CS 6241: Compiler Design

Solving Constant Propagation Data-flow Problem Using Detected Destructive Merges Information

Chayne Thrash and Mansour Alharthi

April 18, 2018

Instructor: Professor Santosh Pande

1 Introduction

In this project, we implemented the algorithm introduced in paper "Comprehensive Path-sensitive Data-flow Analysis" by Thakur and Govindarajan. The algorithm first detects the nodes where two or more paths merge; causing constant variables defined in one or more of those paths no longer feasible for propagating. After that, the algorithm look for the influenced nodes; the nodes of which will be optimized in case the destructive merge is eliminated. Finally, the algorithm duplicates the nodes that are in the *region of influence*, and thus eliminating the destructive merge allowing for the constant propagation to the influenced nodes. The algorithm represents a trade-off between code size and data-flow precision. We apply the algorithm on only the two top fittest destructive merges, as per the definition of fitness in the paper.

2 Test Results

We tested our implementation on two benchmarks. We compare our implementation with sparse conditional constant propagation technique by Wegman-Zadeck implemented in LLVM.

Benchmark	Static size with sparse conditional constant propagation	Static size with destructive merge elimination	% of increase in static size
B1	A	B	C
B2	D	E	F

Table 1: Static size comparison

Benchmark	Constants propagated# with sparse conditional constant propagation	Constants propagated# with destructive merge elimination	% of Constants propagated increased
B1	A	B	C
B2	D	E	F

Table 2: Constants propagated comparison

Benchmark	Execution time with sparse conditional constant propagation	Execution time with destructive merge elimination	% of improvement in Execution time
B1	A	B	C
B2	D	E	F

Table 3: Execution time comparison

3 Work Breakdown

Destructive Merges Detection	Split Graph and CFG Reconstruction	Preliminary Reachability Analysis	Report	Integrating and testing
Mansour	Chayne	Chayne	Mansour	

Table 4: breakdown of work among team members