Computer Languages, Systems & Structures 39 (2013) 95–108

Contents lists available at ScienceDirect

Computer Languages, Systems & Structures

journal homepage: www.elsevier.com/locate/cl

A test-suite reduction approach to improving fault-localization effectiveness

Gong Dandan *, Wang Tiantian, Su Xiaohong, Ma Peijun

School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China

A R T I C L E I N F O

Article history:
Received 14 February 2012
Received in revised form
14 March 2013
Accepted 6 April 2013
Available online 18 April 2013

Keywords:
Software debugging
Fault localization
Test-suite reduction

A B S T R A C T

In order to improve the effectiveness of fault localization, researchers are interested in test-suite reduction to provide suitable test-suite inputs. Different test-suite reduction approaches have been proposed. However, the results are usually not ideal. Reducing the test-suite improperly or excessively can even negatively affect fault-localization effectiveness. In this paper, we propose a two-step test-suite reduction approach to remove the test cases which have little or no effect on fault localization, and improve the distribution evenness of concrete execution paths of test cases. This approach consists of coverage matrix based reduction and path vector based reduction, so it analyzes not only the test cases coverage but also the concrete path information. We design and implement experiments to verify the effect of our approach. The experimental results show that our reduced test-suite can improve fault-localization effectiveness. On average, our approach can reduce the size of a test-suite in 47.87% (for Siemens programs) and 23.03% (for space program). At the same time, on average our approach can improve the fault-localization effectiveness, 2.12 on Siemens programs and 0.13 on space program by Tarantula approach.

1. Introduction

Software debugging, which includes locating and correcting faulty program statements, is an important and expensive activity in the software development and maintenance [1,12]. Traditionally, fault localization is a manual process which is tedious and error-prone. In recent years, there have been considerable researches on automated fault localization techniques [3–24], based on the execution information of both passed and failed test cases. In order to improve the effectiveness of fault localization, some test cases are deleted from test-suite to provide suitable test cases inputs. Thus, the effect of test-suite reduction on fault-localization effectiveness has been widely studied.

In the previous research work, some researchers focus on proposing different approaches to reduce test cases, and then investigating how the reduced test-suite affect the fault-localization effectiveness according to their own experimental artifacts. Abreu et al. [25] and Baudry et al. [26] investigated the relationship between the number of the test cases and fault-localization effectiveness. Zhang et al. [27] found that the effectiveness of the fault-localization technique increases with the distribution evenness of execution traces of test cases. Chen et al. [28] proposed a lightweight test-suite reduction approach based on the conjecture of the covering interaction of requirements to detect more faults. In all the above approaches, researchers only analyze coverage information with respect to each test case, and they use their own experimental artifacts to verify the effect of test-suite reduction on fault-localization effectiveness, but there is not a

* Corresponding author. Tel./fax: +86 451 86412824.
E-mail address: gongdandan0418@126.com (G. Dandan).

1477-8424/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cl.2013.04.001

consensus on how the removal of test cases from the test-suite can affect the fault localization. Hao et al. [29,30] proposed several statement-based reduction strategies and assumed that test cases redundancy or similarity can negatively affect fault-localization effectiveness. Yu et al. [31] presented vector-based reduction techniques and pointed out their strategies are more effective than the statement-based reduction technique. They found that statement-based reduction [29,30] techniques negatively affect the fault-localization effectiveness, and vector-based reduction [31] strategy provides negligible impact on the effectiveness of fault localization.

Different from the previous research work, in this paper, we propose a two-step test-suite reduction technique. Traditional approaches [26–31] only consider the coverage information of test cases. The key intuition of our approach is that it analyzes both the test cases coverage and the concrete path information. Faults usually exist in the execution paths of failed test cases, so the statements in the paths of failed test cases may be more helpful in localizing faults. By contrast, the statements which exist in the paths of all test cases (including passed test cases and failed test cases) have little effect on fault localization. In addition, though researchers have pointed out that improving the distribution evenness of execution traces of test cases benefits fault-localization effectiveness, they only analyzed coverage information rather than concrete path information. Based on the above analysis, fault-localization requirements are proposed to emphasize the statements which are only executed by failed test cases. We propose a coverage matrix based reduction approach to remove the test cases which are not relevant to fault-localization requirements, and a path vector based reduction approach to improve the distribution evenness of concrete execution paths of test cases.

