Mathematical Logics

3. Decision procedure - Examples and usage of MiniSat

Luciano Serafini

Fondazione Bruno Kessler, Trento, Italy

October 16, 2014

DPLL some additional example

Example

Use the DPLL calculus to check satisfiability of the following sets S. In case of a satisfiable set, give (one of) the models encountered by the DPLL search.

1. $\{\{A, \neg B, \neg D\}, \{\neg A, \neg B, \neg C\}, \{\neg A, C, \neg D\}, \{\neg A, B, C\}\}$
2. $\{\{A, B, C\}, \{\neg B, \neg C\}, \{\neg B, \neg A\}, \{\neg A, B, C\}, \{A, B\}$
   $\{A, C\}, \{B, \neg A, \neg C\}\}$
3. $\{\{\neg A, B\}, \{\neg C, D\}, \{\neg E, \neg F\}, \{F, \neg E, \neg B\}\}$

Solution

see https://www.se.tu-darmstadt.de/fileadmin/user_upload/Group_SE/Page_Content/Teaching/ATP13/solutions-07-slides.pdf

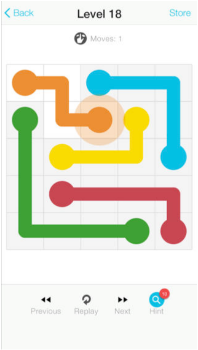
Example of usage of MiniSat

Example

Check if

- $p \vee q, \neg q \vee r \models p \vee r$
- $p \wedge \neg p \models \neg(r \rightarrow q) \wedge (r \rightarrow q)$
- $(p \rightarrow q), (s \rightarrow t) \models (p \vee s) \rightarrow (q \wedge t)$

Formalizing DrawLine game



- The game is played on an $n \times n$ board
- The game starts with a n differently colored pairs of nodes placed on the board
- The goal is to connect each pair of nodes of the same color with a path
- A path is constituted of a non repeating sequence of cells, one nearby the other, that starts with one of the two points and ends with the other of the same color (where nearby means, left, right, above, and below)
- Two paths can never cross each other (i.e., they cannot pass on the same cell)

Formalizing DrawLine 5×5

- we have 5 colors, $C = \{yellow, red, blue, green, orange\}$ and 25 cells, organized in 5 rows $\times$ 5 columns.
- the propositional variable p_{ijk} with $i \in C$, and $j, k \in \{1, 2, 3, 4, 5\}$ is used to represent the fact that a path colored with i passes from the cell jk
- The propositional variable s_{ijk} with $i \in C$, and $j, k \in \{1, 2, 3, 4, 5\}$ is used to represent the fact that a path of color i starts at cell jk
- The propositional variable e_{ijk} with $i \in C$, and $j, k \in \{1, 2, 3, 4, 5\}$ is used to represent the fact that a path of color i ends at cell jk

Formalizing DrawLine 5×5

- The game starts with a n differently colored pairs of nodes placed on the board

for every color there is a starting point and an ending point

$$\bigwedge_{i \in C} \left(\bigvee_{j,k=1}^{5} s_{ijk} \wedge \bigvee_{j,k=1}^{5} e_{ijk} \right) \quad (1)$$

For every colour the sarting point and ending point are different

$$\bigwedge_{i \in C} \bigwedge_{j,k=1}^{5} (s_{ijk} \rightarrow \neg e_{ijk}) \quad (2)$$

Starting points and ending points of a color are part of a path of the same color

$$\bigwedge_{i \in C} \bigwedge_{j,k \in 1}^{5} (s_{ijk} \vee e_{ijk} \rightarrow p_{ijk}) \quad (3)$$

Two paths cannot pass from the same square

$$\bigwedge_{i \neq i' \in C} \bigwedge_{j,k \in 1}^{5} \neg (p_{ijk} \wedge p_{i'jk}) \quad (4)$$

