

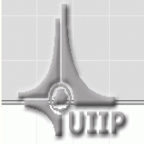


SAT-based approach to verification of logical descriptions with functional indeterminacy

**Liudmila Cheremisinova,
Dmitry Novikov**

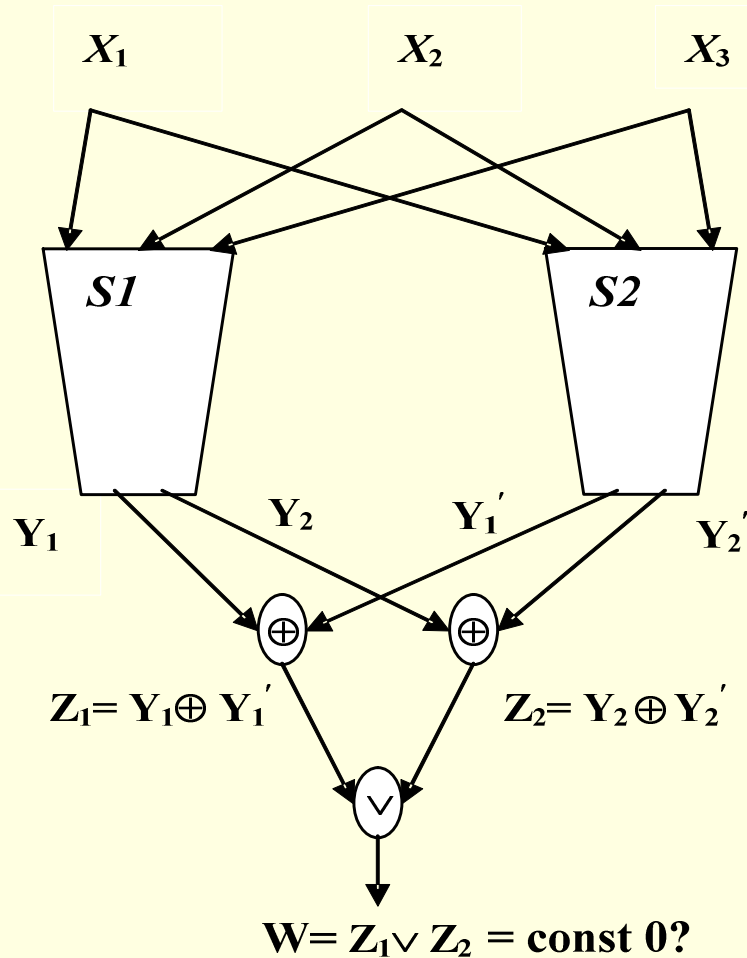
**The United Institute of Informatics Problems of
National Academy of Sciences of Belarus**

8-th International Workshop on Boolean Problems
September 18 - 19, 2008, Freiberg (Sachsen)



Typical task formulation of formal verification

Comparing circuit



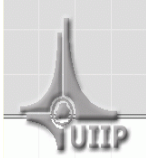
Combinational equivalence checking is verifying functional equivalence of two combinational circuits.

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

ISF system

x_1	x_2	x_3	x_4	x_5	f_1	f_2
-	-	1	1	1	1	-
1	1	-	-	-	1	0
-	0	0	0	-	0	1
0	1	-	1	0	0	0
-	0	1	0	-	-	0
-	1	-	1	1	-	1

