November 2003

Tolerance of control-flow testing criteria

S. A. Vilkomir
University of Wollongong

K. Kapoor
London South Bank University

J. P. Bowen
London South Bank University

Tolerance of control-flow testing criteria

Abstract

Effectiveness of testing criteria is the ability to detect failure in a software program. We consider not only effectiveness of some testing criterion in itself but a variance of effectiveness of different test sets satisfied the same testing criterion. We name this property "tolerance" of a testing criterion and show that, for practical using a criterion, a high tolerance is as well important as high effectiveness. The results of empirical evaluation of tolerance for different criteria, types of faults and decisions are presented. As well as quite simple and well-known control-flow criteria, we study more complicated criteria: full predicate coverage, modified condition/decision coverage and reinforced condition/decision coverage criteria.

Keywords

program testing, software fault tolerance, software performance evaluation, software quality

Disciplines

Physical Sciences and Mathematics

Publication Details

This paper originally appeared as: Vilkomir, A, Kapoor, K & Bowen, JP, Tolerance of control-flow testing criteria, Proceedings 27th Annual International Computer Software and Applications Conference, 3-6 November 2003, 182-187.

Tolerance of Control-Flow Testing Criteria

Sergiy A. Vilkomir *
Decision Systems Laboratory
School of IT and Computer Science
University of Wollongong
NSW 2522, Australia
sergiy@uow.edu.au

Kalpesh Kapoor Jonathan P. Bowen
Centre for Applied Formal Methods
London South Bank University
CISM, Borough Road, London SE1 0AA, UK
www.cafm.sbu.ac.uk
{*kapoork, bowenjp*}*@sbu.ac.uk*

Abstract

Effectiveness of testing criteria is the ability to detect failures in a software program. We consider not only effectiveness of some testing criterion in itself but a variance of effectiveness of different test sets satisfied the same testing criterion. We name this property 'tolerance' of a testing criterion and show that, for practical using a criterion, a high tolerance is as well important as high effectiveness. The results of empirical evaluation of tolerance for different criteria, types of faults and decisions are presented. As well as quite simple and well-known control-flow criteria, we study more complicated criteria: Full Predicate Coverage, Modified Condition/Decision Coverage and Reinforced Condition/Decision Coverage criteria.

Keywords: software testing, testing criteria, tolerance, effectiveness, empirical evaluation, MC/DC, RC/DC.

1 Introduction

Control-flow testing criteria determine how to test logical expressions (*decisions*) in computer programs. Decisions are considered as logical functions of elementary logical predicates (*conditions*) and combinations of conditions' values are used as data for testing of decisions. For example, decision $d = A \wedge (B \vee C)$ contains three conditions A, B, and C; eight combinations of conditions' values $(0,0,0), (0,0,1), \ldots, (1,1,1)$ could be used as testing data.

In practice the number of various combinations of conditions' values could be very large and it is often impossible to test all combinations. For such, situations control-flow criteria establish various testing strategies, which allow sufficient testing coverage using a restricted number of test cases to be achieved.

Control-flow criteria are traditionally considered as program-based and useful for white-box testing [22]. However, they could be also successfully applied in black-box testing as specification-based criteria. In this case, the source of testing data and oracle results is a program specification and decisions are tested without consideration of the program code.

The evaluation of effectiveness of testing criteria has been considered in [5, 6, 9, 12, 19, 21] and other papers. Various objects have been investigated experimentally: real large programs [3], programs with restricted small volume [4] and separate logical expressions [17], for example. Real faults have been considered as well as artificially created faults of different types [16]. However, for control-flow criteria, the objects of investigation have been relatively simple and well-known criteria in the main [10]: Random Coverage (RC), Decision Coverage (DC), Condition Coverage (CC), Decision/Condition Coverage (D/CC), etc.

