SAT-based approach to verification of logical descriptions with functional indeterminacy

Liudmila Cheremisinova, Dmitry Novikov

The United Institute of Informatics Problems of National Academy of Sciences of Belarus

8-th International Workshop on Boolean Problems
September 18 - 19, 2008, Freiberg (Sachsen)
Typical task formulation of formal verification

Comparing circuit

Combinational equivalence checking is verifying functional equivalence of two combinational circuits.

Formal verification: both verified circuits are transformed into a single comparing circuit – a miter.

There is constant 0 on miter output if the compared circuits are equivalent.

The miter circuit is converted into a CNF form.

The task comes to SAT problem – checking whether CNF formula is not satisfiable.
Task formulation for the case of descriptions with functional indeterminacy

Checking whether a given system of incompletely specified Boolean functions (ISF) is implemented by a given combinational circuit

Data representation

<table>
<thead>
<tr>
<th>ISF system</th>
<th>Combinational circuit consisting of gates AND, OR and NOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ x_1 \ x_2 \ x_3 \ x_4 \ x_5 ]</td>
<td>[ f_1 \ f_2 ]</td>
</tr>
<tr>
<td>[ \begin{array}{c} \ - \ - \ 1 \ 1 \</td>
<td>\ 1 \ - \ 1 \ 1 \ 1 \ - \ - \</td>
</tr>
</tbody>
</table>

Input part

Multiple-output cubes \((u_i, t_i)\)

\[ u_i = x_2 x_4 x_5 \]

\[ t_i = f_2 \]
Simulation-based verification

1) Ternary matrix $U$ is transformed into a Boolean matrix $U'$

- $U'$ is obtained by replacing the ternary values $0$, $1$, $-$ with binary values $0$, $1$, respectively.

- For example, the first row of $U$ becomes $1$ $0$ $0$ $1$ $1$, and the second row becomes $0$ $1$ $1$ $1$ $1$.

2) Stimulating inputs of the circuit with binary signals corresponding to the rows of $U'$

3) Propagating signals through the circuit activating the circuit primary outputs

4) Checking whether circuit output signals do not contradict to the specified ones

**Drawback**: exponential growing of the matrix $U'$ when the number of "-" increases
Testing using SAT-solver whether the given combinational circuit implements:

**Subsequent Testing:**
- a **single-output** cube of the given specification;
- a **multiple-output** cube

**Simultaneous Testing:**
- all multiple-output cubes **simultaneously** within the only SAT session.
**CNF encoding of combinational circuit**

*(conventional circuit-to-CNFS transformation)*

1. Construction of CNF-formulas of local functions of gates;
2. Joining local CNFs into the conventional CNF of the circuit

**Combinational circuit**

![Combinational circuit diagram](image)

**Conventional CNF of the circuit**

\[
\begin{align*}
\varphi^\lor(y, z_1 z_2 \ldots z_k) &= (z_1 \lor \bar{y}) (z_2 \lor \bar{y}) \ldots (z_k \lor \bar{y}) (z_1 \lor z_2 \lor \ldots \lor z_k \lor \bar{y});
\end{align*}
\]

**k-input AND:**

\[
\varphi^\land(y, z_1 z_2 \ldots z_k) = (\bar{z_1} \lor y) (\bar{z}_2 \lor y) \ldots (\bar{z_k} \lor y) (\bar{z}_1 \lor \bar{z}_2 \lor \ldots \lor \bar{z}_k \lor y).
\]
SAT-based model of testing multiple-output cubes of ISF system

Multiple-output cube of an ISF system $f(x)$:

$$(u_i, t_i) \in f(x): u_i = x_{i1} x_{i2} \ldots x_{i_k}, t_i = f_{i1} f_{i2} \ldots f_{il}$$

$$u_i \rightarrow t_i: x_{i1} x_{i2} \ldots x_{i_k} \rightarrow f_{i1} f_{i2} \ldots f_{il}$$

A circuit has the same functionality as an ISF system $f(x)$ iff for every input stimulus implied by the input part $u_i$ of any $(u_i, t_i) \in f(x)$ Boolean vector of values of the circuit outputs is covered by the ternary output part $t_i$.

Or in terms of the circuit CNF: for every $(u_i, t_i) \in f(x)$ a partial value assignment $u_i \cup t_i$ of input and output variables should be satisfying assignment for the CNF:

$$\text{CNF} \rightarrow (u_i \rightarrow t_i)$$
Checking whether a circuit implements a single-output cube

The circuit implements the single-output cube, iff the extended CNF is unsatisfiable.
Checking whether the circuit implements a single-output cube: matrix representation

ISF system: 

\[
\begin{array}{cccccc}
\chi_1 & \chi_2 & \chi_3 & \chi_4 & \chi_5 & f_1 & f_2 \\
- & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & - & - & - & - & 1 \\
U = & 0 & 0 & 0 & - & T = & 0 & 1 \\
0 & 1 & - & 1 & 0 & 0 & 4 \\
- & 0 & 1 & 0 & - & - & 0 & 5 \\
- & 1 & - & 1 & - & - & 1 & 6 \\
\end{array}
\]

