Abstract

Path testing a program involves generating all paths through the program, and finding a set of program inputs that will execute every path. Since it is impossible to cover all paths in a program, path testing can be relaxed by selecting a subset of all executable paths that fulfill a certain path selection criterion and finding test data to cover it. The automatic generation of such test paths leads to more test coverage paths thus resulting in efficient and effective testing strategy. This paper presents a structural-oriented technique that uses a genetic algorithm (GA) for automatic generation of a set of test paths that cover the all-uses criterion. In the case of programs that have loops, the proposed technique generates paths according to the ZOT-subset criterion, which requires paths that traverse loops zero, one and two times. The proposed GA uses a binary vector as a chromosome to represent the edges in the DD-graph of the program under test. The set of paths generated by the proposed GA can be passed to a test data generation tool to find program inputs that will execute them. Experiments have been carried out to evaluate the effectiveness of the proposed GA compared to the random test path generation technique.
References

- P. M. S. Bueno and M. Jino, "Automatic Test Data Generation For Program
Automatic Generation of Data Flow Test Paths using a Genetic Algorithm

- I. Hermadi, C. Lokan, and R. Sarker, \textquotedblright;Genetic Algorithm Based Path Testing: Challenges and Key Parameters\textquotedblright; Second WRI World Congress on Software Engineering, 2010.
- S. Rapps and E. J. Weyuker, \textquotedblright;Selecting software test data using data flow information\textquotedblright; IEEE Transactions on Software Engineering, Vol. 11, No. 4, pp. 367-375, 1985.

\textbf{Index Terms}

Computer Science Artifical Intelligence
Keywords

Automatic test path generation  Data flow testing  Genetic algorithms.