VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wānanga o te Ūpoko o te Ika a Māui

School of Engineering and Computer Science
Te Kura Mātai Pūkaha, Pūrorohiko

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Internet: office@ecs.vuw.ac.nz

QuickCheck for Whiley

Janice Chin

Supervisors: David Pearce, Lindsay Groves

Submitted in partial fulfilment of the requirements for
Bachelor of Engineering with Honour.

Abstract

This document is a project proposal for the project, QuickCheck for Whiley. It will describe an engineering problem and how this project will solve the problem. It will also explain how to evaluate my proposed solution and any resource requirements needed for this project.

1. Introduction

This project is about implementing an automated test-case generator for Whiley.

2. The Problem

Testing is an important process in software development as it helps detect the presence of bugs. However, writing and running tests can be tedious and costly. Furthermore, it is difficult to write tests for all possible cases therefore, obscure bugs may not be detected. An automated test-case generator called QuickCheck was implemented in Haskell to solve these issues. This tool generates random input values to create a large number of tests.

Whiley is a programming language, developed to verify code and eliminate errors using formal specifications. Currently, the verifier has limitations with evaluating complex pre- and post-conditions. For example, a post-condition could be identified as not holding by the verifier even though it is a valid post-condition. This project aims to implement an automated test-case generator in Whiley to increase confidence in unverifiable code and ease testing of Whiley code.

3. Proposed Solution

The automated test generator requires reading a Whiley program and creating tests from it.

Timeline

Trimester 1

Output	**Estimated Time**	**Complete by**
Produce bibliography	2 weeks	9/4/2018 (Week 5)
Produce project proposal	2 weeks	9/4/2018 (Week 5)
Implement an automated test-case generator for Whiley programs with only the types: bool, byte, int, real, null	3 weeks	16/4/2018 (Week 6)
Implement automated test-case generation for types: void, array, union and records	3 weeks	14/5/2018 (Week 9)
Produce preliminary report	3 weeks	10/6/2018 (Week 12)
Extend the automatic test generator to be able to generate other types or use methods for better test case distribution (weighting, classification).	3 weeks	2/7/2018 (Exam period)

Trimester 2

Output	Estimated Time	Complete by
Produce slides for presentation of preliminary report	2 weeks	16/7/2018 (Week 1)
Extend the automatic test generator to include other methods of testing (symbolic) or be able to generate other types	3 weeks	6/8/2018 (Week 4)
Produce draft of final report	4 weeks	15/9/2018 (Week 7)
Complete implementation of automated test generator	3 weeks	5/10/2018 (Week 10)
Finalise final report	2 weeks	21/10/2018 (Week 12)
Prepare slides for final presentation	3 weeks	16/11/2018 (Exam period)

4. Evaluating your Solution

To evaluate this tool, we will insert bugs into a small benchmark set and see whether they can be uncovered by the tool.

5. Resource Requirements

No special resources are required. Only the use of the ECS laboratories is required.