ISSN 2287-4348

Vol.2 No.1

SMA 2013 The 2nd International Conference on Smart Media and Applications

October 14-17, 2013

Sutera Harbour Resort Hotel-Pacific, Kota Kinabalu, Malaysia

http://kism.or.kr/SMA2013

Smart Media
KOREAN INSTITUTE OF SMART MEDIA

Oral Presentation 3

Session G : SNS, Mobile Apps & Game (Pacific Function Room 7)
16:00~17:30, Tue, OCT 15, 2013
Session Chair : Jongmyung Choi (Mokpo National Univ.), Inho Ra (Kunsan National Univ.)

Session H : Multimedia & Applications (Pacific Function Room 8)
16:00~17:30, Tue, OCT 15, 2013
Session Chair : Sicheon You (Chosun Univ.), Seill Choi (Honam Univ.)

A Method for Establishing Simplest LTS based on Strong Equivalence

Byungho Park[1], Dongho Kim[1], Yong B. Park[2], Hyun Seung Son[1],R. Young Chul Kim[1]

[1]Dept. of CIC, Hongik University, Sejong Campus, 339-701, Korea
{bhpark, ray, son, bob}@selab.hongik.ac.kr
[2]Dept. of Computer Science, Dankook University, 330-714, Korea
ybpark@dankook.ac.kr

Abstract. The most ideal software testing is to remove errors and recursions through verification in all the sources so as to minimize the bug included in the program to be developed and satisfy the function and accuracy to be required by users. However, if the system to be developed is a large size, it may be difficult to test all of the source codes, so it is universal to try to extract small-sized test cases and satisfy high coverage if possible. This paper suggests a method which can transform original LTS into Simplest LTS at one time. It can expect an advantage to decrease computer sources and testing time epochally because it tests errors after excluding the recursion of software source codes with Strong Equivalence and optimizing them.

Keyword: software testing, simplest LTS, Strong equivalence, test coverage

1 Introduction

It has become very important to remove any danger inherent in a computer software such as nuclear power plants and express trains. So, the importance of testing in software development has been growing more and more.

The verification of the overall source codes in a large system is the most ideal but the real system is so wide that it is general to make and test only a few test cases by the maximum of test coverage. Here, the coverage rate indicates how many coverage criteria it has achieved to be tested.

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. This activity results in the actually expected difference between their effects. In simple words, testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements. [1]

The most ideal software testing is to remove errors and recursions through verification in all the sources so as to minimize the bug included in the program to be developed and satisfy the function and accuracy to be required by users. However, if

the system to be developed is a large size, it may be difficult to test all of the source codes, so it is universal to try to extract small-sized test cases and satisfy high coverage if possible.

This paper suggests a method which can transform original LTS into Simplest LTS at one time. It can expect an advantage to decrease computer sources and testing time epochally because it tests errors after excluding the recursion of software source codes with Strong Equivalence[2,6] and optimizing them.

2 Preliminaries

The equivalence for processes is based on observations. The idea of the equivalence is that when an observer watches behaviors of two processes from an environment, they are equal if the observer cannot distinguish these behaviors [2].

Processes are software systems which are composed of the subset processes. The process can represent semantics by the Labeled Transition System (LTS) [2]. For processes P and Q, if there exists a strong bisimulation R such that P, Q∈R, then P and Q are strongly equivalent, written as P ~ Q. The strong equivalence does not consider the internal actions which cannot be observed from the external environment.

This paper deals only with observable actions because the strong equivalence is enough for the purpose. In brief, the previous study is shown below as Fig. 1.Suppose that the original LTS means developing a system and smallest LTS is to have the smallest number of states and nodes [3].

Even if an original LTS has a tree structured form, the corresponding LTS is not tree-structured but a graph in general. In order to compare two systems using each distinct layer, the minimal LTS must be changed to a tree structured LTS, in other words, the simplest LTS [4].

