Automated Test Data Generation for Control Flow Coverage using Boolean Constraints

Stefan Weiler

Goals

To prevent human error in software development to cause failures in critical software, thorough testing is necessary in software verification. The importance of this task is clear to see in examples like the failed maiden flight of Ariane 5. Coverage criteria are metrics used to measure the grade of coverage of software. Depending on the criticality of a software, defined coverage criteria should be fulfilled to a number of instances where a criterion has been fulfilled divided by the total number of instances the criterion applies to:

\[ \text{grade of fulfillment} = \frac{\text{instances covered}}{\text{instances total}} \]

For critical software it is often stated, that coverage has to be achieved, which means its criteria have to be fulfilled to the grade of 100%. Corresponding regulations exist to specify minimum standards on this, for example in aeronautics.

Control flow coverage is an umbrella term for sets of such coverage criteria. A single set defines a type of control flow coverage. One or more criteria have to be applied for a defined type of control flow coverage. These types of control flow coverage will be discussed in greater detail later on. As the name indicates, the coverage criteria used are based on the control flow, i.e. the structure of a program (For example: Which decisions are made?, Which statements get executed?). To retrieve the actual structure of a program, its source code has to be analyzed. Tests performed on this basis are therefore white box tests.

Finding test cases to achieve a type of control flow coverage is a complex real world problem, which has to be addressed in some way.

Boolean satisfiability (SAT) solving is a technology based on Boolean logic. It can be used on a broad variety of real world problems, which have to be encoded in Boolean formulas.

This thesis aims to approach the problem of achieving certain types of control flow coverage with a new method, using SAT solving techniques. It will provide a tool for automatic test data generation. Further on, this work can also be used for the evaluation of existing test cases derived by other means (for example: requirement based test cases) with respect to their coverage of the control flow. With a second metric it is hence providing a more thorough test of the software as requirements coverage alone.

It has therefore be shown that this problem can be expressed in Boolean formulas. SAT solving was used to find a valid assignment to these formulas. From this assignments sets of test cases were derived. Examples were used to show the applicability of this approach. With the successful implementation of a prototype, this thesis shows a promising approach for the automatic generation of test data.

Background

This thesis is about the task of automatic test data generation in the field of software verification. It is one of many fields in which Boolean constraints and SAT solving can be applied. In white box testing, sets of test data are needed that allow to test for certain types of control flow coverage. For the automatic test data generation this thesis will focus on control flow coverage / structural coverage. The control flow of a program is driven by control structures, including: conditional structures (if/else), iteration structures (loops), selective structures (switch), jump statements, and – as in the Ariane 5 example – exception handling. The selected types of coverage are shown in the following list:

- decision coverage
- condition coverage
- condition/decision coverage
- modified condition/decision coverage (MC/DC)

Other types of coverage are out of scope and will not be discussed in this thesis.

Automatic test data generation, in this context, means to identify input values for a (minimal) set of test cases, leading to the desired control flow coverage of a program when executed.

Finding test data is an important task in software verification. It is important to have generators, which can efficiently provide correct data.

Method of Approach

To show the applicability of the proposed approach, the focus of this thesis is the development of a prototype application. This application can automatically generate test data for selected types of control flow coverage using Boolean constraints.

The technique can potentially be applied to many procedural or object oriented programming languages. Exemplarily it has been implemented for functions in a subset of C.

In general for a real world problem to be solved using SAT, three components are needed: encoder, solver and decoder. The implementation follows this scheme consisting of three major parts:

1. A command line application taking a C source file and options as input, parsing them, and generating a corresponding set of constraints. It also has to give information on mapping the variables of the C source file to the variables of the constraint system.
2. One of the existing SAT solvers being used to solve the generated set of constraints.
3. A second command line application taking the output file of the SAT solver and the variable mapping information as input, deriving a set of test cases.

The common input format (DIMACS CNF) is used to hand over the set of constraints to the SAT solver. Output formats are not standardized. The interface can be adapted to a specific SAT solver.

It is possible to choose between decision coverage, condition coverage, condition/decision coverage, and modified condition/decision coverage.