CS 6241: Compiler Design

Refined Def-Use Analysis Implementation Using Infeasible Paths Information

Chayne Thrash and Mansour Alharthi

April 7, 2018

Instructor: Professor Santosh Pande

1 Introduction

In this project, we first implemented an algorithm that detects both the inter- and intra-procedural infeasible paths using branch correlation. Utilizing this information, we implemented a demand-driven def-use analysis that discards the chains along infeasible paths and therefor give a more accurate data flow information. Both algorithms were introduced in the paper ”Refining Data Flow Information Using Infeasible Paths” by Gupta, Soffa, and Bodik. Finally, we improved the introduced infeasible path detection algorithm by accounting for more cases than the ones mentioned in the paper. We show our improved version with test results in this report.

2 Improved Infeasible Paths Detection

We improved the algorithm presented in the paper by accounting for more cases during the backward propagation while detecting infeasible paths.

3 Test Results

Intra-procedural Infeasible Paths Detection	Intra-procedural Demand-Driven Def-Use Analysis	Inter-procedural Infeasible Paths Detection	Inter-procedural Demand-Driven Def-Use Analysis	Node
Chayne	Mansour	Chayne	Mansour	Chayne

Table 1: breakdown of work among team members

4 Work Breakdown

Intra-procedural Infeasible Paths Detection	Intra-procedural Demand-Driven Def-Use Analysis	Inter-procedural Infeasible Paths Detection	Inter-procedural Demand-Driven Def-Use Analysis	Node
Chayne	Mansour	Chayne	Mansour	Chayne

Table 2: breakdown of work among team members