Testing and Experimental Evaluation of QBF-based Bounded Correctness Checking

Wenhui Zhang
State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences
Beijing, China

1 Introduction

Testing and experimental evaluation of the QBF-based bounded correctness checking implemented in *verbs* version 1.35[1] are reported. The evaluation is relative to the BDD-based model checking implemented in NuSMV version 2.5.0[2], based on two types of random boolean programs[3] (with different numbers of global and local variables) and a set of 24 CTL properties.

Experimental Setup The experimental evaluation is done on a Linux platform with 128 GB memory. For QBF-based bounded correctness checking, the command "verbs -QBF filename" was used (referred to as VERBS-QBF hereafter), and for BDD-based model checking, the command "NuSMV -dcx filename" was used. The option -dcx is for avoiding the generation of counter examples (the functionality for counter example generation has not been implemented in *verbs*).

2 Types of Random Programs and Types of Properties

Two sets of random programs and 24 properties are formulated for the experimental evaluation.

Programs with Concurrent Processes The parameters of the first set of random boolean programs are as follows:

a: number of processes
b: number of all variables
c: number of share variables
d: number of local variables in a process

The shared variables are initially set to a random value in $\{0, 1\}$, and the local variables are initially set to 0. For each process, the shared variables and the local variables are assigned the negation of a variable randomly chosen from these variables.

[1] http://lcs.ios.ac.cn/∼zwh/verbs/
[2] http://nusmv.irst.itc.it/
[3] http://lcs.ios.ac.cn/∼zwh/verbs/tarfiles/cpcsp.tar

Examples An example of such programs in the input format of $NuSMV$ is shown in Appendix A.1. The corresponding program in the input format of *verbs* (called *verbs* verification model, or VVM for short) is shown in Appendix A.2.

Programs with Concurrent Sequential Processes The parameters of the second set of random boolean programs are as follows, in addition to a, b, c, d specified above.

t: number of transitions in a process
p: number of parallel assignments in each transition

Besides the b boolean variables, for each process, there is a local variable representing program locations, with e possible values. The shared variables are initially set to a random value in $\{0, 1\}$, and the local variables are initially set to 0. For each transition of a process, p pairs of shared variables and local variables are randomly chosen among the shared variables and the local variables, such that the first element of such a pair is assigned the negation of the second element of the pair. Transitions are numbered from 0 to $t-1$, and are executed consecutively, and when the end of the sequence of the transitions is reached, it loops back to the execution of the transition numbered 0.

Examples An example of such programs in the input format of $NuSMV$ is shown in Appendix A.3. The corresponding program in VVM is shown in Appendix A.4.

Types of Properties 24 properties are formulated for testing. The properties involve AG, AF properties and properties specified with different combinations of the operators with one or two levels of nesting (with two levels of nesting when AX or EX is involved). Properties p_{01} to p_{12} are shown as follows, in which v_i are global variables. Properties p_{13} to p_{24} are similar to p_{01} to p_{12} with the difference of $\wedge$ and $\bigvee$ replaced by respectively $\vee$ and $\bigwedge$.

p_{01} :	$AG(\bigvee_{i=1}^{c} v_i)$
p_{02} :	$AF(\bigvee_{i=1}^{c} v_i)$
p_{03} :	$AG(v_1 \rightarrow AF(v_2 \wedge \bigvee_{i=3}^{c} v_i))$
p_{04} :	$AG(v_1 \rightarrow EF(v_2 \wedge \bigvee_{i=3}^{c} v_i))$
p_{05} :	$EG(v_1 \rightarrow AF(v_2 \wedge \bigvee_{i=3}^{c} v_i))$
p_{06} :	$EG(v_1 \rightarrow EF(v_2 \wedge \bigvee_{i=3}^{c} v_i))$
p_{07} :	$A(v_1\, U\, A(v_2\, U\, \bigvee_{i=3}^{c} v_i)$
p_{08} :	$A(v_1\, U\, E(v_2\, U\, \bigvee_{i=3}^{c} v_i)$
p_{09} :	$A(v_1\, U\, A(v_2\, R\, \bigvee_{i=3}^{c} v_i)$
p_{10} :	$A(v_1\, U\, E(v_2\, R\, \bigvee_{i=3}^{c} v_i)$
p_{11} :	$A(AXv_1\, R\, AX\, A(v_2\, U\, \bigvee_{i=3}^{c} v_i)$
p_{12} :	$A(EXv_1\, R\, EX\, E(v_2\, U\, \bigvee_{i=3}^{c} v_i)$

3 Experimental Data

Experimental Data for Programs with Concurrent Processes For this type of programs, we test different sizes of the programs with 3 processes ($a = 3$), and let b vary over the set of values $\{12, 24, 36\}$, then set $c = b/2$, $d = c/a$. Each of the 24 properties is tested on 20 test cases for each value of b. For brevity, for each type of properties, a summary of the experimental data is presented as follows (additional details are available in Appendix B), where N is the number of test cases, T is the number of test cases in which the property is true, F is the number of test cases in which the property is false, adv is the number of cases in which VERBS-QBF has an advantage.

property	adv/T	adv/F	adv/N
p_{01}	-	60/60	60/60
p_{02}	60/60	-	60/60
p_{03}	0/3	43/57	43/60
p_{04}	0/60	-	0/60
p_{05}	46/53	1/7	47/60
p_{06}	53/60	-	53/60
p_{07}	60/60	-	60/60
p_{08}	60/60	-	60/60
p_{09}	50/52	5/8	55/60
p_{10}	60/60	-	60/60
p_{11}	13/13	45/47	58/60
p_{12}	60/60	-	60/60
p_{13}	-	60/60	60/60
p_{14}	3/3	55/57	58/60
p_{15}	0/8	38/52	38/60
p_{16}	0/60	-	0/60
p_{17}	46/56	1/4	47/60
p_{18}	53/60	-	53/60
p_{19}	5/5	54/55	59/60
p_{20}	16/21	30/39	46/60
p_{21}	3/3	56/57	59/60
p_{22}	3/3	56/57	59/60
p_{23}	-	60/60	60/60
p_{24}	24/31	11/29	35/60
sum	615/791	575/649	1190/1440

This part of experimental data shows that VERBS-QBF has advantage in 1190 of 1440 test cases. Looking at the relative advantage of verification and falsification, VERBS-QBF has better advantage in the case of falsification.