The main contributions of this paper include:

- We propose the concepts of coverage matrix based reduction and path vector based reduction for fault localization.
- We propose a new test-suite reduction approach, which analyzes both the test cases coverage and the concrete path information.
- We design and implement experiments to verify the effect of our approach, and the experimental results show that our approach can improve the effectiveness of fault localization.

2. Two-step test-suite reduction

The framework of our technique is shown in Fig. 1. A program instrumentor is implemented via Yacc [2]. It records both coverage vector and path vector simultaneously for every test case. The work is divided into two parts in Fig. 1 as follows: (1) The statements only executed by failed test cases are more likely to be faulty, so these statements should be paid attention to. The coverage matrix is established based on fault-localization requirements, and then the passed test cases weakly relevant to the fault-localization requirements are deleted based on the coverage matrix. We call this coverage matrix based reduction (see upper part of Fig. 1). The aim of coverage matrix based reduction is to emphasize the fault-localization requirements on fault localization. (2) The repeat times of loops decrease the distribution evenness of execution paths, and thus affect the effectiveness of fault localization. For the test cases with identical coverage vector, the corresponding path vectors are extracted, and the redundant test cases with identical path vectors are deleted. Next, loop standardization is executed and the test cases with similar paths are deleted. We call this path vector based reduction (see lower part of Fig. 1). The aim of path vector based reduction approach is to increase the distribution evenness of execution paths.

Definition 1. (*Execution path*): We consider each execution path of a program P to be a sequence of statements $S=\langle s_1,s_2,\ldots,s_i,\ldots\rangle$. In this paper, the S executed by test case t is denoted as PATH(t). It is worthwhile to note that the statement s_i may occur several times if it is in the loop structure.

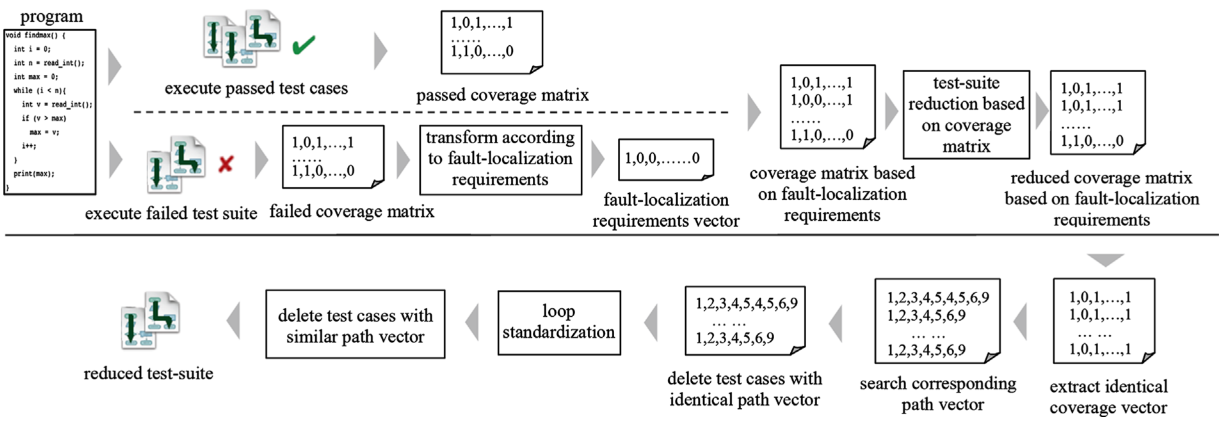

Fig. 1. The framework of two-step test-suite reduction.