Summer 7-9-2012

Generating Minimal T-wise Test Suites

Luis C. Gutierrez  
Department of Computer Science, University of Texas at El Paso, lcgutierrez@miners.utep.edu

Carlos Nieto  
Department of Computer Science, University of Texas at El Paso, canieto@miners.utep.edu

Francisco Zapata  
Department of Computer Science, University of Texas at El Paso, fazapatagonzalez@miners.utep.edu

Martine Ceberio  
Department of Computer Science, University of Texas at El Paso, mceberio@utep.edu

Follow this and additional works at: http://digitalcommons.utep.edu/couri_abstracts_sum12

Recommended Citation
http://digitalcommons.utep.edu/couri_abstracts_sum12/39

This Article is brought to you for free and open access by the COURI Symposium Abstracts at DigitalCommons@UTEP. It has been accepted for inclusion in COURI Symposium Abstracts, Summer 2012 by an authorized administrator of DigitalCommons@UTEP. For more information, please contact lweber@utep.edu.
Generating Minimal T-wise Test Suites

Luis C. Gutierrez, Carlos Nieto, Francisco Zapata, and Martine Ceberio

As the use of computing devices increases every day, users rely on the adequate functioning of software. When software is not tested properly, it can yield erroneous information or a complete failure of the system. The NIST estimates that defective software cost the United States economy close to $60 billion a year. Therefore, there is a need to develop software testing techniques that are time and cost effective. Fully testing software under all possible combinations of parameters values cannot be reduced. However, testing can focus on covering all combinations of subsets of parameters and empirical data shows that doing so for up to groups of 6 parameters can detect up to 100% of failures. This approach is called t-wise testing and can significantly cut companies testing costs. Greedy algorithms are typical approaches to generating t-wise covering test suites: they produce near optimal solutions but are memory-inefficient. As an improvement, we present an algorithm to generate t-wise covering test suites that makes use of rotational patterns and constraint programming techniques, and is memory efficient. In addition, the very nature of our algorithm allows to provide a theoretical proof to a long empirically-known result on the upper bound of the size of optimal test-suites.