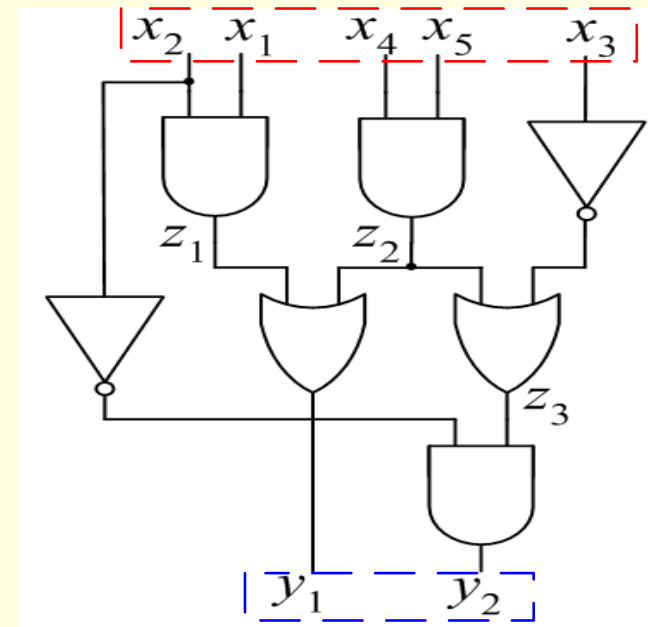
multiple-output
cubes

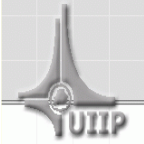
$$(u_i, t_i) \quad u_i = x_2 x_4 x_5$$

output part

$$t_i = f_2$$

Combinational circuit consisting of gates AND, OR and NOT





Simulation-based verification

System of partially defined Boolean functions

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

$U' =$	<table border="1"><tr><th>x_1</th><th>x_2</th><th>x_3</th><th>x_4</th><th>x_5</th></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td colspan="5">.....</td></tr></table>	x_1	x_2	x_3	x_4	x_5	0	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1				
x_1	x_2	x_3	x_4	x_5																											
0	0	1	1	1																											
0	1	1	1	1																											
1	0	1	1	1																											
1	1	1	1	1																											
.....																															

$U =$	<table border="1"><tr><th>x_1</th><th>x_2</th><th>x_3</th><th>x_4</th><th>x_5</th></tr><tr><td>-</td><td>-</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>-</td><td>-</td><td>-</td></tr><tr><td>-</td><td>0</td><td>0</td><td>0</td><td>-</td></tr><tr><td>0</td><td>1</td><td>-</td><td>1</td><td>0</td></tr><tr><td>-</td><td>0</td><td>1</td><td>0</td><td>-</td></tr><tr><td>-</td><td>1</td><td>-</td><td>1</td><td>1</td></tr></table>	x_1	x_2	x_3	x_4	x_5	-	-	1	1	1	1	1	-	-	-	-	0	0	0	-	0	1	-	1	0	-	0	1	0	-	-	1	-	1	1
x_1	x_2	x_3	x_4	x_5																																
-	-	1	1	1																																
1	1	-	-	-																																
-	0	0	0	-																																
0	1	-	1	0																																
-	0	1	0	-																																
-	1	-	1	1																																

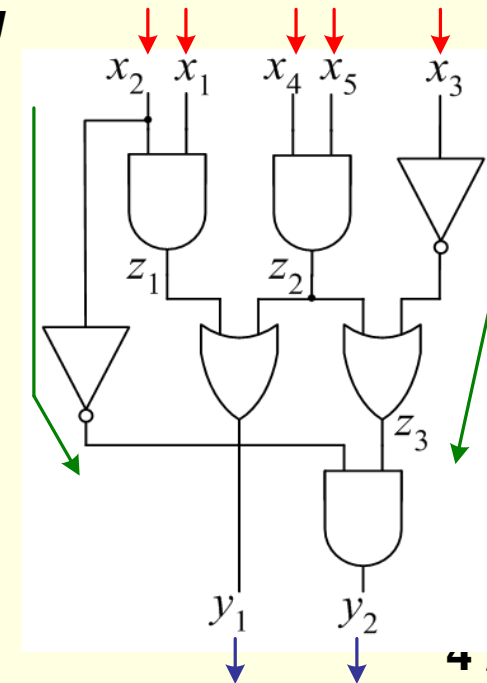
$T =$	<table border="1"><tr><th>f_1</th><th>f_2</th><th></th></tr><tr><td>1</td><td>-</td><td>1</td></tr><tr><td>1</td><td>0</td><td>2</td></tr><tr><td>0</td><td>1</td><td>3</td></tr><tr><td>0</td><td>0</td><td>4</td></tr><tr><td>-</td><td>0</td><td>5</td></tr><tr><td>-</td><td>1</td><td>6</td></tr></table>	f_1	f_2		1	-	1	1	0	2	0	1	3	0	0	4	-	0	5	-	1	6
f_1	f_2																					
1	-	1																				
1	0	2																				
0	1	3																				
0	0	4																				
-	0	5																				
-	1	6																				

The number of rows in U' for every i -th row of U :

$$l = 2^k, \text{ where } k \text{ – the number of “-” in } i\text{-th row of } U$$

- 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





SAT-based verification of logical descriptions with functional indeterminacy

Testing using SAT-solver whether the given combinational circuit implements:

Subsequent Testing:

- ❖ a **single-output** cube of the given specification;
- ❖ a **multiple-output** cube

