

Tutorial on Automated Structural Testing with PathCrawler^{*}

Extended abstract

Nikolai Kosmatov and Nicky Williams

CEA, LIST, Software Safety Laboratory, PC 174, 91191 Gif-sur-Yvette France
`firstname.lastname@cea.fr`

Introduction. Automation of test-case generation brings obvious benefits. In critical systems processes where structural testing is required by the development norm, manually creating tests from the specification fails to achieve complete satisfaction of the coverage criterion. In this case, automatic methods help to reach the objectives which are not covered and provide corresponding path conditions that may be used to refine the specification if needed. They may also determine whether the objectives which are not yet covered are really infeasible. Even when the development process does not impose any structural testing activity, the use of a structural test generation tool is a way to increase the quality of the software with a very low cost overhead. PathCrawler is a concolic test generation tool developed at CEA LIST for structural testing of C programs. It aims to cover all feasible program paths. The new version of PathCrawler is developed in an entirely novel form: that of a test-case generation web service which is freely accessible at PathCrawler-online.com.

The tutorial. The first aim of the tutorial is to show how *C code can quickly and easily be debugged using automatic structural unit testing*. The second aim is to show that tools such as PathCrawler can help to *respect the code coverage required by many development norms*, and report on what cannot be covered. Finally, we will show how structural testing may be used in *combination with static analysis techniques* and enhance their results.

This tutorial is aimed mainly at software engineering professionals and students. They will learn more about the state of the art in automated software testing and how it could help them in their future career in software development or validation. Software engineering lecturers may also be interested in how a tool such as PathCrawler-online.com can help in teaching software testing. The necessary background is some knowledge of the C language.

After a brief introduction to structural testing and the concolic method underlying the PathCrawler tool, the notions of precondition, coverage criterion, and oracle will be explained and illustrated interactively on simple examples. The tool outputs (test cases, results, infeasible paths,...) will be explained and the tutorial students will be guided in the use of these outputs to discover the source of different bugs. Some limits of structural testing and advantages of its combination with static analysis tools will also be illustrated. The tutorial will use the online version of the PathCrawler tool: PathCrawler-online.com.

^{*} This work has been partially funded by several ANR projects.