Software verification for real world applications

Candidate No: 105936

24th December 2018

Contents

1 Introduction

Generally software verification is a very interesting topic in research at the moment, however it is currently limited to the field of researchers and it it is only really used as a part of demonstration software and only as a part of verifiable languages. Therefore this paper will look into how the software verification techniques can be applied to applications designed for use in the real world. This will obviously have advantages as it guarantees code free of fatal runtime errors and reduces the likelihood of other errors.

2 Current Problem

Explain why current software is not verified or written using verifiable languages

3 Existing Software verification techniques

There are three types of static analysis are Abstract static analysis, model checking and, bounded model checking.[1]

3.1 Abstract Static Analysis

3.2 Model Checking

3.3 Bounded Model Checking

4 Existing Verifiable Languages

5 How to bring software verification into the mainstream

[2] shows how to add static checking to java programs but may not go into how to allow for some parts to be checked and other not to be checked.

Come up with an idea about how to bring software verification into the mainstream

6 Conclusion

References

[1] V. D'Silva, D. Kroening, and G. Weissenbacher, "A survey of automated techniques for formal software verification," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 7, pp. 1165–1178, Jul. 2008, ISSN: 0278-0070. DOI: 10.1109/TCAD.2008.923410.

[2] C. Flanagan, K. R. M. Leino, M. Lillibridge, G. Nelson, J. B. Saxe, and R. Stata, "Extended static checking for java," in *Proceedings of the ACM SIGPLAN 2002 Conference on Programming Language Design and Implementation*, ser. PLDI '02, Berlin, Germany: ACM, 2002, pp. 234–245, ISBN: 1-58113-463-0. DOI: 10.1145/512529.512558. [Online]. Available: http://doi.acm.org/10.1145/512529.512558.