MajorSat: A SAT Solver to Majority Logic

Speaker: Ching-Yi Huang
Authors: Yu-Min Chou, Yung-Chih Chen*, Chun-Yao Wang, Ching-Yi Huang
National Tsing Hua University, Taiwan
*Yuan Ze University, Taiwan

Jan. 27, 2016
Outline

- Introduction
- Solving Methods
  - Conflict Analysis
  - Conflict-driven Learning
  - Variable Decision Order Heuristic
- Majority Gate Transformation
- Experimental Results
- Conclusions
The majority function is a concise way to represent a Boolean expression.

The majority function, denoted as \( M(x_1, x_2, \ldots, x_n) \), is an odd-input function which is evaluated as 1 iff more than half of inputs are 1.

- E.g. \( M(a, b, c, d, e) = 1 \) if \( a, b, c \) are 1.

The majority function can express any logic represented by OR or AND operations.

- E.g. \((a \lor b \lor c) \equiv M(a, b, c, 1, 1)\)  
  \((a \land b \land c) \equiv M(a, b, c, 0, 0)\)

Recently, majority logic attracts more attentions and some synthesis algorithms and axiomatic system for majority logic have been proposed.
The Boolean satisfiability (SAT) problem can be expressed in various forms

- E.g. conjunctive-normal-form (CNF), disjunctive-normal-form (DNF), the conjunction of majority functions, … etc

The conjunctive-normal-form (CNF) solvers for the SAT problem have had a remarkable achievement and have been widely used in the domains of synthesis and verification of logic circuit

To express specific logic functions such as majority decision problems, majority functions can be more compact and expressive compared to traditional CNF
Motivation (1/2)

- It’s impractical to convert the large-size majority function into the CNF for been solved by CNF SAT solvers
  - The time required to convert the majority function to the CNF grows exponentially with the size of the majority function
  - Modern CNF SAT solvers may be not able to store so many clauses
  - E.g. \( M(a, b, c, d, e) = (a \lor b \lor c) \land (a \lor b \lor d) \land (a \lor b \lor e) \land (a \lor c \lor d) \land (a \lor c \lor e) \land (a \lor d \lor e) \land (b \lor c \lor d) \land (b \lor c \lor e) \land (b \lor d \lor e) \land (c \lor d \lor e), \left( \frac{n}{2} \right) \) clauses, where \( n \) is the size of the majority function
Motivation (2/2)

- Therefore, we propose a new SAT solver – **MajorSat**, which can directly solve the instances with majority functions and CNF clauses.

- Definition 1: A majority expression, denoted as ME, is a conjunction of majority functions.
  - $M(a, b, \overline{c}) \land M(\overline{d}, \overline{e}, 0, f, g)$ is an ME.
Outline

- Introduction
- **Solving Methods**
  - Conflict Analysis
  - Conflict-driven Learning
  - Variable Decision Order Heuristic
- Majority Gate Transformation
- Experimental Results
- Conclusions
Preprocessing

Property 1:

- A majority function with size $n$ can be simplified to a majority function with size $(n - 2)$ by removing two inputs that are either the same variable but with different phases, or 0 and 1, while preserving the same result.

E.g. $M(a, a, b, c, \bar{a}, 0, 1)$ can be reduced to $M(a, b, c, 0, 1)$, and $M(a, b, c, 0, 1)$ can be further reduced to $M(a, b, c)$.
Conflicts Analysis (1/6)

- Property 2:
  - The ME is **UNSAT** if it satisfies the following conditions simultaneously
    - (1) There exists a majority function of size $n$ with an input $x \in \{a, \bar{a}, 0, 1\}$ appearing more than $\lceil n/2 \rceil$ times
    - (2) There exists another majority function of size $m$ with an input $y \in \{a, \bar{a}, 0, 1\}$ appearing more than $\lceil m/2 \rceil$ times
    - (3) $x = \bar{y}$