Formalizing DrawLine 5×5

- A starting point of a color i is closed only to a single point of the same color

$$s_{i00} \vee e_{i00} \rightarrow p_{i01} \wedge \neg p_{i10} \vee \neg p_{i01} \wedge p_{i10}$$

$$s_{i05} \vee e_{i05} \rightarrow p_{i04} \wedge \neg p_{i15} \vee \neg p_{i04} \wedge p_{i15}$$

$$s_{i05} \vee e_{i50} \rightarrow p_{i40} \wedge \neg p_{i51} \vee \neg p_{i40} \wedge p_{i51}$$

$$s_{i55} \vee e_{i55} \rightarrow p_{i54} \wedge \neg p_{i45} \vee \neg p_{i54} \wedge p_{i45}$$

$$\bigwedge_{j=2}^{4} s_{ij0} \vee e_{ij0} \rightarrow \begin{array}{l} p_{i(j-1)0} \wedge \neg p_{ij1} \wedge \neg p_{i(j+1)0} \vee \\ \neg p_{i(j-1)0} \wedge p_{ij1} \wedge \neg p_{i(j+1)0} \vee \\ \neg p_{i(j-1)0} \wedge \neg p_{ij1} \wedge p_{i(j+1)0} \end{array}$$

$$\bigwedge_{k=2}^{4} s_{i0k} \vee e_{i0k} \rightarrow \begin{array}{l} p_{i0(k-1)} \wedge \neg p_{i1k} \wedge \neg p_{i0(k+1)} \vee \\ \neg p_{i0(k-1)} \wedge p_{i1k} \wedge \neg p_{i0(k+1)} \vee \\ \neg p_{i0(k-1)} \wedge \neg p_{i1k} \wedge p_{i0(k+1)} \end{array}$$

$$\bigwedge_{j=2}^{4} s_{ij5} \vee e_{ij5} \rightarrow \begin{array}{l} p_{i(j-1)5} \wedge \neg p_{ij4} \wedge \neg p_{i(j+1)5} \vee \\ \neg p_{i(j-1)5} \wedge p_{ij4} \wedge \neg p_{i(j+1)5} \vee \\ \neg p_{i(j-1)5} \wedge \neg p_{ij4} \wedge p_{i(j+1)5} \end{array}$$

$$\bigwedge_{k=2}^{4} s_{i5k} \vee e_{i5k} \rightarrow \begin{array}{l} p_{i5(k-1)} \wedge \neg p_{i4k} \wedge \neg p_{i5(k+1)} \vee \\ \neg p_{i5(k-1)} \wedge p_{i4k} \wedge \neg p_{i5(k+1)} \vee \\ \neg p_{i5(k-1)} \wedge \neg p_{i4k} \wedge p_{i5(k+1)} \end{array}$$

$$\bigwedge_{i,k=2}^{4} s_{ijk} \vee e_{ijk} \rightarrow \begin{array}{l} p_{ij(k-1)} \wedge \neg p_{i(j-1)k} \wedge \neg p_{ij(k+1)} \wedge \neg p_{i(j+1)k} \vee \\ \neg p_{ij(k-1)} \wedge p_{i(j-1)k} \wedge \neg p_{ij(k+1)} \wedge \neg p_{i(j+1)k} \vee \\ \neg p_{ij(k-1)} \wedge \neg p_{i(j-1)k} \wedge p_{ij(k+1)} \wedge \neg p_{i(j+1)k} \vee \\ \neg p_{ij(k-1)} \wedge \neg p_{i(j-1)k} \wedge \neg p_{ij(k+1)} \wedge p_{i(j+1)k} \end{array}$$

