Compilers - Second Assignment

Authors

Menozzi Matteo
Turci Gabriele
Turci Sologni Enrico

April 8, 2025

Contents

1 Very Busy Expressions

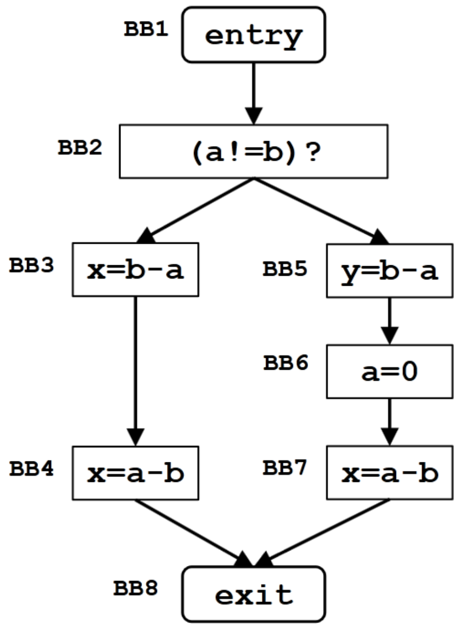
The first point asks to create a general framework for identifying Very Busy Expressions (VBE) and apply this framework to a given problem, which is described using a Data Flow Graph (DFG). The graph can be found in the assignment slides.

1.1 Problem Definition

An expression is considered "very busy" at a point p if, no matter which path is taken from p, it is calculated before any of its operands get a new value.

For example, the expression $b - a$ is very active at point p if it is always calculated along every possible path from p to EXIT, without a or b being redefined along those paths.

We are interested in finding the set of expressions that are available at the start of block B. This set depends on the paths that lead from p to block B.



1.2 Dataflow Analysis

Very Busy Expressions - DFA Framework	
Domain	Sets of Expressions
Direction	Backward: $in[b] = f_b(out[b])$ $out[b] = \wedge\ in[succ(b)]$
Transfer function	$f_b(x) = Gen_b \cup (x - Kill_b)$
Meet operation ($\wedge$)	$\cap$
Boundary Condition	$in[exit] = \varnothing$
Initial interior points	$in[b] = \mathbb{U}$

Very Busy Expressions - Iterations				
	1° Iteration		**2° Iteration**	
	IN[B]	OUT[B]	IN[B]	OUT[B]
BB1	$\{b-a\}$	$\{b-a\}$	$\{b-a\}$	$\{b-a\}$
BB2	$\{b-a\}$	$\{b-a\}$	$\{b-a\}$	$\{b-a\}$
BB3	$\{a-b,\ b-a\}$	$\{a-b\}$	$\{a-b,\ b-a\}$	$\{a-b\}$
BB4	$\{a-b\}$	$\{\varnothing\}$	$\{a-b\}$	$\{\varnothing\}$
BB5	$\{\ b-a\}$	$\{\varnothing\}$	$\{\ b-a\}$	$\{\varnothing\}$
BB6	$\{\varnothing\}$	$\{a-b\}$	$\{\varnothing\}$	$\{a-b\}$
BB7	$\{a-b\}$	$\{\varnothing\}$	$\{a-b\}$	$\{\varnothing\}$
BB8	$\{\varnothing\}$	$\{\varnothing\}$	$\{\varnothing\}$	$\{\varnothing\}$

The algorithm stops after the second iteration because the input set from each basic block doesn't change between the first and second iterations.

2 Dominator Analysis

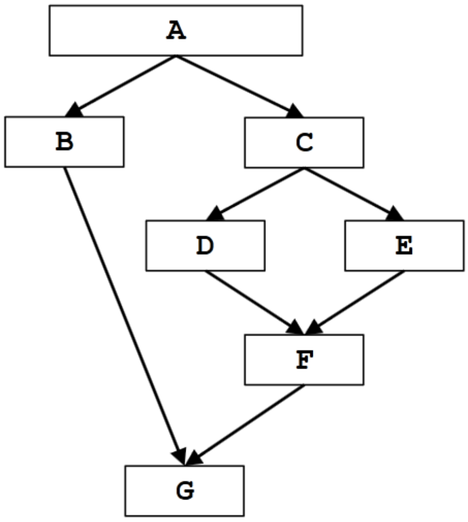
The second point asks to create a general framework to perform Dominator Analysis (DA) and apply this framework to a given problem, which is described using a Data Flow Graph (DFG). The graph can be found in the assignment slides.

2.1 Problem Definition

In a control flow graph (CFG), we say that a node A dominates another node B if A is present in every possible path from the ENTRY block to B.

Each BasicBlock B_i is associated to a set $DOM[B_i]$. A block B_j is included in $DOM[B_i]$ if B_j dominates B_i.

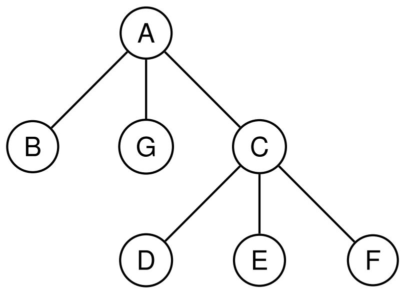
It's important that a node always dominates itself; B_i will always be part of $DOM[B_i]$.



