, 14; 9, 10, 16, 15;$$

Implement a SAT based Hidoku solver. For a working solver you get 12 points. The fastest solver will receive a bonus of 12 points. Some examples for testing can be found online at `https://satlecture.github.io/kit2024/exercises/hidoku/hidokus.txt`