This paper has the following specific features:

- As well as the above mentioned simple control-flow criteria, we study more complicated criteria – Full Predicate Coverage criterion (FPC) [11], Modified Condition/Decision Coverage criterion (MC/DC) [2, 13], and a new Reinforced Condition/Decision Coverage criterion (RC/DC) [15] – that have not been studied extensively before in an experimental framework.

- The main object of our investigation is not only the effectiveness of an individual testing criterion but also the variance of effectiveness of different test sets satisfying the same testing criterion. We name this property *tolerance* of a testing criterion and show that, for practical use of a criterion, a high tolerance (low variance) is as important as high effectiveness.

*Work undertaken while at London South Bank University. The support of the UK EPSRC FORTEST Network [1] on formal methods and testing (GR/R43150/01) and helpful interaction with colleagues on this network is gratefully acknowledged. See: *www.fortest.org.uk*

This paper is structured as follows. Section 2 presents a brief review of definitions of the main control-flow testing criteria evaluated later in Section 5. In Section 3 we consider approaches for evaluating testing criteria effectiveness in general case as well as for control-flow criteria in particular. Section 4 presents the notion of testing criteria tolerance and considers differences of tolerance, citing as an example CC and RC/DC criteria. Section 5 contains the results of empirical evaluation of tolerance for six different criteria, three types of faults and sixteen decisions. General conclusions and directions for future work are addressed in Section 6.

2 Definitions of control-flow criteria

The definition of every control-flow criteria traditionally includes a statement coverage requirement as a component part: every statement in the program has been executed at least once. Because this requirement is not directly connected with the main parts of the criteria and is not important for our consideration, we omit mention of it hereafter.

We use definitions of the DC, CC, and D/CC criteria in accordance with [10]:

- DC criterion: every decision in the program has taken all possible outcomes at least once;
- CC criterion: every condition in each decision has taken all possible outcomes at least once;
- D/CC criterion: every decision in the program has taken all possible outcomes at least once and every condition in each decision has taken all possible outcomes at least once.

The requirements of these criteria are quite weak and often not sufficient for safety-critical software testing. In these cases, using of more complicated and stronger criteria (like FPC, MC/DC, and RC/DC) could be useful.

The FPC criterion was originally formulated in slightly different terms [11] but it is possible to reformulate it for the purpose of maintaining uniformity:

> *FPC criterion: each condition in a decision has taken all possible outcomes where the value of a decision is directly correlated with the value of a condition.*

This means that a decision changes every time a condition changes. The difference between D/CC and FPC is that, for D/CC, a test set could contain only two test cases with different outcomes of a decision and test cases with different values for each condition could be chosen irrespective of the values of a decision. For FPC, test cases, chosen for testing a condition, should provide different outcomes for the decision at the same time.

MC/DC [2, 13] is stronger than FPC and contains additional requirements for each pair of test cases chosen for testing a condition:

> *MC/DC criterion: every condition in a decision in the program has taken on all possible outcomes at least once, every decision in the program has taken all possible outcomes at least once, and each condition in a decision has been shown to independently affect the decision's outcome. A condition is shown to independently affect a decision's outcome by varying just that condition while holding fixed all other possible conditions.*

In the RTCA/DO-178B standard [13], where the MC/DC criterion has been firstly proposed, multiple occurrences of a condition in a decision were considered as different conditions; this creates some problems during the practical use of MC/DC. In this paper we consider every condition only once, since this seems more natural, and we consider a decision as a function of conditions. This approach has been reflected in the formal definitions of MC/DC using the Z notation [14, 15].