Experimental Data for Programs with Concurrent Sequential Processes For this type of programs, we test different sizes of the programs with 2 processes ($a = 2$), and let b vary over the set of values $\{12, 16, 20\}$, and then set $c = b/2$, $d = c/a$,

$t = c$, and $p = 4$. Similarly, each property is tested on 20 test cases for each value of b, and a summary of the experimental data is presented as follows.

property	adv/T	adv/F	adv/N
p_{01}	0/53	2/7	2/60
p_{02}	60/60	-	60/60
p_{03}	0/10	1/50	1/60
p_{04}	0/60	-	0/60
p_{05}	0/46	0/14	0/60
p_{06}	0/60	-	0/60
p_{07}	60/60	-	60/60
p_{08}	60/60	-	60/60
p_{09}	36/54	0/6	36/60
p_{10}	53/60	-	53/60
p_{11}	33/47	0/13	33/60
p_{12}	52/60	-	52/60
p_{13}	-	60/60	60/60
p_{14}	4/4	4/56	8/60
p_{15}	0/10	1/50	1/60
p_{16}	0/60	-	0/60
p_{17}	0/48	0/12	0/60
p_{18}	0/60	-	0/60
p_{19}	8/8	52/52	60/60
p_{20}	13/18	27/42	40/60
p_{21}	4/4	54/56	58/60
p_{22}	4/4	55/56	59/60
p_{23}	-	59/60	59/60
p_{24}	8/16	29/44	37/60
sum	395/862	344/578	739/1440

This part of experimental data shows that VERBS-QBF has advantage in 739 of 1440 test cases. Looking at the relative advantage of verification and falsification, VERBS-QBF has also better advantage in falsification in this part of the evaluation.

4 Summary

Two types of programs are tested. Looking at the relative advantage of verification and falsification, VERBS-QBF has better advantage in falsification in both of the types of programs. In summary, the bounded correctness checking has advantage in more than 50 percent of the test cases, which are well distributed among verification and falsification of universal properties. In this sense, bounded correctness checking and BDD based traditional symbolic model checking may be considered complementary with their own advantages.

Note The efficiency also depends very much on the QBF-solving techniques. External QBF-solvers may be used to increase the efficiency of the verification. For ACTL formulas, special considerations are possible [2], and the use of SAT-solving techniques may be more efficient for this kind of problems.

References

1. A. Cimatti, E. M. Clarke, F. Giunchiglia, M. Roveri. NUSMV: A New Symbolic Model Verifier. CAV 1999: 495-499.
2. W. Zhang. Bounded Semantics of CTL and SAT-based Verification. Lecture Notes in Computer Science 5885 (ICFEM 2009):286-305. Springer-Verlag. 2009.

A Examples of SMV and VVM Programs

Examples of SMV programs and VVM programs are presented. The contents are as follows:

Random concurrent processes A random SMV program with 6 global variables, 3 processes and 2 local variables for each of the processes, and the corresponding VVM program;

Random concurrent sequential processes A random SMV program with 6 global variables, 2 processes and 4 local variables (of which 3 are boolean variables) for each of the processes, and the corresponding VVM program.

A.1 An SMV Program with Concurrent Processes

```
MODULE main
VAR    v1:boolean; v2:boolean; v3:boolean;
       v4:boolean; v5:boolean; v6:boolean;
       v7:boolean; v8:boolean;
       p1:process p1t(v1,v2,v3,v4,v5,v6);
       p2:process p2t(v1,v2,v3,v4,v5,v6);
ASSIGN init(v1):=1; init(v2):=0; init(v3):=1;
       init(v4):=1; init(v5):=1; init(v6):=1;
       init(v7):=0; init(v8):=0;
       next(v1):=!v5; next(v2):=!v6; next(v3):=!v4;
       next(v4):=!v4; next(v5):=!v7; next(v6):=!v3;
       next(v7):=!v1; next(v8):=!v1;
SPEC   AG((v1=1|v2=1|v3=1|v4=1|v5=1|v6=1))

MODULE p1t(v1,v2,v3,v4,v5,v6)
VAR    v9:boolean; v10:boolean;
ASSIGN init(v9):=0; init(v10):=0;
       next(v1):=!v6; next(v2):=!v3; next(v3):=!v4;
       next(v4):=!v10; next(v5):=!v3; next(v6):=!v3;
       next(v9):=!v10; next(v10):=!v1;

MODULE p2t(v1,v2,v3,v4,v5,v6)
VAR    v11:boolean; v12:boolean;
ASSIGN init(v11):=0; init(v12):=0;
       next(v1):=!v3; next(v2):=!v6; next(v3):=!v12;
       next(v4):=!v2; next(v5):=!v5; next(v6):=!v3;
       next(v11):=!v11; next(v12):=!v5;
```