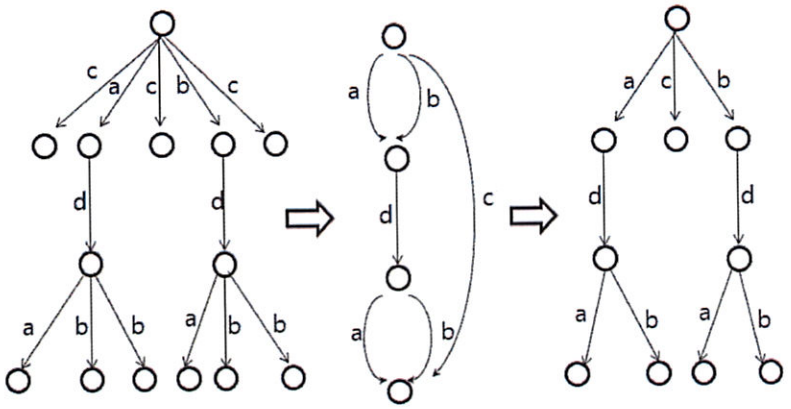

Fig. 1. Simplest LTS via Smallest LTS from Original LTS [4]

3 Improved Simplest Method Transformation

The above algorithm needs to be performed twice even though it can make the simplest LTS. In other words, as it is shown in the figure, leaf nodes need to have only one edge in order to compare the simplest LTS optimization contraction and the equivalence of each layer between the two LTS's, and also, each node has to go through the transition process to the simplest LTS to transform so that it can only one parent.

It requires making efforts twice, so it needs improvements. Therefore, this paper suggests an improved method, which can be transformed into Original LTS and the Simplest LTS based on Strong Equivalence. An example below shows that it changed from original LTS to Simplest LTS using a Strong equivalence.

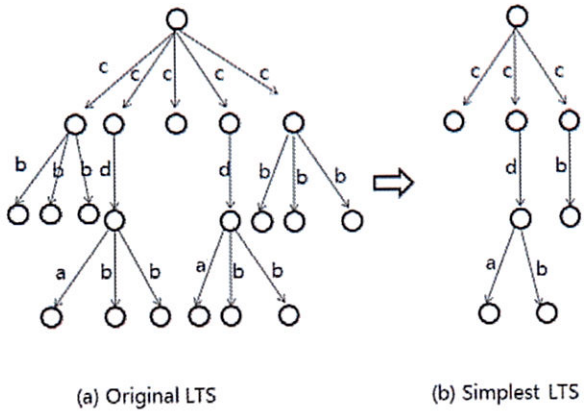

Fig. 2. The Simplest LTS from the Original One at once

As shown in Fig. 2, it is recognized that if the Simplest LTS is used and test case is generated in this method, it is useful because computer resource and testing time can be decreased in a large scale.

4 Conclusion

The most ideal software testing is to remove errors and recursions through verification in all the sources so as to minimize the bug included in the program to be developed and satisfy the function and accuracy to be required by users. However, if the system to be developed is a large size, it may be difficult to test all of the source codes, so it is universal to try to extract small-sized test cases and satisfy high coverage if possible.

This paper suggests a method which can transform original LTS into Simplest LTS at one time. It can expect an advantage to decrease computer sources and testing time epochally because it tests errors after excluding the recursion of software source codes with Strong Equivalence and optimizing them.

Future studies are expected to find out a method to exclude the recursion about the same type existing in the optimized LTS hierarchically.

Acknowledgments. This work was supported by the IT R&D Program of MKE/KEIT [10035708, "The Development of CPS(Cyber-Physical Systems) Core Technologies for High Confidential Autonomic Control Software"] and Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2013R1A1A2011601)

References

1. http://tutorialspoint.com,"SOFTWARE TESTING TUTORIAL"
2. Robin, M.:Communication and Concurrency. Prentice hall, 1989
3. Shiratori, N., Kaminaga, H., Takahashi, K., Noguchi, S.:A Verification Method for LOTOS Specifications and its Application. Protocol Specification, Testing and Verification IX, pp. 59-70, North-Holland, 1990
4. Park, B., Kimura, S., Lee, E., Shiratori, N.: Process Error Detection of Education Support Environment for LOTOS. IEICE Trans. Vol. J80-D-II, No.4, pp. 961-971, 1997
5. J. Myers, G.:The Art of Software Testing, Second Edition. Word Association, Inc. 2004
6. Valmari, A.:External Behaviour of Systems of State Machines with Variables, Transactions on Petri Nets and Other Models of Concurrency VII, LNCS 7480, pp. 255-299, 2013
7. A. Reniers, M., Schoren, R., A.C. Willemse, T.:Results on Embeddings Between State-Based and Event-Based Systems, The Computer Journal, pp. 1-20, 2012

Smart Media

KOREAN INSTITUTE OF SMART MEDIA