Summary of
"A Comparison of Constraint-based and Sequence-based Generation of Complex Input Data Structures"

Nikhil Satish Pai Vishal Mishra Nikhil Anand

7th September 2015

Important Keywords:

ii1. Bounded exhaustive testing :

It is an automated testing technique to verify the functionality of the code for all possible cases within a given bound specified by a user. The assumption is that bugs can be detected within the specified bounds with extensive testing so that all possible scenarios are covered.

ii2. Constraint based generation:

It is a test case generation method where the user encodes an invariant for the input data structures and a tool generates all possible variants of the same. The outputs generated must be non isomorphic. As the structure is already generated, different operations can be tested in isolation from each other as they do not depend on the previous ones.

ii3. Sequence based generation

The data structures will be generated by a sequence of operations. For example a sequence of add and delete operations on a linked list, to generate all possible output structures different combinations of the operations must be executed. It may leave out certain combinations of the data structure.

ii4. False sense of confidence:

False positives or False alarm is a result that indicates a given condition has been fulfilled, when it actually has not been fulfilled, whereas False negative implies a test result indicates that a condition failed, while it actually was successful ; These kind of results provide a false sense of confidence on the result collected out of a test.

Brief notes:

iii1. Motivation: Manual generation of complex input data structures to be used for testing is very tedious and error prone. Automated generation of data structures that can be used for testing can be achieved using two approaches : constraint based and search based generation. Knowing the circumstances where each of these approaches have their relative advantages and disadvantages will be really useful in generating automated test cases.

iii2. Commentary : The data structures were implemented using both Korat and Java PathFinder. JPF checks Java software models for all property violations and unhandled exceptions along all possible execution paths. Korat contains a Java method called repOK and the constraints to be encoded in this method are readily available.

iii3. Related Work : Generation of complex data inputs for better coverage using symbolic execution was introduced by Khurshid et al *. Extension of the aforementioned work was done by Pasareanu ** where they worked on model checking (using JPF) and abstraction on container data structures. d'Amorim et al. *** compared random generation and symbolic execution.

iii4. Future Work : Both the approaches (Constraint based and Sequence based) have their relative advantages under different circumstances. The proposed future work is on combining these two approaches; sequence based generation can be used to search for method sequences that create the data structures that are found interesting by using constraint based generation.

Area of Improvement:

iv1. The paper suggests methods of generation of complex data structures for automated testing but does not consider the time taken in generating all exhaustive structures.

iv2. More data structures like skip lists, B trees can be included for evaluation of the two methods.

Relation to last Paper :*The IntAVLTreeMap, IntRedBlackTree and the HeapArray data structures used in the original paper were generated by referring to this paper.*

**S. Khurshid, C. Pasareanu, and W. Visser, "Generalized symbolic execution for model checking and testing," in Proc. 9th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS), 2003.*

*** W. Visser, C. S. Pasareanu, and R. Pelanek, "Test input generation for red-black trees using abstraction." in Proc. 20th International Conference on Automated Software Engineering (ASE), 2005*

****M. d'Amorim, C. Pacheco, T. Xie, D. Marinov, and M. D. Ernst, "An empirical comparison of automated generation and classification techniques for object-oriented unit testing," in Proc. 21st International Conference on Automated Software Engineering (ASE), 2006.*

Reference to the Paper:

[1] R. Sharma, M. Gligoric, V. Jagannath, and D. Marinov, “A Comparison of Constraint-Based and Sequence-Based Generation of Complex Input Data Structures,” 2nd Workshop on Constraints in Software Testing, Verification and Analysis, Apr. 2010.

References:

http://javapathfinder.sourceforge.net

http://mir.cs.illinois.edu/astgen/JagannathETAL09ReducingCosts.pdf