A.2 A VVM Program with Concurrent Processes

```
VVM
VAR     v1:(0..1); v2:(0..1); v3:(0..1);
        v4:(0..1); v5:(0..1); v6:(0..1);
INIT    v1=1; v2=0; v3=1; v4=1; v5=1; v6=1;
PROC    p0:p0t(v1,v2,v3,v4,v5,v6);
        p1:p1t(v1,v2,v3,v4,v5,v6); p2:p2t(v1,v2,v3,v4,v5,v6);
SPEC    AG((v1=1|v2=1|v3=1|v4=1|v5=1|v6=1))

MODULE p0t(v1,v2,v3,v4,v5,v6)
VAR     v7:(0..1); v8:(0..1);
INIT    v7=0; v8=0;
TRANS   TRUE: (v1,v2,v3,v4,v5,v6,v7,v8):=
              (1-v5,1-v6,1-v4,1-v4,1-v7,1-v3,1-v1,1-v1);

MODULE p1t(v1,v2,v3,v4,v5,v6)
VAR     v9:(0..1); v10:(0..1);
INIT    v9=0; v10=0;
TRANS   TRUE: (v1,v2,v3,v4,v5,v6,v9,v10):=
              (1-v6,1-v3,1-v4,1-v10,1-v3,1-v3,1-v10,1-v1);

MODULE p2t(v1,v2,v3,v4,v5,v6)
VAR     v11:(0..1); v12:(0..1);
INIT    v11=0; v12=0;
TRANS   TRUE: (v1,v2,v3,v4,v5,v6,v11,v12):=
              (1-v3,1-v6,1-v12,1-v2,1-v5,1-v3,1-v11,1-v5);
```

A.3 An SMV Program with Concurrent Sequential Processes

```
MODULE main
VAR    v1:boolean; v2:boolean; v3:boolean;
       v4:boolean; v5:boolean; v6:boolean;
       cp:{c0,c1,c2,c3,c4,c5}; v7:boolean; v8:boolean; v9:boolean;
       p1:process p1t(v1,v2,v3,v4,v5,v6);
ASSIGN init(v1):=1; init(v2):=0; init(v3):=1;
       init(v4):=1; init(v5):=1; init(v6):=1;
       init(cp):=c0; init(v7):=0; init(v8):=0; init(v9):=0;
       next(cp):=case cp=c0:c1; cp=c1:c2;
                   cp=c2:c3; cp=c3:c4; cp=c4:c5; cp=c5:c0; esac;
       next(v1):=case cp=c0:!v9; cp=c3:!v6; cp=c5:!v8; 1:v1; esac;
       next(v2):=case cp=c0:!v4; cp=c3:!v9; 1:v2; esac;
       next(v3):=case cp=c2:!v6; cp=c3:!v8; cp=c4:!v1; 1:v3; esac;
       next(v4):=case cp=c1:!v8; cp=c2:!v2; cp=c4:!v5; 1:v4; esac;
       next(v5):=case cp=c0:!v8; cp=c1:!v2;
                   cp=c2:!v8; cp=c3:!v7; 1:v5; esac;
       next(v6):=case cp=c1:!v1; cp=c5:!v4; 1:v6; esac;
       next(v7):=case cp=c0:!v6; cp=c2:!v1;
                   cp=c4:!v6; cp=c5:!v3; 1:v7; esac;
       next(v8):=case cp=c4:!v9; 1:v8; esac;
       next(v9):=case cp=c1:!v7; cp=c5:!v2; 1:v9; esac;
SPEC   AG((v1=1|v2=1|v3=1|v4=1|v5=1|v6=1))

MODULE p1t(v1,v2,v3,v4,v5,v6)
VAR    cp:{c0,c1,c2,c3,c4,c5}; v10:boolean; v11:boolean; v12:boolean;
ASSIGN init(cp):=c0; init(v10):=0; init(v11):=0; init(v12):=0;
       next(cp):=case cp=c0:c1; cp=c1:c2;
                   cp=c2:c3; cp=c3:c4; cp=c4:c5; cp=c5:c0; esac;
       next(v1):=case cp=c0:!v4; cp=c2:!v11; cp=c5:!v11; 1:v1; esac;
       next(v2):=case cp=c4:!v3; 1:v2; esac;
       next(v3):=case cp=c0:!v2; cp=c1:!v11;
                   cp=c2:!v4; cp=c5:!v2; 1:v3; esac;
       next(v4):=case cp=c3:!v3; 1:v4; esac;
       next(v5):=case cp=c1:!v1; cp=c2:!v10; 1:v5; esac;
       next(v6):=case cp=c2:!v2; cp=c4:!v1; cp=c5:!v4; 1:v6; esac;
       next(v10):=case cp=c0:!v11; cp=c1:!v6;
                   cp=c3:!v6; cp=c4:!v5; 1:v10; esac;
       next(v11):=case cp=c3:!v5; cp=c4:!v12; 1:v11; esac;
       next(v12):=case cp=c0:!v6; cp=c1:!v2;
                   cp=c3:!v1; cp=c5:!v5; 1:v12; esac;
```