RC/DC [15] contains MC/DC as a part of its requirements and mandates additional test cases with a view to considering all safety-critical situations:

> *RC/DC criterion: ... each condition in a decision has been shown to independently affect the decision's outcome, and each condition in a decision has been shown to independently keep the decision's outcome. A condition is shown to independently affect and keep a decision's outcome by varying just that condition while holding fixed (if it is possible) all other conditions.*

3 Effectiveness of testing criteria

3.1 General approach

Effectiveness of testing criteria is usually understood as *the ability to detect failures in a software program.* When we consider one specific test case for a particular program, containing one or more failures, only two possibilities exist: either this test case detects a failure or it does not. Thus, effectiveness of one specific test case equals either 1 (100%) or 0. A test set (a set of several test cases) detects a failure when at least one test case from this set detects a failure. So effectiveness of one specific test set also equals either 1 or 0.

However, it is interesting to consider some more generalized measures of effectiveness. This generalization could be carried out in several directions:

- Effectiveness of testing strategies (average effectiveness of test sets satisfying a specific test criterion);
- Effectiveness in detecting faults of some specific type;
- Effectiveness averaged for different programs.

These approaches have been considered in many papers. Thus, the following measures have been suggested for the evaluation of effectiveness of a subdomain-based criterion C (which divides the input domain into subdomains and requires the selection of one test case or set from each subdomain) for a specific program P and specification S [18]:

$$M(C,P,S) = 1 - \prod_{i=1}^{n}(1 - \frac{m_i}{d_i}) \quad (1)$$

$$E(C,P,S) = \sum_{i=1}^{n} \frac{m_i}{d_i} \quad (2)$$

where $d_i = |D_i|$ – size of subdomain D_i, m_i – number of inputs in D_i, which detect the failure.

However, as was pointed out in [18], the information required for measures M and E is typically not available. Besides that, these measures are considered for a specific program with specific faults. But for practical use it is desirable to have general measures of effectiveness for the comparison of different testing criteria before testing. Further generalization requires understanding of a typical program and typical faults, which is not normally possible in the general case.

3.2 Effectiveness of control-flow criteria

We now consider using the approaches mentioned in Section 3.1 for the more specific case of control-flow criteria. It is possible to regard these criteria as subdomain-based, where each domain is formed by test sets for testing one specific decision in a program. But it is often convenient to consider using control-flow criteria separately for each decision. In this case, the specification S is a correct version of this decision and P is a concrete program realization of this decision which may contains faults. In this situation, the values of M and E (see formulas (1) and (2)) are equal: $E(C,P,S) = M(C,P,S) = m_p/d_s$, where d_s is the number of test sets satisfying a criterion C and m_p is the number of test sets from d_s which detect faults in P.

Moving on to general measures of control-flow criteria effectiveness, consider effectiveness for different faults. Above all, notice that it is practically impossible to consider effectiveness for *all* possible faults. There are 2^{2^n} possible realizations of a decision containing n conditions. One of these realizations is correct (i.e., coincides with the specification S) and we can consider all others as containing faults. If we consider all these faults as possible and equally probable, no strategy of choosing test cases could give an advantage and effectiveness of a testing criterion depends only on the required size of a test set[1]. So consideration of effectiveness of testing criteria can be done for specific types of faults that are typical in practice.

Various typical types of faults in decisions have been considered; see for example [7,17]. We study some of them in Section 5 but here let us consider any given type of fault F. Let k be the number of all possible faults of type F for the decision's specification S, or, equivalently, the number of different realizations p_i of specification S, for which the difference between p_i and S relates to type F. Then the effectiveness of criterion C for faults of type F is

$$E(C,F,S) = \frac{1}{k}\sum_{i=1}^{k}\frac{m_{p_i}}{d_s} = \frac{1}{kd_s}\sum_{i=1}^{k} m_{p_i} \quad (3)$$

Thus, first we find the effectiveness of a criterion for each concrete fault and next we find the average effectiveness for all faults of the given type. It is possible to find the same effectiveness by calculating it in the reversed order: first find the effectiveness of each concrete test set for all faults of the given type and then find the average effectiveness for all possible test sets satisfying a criterion. In more detail, if m_{t_i} faults are detected by using one specific test set t_i for k realizations of the decision with faults of type F, then $E_{t_i}(C,F,S) = m_{t_i}/k$ is the effectiveness of test set t_i and the effectiveness of criterion C is determined as

$$E(C,F,S) = \frac{1}{d_s}\sum_{i=1}^{d_s} E_{t_i}(C,F,S) = \frac{1}{kd_s}\sum_{i=1}^{d_s} m_{t_i} \quad (4)$$

The values of effectiveness calculated by formulas (3) and (4) are equal.