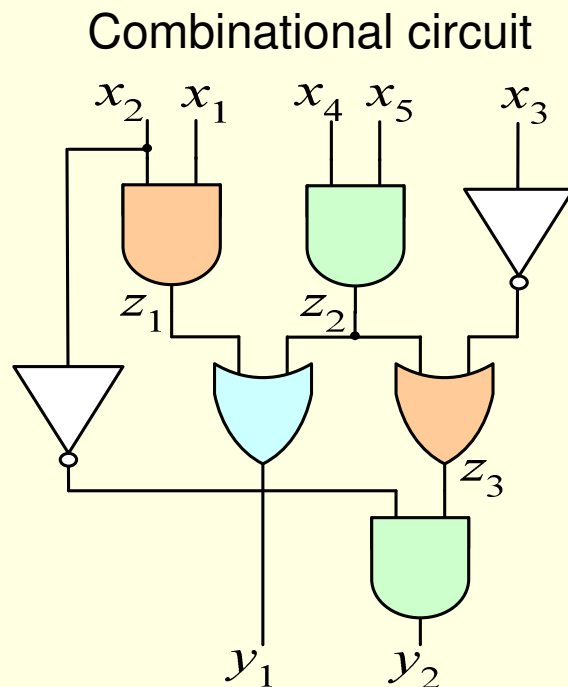
Simultaneous Testing:

- ❖ all multiple-output cubes **simultaneously** within the only SAT session.



CNF encoding of combinational circuit (conventional circuit-to-CNF transformation)

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

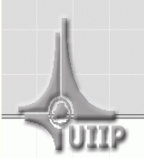


Conventional CNF of the circuit

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
—	0	—	—	—	—	—	—	—	0	13
—	—	—	—	—	—	—	1	—	0	14
—	1	—	—	—	—	—	0	—	1	15

k -input AND: $\varphi^{\wedge}(y, z_1, z_2, \dots, z_k) = (z_1 \vee \bar{y})(z_2 \vee \bar{y}) \dots (z_k \vee \bar{y})(\bar{z}_1 \vee \bar{z}_2 \vee \dots \vee \bar{z}_k \vee y)$;

k -input OR: $\varphi^{\vee}(y, z_1, z_2, \dots, z_k) = (\bar{z}_1 \vee y)(\bar{z}_2 \vee y) \dots (\bar{z}_k \vee y)(z_1 \vee z_2 \vee \dots \vee z_k \vee \bar{y})$. / 13



SAT-based model of testing multiple-output cubes of ISF system

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

$$(\mathbf{u}_i, \mathbf{t}_i) \in f(x): \mathbf{u}_i = x_{i1} x_{i2} \dots x_{ik}, \mathbf{t}_i = f_{i1} f_{i2} \dots f_{il}$$

$$\mathbf{u}_i \rightarrow \mathbf{t}_i: x_{i1} x_{i2} \dots x_{ik} \rightarrow f_{i1} f_{i2} \dots f_{il}$$

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

Or in terms of the circuit CNF : for every $(\mathbf{u}_i, \mathbf{t}_i) \in f(x)$ a partial value assignment $\mathbf{u}_i \cup \mathbf{t}_i$ of input and output variables should be satisfying assignment for the CNF:

$$\text{CNF} \rightarrow (\mathbf{u}_i \rightarrow \mathbf{t}_i)$$



Checking whether a circuit implements a single-output cube

ISF system:

X_1	X_2	X_3	X_4	X_5	f_1	f_2			
-	-	1	1	1	1	-	1		
1	1	-	-	-	1	0	2		
$U =$	-	0	0	0	-	$T =$	0	1	3
	0	1	-	1	0		0	0	4
	-	0	1	0	-		-	0	5
	-	1	-	1	1		-	1	6

(u_6, t_6)

X_1	X_2	X_3	X_4	X_5	f_1	f_2
-	1	-	1	1	-	1

Extending the conventional circuit CNF:

Property: if $x_2 = x_4 = x_5 = 1$ then $f_2 = 1$
 Formally: $(\text{CNF} \rightarrow (x_2 \wedge x_4 \wedge x_5 \rightarrow f_2)) = 1$
 $\neg \text{CNF} \vee \neg(x_2 \wedge x_4 \wedge x_5) \vee f_2 = 1$
 $\boxed{\text{CNF} \wedge x_2 \wedge x_4 \wedge x_5 \wedge \neg f_2} = 0$

extended CNF

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:

X_1	X_2	X_3	X_4	X_5	f_1	f_2
-	-	1	1	1	1	-
1	1	-	-	-	1	0
-	0	0	0	-	0	1
0	1	-	1	0	0	0
-	0	1	0	-	-	0
-	-	1	-	1	-	1

(u_6, t_6)

X_1	X_2	X_3	X_4	X_5	f_1	f_2
-	1	-	1	1	-	1

Extended conventional CNF of the circuit

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
-	0	-	-	-	-	-	-	-	0	13
-	-	-	-	-	-	-	1	-	0	14
-	1	-	-	-	-	-	0	-	1	15
-	-	-	-	1	-	-	-	-	-	16
-	-	-	1	-	-	-	-	-	-	17
-	1	-	-	-	-	-	-	-	-	18
-	-	-	-	-	-	-	-	-	0	19

Searching for a satisfying assignment proves that there exists a counter-example for (u_6, t_6) :
 1 1 - 1 1 1 1 1 0.