A.4 A VVM Program with Concurrent Sequential Processes

```
VVM
VAR    v1:(0..1); v2:(0..1); v3:(0..1);
       v4:(0..1); v5:(0..1); v6:(0..1);
INIT   v1=1; v2=0; v3=1; v4=1; v5=1; v6=1;
PROC   p0:p0t(v1,v2,v3,v4,v5,v6); p1:p1t(v1,v2,v3,v4,v5,v6);
SPEC   AG((v1=1|v2=1|v3=1|v4=1|v5=1|v6=1))

MODULE p0t(v1,v2,v3,v4,v5,v6)
VAR    cp:{c0,c1,c2,c3,c4,c5}; v7:(0..1); v8:(0..1); v9:(0..1);
INIT   cp=c0; v7=0; v8=0; v9=0;
TRANS  cp=c0: (v2,v7,v5,v1,cp):=(1-v4,1-v6,1-v8,1-v9,c1);
       cp=c1: (v9,v5,v6,v4,cp):=(1-v7,1-v2,1-v1,1-v8,c2);
       cp=c2: (v5,v7,v3,v4,cp):=(1-v8,1-v1,1-v6,1-v2,c3);
       cp=c3: (v2,v1,v5,v3,cp):=(1-v9,1-v6,1-v7,1-v8,c4);
       cp=c4: (v4,v3,v7,v8,cp):=(1-v5,1-v1,1-v6,1-v9,c5);
       cp=c5: (v6,v7,v9,v1,cp):=(1-v4,1-v3,1-v2,1-v8,c0);

MODULE p1t(v1,v2,v3,v4,v5,v6)
VAR    cp:{c0,c1,c2,c3,c4,c5}; v10:(0..1); v11:(0..1); v12:(0..1);
INIT   cp=c0; v10=0; v11=0; v12=0;
TRANS  cp=c0: (v1,v3,v12,v10,cp):=(1-v4,1-v2,1-v6,1-v11,c1);
       cp=c1: (v3,v12,v10,v5,cp):=(1-v11,1-v2,1-v6,1-v1,c2);
       cp=c2: (v5,v3,v1,v6,cp):=(1-v10,1-v4,1-v11,1-v2,c3);
       cp=c3: (v4,v12,v11,v10,cp):=(1-v3,1-v1,1-v5,1-v6,c4);
       cp=c4: (v2,v10,v11,v6,cp):=(1-v3,1-v5,1-v12,1-v1,c5);
       cp=c5: (v12,v1,v3,v6,cp):=(1-v5,1-v11,1-v2,1-v4,c0);
```

B Details of Experimental Evaluation

This section contain experimental evaluation based on 2 types of random boolean programs: programs with concurrent processes and programs with concurrent sequential processes. The maximum time for running VERBS-QBF is set to 300 seconds. The sign $-$ in the tables either means timeout or the data are not available.

B.1 Programs with Concurrent Processes

For the programs with concurrent processes, we let b vary over the set of values $\{12, 24, 36\}$, $a = 3$, $c = b/2$, and by locating each process with the same number of local variables, we obtain $d = (b-c)/a = c/a$. Each property is tested initially on 20 test cases for each $b \in \{12, 24, 36\}$. The details are shown in the following tables, where the left part of the table is the data obtained for model checking with $NuSMV$, using the command "NuSMV -dcx filename". The right part of the table is the data obtained for bounded correctness checking with VERBS-QBF, using the command "verbs -QBF filename".

Meaning of the Columns Let $\tau(b, i)$ be the time for verification of the i-th ($i \in \{1, ..., 20\}$) test case with b variables, and let $lt.aver$ and $rt.aver$ denote the $aver$ value in respectively the left part and the right part of the table. The explanation of the meaning of different columns of the data is as follows.

$$\begin{aligned} min(b) &= \min_{1 \leq i \leq 20} \tau(i, b) \\ min(b) &= \max_{1 \leq i \leq 20} \tau(i, b) \\ aver(b) &= \textstyle\sum_{1 \leq i \leq 20} \tau(i, b)/20 \\ ratio(b) &= lt.aver(b)/rt.aver(b) \end{aligned}$$

The $ratio$ is marked with 0.00 when the max of the right part of the table is not available (the time exceeds 300 seconds). NuSMV has advantage in cases where $ratio < 1.00$ (possibly except the case of p_{09}, due to that no maximum time was set for running NuSMV).

p_{01}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.0	0.0	3.91
24	0.4	18.4	4.3	0.0	0.0	0.0	332.79
36	27.8	986.3	295.9	0.0	0.2	0.1	5690.86

p_{02}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.0	0.0	7.87
24	0.4	15.7	3.8	0.0	0.0	0.0	1158.66
36	26.1	945.7	282.3	0.0	0.0	0.0	62037.63

p_{03}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	-	-	0.00
24	0.4	19.9	4.5	0.0	-	-	0.00
36	28.5	1226.8	335.6	0.2	-	-	0.00

p_{04}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	-	-	-	0.00
24	0.5	15.7	3.8	-	-	-	0.00
36	26.3	924.6	280.9	-	-	-	0.00

p_{05}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	-	-	0.00
24	0.4	18.9	4.2	0.0	-	-	0.00
36	26.7	1231.3	346.1	0.1	-	-	0.00

p_{06}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.0	0.0	1.09
24	0.4	16.4	4.0	0.0	0.5	0.2	25.96
36	29.3	970.8	300.5	0.1	8.5	1.7	177.54

p_{07}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.0	0.0	8.39
24	0.4	16.0	3.9	0.0	0.0	0.0	1191.78
36	25.9	982.9	297.3	0.0	0.0	0.0	57720.62

p_{08}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.0	0.0	7.67
24	0.4	16.1	3.9	0.0	0.0	0.0	1416.16
36	27.4	956.2	300.2	0.0	0.0	0.0	61900.18

p_{09}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.0	0.0	3.06
24	0.4	19.4	4.3	0.0	5.2	0.4	10.82
36	26.3	1023.8	296.4	0.0	-	-	0.00

p_{10}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.0	0.0	6.44
24	0.4	17.9	4.1	0.0	0.0	0.0	1020.89
36	29.6	1129.0	332.9	0.0	0.0	0.0	54570.15

p_{11}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.0	0.0	2.20
24	0.5	20.4	4.6	0.0	0.3	0.1	62.90
36	29.4	1038.1	326.0	0.0	3.8	0.6	546.90

p_{12}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.0	0.0	5.74
24	0.4	16.2	4.0	0.0	0.0	0.0	952.73
36	30.2	955.2	312.7	0.0	0.0	0.0	53910.67

