Undergraduate Research Opportunity Program

(UROP) Project Report

**Static Visualization of Program Dynamics**

By

Qiao Li

Department of Computer Science

School of Computing

National University of Singapore

2009/2010

Project No: 10

Advisor: Prof Chin Wei-Ngan

Deliverables:

Report: 1 Volume

Program: 1 Diskette

**Abstract**

Static debugging has been a challenge for many years as accuracy and completeness is often non-deterministic. Verification with specifications is one technique that uses formal rules to achieve a greater completeness for debugging at compile time. Our automated verification has proved effectiveness in many examples on verifying pointer-based programs and tracking shape, size and bag properties of various data structures. We propose to make use of the available information provided by the existing system to build a static debugger prototype. We started with a basic Graphical User Interface to visualize information of program dynamics in a more user-friendly manner. This GUI aims to help the development of our verification system because its features are implemented based on our own experiences of developing and using the system. With minimal changes on the existing system, the prototype uses program mark-ups such as highlighting to present debugging information. At the same time, we also explore other ways to improve on the current prototype to build a static debugger with our verification techniques.

**Subject Descriptors:**

D.2.4 Software/Program Verification

D.2.5 Testing and Debugging

H.5.2 User Interfaces

**Keywords:**

Verification, debugging, GUI

**Acknowledgement**

Firstly, I have to thank my advisor, Associate Prof. Chin Wei-Ngan for taking so much of his valuable time in clarifying my doubts and guiding me on this project. Without his help, I would not be able to learn the foundations in functional programming languages, understand the features of the automated system or even get started properly on this project. This report would not be possible without his guidance too. Also, I would like to thank Christian Gherghina for assisting me with hardware problems, understanding the implementation of the system and help adding in labels. Lastly, Cristina David has been a nice lab mate in helping me with some syntax problems in functional programming languages at the beginning of the project. I indeed appreciate this learning experience and exploring process.

**List of Figures**

Figure 1 5

Figure 2 : Integrating GUI to the existing system 7

Figure 3: Illustration of program states for a sub-expression 10

Figure 4: Three main windows of the GUI 13

Figure 5: Verification Results Window 13

Figure 6: Highlighting expressions with different colors 15

Figure 7: Displaying program states and highlighting sub-expression 15

Figure 8: Zoom in/out at the same cursor position 16

Figure 9: Highlighting the problematic specifications with different colors 16

Figure 10: Focus on only one specific pre/postconditions 18

Figure 11: Sample predicates Appendix-A 1

Figure 12: Sample Procedures Appendix-A 1

Figure 13: Open Dialogue Appendix-A 2

Figure 14: Verification Results Window Appendix-A 2

Figure 15: Verification Results after Debugging Appendix-A 3

Figure 16: Function of Heap States Appendix-A 3

Figure 17: Command of assume sample Appendix-A 4

Figure 18: Sample information of heap states Appendix-A 5

Figure 19: Select a specification to focus on Appendix-A 5

Figure 20: Core program and Core window Appendix-A 6