The effectiveness $E_{t_i}(C,F,S)$ of one test set t_i is interesting in itself. When effectiveness of individual test sets is considered for all test sets satisfying the same criterion, the following question naturally emerges: to what extent does effectiveness vary for different test sets? The character of this variation could differ for various criteria. Test sets can have significant variance of effectiveness for some testing criteria and small variance for others. We name this property *tolerance* of a testing criterion and study it below in Sections 4 and 5.

[1] Mathematical reasoning about this is given in [20] in the context of the MC/DC criterion

4 Tolerance of testing criteria

4.1 Definition of tolerance

The application of some testing criterion for practical testing presupposes that it is sufficient to use any one test set that satisfies the given criterion. Knowledge of the effectiveness of a criterion cannot be enough for prediction of the effectiveness of this one specific test set. This effectiveness also depends on the distribution of effectiveness for all test sets satisfying a criterion.

We define tolerance of a testing criterion as *the ability of every test set satisfying this criterion to provide a similar level of effectiveness*. For criteria with *high tolerance*, effectiveness of separate test sets does not vary much and is sufficiently close to the average effectiveness. For criteria with *low tolerance* effectiveness of separate test sets can vary significantly. So in the latter case high average effectiveness does not guarantee the same effectiveness of the chosen test set. For example, let some hypothetical criterion have for some type of fault's effectiveness equal to 0.5 but effectiveness of separate test sets have a uniform distribution. Then it is not possible to predict real effectiveness of testing since it could as well be high as low with equal probability. So it is expedient to use testing criteria not only with high effectiveness but also with high tolerance.

One of possible measures of tolerance $T(C, F, S)$ is the standard deviation of a distribution of effectiveness, where

$$T^2(C, F, S) = \frac{1}{d_s} \sum_{i=1}^{d_s} (E_{t_i}(C, F, S) - E(C, F, S))^2 \quad (5)$$

According the above, if $T(C_1, F, S) < T(C_2, F, S)$ then criterion C_1 has a higher tolerance than criterion C_2. An example in the next section shows how significant the difference of tolerance for various criteria can be in practice.

4.2 An example

Consider tolerance of two control-flow criteria, CC and RC/DC, for the following decision s, containing eight conditions denoted by capital letters from A to H:

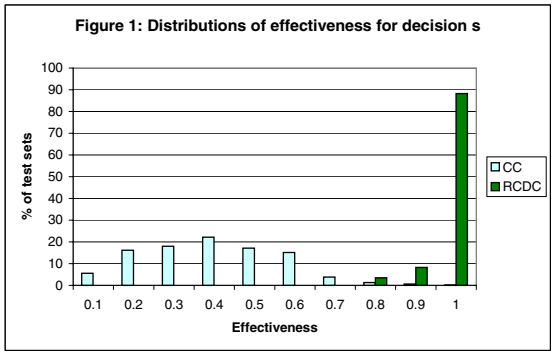
$$\neg(A \wedge B) \wedge (D \wedge \neg E \wedge \neg F \vee \neg D \wedge E \wedge \neg F \vee \neg D \wedge \neg E \wedge \neg F) \wedge ((A \wedge C \wedge (D \vee E) \wedge H \vee A \wedge (D \vee E) \wedge \neg H) \vee B \wedge (E \vee F))$$

This decision is one of the Boolean expressions studied in [12, 17] which were originally the specifications of the Traffic Alert and Collision Avoidance System, TCAS II [8].

