Constraint-based Software Verification of Program:What can be done?

Shubhra Datta and Martine Ceberio
Department of Computer Science
University of Texas at El Paso

October 26, 2010

Abstract

Verification and validation are two components of the software engineering process critical to achieve reliability that can account for up to 50% of the cost of software development [1]. Numerous techniques ranging from formal proofs to testing methods exist to verify whether programs conform to their specifications. Recently constraint programming techniques have been emerged [1, 2]. They use the idea of proof by contradiction. They consider the conjunction of the constraint systems derived from the program and the negation of the specifications as a Constraint Satisfaction Problem (CSP). They typically aim at proving that the CSP has no solution, which means that the software conforms to its specifications.

Although the framework seems straightforward, the number of generated constraints can be high.

In this work we propose ideas for improvement based on symbolic manipulation of the constraints to be solved. We design a tree structure based on the program flow and then we do a DFS traversal of the tree when at the same time we try to weakly eliminate inconsistent constraints along the path. If any conditional statement is encountered (decision point), we check the consistency of the constraints we have so far with the specification. If the test fails then we cut that branch from the tree. If the test succeeds we continue with the traversal. In the worst case, meaning when the code is correct, we have to create and traverse the whole tree. So we try to analyze how much practical the approach is if the program is correct and if the program is incorrect: can we make it faster by applying our elimination rules? We will use CPBPV as a benchmark to verify and compare our approach. Our future work is about proposing a bi-directional framework to reduce the size of the generated and traversed tree.

References

[1] H. Collavizza,and M. Reuher, "Exploration of the capabilities of constraint programming for software verification," in *TACAS*,2006.

[2] M. Ceberio, C. Acosta, and C. Servin, "A Constraint-based approach to Verification of Programs with Floating-point Numbers," in *the 2008 International Conference of Software Engineering Research and Practice*, Las Vegas, Nevada, USA, CSREA Press, pp. 225-230, 2008.

[3] H. Collavizza,M. Reuher, and P. Van Hentenryck, "CPBPV: A constraint-programming framework for bounded program verification," in *Proceedings of the CP2008,LNCS 5202*, 2008.