- E.g. An ME: $M(a, a, a, b, c) \land M(\bar{a}, \bar{a}, \bar{a}, a, d, e, f)$
  
  \[
  M(a, a, a, b, c) \hspace{1cm} \text{conflicts with} \hspace{1cm} M(\bar{a}, \bar{a}, \bar{a}, a, d, e, f)
  \]

  \[
  3 > \lceil 5/2 \rceil = 2 \hspace{1cm} 4 > \lceil 7/2 \rceil = 3
  \]
Definition 2: Given a majority function $s$, the minimum number of variables required to be assigned such that $s$ is 1 is denoted as $\text{MinV}_s$.

- E.g. For a majority function $s$: $M(a, a, b, c, d)$, $\text{MinV}_s$ is 2, i.e., $(a, b)$, $(a, c)$, or $(a, d) = (1, 1)$

Property 3:

- Given two majority functions $s$ and $t$, the $\text{ME} = s \land t$ is **UNSAT** if
  - $s$ and $t$ have the same variable set with size $w$, and the phase of each variable in $s$ is opposite to the phase of that in $t$
  - Both $\text{MinV}_s$ and $\text{MinV}_t > \lfloor w/2 \rfloor$
Conflicts Analysis (3/6)

- **Example:**

  ✓ $M(a, b, c) \land M(\bar{a}, \bar{b}, \bar{c})$ is UNSAT

  $w = 3, \min V_{M(a, b, c)} = \min V_{M(\bar{a}, \bar{b}, \bar{c})} = 2 > \lfloor w/2 \rfloor = 1$

  ✓ $M(a, b, c) \land M(\bar{a}, \bar{b}, \bar{b}, \bar{b}, \bar{c})$ doesn’t work with property 3

  $w = 3, \min V_{M(\bar{a}, \bar{b}, \bar{b}, \bar{b}, \bar{c})} = 1 \neq \lfloor w/2 \rfloor = 1$

  ✓ $M(a, b, c) \land M(\bar{a}, \bar{b}, \bar{b}, \bar{b}, \bar{c}, \bar{c}, \bar{c})$ is UNSAT

  $w = 3, \min V_{M(a, b, c)} = \min V_{M(\bar{a}, \bar{b}, \bar{c}, \bar{c}, \bar{c})} = 2 > \lfloor w/2 \rfloor = 1$
Property 4:

- Given a majority function of size $n$, resolving a literal $a$ (assigning 0 to the literal $a$) that occurs $k$ times in the function implies all the other literals that occur $\geq \lfloor n/2 \rfloor - k$ times for satisfying the majority function.

- E.g. In $M(a, a, b, b, \overline{c}, \overline{c}, d)(n=7)$, resolving the literal $a$ ($a = 0, k = 2$) implies both literals $b$ and $\overline{c}$ to 1.

Construct the implication graph based on Property 4:

- The ME is **UNSAT** if there exists a *strongly-connected component* in the implication graph containing nodes of a variable with opposite phases.
Conflicts Analysis (5/6)

Consider the ME containing only 3-input majority functions:

\[ F = M(a, b, c) \land M(\bar{b}, \bar{c}, d) \land M(\bar{a}, \bar{c}, \bar{d}) \]

The potential conflicts hidden in the ME can be extracted by forming the following implications:

\[ \bar{a} \rightarrow (b \land c), \quad \bar{b} \rightarrow (a \land c), \quad \bar{c} \rightarrow (a \land b) \]
\[ b \rightarrow (\bar{c} \land d), \quad c \rightarrow (\bar{b} \land d), \quad \bar{d} \rightarrow (\bar{b} \land \bar{c}) \]
\[ a \rightarrow (\bar{c} \land \bar{d}), \quad c \rightarrow (\bar{a} \land \bar{d}), \quad d \rightarrow (\bar{a} \land \bar{c}) \]
Assigning variable $a = 1$ leads to $\overline{a} = 1$, and assigning variable $\overline{a} = 1$ leads to $a = 1$