Formalizing DrawLine 5×5

- A path is constituted of a non repeating sequence of cells, one nearby the other, that starts with one of the two points and ends with the other of the same color (where nearby means, left, right, above, and below)

$$p_{i00} \wedge \neg s_{i00} \wedge \neg e_{i00} \rightarrow p_{i01} \wedge p_{i10}$$
$$p_{i05} \wedge \neg s_{i05} \wedge \neg e_{i05} \rightarrow p_{i04} \wedge p_{i15}$$

$$p_{i05} \wedge \neg s_{i05} \wedge \neg e_{i50} \rightarrow p_{i40} \wedge p_{i51}$$
$$p_{i55} \wedge \neg s_{i55} \wedge \neg e_{i55} \rightarrow p_{i54} \wedge p_{i45}$$

$$\bigwedge_{j=2}^{4} p_{ij0} \wedge \neg s_{ij0} \wedge \neg e_{ij0} \rightarrow \begin{array}{l} p_{i(j-1)0} \wedge p_{ij1} \wedge \neg p_{i(j+1)0} \vee \\ \neg p_{i(j-1)0} \wedge p_{ij1} \wedge p_{i(j+1)0} \vee \\ p_{i(j-1)0} \wedge \neg p_{ij1} \wedge p_{i(j+1)0} \end{array}$$

$$\bigwedge_{k=2}^{4} p_{i0k} \wedge \neg s_{i0k} \wedge \neg e_{i0k} \rightarrow \begin{array}{l} p_{i0(k-1)} \wedge p_{i1k} \wedge \neg p_{i0(k+1)} \vee \\ \neg p_{i0(k-1)} \wedge p_{i1k} \wedge p_{i0(k+1)} \vee \\ p_{i0(k-1)} \wedge \neg p_{i1k} \wedge p_{i0(k+1)} \end{array}$$

$$\bigwedge_{j=2}^{4} p_{ij5} \wedge \neg s_{ij5} \wedge \neg e_{ij5} \rightarrow \begin{array}{l} p_{i(j-1)5} \wedge p_{ij4} \wedge \neg p_{i(j+1)5} \vee \\ \neg p_{i(j-1)5} \wedge p_{ij4} \wedge p_{i(j+1)5} \vee \\ p_{i(j-1)5} \wedge \neg p_{ij4} \wedge p_{i(j+1)5} \end{array}$$

$$\bigwedge_{k=2}^{4} p_{i5k} \wedge \neg s_{i5k} \wedge \neg e_{i5k} \rightarrow \begin{array}{l} p_{i5(k-1)} \wedge p_{i4k} \wedge \neg p_{i5(k+1)} \vee \\ \neg p_{i5(k-1)} \wedge p_{i4k} \wedge p_{i5(k+1)} \vee \\ p_{i5(k-1)} \wedge \neg p_{i4k} \wedge p_{i5(k+1)} \end{array}$$

$$\bigwedge_{i,k=2}^{4} p_{ijk} \wedge \neg s_{ijk} \wedge \neg e_{ijk} \rightarrow \begin{array}{l} p_{ij(k-1)} \wedge p_{i(j-1)k} \wedge \neg p_{ij(k+1)} \wedge \neg p_{i(j+1)k} \vee \\ p_{ij(k-1)} \wedge \neg p_{i(j-1)k} \wedge p_{ij(k+1)} \wedge \neg p_{i(j+1)k} \vee \\ p_{ij(k-1)} \wedge \neg p_{i(j-1)k} \wedge \neg p_{ij(k+1)} \wedge p_{i(j+1)k} \vee \\ \neg p_{ij(k-1)} \wedge p_{i(j-1)k} \wedge p_{ij(k+1)} \wedge \neg p_{i(j+1)k} \vee \\ \neg p_{ij(k-1)} \wedge p_{i(j-1)k} \wedge \neg p_{ij(k+1)} \wedge p_{i(j+1)k} \vee \\ \neg p_{ij(k-1)} \wedge \neg p_{i(j-1)k} \wedge p_{ij(k+1)} \wedge p_{i(j+1)k} \end{array}$$

MiniSat http://minisat.org

About

MINISAT is a minimalistic, open-source SAT solver, developed to help researchers and developers alike to get started on SAT. It is released under the MIT licence, and is currently used in a number of projects (see "Links"). On this page you will find binaries, sources, documentation and projects related to MINISAT, including the Pseudo-boolean solver MINISAT+ and the CNF minimizer/preprocessor SATELITE.

