· U C ·

UNIVERSIDADE DE COIMBRA
Faculty of Science and Technology
Department of Informatics Engineering

Master in Informatics Engineering

System Modelling and Analysis
Assignment 1 – Optimisation Models

Coimbra, 21 October 2015

Path-set selection in control-flow testing

Control-flow testing is a technique used to test the implementation of a program. It is a form of white-box testing, since the structure of the program to be tested is assumed to be known. In particular, the control-flow graph of the program to be tested is considered.

The control-flow graph (CFG) is a directed graph (or digraph) where nodes represent code segments to be executed sequentially without branching (or basic blocks), and edges represent transfer of control flow from one code segment to another due to a branch instruction at the end of a code segment. The graph is assumed to have a single entry (or source) node, and a single exit (or terminal) node.

A particular execution of the program, given some sort of input (a test case), may then be represented as a path from the source node to the terminal node of the CFG, where nodes and edges may or may not be visited multiple times. Such a path is said to *cover* a node (or an edge) if such a node (or edge) is contained in the path. Since there may be no path that covers all nodes (or edges) of the program, a set of test cases is usually required in order to achieve sufficient coverage.

Although determining all paths in the graph from the source node to the terminal node may be possible, all-path testing may be too costly in practice, and a smaller set of paths is usually sought to provide the desired coverage. Chung and Lee [1] define the *optimal path set selection problem* as the problem of finding a smallest set of paths satisfying a required coverage criterion. Common coverage criteria include node coverage (all nodes should be executed at least once), edge coverage (all edges should be traversed at least once), and consecutive edge-pair coverage (all pairs of consecutive edges in the graph should be covered). Such criteria may be relaxed to allow a given (small) percentage of nodes, edges or edge pairs to be left uncovered, at the expense of lower testing quality. Also, if the cost of executing individual paths is taken into account, a minimal-cost path set may be sought instead of a minimal-size set. Stronger testing requirements, such as those based on path linear independence and cyclomatic complexity are out of the scope of this assignment.

ILP modelling

In this assignment, you are asked to produce integer linear programming (ILP) models that allow you to enumerate paths in a control flow graph and to find exact solutions to the optimal (weighted) path-set selection problem defined above by using a standard ILP solver.

Data

Control-flow graph data is provided as a set of pairs of integers (i, j) denoting the unweighted, directed edges of the CFG (from i to j, files `cfg?.dat`). In all cases, the source node and the terminal node are considered to be the nodes with smallest and largest index, respectively. Note that some of the CFGs provided include cycles.

Test-path generation

For each CFG, enumerate all paths from the source node to the terminal node by formulating and solving a sequence of constrained shortest-path ILP problem formulations, and record the weight of each path as the number of nodes it contains (including the source and terminal nodes).

Note that, if the graph has cycles, there is an infinite number of possible paths. In that case, all paths such that so-called *basis edges* are traversed at most once should be determined. The first edge in the shortest path from each node to the terminal node is called a *non-basic* edge [2]. Basis edges are edges that are not non-basic. If, as a result of a cycle, the same node appears multiple times in the path, this should be accounted for when computing the path weight.

Path-set selection

Formulate optimal weighted path-set selection, where the cost of a path set is the sum of the costs of the paths contained in it, as an ILP problem. Consider three different coverage criteria: node-coverage, edge-coverage, and consecutive edge-pair coverage, and solve the corresponding problems.

Path-set cost/coverage trade-off

Modify the previous model in order to allow path-set selection under partial (instead of full) coverage constraints. For example, it may be possible to reduce the cost of the test set by requiring only 90% of CFG nodes (or edges) to be covered, *without* specifying which actual nodes or edges should be left out. Use this new, more general, model to obtain cost-coverage trade-offs for one of the CFGs provided.

Implementation

All ILP models developed are to be implemented in the GNU MathProg modelling language, and results should be obtained using the `glpk` solver. Pre-processing, post-processing, and visualisation of input and output data may be done using other tools of your choice.

Reporting

Working in groups of (exactly) three students, write a brief report detailing and justifying the ILP models developed, and discussing the results obtained with them on the data provided. A zip file containing the report in pdf format as well as the ILP models and any other source code produced must be submitted through inforestudante.

Reference

[1] C.-G. Chung and J.-G. Lee, "An Enhanced Zero-One Optimal Path Set Selection Method," *Journal of Systems Software*, vol. 39, pp. 145–164, 1997.
[2] A. H. Watson and T. J. McCabe, "Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric," NIST Special Publication 500-235, 1996.

CMF