Consider, for example, effectiveness and tolerance in relation to *Operator Reference Faults* (ORF) [7, 17], when one Boolean operator is mistakenly replaced with another, in this case, operator '$\wedge$' replaced operator '$\vee$' and vice versa. We have generated all possible faulty decisions with this type of fault (total number $k = 22$) and test sets for CC (total number $d_s = 1500$) and RC/DC (total number $d_s = 2000000$). The use of all these test sets for testing all faulty decisions shows the following effectiveness of testing criteria:

$$E(CC, ORF, s) = 0.34; \quad E(RC/DC, ORF, s) = 0.92$$

The effectiveness of RC/DC is significantly higher partly because the average size of a test set for RC/DC is bigger than for CC. But our aim is not proper comparison of effectiveness (although that is an important separate task) but comparison of tolerance of criteria. Experimental data, which describe the distribution of effectiveness of test sets, is shown in Figure 1:



The spread of effectiveness for CC is significantly bigger than for RC/DC. The calculations by formula (5) give the following values of tolerance

$$T(CC, ORF, s) = 0.17; \quad T(RC/DC, ORF, s) = 0.05$$

So tolerance of RC/DC is more than three times higher. It signifies that, in contrast to CC, the value of average effectiveness of RC/DC characterizes effectiveness of testing quite well even when only one test set is used.

5 Results of empirical evaluation of tolerance

Empirical data, considered in this section, reflect the results for the first stage of our experimental evaluation and do not pretend to be a complete investigation. The main aim of this presentation is to give a feeling for the variance of tolerance in realtion to different testing criteria and to outline the main directions for further evaluation.

We analyzed 16 different decisions which, as for the decision s from the example in Section 4.2, are specifications from the TCAS II System. The list of these decisions is available in [12, 17], where they were used for comparison of different testing strategies. The tolerance of six control-flow criteria (see the definitions in Section 2) were considered for three different types of faults: ORF (see Section

	DC	CC	D/CC	FPC	MC/DC	RC/DC
ORF	31.63	35.25	46.44	46.50	94.31	95.38
ENF	64.13	50.56	71.69	71.81	98.88	99.13
VNF	39.88	24.38	47.81	47.94	94.94	96.00
Average	45.21	36.73	55.31	55.42	96.04	96.83

Table 1. Effectiveness of control-flow criteria (x 100 %).

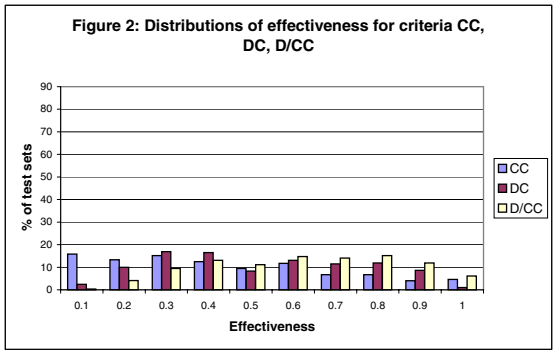
	DC	CC	DCC	FPC	MCDC	RCDC
ORF	16.47	22.09	19.28	19.38	6.17	5.44
ENF	19.16	27.17	17.98	17.97	4.40	4.43
VNF	20.46	23.26	20.56	20.69	6.72	5.57
Average	18.70	24.17	19.27	19.35	5.76	5.14

Table 2. Tolerance of control-flow criteria (x 100 %).

4.2 above), Variable Negation Faults (VNF), and Expression Negation Faults (ENF). A VNF type fault replaces one occurrence of a variable by its negation and an ENF type fault replaces an expression by its negation. These types of faults were considered in [17] for comparison of different testing strategies and in [7], where the hierarchy of fault classes was studied.

The effectiveness of the criteria for each type of fault separately and on average is shown in Table 1. These data show that effectiveness of MC/DC and RC/DC is significantly higher than the effectiveness of other criteria. But, as in the case of the example from Section 4.2, we need to notice that the size of test sets for MC/DC and RC/DC is bigger than for other criteria. This fact should be taken into account when choosing a criterion for practical use.