Node $a$ and node $\overline{a}$ belong to the same **strongly-connected component**, which means the original formula $M(a, b, c) \land M(b, \overline{c}, d) \land M(\overline{a}, \overline{c}, \overline{d})$ is **UNSAT**
Outline

- Introduction
- **Solving Methods**
  - Conflict Analysis
  - Conflict-driven Learning
  - Variable Decision Order Heuristic
- Majority Gate Transformation
- Experimental Results
- Conclusions
Searching with Conflict-driven Learning Technique

**Searching procedure**

- Determine the values of variables one by one
- Record the reasons of conflicts
  - Can help in **pruning** the search space

clause: $\bar{x}_1 \lor x_3 \lor \bar{x}_5$
Majority Propagation

Property 5 (Majority Propagation):

- During the searching procedure, when a majority function \( s \) of size \( n \) with \( k \) inputs have been assigned to 0

  Any unassigned literal in \( s \) that occurs \( \geq \left\lfloor \frac{n}{2} \right\rfloor - k \) times will be implied to 1 for satisfying \( s \)

- E.g. In \( M(a, a, a, b, c, e, e, f, g) \), if literals \( e \) and \( g \) have been assigned 0 (\( k = 3 \)), the literal \( a \) is implied to 1 since \( a \) occurs three times, which \( \geq \left\lfloor \frac{9}{2} \right\rfloor - 3 = 2 \)
The following example shows the procedure of the searching with conflict-driven learning technique:

\[ M(a, a, a, b, c, c, e, 1, 1) \land M(\bar{a}, \bar{b}, c, e, 1) \land M(d, \bar{e}, f, g, h) \]

Decide \( f = 0 \)

\[ M(a, a, a, b, c, c, e, 1, 1) \land M(\bar{a}, \bar{b}, c, e, 1) \land M(d, \bar{e}, f, g, h) \]

Decide \( g = 0 \)

\[ M(a, a, a, b, c, c, e, 1, 1) \land M(\bar{a}, \bar{b}, c, e, 1) \land M(d, \bar{e}, f, g, h) \]

Majority Propagation

Imply \( d = 1, \bar{e} = 1, \) and \( h = 1 \)
Learning Example (2/3)

\[
M(a, a, a, b, c, c, e, 1, 1) \land M(\bar{a}, \bar{b}, c, e, 1) \land M(d, \bar{e}, f, g, h)
\]

Decide \( c = 0 \)

\[
M(a, a, a, b, c, c, e, 1, 1) \land M(\bar{a}, \bar{b}, c, e, 1) \land M(d, \bar{e}, f, g, h)
\]

Majority Propagation

Imply \( a = 1, \bar{a} = 1, \) and \( \bar{b} = 1 \)

Conflict!
Learning Example (3/3)

The clause \((e \lor c)\) is learned, and is added to the original ME.
Outline

- Introduction
- **Solving Methods**
  - Conflict Analysis
  - Conflict-driven Learning
  - Variable Decision Order Heuristic
- Majority Gate Transformation
- Experimental Results
- Conclusions
Variable Decision Order Heuristic (1/4)

Definition 3: \( threshold_m \) and \( weight_m(x) \) are defined as \([n/2]\) and the appearance time of the variable \( x \) in a majority function \( m \) of size \( n \)

Definition 4: The score function of variable \( x \) in a majority function \( m \) and a clause \( c \) is denoted as \( scoreM_m(x) \) and \( scoreC_c(x) \), which are

\[
\begin{align*}
\text{score}M_m(x) &= \begin{cases} 
0 & \text{if } x \text{ is absent in } m \\
1 - (threshold_m - weight_m(x))/\text{size of } m & \text{if } x \text{ is in } m
\end{cases} \\
\text{score}C_c(x) &= \begin{cases} 
0 & \text{if } x \text{ is absent in } c \\
1 & \text{if } x \text{ is in } c
\end{cases}
\end{align*}
\]

