Control flow graph, formal verification and constraint programming techniques

Jesse Deveza [1,*], Lanier Santos [1], Rosiane de Freitas [1], Lucas Cordeiro[2,1]
[1] Institute of Computing - UFAM, Brazil
[2] University of Manchester, UK

Keywords: constraint programming, directed graph, SAT problem.

Formal program verification is a generally undecidable problem. Bounded Model Checking (BMC) is one method that can achieve decidability by searching for violations of properties of a program up to a bound k. BMC reduces the program verification problem to the classic NP-complete Boolean Satisfiability (SAT). However, it can still lead to an exponential state-space exploration due to the program's large and possibly unbounded loops. In this case, there might be many execution paths to traverse through a program during its symbolic execution. Therefore, the control flow or computation during the program's execution, mainly in symbolic execution, can be represented as a directed graph named Control Flow Graph (CFG). In this work, we present the properties of the CFG and discuss the application of constraint programming techniques to reduce variable domains as a preprocessing step or during the BMC process for verifying software systems. We also describe how constraint programming can be exploited to prove the (partial) correctness of the program via proof by induction built on top of BMC.

References

1. Cordeiro, L., Fischer, B., and Marques-Silva, J. (2009). Smt-based bounded model checking for embedded ansi-c software. In ASE, pp. 137-–148, 2009.
2. Rossi, F., van Beek, P., and Walsh, T. (2006). Handbook of Constraint Programming (Foundations of Artificial Intelligence). Elsevier Science, 2006.
3. Gadelha, M., Ismail, H., Cordeiro, L.: Handling loops in bounded model checking of C programs via k-induction. In STTT, 19(1), pp. 97–114, 2017.