Checking whether the circuit implements a multiple-output cube

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

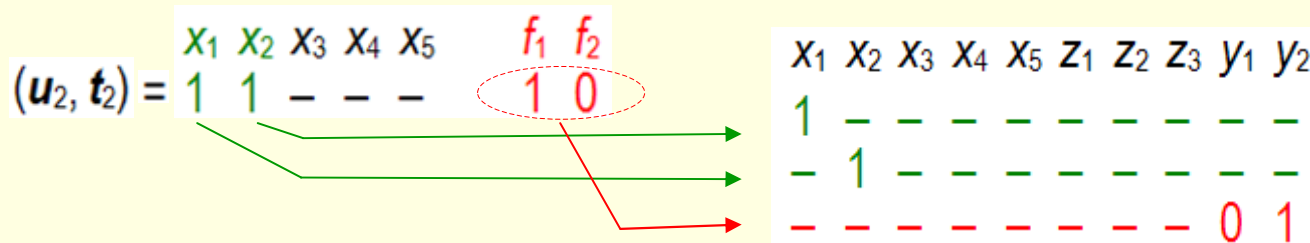
Extended CNF \rightarrow

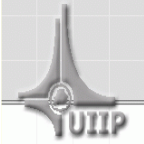
$$\begin{aligned} & \text{"(CNF} \rightarrow (u_i, t_i)) = 1" \quad (\text{CNF} \rightarrow (u_i \rightarrow t_i)) = 1 \\ & (\text{CNF} \rightarrow ((x_{i1}^{\gamma_1} \wedge x_{i2}^{\gamma_2} \wedge \dots \wedge x_{in}^{\gamma_n}) \rightarrow (y_{i1}^{\sigma_1} \wedge y_{i2}^{\sigma_2} \wedge \dots \wedge y_{ik}^{\sigma_k}))) = 1 \\ & \neg \text{CNF} \vee \neg (x_{i1}^{\gamma_1} \wedge x_{i2}^{\gamma_2} \wedge \dots \wedge x_{in}^{\gamma_n}) \vee (y_{i1}^{\sigma_1} \wedge y_{i2}^{\sigma_2} \wedge \dots \wedge y_{ik}^{\sigma_k}) = 1 \\ & \boxed{\text{CNF} \wedge x_{i1}^{\gamma_1} \wedge x_{i2}^{\gamma_2} \wedge \dots \wedge x_{in}^{\gamma_n} \wedge (y_{i1}^{\bar{\sigma}_1} \vee y_{i2}^{\bar{\sigma}_2} \vee \dots \vee y_{ik}^{\bar{\sigma}_k})} = 0 \end{aligned}$$

We add to CNF $n + 1$ clauses: n clauses of the type $x_j^{\sigma_j}$ ($x_j \in u_i$) and a clause $y_{i1}^{\bar{\sigma}_1} \vee y_{i2}^{\bar{\sigma}_2} \vee \dots \vee y_{ik}^{\bar{\sigma}_k}$ of size k

Multi-output cube

Extension of circuit CNF





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 generated by (u_i, t_i) :

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

$$\text{ext}(u_i, t_i) = x_{i1}^{\gamma_1} \wedge x_{i2}^{\gamma_2} \wedge \dots \wedge x_{in}^{\gamma_n} \wedge (y_{i1}^{\bar{\sigma}_1} \vee y_{i2}^{\bar{\sigma}_2} \vee \dots \vee y_{ik}^{\bar{\sigma}_k})$$

$$w_i \rightarrow \text{ext}(u_i, t_i) = w_i \vee \text{ext}(u_i, t_i) =$$

$$= (x_{i1}^{\gamma_1} \vee \bar{w}_i) \wedge (x_{i2}^{\gamma_2} \vee \bar{w}_i) \wedge \dots \wedge (x_{in}^{\gamma_n} \vee \bar{w}_i) \wedge (y_{i1}^{\bar{\sigma}_1} \vee y_{i2}^{\bar{\sigma}_2} \vee \dots \vee y_{ik}^{\bar{\sigma}_k} \vee \bar{w}_i) \quad (\bullet)$$

The circuit CNF is appended with:

- 1) groups of clauses (\bullet) for testing all multiple-output cubes;
 - 2) an additional clause $w_1 \vee w_2 \vee \dots \vee w_l$ 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:

$$\text{CNF} \wedge (w_1 \vee w_2 \vee \dots \vee w_l) \wedge (w_1 \rightarrow \text{ext}(u_1, t_1)) \wedge (w_2 \rightarrow \text{ext}(u_2, t_2)) \wedge \dots \wedge (w_l \rightarrow \text{ext}(u_l, t_l))$$



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