p_{13}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.0	0.0	10.62
24	0.4	15.4	4.0	0.0	0.0	0.0	1213.17
36	26.9	923.4	285.1	0.0	0.0	0.0	58776.45

p_{14}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.0	0.0	2.10
24	0.4	16.8	4.0	0.0	0.2	0.0	81.22
36	27.2	1046.8	311.0	0.0	8.3	1.1	294.33

p_{15}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	-	-	0.00
24	0.4	20.0	4.5	0.0	-	-	0.00
36	29.9	1303.9	340.1	0.3	-	-	0.00

p_{16}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	-	-	-	0.00
24	0.4	15.9	4.0	-	-	-	0.00
36	26.9	932.9	282.9	-	-	-	0.00

p_{17}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.3	0.0	0.47
24	0.5	19.9	4.3	0.0	-	-	0.00
36	29.1	1305.9	348.2	0.1	-	-	0.00

p_{18}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.0	0.0	1.07
24	0.4	16.6	4.0	0.0	0.5	0.2	25.23
36	28.3	975.7	304.6	0.1	8.6	1.8	168.33

p_{19}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.0	0.0	3.54
24	0.4	18.3	4.2	0.0	0.0	0.0	722.72
36	31.6	1077.6	352.7	0.0	0.0	0.0	35099.05

p_{20}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.0	0.0	3.33
24	0.5	18.4	4.4	0.0	-	-	0.00
36	28.5	1014.0	321.7	0.0	-	-	0.00

p_{21}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.0	0.0	4.41
24	0.5	17.0	4.2	0.0	0.0	0.0	866.29
36	30.2	1007.1	328.9	0.0	0.0	0.0	40350.71

p_{22}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.0	0.0	4.66
24	0.5	16.2	4.0	0.0	0.0	0.0	915.27
36	28.2	1115.0	343.5	0.0	0.0	0.0	42673.31

p_{23}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.0	0.0	6.15
24	0.5	17.5	4.2	0.0	0.0	0.0	1034.65
36	31.5	1239.3	363.4	0.0	0.0	0.0	43526.82

p_{24}

b	min	max	aver	min	max	aver	ratio
12	0.0	0.0	0.0	0.0	0.1	0.0	1.74
24	0.4	17.4	4.3	0.0	-	-	0.00
36	27.6	964.1	299.9	0.4	-	-	0.00

Summary For brevity, we present a summary for $b = 36$ as follows.

p	min	max	aver	min	max	aver	ratio
p_{01}	27.8	986.3	295.9	0.0	0.2	0.1	5690.86
p_{02}	26.1	945.7	282.3	0.0	0.0	0.0	62037.63
p_{03}	28.5	1226.8	335.6	0.2	-	-	0.00
p_{04}	26.3	924.6	280.9	-	-	-	0.00
p_{05}	26.7	1231.3	346.1	0.1	-	-	0.00
p_{06}	29.3	970.8	300.5	0.1	8.5	1.7	177.54
p_{07}	25.9	982.9	297.3	0.0	0.0	0.0	57720.62
p_{08}	27.4	956.2	300.2	0.0	0.0	0.0	61900.18
p_{09}	26.3	1023.8	296.4	0.0	-	-	0.00
p_{10}	29.6	1129.0	332.9	0.0	0.0	0.0	54570.15
p_{11}	29.4	1038.1	326.0	0.0	3.8	0.6	546.90
p_{12}	30.2	955.2	312.7	0.0	0.0	0.0	53910.67
p_{13}	26.9	923.4	285.1	0.0	0.0	0.0	58776.45
p_{14}	27.2	1046.8	311.0	0.0	8.3	1.1	294.33
p_{15}	29.9	1303.9	340.1	0.3	-	-	0.00
p_{16}	26.9	932.9	282.9	-	-	-	0.00
p_{17}	29.1	1305.9	348.2	0.1	-	-	0.00
p_{18}	28.3	975.7	304.6	0.1	8.6	1.8	168.33
p_{19}	31.6	1077.6	352.7	0.0	0.0	0.0	35099.05
p_{20}	28.5	1014.0	321.7	0.0	-	-	0.00
p_{21}	30.2	1007.1	328.9	0.0	0.0	0.0	40350.71
p_{22}	28.2	1115.0	343.5	0.0	0.0	0.0	42673.31
p_{23}	31.5	1239.3	363.4	0.0	0.0	0.0	43526.82
p_{24}	27.6	964.1	299.9	0.4	-	-	0.00

Number of Cases in which VERBS-QBF has Advantages

The following tables present the data in a way that the number of cases in which VERBS-QBF has advantage and that in which NuSMV has advantage are explicitly shown, where N is the number of test cases, T is the number of test cases in which the property is true, F is the number of test cases in which the property is false, adv is the number of cases in which VERBS-QBF has an advantage.