Definition 5: The score function \( score(x) \) is to decide the variable decision, which is

\[
score(x) = \sum_{m \in M} scoreM_m(x) + \sum_{c \in C} scoreC_c(x)
\]
Variable Decision Order Heuristic (2/4)

- According to **Definition 4**, the scores of variables are related to their appearance times in majority functions
  - Choosing the variable of a higher score can increase the probability of reaching the **satisfiable branch**
- E.g. Given an expression $F: M(a, a, a, b, b, \bar{c}, d) \land M(\bar{b}, \bar{c}, \bar{d}) \land (\bar{a} \lor b \lor c)$,

\[
\begin{align*}
score(a) &= (1 - (4 - 3)/7) + 0 + 1 = 13/7 = 39/21 \\
score(b) &= (1 - (4 - 2)/7) + (1 - (2 - 1)/3) + 1 = 50/21 \\
score(c) &= (1 - (4 - 1)/7) + (1 - (2 - 1)/3) + 1 = 47/21 \\
score(d) &= (1 - (4 - 1)/7) + (1 - (2 - 1)/3) + 0 = 26/21
\end{align*}
\]

$score(b) > score(c) > score(a) > score(d)$, which indicates that the variable decision order is $b > c > a > d$
Variable Decision Order Heuristic (3/4)

- Update scores of variables when conflicts happen
  - Add 1 to the scores of variables on the paths from conflict nodes to decision nodes
  - Recompute the variable decision order
  - Lead the search to unsatisfiable branches
    - Help in learning more conflict clauses
Variable Decision Order Heuristic (4/4)

Variable decided
- $f = 0$
- $g = 0$
- $c = 0$

Variable implied
- $d = 1$
- $e = 0$
- $h = 1$
- $b = 0$
- $a = 0$
- $a = 1$

level: 1 level: 2 level: 3

score(a), score(c), score(e), score(f), and score(g) are added by 1 after the conflict happens
Outline

- Introduction
- Solving Methods
  - Conflict Analysis
  - Conflict-driven Learning
  - Variable Decision Order Heuristic
- Majority Gate Transformation
- Experimental Results
- Conclusions
The characteristic function of a majority gate can be expressed by an ME:

\[
M(a, b, c, d, d, d, 1, 1, 1) \land M(\bar{a}, \bar{b}, \bar{c}, d, d, d, 1, 1, 1)
\]
Therefore, the characteristic function of a majority network can be also expressed as an ME:

\[ M(a, b, c, d, 1, 1, 1) \land M(\bar{a}, \bar{b}, \bar{c}, d, d, d, 1, 1, 1) \land M(d, e, f, \bar{g}, \bar{g}, \bar{g}, 1, 1, 1) \land M(\bar{d}, \bar{e}, \bar{f}, g, g, g, 1, 1, 1) \land M(d, i, \bar{k}, \bar{k}, \bar{k}, 1, 1, 1) \land M(\bar{d}, \bar{i}, j, k, k, k, 1, 1, 1) \land M(g, i, k, \bar{m}, \bar{m}, \bar{m}, 1, 1, 1) \land M(\bar{g}, \bar{i}, \bar{k}, m, m, m, 1, 1, 1) \land m \]

The satisfiability of the above network can be evaluated through an ME.
Outline

- Introduction
- Solving Methods
  - Conflict Analysis
  - Conflict-driven Learning
  - Variable Decision Order Heuristic
- Majority Gate Transformation
- Experimental Results
- Conclusions
Experimental Environment

- **Platform**
  - Intel Xeon® E5530 2.40GHz CentOS 4.6 platform with 64GB memory
  - C++

- **Benchmarks**
  - CNF benchmarks from SATLIB for verifying the correctness
  - Randomly-generated benchmarks of ME with different scales for testing the efficiency
Experimental Results (1/2)