Extended conventional CNF of the circuit:

\[
\begin{array}{cccccccccccc}
x_1 & x_2 & x_3 & x_4 & x_5 & z_1 & z_2 & z_3 & y_1 & y_2 \\
1 & - & - & - & - & 0 & - & - & - & 1 \\
- & 1 & - & - & - & 0 & - & - & - & 2 \\
0 & 0 & - & - & - & 1 & - & - & - & 3 \\
- & - & - & 1 & - & - & 0 & - & - & 4 \\
- & - & - & 1 & - & 0 & - & - & - & 5 \\
- & - & - & 0 & 0 & 1 & - & - & - & 6 \\
- & - & - & 0 & - & - & 1 & 0 & - & 7 \\
- & - & - & 1 & - & - & - & 1 & - & 8 \\
- & - & - & - & - & 0 & 1 & - & - & 9 \\
- & - & - & - & 1 & 1 & - & 0 & - & 10 \\
- & - & - & - & - & 0 & - & - & 1 & 11 \\
- & - & - & - & - & 0 & - & - & 1 & 12 \\
\end{array}
\]

Searching for a satisfying assignment proves that there exists a counter-example for \((u_6, t_6)\):

\[1 1 - 1 1 1 1 1 1 0.\]
Checking whether the circuit implements a multiple-output cube

In general case the output part $t_i = y_{i1}^{\sigma_1} y_{i2}^{\sigma_2} ... y_{ik}^{\sigma_k}$ of a multiple-output cube $(u_i, t_i)$ consists of more than one component having definite value.

We add to CNF $n + 1$ clauses: $n$ clauses of the type $x_j^{\sigma_j} (x_j \in u_i)$ and a clause $y_{i1}^{\sigma_1} \lor y_{i2}^{\sigma_2} \lor ... \lor y_{ik}^{\sigma_k}$ of size $k$.
Checking whether the circuit implements a set of multiple-output cubes simultaneously

For each multiple-output cube \((u_i, t_i)\) we introduce new variable \(w_i\) which implies extension of the CNF generating by \((u_i, t_i)\):

\[
\text{ext}(u_i, t_i) = u_i \cup \neg t_i \quad \text{or}
\]

\[
\text{ext}(u_i, t_i) = x_1^{y_1} \land x_2^{y_2} \land \ldots \land x_n^{y_n} \land (y_{i1}^{\sigma_1} \lor y_{i2}^{\sigma_2} \lor \ldots \lor y_{ik}^{\sigma_k})
\]

\[
w_i \rightarrow \text{ext}(u_i, t_i) = w_i \lor \text{ext}(u_i, t_i) =
\]

\[
= (x_1^{y_1} \lor \neg w_i) \land (x_2^{y_2} \lor \neg w_i) \land \ldots \land (x_n^{y_n} \lor \neg w_i) \land (y_{i1}^{\sigma_1} \lor y_{i2}^{\sigma_2} \lor \ldots \lor y_{ik}^{\sigma_k} \lor \neg w_i)
\]

\(\bullet\)

The circuit CNF is appended with:
1) groups of clauses \(\bullet\) for testing all multiple-output cubes;
2) an additional clause \(w_1 \lor w_2 \lor \ldots \lor w_l\) to allow SAT-solver to seek for satisfying assignment for at least of one of introduced groups of clauses.

CNF formula for testing a set of multiple-output cubes simultaneously:

\[
\text{CNF} \land (w_1 \lor w_2 \lor \ldots \lor w_l) \land (w_1 \rightarrow \text{ext}(u_1, t_1)) \land (w_2 \rightarrow \text{ext}(u_2, t_2)) \land \ldots \land (w_l \rightarrow \text{ext}(u_l, t_l))
\]
The example of checking whether the circuit implements all multiple-output cubes

A set of multi-output cubes:

<table>
<thead>
<tr>
<th>x₁, x₂, x₃, x₄, x₅</th>
<th>f₁, f₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>(u₂, t₂) = 1 1 1 0 0</td>
<td>1 0</td>
</tr>
<tr>
<td>(u₆, t₆) = 1 1 0 1 0</td>
<td>0 1</td>
</tr>
</tbody>
</table>

Extended CNF:

<table>
<thead>
<tr>
<th>x₁, x₂, x₃, x₄, x₅, z₁, z₂, z₃, y₁, y₂, w₂, w₆</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 1 1 1 1 0 0 0 0 0 0 1</td>
</tr>
</tbody>
</table>

Additional clause: $w₂ \lor w₆$

One of satisfying assignments:

CNF does not imply one of these multiple-output cubes
Conclusion

The following contributions to the problem of verification are proposed:

- A case is considered when one of the compared descriptions is incompletely specified.
- It is shown how it is possible to use SAT tools for the considered case.
- An effective way of reducing the complexity and speeding up verification procedure is supposed that organizes simultaneous checking of multiple-output cubes of ISF system.

Thank you for your attention