

Constraint Based Software Verification with Floating Point Numbers

Shubhra Datta and Martine Ceberio

Department of Computer Science

University of Texas at El Paso

El Paso, Texas 79968

Email: sdatta2@miners.utep.edu and mceberio@utep.edu

March 20, 2010

Abstract

Software plays an important role in our lives. Verification and validation are two components of the software engineering process critical to achieve reliability. Verification and validation are two of the most critical issues in the software engineering process. These expensive and difficult tasks can account for up to 50% of the cost of software development [2].

Real-time applications that take advantage of constraint programming (CP) techniques keep increasing. Recently, CP techniques have been used in software verification and applied to automatically generating test cases. Numerous techniques ranging from formal proofs to testing methods exist to verify whether programs conform to their specifications. Some of these techniques (CP techniques) focus on demonstrating that the union of the constraints derived from the program and the negation of the constraints derived from its specification is inconsistent.

However, computations with floating-point numbers are a major source of failures of software systems, making verification of critical software that use floating point numbers essential. There are other challenges also including risk of getting false positives and missing solutions and solving disjunctive CSPs. Previous work [1] in this field, suggested an approach to this problem which, if implemented, can make software verification easier and simplified.

Since software testing is a crucial stage in software development, the proposed approach will deeply impact the software industry by dramatically cutting costs. Unfortunately, to date, no implementation has been completed. This work if improved for practicality and implemented, will constitute a new direction towards software verification, which is a long-lived and expensive problem in software industry.

My work plan is as follows:

- a. Studying and implementing the current approach to be able to compare its efficiency against our future systems
- b. Investigating ways to integrate existing techniques for solving union of CSPs into a faster verification tool
- c. Solving disjunctive CSPs as well as disjunctions of CSPs, providing rules to control the explosion of computing complexity
- d. Designing rules to build CSPs for software verification with floating-point numbers

The end product will be a complete software verification tool able to handle floating-point programs.

References

- [1] Martine Ceberio, Carlos Acosta, and Christian Servin. A constraint-based approach to verification of programs with floating-point numbers. In *International Conference of Software Engineering Research and Practice*, pages 225–230, Las Vegas, Nevada, USA, 2008. CSREA Press.
- [2] H el ene Collavizza and Michel Rueher. Exploration of the capabilities of constraint programming for software verification. In *TACAS*, 2006.