SELECTION OF AN INTERMEDIATE REPRESENTATION FOR PROGRAM ANALYSIS AND OPTIMIZATION

by

Amar Man Maharjan

A dissertation submitted to the
Central Department of Computer Science and Information Technology,
Tribhuvan University
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Information Technology

TRIBHUVAN UNIVERSITY
Kirtipur, Kathmandu
Nepal

December 2007

Acknowledgements

I am truly thankful to my supervisor and respected teacher Prof. Dr. Shasidhar Ram Joshi, Department of Electronics and Computer Engineering, Institute of Engineering, Pulchowk, for his immeasurable guidance and support to this thesis. I also thank to the Head of the Central Department of Computer Science and Information Technology (CDCSIT) Dr. Tanka Nath Dhamala. Special thanks to my co-supervisor Mr. Samujjwal Bhandari for his kind cooperation, insightful ideas, advice and suggestions.

Many friends and colleagues have contributed greatly to this thesis. Mr. Jagdish Bhatta is the special one who helped me in great deal. Much gratitude to my friends Mr. Bardan S.J.B. Rana, and Mr. Shishir Paudyal also for their incomparable support.

I thank my teachers Prof. Dr. Devi Dutta Paudyal, Prof. Dr. Onkar P. Sharma (Marist College, USA), Asst. Prof. Dr. Subarna Shakya, Asst. Prof. Sudharshan Karanjit, Asst. Prof. Arun Timalsina, Mr. Bishnu Gautam, Mr. Hemanta Bahadur G.C. for their invaluable knowledge, support and confidence during my master degree.

CDCSIT's laboratory and its members have provided an excellent environment. In particular, my appreciation goes to Mr. Niraj Manandhar for his technical support. I also thank to all the people who helped me in this thesis directly or indirectly.

And finally, I thank my loving parents and family for their tireless support and patience throughout my life and studies.

Amar Man Maharjan
2007/12/4

Abstract

An Intermediate Representation (IR) is an important part of a compiler. Selecting the right IR can significantly improve not only analyses and optimizations processes of a compiler but also reduce overall time of compiler design. There are many IRs found today but selecting the right IR for compiler is difficult job because different IRs have different properties. In this dissertation, two important IRs, Static Single Assignment (SSA) and Program Dependence Graph (PDG), are studied and presented comparative analyses between PDG and three flavors of SSA form: minimal, pruned and semi-pruned. Selected IRs are implemented in the Machine SUIF compiler infrastructure. PDG pass is implemented in this work but has used Machine SUIF Static Single Assignment Library of Machine SUIF for SSA form. Selected IRs are tested and analyzed with benchmark programs. The results showed that the comparative study presented in this work is very useful to the compiler designer for selecting appropriate IR.

TABLE OF CONTENTS

List of Figures

List of Table

List of Abbreviations

IR	Intermediate Representation
SSA	Static Single Assignment
PDG	Program Dependence Graph
SUIF	Stanford University Intermediate Format
MIR	Middle Level Intermediate Representation
AST	Abstract Syntax Tree
DAG	Directed Acyclic Graph
CFG	Control Flow Graph
DDG	Data Dependence Graph
SSI	Static Single Information
VDG	Value Dependence Graph
DFG	Dependence Flow Graph
PDW	Program Dependence Web
VCG	Visualization for Compiler Graphs
OPI	Optimization Programming Interface
NCI	National Compiler Infrastructure
FORTRAN	Formula Translation
SuifEnv	SUIF Environment