p_{01}

b	adv/T	adv/F	adv/N
12	-	20/20	20/20
24	-	20/20	20/20
36	-	20/20	20/20
sum	-	60/60	60/60

p_{02}

b	adv/T	adv/F	adv/N
12	20/20	-	20/20
24	20/20	-	20/20
36	20/20	-	20/20
sum	60/60	-	60/60

p_{03}

b	adv/T	adv/F	adv/N
12	0/2	10/18	10/20
24	0/1	17/19	17/20
36	-	16/20	16/20
sum	0/3	43/57	43/60

p_{04}

b	adv/T	adv/F	adv/N
12	0/20	-	0/20
24	0/20	-	0/20
36	0/20	-	0/20
sum	0/60	-	0/60

p_{05}

b	adv/T	adv/F	adv/N
12	9/16	0/4	9/20
24	18/18	1/2	19/20
36	19/19	0/1	19/20
sum	46/53	1/7	47/60

p_{06}

b	adv/T	adv/F	adv/N
12	13/20	-	13/20
24	20/20	-	20/20
36	20/20	-	20/20
sum	53/60	-	53/60

p_{07}

b	adv/T	adv/F	adv/N
12	20/20	-	20/20
24	20/20	-	20/20
36	20/20	-	20/20
sum	60/60	-	60/60

p_{08}

b	adv/T	adv/F	adv/N
12	20/20	-	20/20
24	20/20	-	20/20
36	20/20	-	20/20
sum	60/60	-	60/60

p_{09}

b	adv/T	adv/F	adv/N
12	14/14	3/6	17/20
24	17/18	2/2	19/20
36	19/20	-	19/20
sum	50/52	5/8	55/60

p_{10}

b	adv/T	adv/F	adv/N
12	20/20	-	20/20
24	20/20	-	20/20
36	20/20	-	20/20
sum	60/60	-	60/60

p_{11}

b	adv/T	adv/F	adv/N
12	3/3	15/17	18/20
24	7/7	13/13	20/20
36	3/3	17/17	20/20
sum	13/13	45/47	58/60

p_{12}

b	adv/T	adv/F	adv/N
12	20/20	-	20/20
24	20/20	-	20/20
36	20/20	-	20/20
sum	60/60	-	60/60

p_{13}

b	adv/T	adv/F	adv/N
12	-	20/20	20/20
24	-	20/20	20/20
36	-	20/20	20/20
sum	-	60/60	60/60

p_{14}

b	adv/T	adv/F	adv/N
12	2/2	16/18	18/20
24	1/1	19/19	20/20
36	-	20/20	20/20
sum	3/3	55/57	58/60

p_{15}

b	adv/T	adv/F	adv/N
12	0/5	6/15	6/20
24	0/2	16/18	16/20
36	0/1	16/19	16/20
sum	0/8	38/52	38/60

p_{16}

b	adv/T	adv/F	adv/N
12	0/20	-	0/20
24	0/20	-	0/20
36	0/20	-	0/20
sum	0/60	-	0/60

p_{17}

b	adv/T	adv/F	adv/N
12	9/19	0/1	9/20
24	18/18	1/2	19/20
36	19/19	0/1	19/20
sum	46/56	1/4	47/60

p_{18}

b	adv/T	adv/F	adv/N
12	13/20	-	13/20
24	20/20	-	20/20
36	20/20	-	20/20
sum	53/60	-	53/60

p_{19}

b	adv/T	adv/F	adv/N
12	4/4	15/16	19/20
24	1/1	19/19	20/20
36	-	20/20	20/20
sum	5/5	54/55	59/60

p_{20}

b	adv/T	adv/F	adv/N
12	11/11	7/9	18/20
24	4/6	12/14	16/20
36	1/4	11/16	12/20
sum	16/21	30/39	46/60

p_{21}

b	adv/T	adv/F	adv/N
12	2/2	17/18	19/20
24	1/1	19/19	20/20
36	-	20/20	20/20
sum	3/3	56/57	59/60

p_{22}

b	adv/T	adv/F	adv/N
12	2/2	17/18	19/20
24	1/1	19/19	20/20
36	-	20/20	20/20
sum	3/3	56/57	59/60

p_{23}

b	adv/T	adv/F	adv/N
12	-	20/20	20/20
24	-	20/20	20/20
36	-	20/20	20/20
sum	-	60/60	60/60

p_{24}

b	adv/T	adv/F	adv/N
12	14/15	3/5	17/20
24	7/10	5/10	12/20
36	3/6	3/14	6/20
sum	24/31	11/29	35/60

B.2 Programs with Concurrent Sequential Processes

For the programs with concurrent sequential processes,we let b vary over the set of values $\{12, 16, 20\}$, $a = 2$, $c = b/2$, $e = c$, $f = 4$, and $d = (b - c)/a = c/a$. Each property is tested on 20 test cases for each $b \in \{12, 16, 20\}$. The details are shown in the following tables.

p_{01}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.2	0.1	-	-	-	0.00
16	0.6	7.8	2.6	0.3	-	-	0.00
20	8.9	79.1	37.4	-	-	-	0.00

p_{02}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.2	0.1	0.0	0.0	0.0	26.91
16	0.7	3.7	1.8	0.0	0.0	0.0	258.64
20	9.2	82.1	37.8	0.0	0.0	0.0	3356.16

p_{03}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.8	0.4	1.4	-	-	0.00
16	1.2	14.0	6.0	13.6	-	-	0.00
20	20.3	1028.9	256.0	-	-	-	0.00

p_{04}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.4	0.2	-	-	-	0.00
16	0.9	5.4	2.7	-	-	-	0.00
20	13.8	150.1	61.1	-	-	-	0.00

p_{05}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.7	0.3	1.0	-	-	0.00
16	1.7	29.8	8.7	129.9	-	-	0.00
20	83.9	2429.2	534.5	-	-	-	0.00

p_{06}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.4	0.2	1.0	-	-	0.00
16	1.0	7.9	3.6	59.5	-	-	0.00
20	19.4	215.4	88.5	-	-	-	0.00

p_{07}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.3	0.1	0.0	0.0	0.0	25.39
16	0.7	3.7	1.8	0.0	0.0	0.0	261.09
20	9.2	83.2	37.9	0.0	0.0	0.0	3387.83

p_{08}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.3	0.2	0.0	0.0	0.0	31.56
16	0.8	6.0	2.7	0.0	0.0	0.0	382.11
20	14.9	149.2	65.6	0.0	0.0	0.0	6070.82

