Abstract

With the success of formal verification techniques like equivalence checking and model checking for hardware designs, there has been growing interest in applying such techniques for formal analysis and automatic verification of software programs. The majority of work carried out in the formal methods community throughout the last three decades has been devoted to special languages designed to make it easier to experiment with mechanized formal methods such as theorem provers, proof checkers and model checkers. With this philosophy, in this paper, it is proposed to develop a verification and testing environment for checking user given safety properties in C Programs, which uses model checking. The reachability properties for verification have to be considered in particular whether certain labeled statements are reachable or not in a program. Also, checkers are included for a set of standard programming
bugs such as array bound violations, NULL pointer dereferences, use of uninitialized variables, memory leaks, lock/unlock violations, division by zero etc.

References

- Dijkstra, E.W., "Guarded commands, nondeterminacy and formal derivation of programs." CACM 18, 8 (1975), 453-457.
- Holzmann, G.J., "Algorithms for automated protocol verification"
- Dijkstra, E.W., "Solution to a problem in concurrent programming control." CACM 8, 9 (1965), 569.

Index Terms

Computer Science
Emerging Trends in Technology

Keywords
Lex and Yaac  SPIN  Formal Methods