- CNF benchmarks with different numbers of variables and clauses

| Benchmarks | |variable| |clause| |Golden Result| |Solving Result|
|------------|-------|-------|------|------|--------------|------------|
|            |       |       |      |      | CNF | ME |
| uf20-91    | 20    | 91    |      |      | SAT | SAT | SAT |
| uf50-218   | 50    | 218   |      |      | SAT | SAT | SAT |
| uf75-325   | 75    | 325   |      |      | SAT | SAT | SAT |
| uf100-430  | 100   | 430   |      |      | SAT | SAT | SAT |
| uf125-538  | 125   | 538   |      |      | SAT | SAT | SAT |
| uf150-645  | 150   | 645   |      |      | SAT | SAT | SAT |
| uuf50-218  | 50    | 218   |      |      | UNSAT | UNSAT | UNSAT |
| uuf75-325  | 75    | 325   |      |      | UNSAT | UNSAT | UNSAT |
| uuf100-430 | 100   | 430   |      |      | UNSAT | UNSAT | UNSAT |
| uuf125-538 | 125   | 538   |      |      | UNSAT | UNSAT | UNSAT |
| uuf150-645 | 150   | 645   |      |      | UNSAT | UNSAT | UNSAT |
Experimental Results (2/2)

- The experiments on randomly-generated ME benchmarks of different scales
  - \((\text{number of variables})_\text{number of majority functions}_\text{size of majority function})\)

- The solving time of MajorSat is less than the time of converting ME into CNF coupled with the solving time of CNF solvers

<table>
<thead>
<tr>
<th>Benchmarks</th>
<th>MajorSat</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>t_{sol}(s)</td>
<td>t_{conv}(s)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75_75_17</td>
<td>2.37</td>
<td>1.37</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75_75_19</td>
<td>9.51</td>
<td>5.53</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75_75_21</td>
<td>12.38</td>
<td>22.80</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75_75_23</td>
<td>20.16</td>
<td>97.58</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75_75_25</td>
<td>42.37</td>
<td>410.07</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75_75_27</td>
<td>118.14</td>
<td>&gt; 1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75_75_29</td>
<td>158.01</td>
<td>&gt; 1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100_100_11</td>
<td>0.15</td>
<td>0.04</td>
<td>3.55</td>
<td>3.59</td>
<td>10.70</td>
<td>10.74</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100_100_13</td>
<td>2.81</td>
<td>0.14</td>
<td>475.10</td>
<td>475.24</td>
<td>404.40</td>
<td>404.54</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100_100_15</td>
<td>12.94</td>
<td>0.43</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100_100_17</td>
<td>59.59</td>
<td>2.10</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100_100_19</td>
<td>140.05</td>
<td>8.10</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100_100_21</td>
<td>894.18</td>
<td>30.36</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>125_125_11</td>
<td>2.88</td>
<td>0.04</td>
<td>895.80</td>
<td>895.84</td>
<td>237.60</td>
<td>237.64</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>125_125_13</td>
<td>11.08</td>
<td>0.17</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>125_125_15</td>
<td>152.87</td>
<td>0.59</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td>&gt; 1000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Outline

- Introduction
- Solving Methods
  - Conflict Analysis
  - Conflict-driven Learning
  - Variable Order Decision Heuristic
- Majority Gate Transformation
- Experimental Results
- Conclusions
Conclusions

- We propose a new SAT solver – MajorSat – for solving majority logic
- Several properties about majority functions are also investigated to increase the efficiency of MajorSat
- The experimental results show that MajorSat is more efficient in solving majority expressions than CNF solvers
Thanks for Attention

• Q&A
The Overall Flow of MajorSat

1. Start
2. Conflict Analysis
3. Conflict exists?
   - yes
   - no
4. Conflict-driven Learning with Variable Decision Order Heuristic
5. Find an assignment?
   - yes
     - SAT
   - no
8. UNSAT