How to use MiniSat

Input format

MINISAT, like most SAT solvers, accepts its input in a simplified "DIMACS CNF" format, which is a simple text format. Every line beginning "c" is a comment. The first non-comment line must be of the form:

```
p cnf NUMBER_OF_VARIABLES NUMBER_OF_CLAUSES
```

Each of the non-comment lines afterwards defines a clause. Each of these lines is a space-separated list of variables; a positive value means that corresponding variable (so 4 means x4), and a negative value means the negation of that variable (so -5 means -x5). Each line must end in a space and the number 0.

```
c Here is a comment
p cnf 5 3
1 -5 4 0
-1 5 3 4 0
-3 -4 0
```

is the representation of the CNF

$$\{\{x_1, \neg x_5, x_4\}, \{\neg x_1, x_5, x_3, x_4\}, \{\neg x_3, \neg x_4\}\}$$

Invoking MiniSat

MiniSAT's usage is:

```
minisat [options] [INPUT-FILE [RESULT-OUTPUT-FILE]]
```

MiniSat output format

- When run, miniSAT sends to standard error a number of different statistics about its execution. It will output to standard output either "SATISFIABLE" or "UNSATISFIABLE" (without the quote marks), depending on whether or not the expression is satisfiable or not.
- If you give it a RESULT-OUTPUT-FILE, MiniSat will write text to the file. The first line will be "SAT" (if it is satisfiable) or "UNSAT" (if it is not). If it is SAT, the second line will be set of assignments to the boolean variables that satisfies the expression. (There may be many others; it simply has to produce one assignment).
- for example the output file of the previous example is

  ```
  SAT
  1 2 -3 4 5 0
  ```

  This means that it is satisfiable, with the model $\mathcal{I}$ with $\mathcal{I}(x_1) = \mathit{true}, \mathcal{I}(x_2) = \mathit{true}, \mathcal{I}(x_3) = \mathit{false}, \mathcal{I}(x_4) = \mathit{true}$ and $\mathcal{I}(x_5) = \mathit{true}$.

Translating CNF into MiniSat input format

Example

- $\{\{A, \neg B, \neg D\}, \{\neg A, \neg B, \neg C\}, \{\neg A, C, \neg D\}, \{\neg A, B, C\}\}$

```
c A -> 1, B -> 2, C -> 3, D -> 4
p cnf 4 4
1 -2 -4 0
-1 -2 -3 0
-1 3 -4 0
-1 2 3 0
```

Translating CNF into MiniSat input format

Example

- $\{\{A, B, C\}, \{\neg B, \neg C\}, \{\neg B, \neg A\}, \{\neg A, B, C\}, \{A, B\}$
  $\{A, C\}, \{B, \neg A, \neg C\}\}$

```
c A -> 1, B -> 2, C -> 3
p CNF 3 7
1 2 3 0
-2 -3 0
-1 -2 0
-1 2 3 0
1 2 0
1 3 0
-1 2 -3 0
```

Translating CNF into MiniSat input format

exercise

Rewrite in MiniSat input format (DIMACS) the following set of clauses:

- $\{\{\neg A, B\}, \{\neg C, D\}, \{\neg E, \neg F\}, \{F, \neg E, \neg B\}\}$

Exercise

Write a program that takes in input a natural number n and generates the DIMACS format for the DrowLine formalization.

Obtaining more than one assignment from MiniSat

- MiniSat searches for one assignment that satisfies a CNF, and if there is one, it is returned.
- Question: How can I obtain more than one assignment?
- Answer: Suppose that MiniSat $C_1, \dots, C_n$ returns $l_1, \dots, l_n$. To check if there is another assignment, different from $l_1, \dots, l_n$ we can check if $C_1, \dots, C_n \wedge \neg(l_1 \wedge \dots \wedge l_n)$ is satisfiable
- Notice that $\neg(l_1 \wedge \dots \wedge l_n)$ is the clauses $\neg l_1 \vee \dots \vee \neg l_n$
- In practice we can rerun MiniSat on $C_1, \dots, C_n, \{\neg l_1, \dots, \neg l_n\}$