COBSlicer : A tool to identify IT Rules in COBOL Programs

By

Manthan Doshi (ID No. 106017)

A project report submitted
In
Partial fulfillment of the requirement
for the degree of
BACHELOR OF TECHNOLOGY
In
Computer Engineering

Internal Guide

Dr. C. K. Bhensdadia

Professor and Head of Department,

Department of Computer Engineering,

Faculty Of Technology,

DDU, Nadiad.

External Guide

Mr. Pavan Chittimalli

Research Scientist

Tata Research Design and Development Centre, TCS Pune.

Faculty of Technology

Department of Computer Engineering

Dharmsinh Desai University.

April, 2014

Certificate

This is to certify that the project work titled,

COBSlicer : A tool to identify IT Rules in COBOL Programs

Is the bona fide work of

Manthan H. Doshi (ID No. 106017)

Carried out in the partial fulfillment of the degree of Bachelor of Technology in Computer Engineering at Dharmsinh Desai University in the academic session December 2013 to April 2014.

Dr. C. K. Bhensdadia
Professor and Head of Department,
Department of Computer Engineering,
Faculty of Technology,
Dharmsinh Desai University,
Nadiad-387001.

Dr. C. K. Bhensdadia
Head of Department,
Department of Computer Engineering,
Faculty of Technology,
Dharmsinh Desai University,
Nadiad-387001.

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University

ABSTRACT

Large Software programs frequently require decomposition for understanding and manipulation by people. However, not just any decomposition is useful to people. If decomposition is done with some context (viz., Data Flow, Control Flow) then the purpose of decomposition can be met effectively. Program Slicing is one such Static Program Analysis technique of decomposition wherein some behaviour of software is taken as input along with some context like Data flow or Control flow, and part of the behaviour that is compliant with specified context is conceded as output.

Program Slicing originally described by Mark Weiser, has underwent many changes and today many approaches are being used to achieve the same. In this study I have used the approach of Program Dependency Graph(PDG) and System Dependency Graph (SDG) based Program Slicing as first suggested by Horwitz *et al*. [2] and COBOL as input program Language. This approach builds PDG per procedure of COBOL project and a SDG for entire project(multi-procedure). This COBOL project may encapsulate part or whole of software behaviour. In Layman's term, PDG or SDG is a graph with nodes representing program statements and edges representing data and control flow among those nodes. Once graphs are built, slicing is analogous to conditional traversal of graph, beginning from some seed statement and context.

Program Slicing using Program Dependency Graph as intermediate representation was found to be more convenient to apply , compared to original technique that used Control Flow Graph as intermediate representation. Precision of slices obtained could be improved greatly by appropriately labelling edges of graph. Finding different slices like Data Slice, Control Slice and Program Slice using PDG, then merely requires modifying traversal logic of already built PDG.

PDG based Program Slicing could be used to determine which statements of code affect given seed statement and which statements get affected by seed statement. Thus, providing a way to understand dependency among code statements in both forward as well as backward direction. Due to such convenience, program slicing finds wide application in areas like Debugging, Regression Testing, Reverse Engineering.

ACKNOWLEDGEMENTS

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely fortunate to have received constant support all along.

I take this opportunity to express my profound gratitude and deep regards to my external mentors **Pavan Chittimalli** and **Ravindra Naik** for their exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by them from time to time shall carry me long way in journey of life on which I am about to embark.

I also take the opportunity to express deep sense of gratitude to the staff members of **TRDDC**, Pune, for the valuable support provided by them in their respective fields. I am grateful for their cooperation during the course of my project.

I am obliged to my Prof. Dr. C.K.Bhensdadia , Head of Department Of Computer Science, DDIT, for considering me and giving me a chance to persue internship at extremely prestigious Research Facility of TCS.

Lastly, I am grateful to every individual whom I came across during my course and who encouraged me one or other way.

Manthan H.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF CODE SNIPPETS