The distribution of effectiveness of test sets for CC, DC and D/CC is shown in Figure 2.



All these criteria have a large range of effectiveness of separate test sets and therefore have a low tolerance. This fact casts doubt on the practicability these criteria for testing safety-critical systems.

The distribution of the effectiveness of test sets for FPC, MC/DC and RC/DC is shown in Figure 3. The range of effectiveness for FPC is similar to CC, DC and D/CC, indicating the low tolerance of FPC. At the same time, the dispersion of effectiveness of test sets for MC/DC and RC/DC is very low, demonstrating the high tolerance of these two criteria.

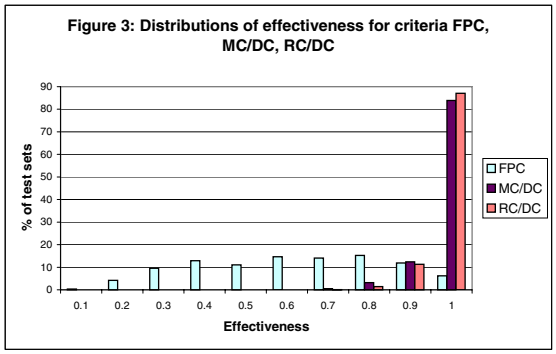


Table 2 gives numerical values of tolerance of all the studied criteria for each fault type and on average. These data show that DC, CC, D/CC and FPC have a similar (quite low) level of tolerance and this level has no significant differences for the various types of faults. At the same time MC/DC and RC/DC have very high level of tolerance, which guarantees a stable effectiveness when they are used.

6 Conclusion

In this paper we have addressed the different aspects of effectiveness of software testing criteria and in particular we have considered the effectiveness of one separate test set relative to a specific type of fault. The main part of the paper

introduced a new concept of *tolerance of a testing criterion* that characterizes the ability of every test set, satisfying this criterion, to provide a similar level of effectiveness. A high level of tolerance guarantees a stable effectiveness during use of a criterion.

Our preliminary empirical evaluation shows the low level of tolerance for such criteria as DC, CC, D/CC, and FPC and, in contrast, the high level of tolerance for MC/DC and RC/DC. Because both criteria have also a high effectiveness it may be expedient to use MC/DC and RC/DC for practical software testing, especially in high integrity systems.

References

[1] Bowen, J. P., Bogdanov, K., Clark, J., Harman, M., Hierons, R., Krause, P. FORTEST: Formal methods and testing. *Proceedings of 26th Annual International Computer Software and Applications Conference (COMPSAC 02)*, Oxford, UK, August 26–29, 2002, IEEE Computer Society Press, pp. 91–101.

[2] Chilenski, J., Miller, S. Applicability of Modified Condition/Decision Coverage to software testing. *Software Engineering Journal*, September 1994, pp. 193–200.

[3] Frankl, P., Iakounenko, O. Further empirical studies of test effectiveness. *ACM SIGSOFT Software Engineering Notes, Proceedings of ACM SIGSOFT 6th International Symposium on Foundations of Software Engineering*, November 1998, Vol. 23, No. 6, pp. 153–162.

[4] Frankl, P., Weiss, S. An experimental comparison of the effectiveness of branch testing and data flow testing. *IEEE Transactions on Software Engineering*, Vol. 19, No. 8, August 1993, pp. 774–787.

[5] Frankl, P., Weyuker E. A formal analysis of the fault-detecting ability of testing methods. *IEEE Transactions on Software Engineering*, Vol. 19, No. 3, March 1993, pp. 202–213.

[6] Hutchins, M., Foster, H., Goradia, T., Ostrand, T. Experiments on the effectiveness of dataflow- and control-flow-based test adequacy criteria. *Proceedings of 16th International Conference on Software Engineering (ICSE-16)*, 1994, pp. 191–200.