p_{09}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.4	0.2	0.0	-	-	0.00
16	0.7	5.2	2.8	0.0	-	-	0.00
20	9.1	796.5	99.4	0.0	-	-	0.00

p_{10}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.6	0.2	0.0	74.0	3.8	0.06
16	1.1	9.5	4.1	0.0	40.5	2.3	1.76
20	25.6	414.4	122.7	0.0	22.6	1.3	92.22

p_{11}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.6	0.2	0.0	-	-	0.00
16	0.8	7.4	2.7	0.0	-	-	0.00
20	11.1	192.0	62.8	0.0	-	-	0.00

p_{12}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.3	0.2	0.0	-	-	0.00
16	0.9	8.1	3.4	0.0	-	-	0.00
20	21.9	239.0	102.7	0.0	-	-	0.00

p_{13}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.3	0.1	0.0	0.0	0.0	27.35
16	0.7	5.7	2.4	0.0	0.0	0.0	332.91
20	14.3	143.3	64.2	0.0	0.0	0.0	5557.25

p_{14}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.2	0.1	0.0	-	-	0.00
16	0.7	7.4	2.7	0.0	-	-	0.00
20	14.5	142.2	64.8	0.0	-	-	0.00

p_{15}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.8	0.4	1.3	-	-	0.00
16	1.3	16.0	6.5	11.2	-	-	0.00
20	20.3	1007.9	292.9	-	-	-	0.00

p_{16}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.4	0.2	-	-	-	0.00
16	1.0	5.5	2.7	-	-	-	0.00
20	14.0	151.4	61.3	-	-	-	0.00

p_{17}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.7	0.3	1.1	-	-	0.00
16	1.8	27.4	9.0	140.8	-	-	0.00
20	82.6	2959.8	601.7	-	-	-	0.00

p_{18}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.4	0.2	0.9	-	-	0.00
16	1.3	8.0	3.8	54.2	-	-	0.00
20	19.9	218.1	88.6	240.0	-	-	0.00

p_{19}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.8	0.3	0.0	0.2	0.0	7.04
16	1.6	13.9	6.1	0.0	1.6	0.4	14.64
20	41.1	743.5	205.7	0.0	11.0	1.3	163.42

p_{20}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.6	0.3	0.0	-	-	0.00
16	1.1	14.7	5.6	0.0	-	-	0.00
20	25.5	2615.4	376.0	0.0	-	-	0.00

p_{21}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.5	0.2	0.0	0.2	0.0	6.01
16	1.1	10.8	4.5	0.0	2.8	0.4	11.00
20	29.9	351.3	137.5	0.0	11.1	1.1	121.29

p_{22}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.5	0.2	0.0	0.5	0.1	4.05
16	1.0	9.8	3.9	0.0	1.8	0.3	14.39
20	25.6	285.0	101.5	0.0	7.5	0.7	138.13

p_{23}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.6	0.2	0.0	0.1	0.0	15.48
16	1.3	13.6	5.7	0.0	0.3	0.0	141.37
20	37.5	635.0	190.8	0.0	120.8	6.2	31.00

p_{24}

b	min	max	aver	min	max	aver	ratio
12	0.1	0.4	0.2	0.0	-	-	0.00
16	0.9	6.6	3.4	0.0	-	-	0.00
20	16.6	239.9	87.8	0.0	-	-	0.00

Summary For brevity, we present a summary for $b = 20$ as follows.

p	min	max	aver	min	max	aver	ratio
p_{01}	8.9	79.1	37.4	-	-	-	0.00
p_{02}	9.2	82.1	37.8	0.0	0.0	0.0	3356.16
p_{03}	20.3	1028.9	256.0	-	-	-	0.00
p_{04}	13.8	150.1	61.1	-	-	-	0.00
p_{05}	83.9	2429.2	534.5	-	-	-	0.00
p_{06}	19.4	215.4	88.5	-	-	-	0.00
p_{07}	9.2	83.2	37.9	0.0	0.0	0.0	3387.83
p_{08}	14.9	149.2	65.6	0.0	0.0	0.0	6070.82
p_{09}	9.1	796.5	99.4	0.0	-	-	0.00
p_{10}	25.6	414.4	122.7	0.0	22.6	1.3	92.22
p_{11}	11.1	192.0	62.8	0.0	-	-	0.00
p_{12}	21.9	239.0	102.7	0.0	-	-	0.00
p_{13}	14.3	143.3	64.2	0.0	0.0	0.0	5557.25
p_{14}	14.5	142.2	64.8	0.0	-	-	0.00
p_{15}	20.3	1007.9	292.9	-	-	-	0.00
p_{16}	14.0	151.4	61.3	-	-	-	0.00
p_{17}	82.6	2959.8	601.7	-	-	-	0.00
p_{18}	19.9	218.1	88.6	240.0	-	-	0.00
p_{19}	41.1	743.5	205.7	0.0	11.0	1.3	163.42
p_{20}	25.5	2615.4	376.0	0.0	-	-	0.00
p_{21}	29.9	351.3	137.5	0.0	11.1	1.1	121.29
p_{22}	25.6	285.0	101.5	0.0	7.5	0.7	138.13
p_{23}	37.5	635.0	190.8	0.0	120.8	6.2	31.00
p_{24}	16.6	239.9	87.8	0.0	-	-	0.00

Number of Cases in which VERBS-QBF has Advantages

The following tables present the data in a way that the number of cases in which VERBS-QBF has advantage and that in which NuSMV has advantage are explicitly shown.

