On Comparing Testing Criteria for Logical Decisions

Man Fai Lau¹ and Yuen Tak Yu²,⋆⋆

¹ Faculty of Information and Communication Technologies, Swinburne University of Technology Hawthorn, Victoria, AUSTRALIA, 3122 elau@swin.edu.au
² Department of Computer Science, City University of Hong Kong Kowloon Tong, HONG KONG csytyu@cityu.edu.hk

Abstract. Various test case selection criteria have been proposed for quality testing of software. It is a common phenomenon that test sets satisfying different criteria have different sizes and fault-detecting ability. Moreover, test sets that satisfy a stronger criterion and detect more faults usually consist of more test cases. A question that often puzzles software testing professionals and researchers is: when a testing criterion $C_1$ helps to detect more faults than another criterion $C_2$, is it because $C_1$ specifically requires test cases that are more fault-sensitive than those for $C_2$, or is it essentially because $C_1$ requires more test cases than $C_2$? In this paper, we discuss several methods and approaches for investigating this question, and empirically compare several common coverage criteria for testing logical decisions, taking into consideration the different sizes of the test sets required by these criteria. Our results demonstrate evidently that the stronger criteria under study are more fault-sensitive than the weaker ones, not solely because the former require more test cases. More importantly, we have illustrated a general approach, which takes into account the size factor of the generated test sets, for demonstrating the superiority of a testing criterion over another.

Keywords: Boolean expression, condition coverage, condition/decision coverage, control flow criteria, decision coverage, modified condition/decision coverage (MC/DC), software testing.

1 Introduction

Various test case selection criteria have been proposed in the literature, and some criteria are known to be more stringent than others. For example, basis path

* The work described in this paper was supported in part by a grant from the Australian Research Council, Australia (project id. DP0558597), and in part by a grant from the Research Grants Council of Hong Kong Special Administrative Region, China (project no. 123206).

** Corresponding author.
coverage \cite{1, 2} (that is, coverage of all basis paths) is known to be more stringent than branch coverage. In choosing a testing criterion, two competing factors are often considered. A primary consideration is the fault-detecting ability of the test sets selected by the criterion. Another important consideration is the number of test cases required. For example, the basis path coverage criterion exercises a wider variety of control flows of the program under test as compared to the branch coverage criterion, and hence the former is often thought to be more fault-sensitive than the latter. However, the basis path coverage criterion often requires many more test cases. Thus, comparing the fault-detecting ability of two testing criteria without considering the number of test cases required would not be adequate.

A common way of comparing the effectiveness of test case selection criteria is to compare the percentage of (non-equivalent) mutants that can be detected by the test sets that satisfy each criterion. A mutant of a program is another program which differs from the original one by some syntactic changes. Given two test case selection criteria \(C_1\) and \(C_2\), \(C_1\) is considered to be more fault-sensitive than criterion \(C_2\) if \(C_1\) requires test sets that detect a higher percentage of mutants than \(C_2\). However, such a comparison does not take into consideration the factor of size of the test sets required by the two criteria. Of course, if the size of the test sets required by \(C_1\) is smaller than that of \(C_2\), we can be sure that the higher fault detecting ability of \(C_1\) is due to the characteristics of the test cases required to satisfy \(C_1\). Otherwise, if the size of the test sets required by \(C_1\) is larger than that of \(C_2\), then the question is, “Does \(C_1\) help to detect more faults because it specifically requires test cases that are more fault-sensitive than those for \(C_2\), or is it essentially because \(C_1\) requires more test cases than \(C_2\) does?”

In this paper, we shall discuss several methods for investigating the above question (Section 2), compare several common coverage criteria for testing logical decisions (Section 3) by means of an empirical study (Section 4), and finally conclude with a brief summary of our findings (Section 5).

### 2 Methods for Comparing Testing Criteria

To compare two testing criteria \(C_1\) and \(C_2\), where \(C_1\) requires larger test sets than those for \(C_2\), there are several general approaches to bring the two test sets to a common ground for fair comparison:

1. Compare the “efficiency" of the two criteria \(C_1\) and \(C_2\), defined as the “effectiveness per test case” \cite{2}. Despite its apparent intuitive appeal, such a comparison misses the important goal in testing, which is to discover as many faults/failures as possible \cite{4}. For example, suppose that a software contains 5 faults. Suppose further that \(C_1\) requires only 2 test cases and can only detect one fault (hence its “efficiency” is 0.5 fault per test case), whereas \(C_2\) requires 20 test cases but can detect all 5 faults in the software (hence its “efficiency” is 0.25 fault per test case). Then even though the latter is apparently “less efficient”, it is actually much more useful than the former. In other words, from the quality assurance point of view, fault