BRNO UNIVERSITY OF TECHNOLOGY

FIT

FACULTY OF INFORMATION TECHNOLOGY

Brno, July 31, 2012

Review of Doctoral Thesis submitted to the Brno University of Technology, Faculty of Information Technology

Thesis title: Analysis and Testing of Concurrent Programs
Author: Ing. Zdeněk Letko

Topics of the thesis

This doctoral thesis deals with concurrency-related errors and noise injection techniques in concurrent programs. In particular, it proposes new concurrency coverage metrics, noise injection heuristics and search based noise injection methods. I consider the theme of the thesis **as topical and relevant** because the requirement on fast and accurate testing is especially important nowadays when parallel programs become omnipresent. Search-based software engineering is now one of the hot topics for the software engineering/testing community as well as the evolutionary computation community.

Organization of the thesis

The thesis contains 119 pages of regular text. Additional 20 pages are devoted to two appendixes dealing with experiments with genetic algorithm and description of the developed software tool for testing of concurrent programs. The thesis is divided into 6 chapters. The thesis is clearly written. The text is well organized and structured.

Chapter 1 provides an introduction to verification of multi-threaded software, search techniques, goals and structure of the thesis. The goal of research is briefly formulated in Chapter 1.3 as “to improve the testing process of real-world multi-threaded programs”. The subgoals are not formulated explicitly, but can be derived from the description given in the same chapter.

Chapter 2 is devoted to verification of concurrent programs. It provides a uniform taxonomy of concurrency errors which can be considered as an original work of the author. Special subchapters deal with noise injection techniques and dynamic analysis algorithms as key parts of the solutions proposed in the following chapters of the thesis. The semi-formal approach used by the author to explain the key concepts is appropriate and makes reading easier. However, despite it is probably very difficult, I would expect a quantitative comparison of mentioned algorithms.

Chapter 3 deals with a methodology of deriving new coverage metrics for testing of concurrent programs based on dynamic analysis. The experimental comparison is performed using a set of benchmark programs created by the author. As there is no suitable benchmark set available for comparisons it makes sense to create a new one. With respect to objectives given and available computing resources the proposed set of benchmark programs seems to be composed correctly. The author should explain why he has used the given setting of experiments (the number of executions etc.). It is also confusing that the author has used different colors in the performance curves (Fig. 3.1) for the same test program.

Chapter 4 proposes new noise injection heuristics. The heuristics are experimentally evaluated in greater detail. Here, I would expect more statistical measures (the standard deviation, at least). It is not clear whether the average values are sufficient for performance analyses. Despite the fact that no single best-performing solution has been discovered, I appreciated that the author has done a lot of experimental work to learn about the problem. The results are very valuable for future research.

Search-based methods, in particular genetic algorithms (GA), are introduced to concurrent program testing in Chapter 5. Similarly to Chapter 4, this chapter contains numerous valuable experimental results. On page 91, I would expect a mathematical (formal) definition of the fitness function. Without it, it is difficult to analyze the results. It is natural to formulate the problem as multi objective (page 100); however, a truly multi objective GA should be used instead of the weighting function where the given setting of the weights is not supported by any experimental evidence. I would also expect more detailed statistical evaluation of the proposed methods. Again, the setting of same parameters is just mentioned without any explanation (page 92: Why did you evaluate each state of the space 100 times?). On page 93, the average number of steps is counted. However, it is unclear what the steps really mean. The thesis contains a detailed analysis of the role of GA parameters which is very useful.

Chapter 6 summarizes the obtained results and gives some directions for future research.

Contribution and methodology

The author has developed various techniques which utilize and combine program analysis methods (static/dynamic etc.), noise injection methods and search methods in order to improve detecting of concurrency related errors in Java programs.

Deriving new coverage metrics for discovering bugs in concurrent programs, introducing new noise injection heuristics and controlling the parameters of these heuristics using search algorithms can be considered as **original contributions** of the author. It is also important that proposed methods have been carefully evaluated and compared with existing approaches. Formalizing the test and noise configuration search problems has opened a new way to concurrent programs testing. Although the methodology is very time consuming and does not work perfectly, it looks promising for future.

The author has demonstrated excellent knowledge of formal methods and a deep expertise in the areas of concurrent software programming, analyzing and testing. I also highly appreciate his productive collaboration with other members of Prof. Vojnar's team and companies such as IBM. The proposed solutions have been implemented in a software tool called SearchBestie. Experimental evaluation was carried out using this software and a set of benchmark programs.

To summarize, the presented research results were obtained using **correct and rigorous methodology**.

Minor comments

(1) The thesis is written in such a way that it looks like a result of a group of researchers instead of a single PhD candidate's work. E.g. "we present our new…", "is inspired by our work on…", "presents our proposal… " on page 8; "Our metrics…", "our methodology…" on page 38. The list of publications does not contain any information about candidate's contribution to the publications. The candidate should clearly specify his contribution during the defense of the thesis.

(2) Some claims are too strong or imprecise and should be reformulated, e.g.

- Page 6: "However, metaheuristics deliver satisfactory solutions for complex problems in a reasonable time." I would use "can deliver".
- Page 7: "The crossover operation combines two parents, and the mutation operation modifies (mutates) the result of the crossover." Mutation can be applied without crossover. The crossover operation can have more than two parents.

(3) I consider the language level of the thesis as very good. I found only a few small mistakes and typos.

(4) Other mistakes:

- Page 6: “as optimal as possible” makes no sense.
- Page 63, 75 etc.: “The table illustrates…” is confusing. The number of the table has to be used explicitly.
- Table 5.2: The title is too general. The reader should be able to understand the content from the title.

Author’s publications

The list of author’s publications contains 8 peer-reviewed conference/workshop publications, 3 technical reports and 3 other papers, in most cases published by ACM or Springer. The author has developed 2 software tools and participated on several research projects, including international research projects. His work has been cited 38 times which is a very good result for a PhD candidate. I found all the papers relevant for the topic of the thesis. The number of publications is more than sufficient. Nevertheless, I would like to see a journal paper. Unfortunately, author’s contribution to the multi-author papers has not been given in the report.

Questions

(1) In some cases it is not explained why a particular setting of parameters has been applied (e.g. 100 evaluations of each state on page 92; 100 runs on page 92). Please justify your setting.
(2) What do you mean by the “steps” when speaking about the hill climbing algorithm on page 93?
(3) When designers/testers should use the proposed search based fault injection? Every time when a new piece of code has to be tested or after some preliminary testing using standard methods?
(4) What is your contribution to the multi-author papers?

Summary

Despite some of my comments given above, this is a well-written doctoral dissertation presenting new methods and results in the area of concurrent programs testing. The proposed solution seems to be applicable for industrial purposes as well as academic experimental work. I highly appreciate that the candidate has done a lot of implementation and experimental work. In my opinion, the author of the thesis proved his ability to perform research and achieve scientific results. The thesis contains **new and original results** that have already **been published** on recognized international conferences. I **do recommend** the thesis for presentation/defense with the aim of receiving the doctoral degree.

Prof. Ing. Lukáš Sekanina, Ph.D.

Faculty of Information Technology
Brno University of Technology
Božetěchova 2
612 66 Brno
http://www.fit.vutbr.cz/~sekanina
sekanina@fit.vutbr.cz