[7] Kuhn, D. Fault classes and error detection capability of specification-based testing. *ACM Transactions on Software Engineering and Methodology*, Vol. 8, No. 4, October 1999, pp. 411–424.

[8] Leveson, N. G., Heimdahl, M. P. E., Hildreth, H., Reese, J. D. Requirements specification for process-control systems. *IEEE Transactions on Software Engineering*, Vol. 20, No. 9, September 1994, pp. 684–707.

[9] Ntafos, S. C. On comparisons of random, partition, and proportional partition testing. *IEEE Transactions on Software Engineering*, Vol. 27, No. 10, October 2001, pp. 949–960.

[10] Myers, G. *The Art of Software Testing.* Wiley-Interscience, 1979.

[11] Offutt, A. J., Xiong, Y., Liu, S. Criteria for generating specification-based tests. *Proceedings of 5th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'99)*, Las Vegas, Nevada, USA, October 18–21, 1999, IEEE Computer Society Press, pp. 119–129.

[12] Paradkar, A., Tai K. Test generation for Boolean expressions. *Proceedings of 6th International Symposium on Software Reliability Engineering (ISSRE'95)*, Toulouse, France, 1995, pp. 106–115.

[13] RTCA/DO-178B. *Software Considerations in Airborne Systems and Equipment Certification*, RTCA, Washington D.C., USA, 1992.

[14] Vilkomir, S. A., Bowen, J. P. Formalization of software testing criteria using the Z notation, *Proceedings of 25th IEEE Annual International Computer Software and Applications Conference (COMPSAC 01)*, Chicago, Illinois, USA, October 8–12, 2001, IEEE Computer Society Press, pp. 351–356.

[15] Vilkomir, S. A., Bowen, J. P. Reinforced Condition/Decision Coverage (RC/DC): A new criterion for software testing. In Bert, D., Bowen, J. P., Henson, M. C., Robinson, K. (eds.), *ZB2002: Formal Specification and Development in Z and B, Proceedings of 2nd International Conference of B and Z Users*, Grenoble, France, January 23–25, 2002, Springer-Verlag, LNCS 2272, pp. 295–313.

[16] Vouk, M., Tai, K. C., Paradkar, A. Empirical studies of predicate-based software testing. *Proceedings of 5th International Symposium on Software Reliability Engineering*, 1994, pp. 55–64.

[17] Weyuker, E., Goradia, T., Singh, A. Automatically generating test data from a Boolean specification. *IEEE Transactions on Software Engineering*, Vol. 20, No. 5, May 1994, pp. 353–363.

[18] Weyuker, E. Can we measure software testing effectiveness? *Proceedings of 1st International Software Metrics Symposium*, Baltimore, USA, May 21–22, 1993, pp. 100–107.

[19] Weyuker, E. Thinking formally about testing without a formal specification. *Proceedings of Formal Approaches to Testing of Software (FATES'02)*, A Satellite Workshop of CONCUR'02, Brno, Czech Republic, August 24, 2002, pp. 1–10.

[20] White, A. Comments on Modified Condition/Decision Coverage for software testing. *2001 IEEE Aerospace Conference Proceedings*, Vol. 6, Big Sky, Montana, USA, March 10–17, 2001, IEEE Computer Society Press, pp. 2821–2828.

[21] Wong, W., Horgan, J., Mathur, A., Pasquini, A. Test set size minimization and fault detection effectiveness: A case study in a space application. *Proceedings of 21st Annual International Computer Software and Applications Conference (COMPSAC 97)*, Washington, DC, USA, August 13–15, 1997, IEEE Computer Society Press, pp. 522–528.

[22] Zhu, H., Hall P. A., and May, H. R. Software unit test coverage and adequacy. *ACM Computing Surveys*, Vol. 29, No. 4, December 1997, pp. 336–427.