p_{01}

b	adv/T	adv/F	adv/N
12	0/20	-	0/20
16	0/14	2/6	2/20
20	0/19	0/1	0/20
sum	0/53	2/7	2/60

p_{02}

b	adv/T	adv/F	adv/N
12	20/20	-	20/20
16	20/20	-	20/20
20	20/20	-	20/20
sum	60/60	-	60/60

p_{03}

b	adv/T	adv/F	adv/N
12	0/6	0/14	0/20
16	0/3	1/17	1/20
20	0/1	0/19	0/20
sum	0/10	1/50	1/60

p_{04}

b	adv/T	adv/F	adv/N
12	0/20	-	0/20
16	0/20	-	0/20
20	0/20	-	0/20
sum	0/60	-	0/60

p_{05}

b	adv/T	adv/F	adv/N
12	0/17	0/3	0/20
16	0/15	0/5	0/20
20	0/14	0/6	0/20
sum	0/46	0/14	0/60

p_{06}

b	adv/T	adv/F	adv/N
12	0/20	-	0/20
16	0/20	-	0/20
20	0/20	-	0/20
sum	0/60	-	0/60

p_{07}

b	adv/T	adv/F	adv/N
12	20/20	-	20/20
16	20/20	-	20/20
20	20/20	-	20/20
sum	60/60	-	60/60

p_{08}

b	adv/T	adv/F	adv/N
12	20/20	-	20/20
16	20/20	-	20/20
20	20/20	-	20/20
sum	60/60	-	60/60

p_{09}

b	adv/T	adv/F	adv/N
12	12/17	0/3	12/20
16	10/18	0/2	10/20
20	14/19	0/1	14/20
sum	36/54	0/6	36/60

p_{10}

b	adv/T	adv/F	adv/N
12	16/20	-	16/20
16	17/20	-	17/20
20	20/20	-	20/20
sum	53/60	-	53/60

p_{11}

b	adv/T	adv/F	adv/N
12	10/15	0/5	10/20
16	11/16	0/4	11/20
20	12/16	0/4	12/20
sum	33/47	0/13	33/60

p_{12}

b	adv/T	adv/F	adv/N
12	18/20	-	18/20
16	15/20	-	15/20
20	19/20	-	19/20
sum	52/60	-	52/60

p_{13}

b	adv/T	adv/F	adv/N
12	-	20/20	20/20
16	-	20/20	20/20
20	-	20/20	20/20
sum	-	60/60	60/60

p_{14}

b	adv/T	adv/F	adv/N
12	2/2	2/18	4/20
16	1/1	0/19	1/20
20	1/1	2/19	3/20
sum	4/4	4/56	8/60

p_{15}

b	adv/T	adv/F	adv/N
12	0/6	0/14	0/20
16	0/3	1/17	1/20
20	0/1	0/19	0/20
sum	0/10	1/50	1/60

p_{16}

b	adv/T	adv/F	adv/N
12	0/20	-	0/20
16	0/20	-	0/20
20	0/20	-	0/20
sum	0/60	-	0/60

p_{17}

b	adv/T	adv/F	adv/N
12	0/19	0/1	0/20
16	0/15	0/5	0/20
20	0/13	0/6	0/19
sum	0/47	0/12	0/59

p_{18}

b	adv/T	adv/F	adv/N
12	0/20	-	0/20
16	0/20	-	0/20
20	0/20	-	0/20
sum	0/60	-	0/60

p_{19}

b	adv/T	adv/F	adv/N
12	4/4	16/16	20/20
16	2/2	18/18	20/20
20	2/2	18/18	20/20
sum	8/8	52/52	60/60

p_{20}

b	adv/T	adv/F	adv/N
12	6/7	7/13	13/20
16	5/6	8/14	13/20
20	2/5	12/15	14/20
sum	13/18	27/42	40/60

p_{21}

b	adv/T	adv/F	adv/N
12	2/2	17/18	19/20
16	1/1	18/19	19/20
20	1/1	19/19	20/20
sum	4/4	54/56	58/60

p_{22}

b	adv/T	adv/F	adv/N
12	2/2	17/18	19/20
16	1/1	19/19	20/20
20	1/1	19/19	20/20
sum	4/4	55/56	59/60

p_{23}

b	adv/T	adv/F	adv/N
12	-	20/20	20/20
16	-	20/20	20/20
20	-	19/20	19/20
sum	-	59/60	59/60

p_{24}

b	adv/T	adv/F	adv/N
12	5/8	6/12	11/20
16	2/3	11/17	13/20
20	1/5	12/15	13/20
sum	8/16	29/44	37/60

B.3 Summary

The data are grouped into 12 groups, one for each type of properties where p_i and p_{i+12} are considered as of the same type. Since p_{05} and p_{06} are existential properties (and there is only one initial state in the models), we present and interpret the data according to the verification and falsification of $\neg p_{05}$ and $\neg p_{06}$. The following is a summary of the experimental data.

properties	adv/T	adv/F	adv/N
p_{01}, p_{13}	0/53	182/187	182/240
p_{02}, p_{14}	127/127	59/113	186/240
p_{03}, p_{15}	0/31	83/209	83/240
p_{04}, p_{16}	0/240	0/0	0/240
$\neg p_{05}, \neg p_{17}$	2/37	92/202	94/239
$\neg p_{06}, \neg p_{18}$	0/0	106/240	106/240
p_{07}, p_{19}	133/133	106/107	239/240
p_{08}, p_{20}	149/159	57/81	206/240
p_{09}, p_{21}	93/113	115/127	208/240
p_{20}, p_{22}	120/127	111/113	231/240
p_{21}, p_{23}	46/60	164/180	210/240
p_{21}, p_{24}	144/167	40/73	184/240
sum	814/1247	1115/1632	1929/2880

The details of the experimental data do not indicate how the sizes of problems may affect the advantage/disadvantage of bounded verification. Rather, the type of the verification problems matters.