2.2 Dataflow Analysis

Dominator Analysis - DFA Framework	
Domain	Sets of Basic Blocks
Direction	Forward: $out[b] = f_b(in[b])$ $in[b] = \wedge\ out[pred(b)]$
Transfer function	$f_b(x) = Def_b \cup x$
Meet operation ($\wedge$)	$\cap$
Boundary Condition	$out[entry] = \varnothing$
Initial interior points	$out[b] = \mathbb{U}$

Dominator Analysis - Iterations				
	1° Iteration		2° Iteration	
	IN[B]	OUT[B]	IN[B]	OUT[B]
A	$\{\varnothing\}$	$\{A\}$	$\{\varnothing\}$	$\{A\}$
B	$\{A\}$	$\{A, B\}$	$\{A\}$	$\{A, B\}$
C	$\{A\}$	$\{A, C\}$	$\{A\}$	$\{A, C\}$
D	$\{A, C\}$	$\{A, C, D\}$	$\{A, C\}$	$\{A, C, D\}$
E	$\{A, C\}$	$\{A, C, E\}$	$\{A, C\}$	$\{A, C, E\}$
F	$\{A, C\}$	$\{A, C, F\}$	$\{A, C\}$	$\{A, C, F\}$
G	$\{A\}$	$\{A, G\}$	$\{A\}$	$\{A, G\}$



Associated dominator tree.

The algorithm stops after the second iteration because the output set from each basic block doesn't change between the first and second iterations.

3 Constant Propagation

The third point asks to create a general framework for performing Constant Propagation (CP) analysis and apply this framework to a given problem, which is described using a Data Flow Graph (DFG). The graph can be found in the assignment slides.

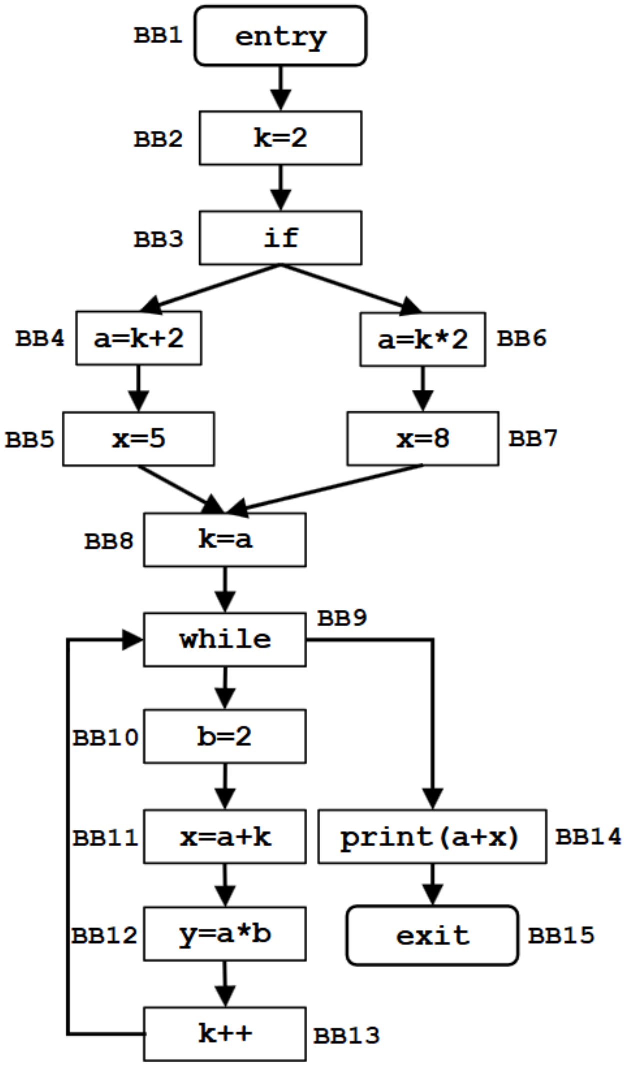
3.1 Problem Definition

Constant propagation is a technique aimed at identifying where variables in a program have fixed values.

For each node in the control flow graph (CFG), we calculate an associated set of $(variable, constant)$ pairs.

When a pair (v, c) is present at a node, it means that v will always have the value c whenever that particular node is reached during the program's execution.

Constant Propagation can determine the constant value of binary expressions where one or both operands are variables with known constant values.



3.2 Dataflow Analysis

Constant Propagation - DFA Framework	
Domain	Sets of Pairs $(var, const)$
Direction	Forward: $out[b] = f_b(in[b])$ $in[b] = \wedge\ out[pred(b)]$
Transfer function	$f_b(x) = Gen_b \cup (x - Kill_b)$
Meet operation ($\wedge$)	$\cap$
Boundary Condition	$out[entry] = \varnothing$
Initial interior points	$out[b] = \mathbb{U}$

Constant Propagation - Iterations				
	1° Iteration		2° Iteration	
	IN[B]	OUT[B]	IN[B]	OUT[B]
BB1	$\{\varnothing\}$	$\{\varnothing\}$	$\{\varnothing\}$	$\{\varnothing\}$
BB2	$\{\varnothing\}$	$\{(k,2)\}$	$\{\varnothing\}$	$\{(k,2)\}$
BB3	$\{(k,2)\}$	$\{(k,2)\}$	$\{(k,2)\}$	$\{(k,2)\}$
BB4	$\{(k,2)\}$	$\{(k,2),(a,4)\}$	$\{(k,2)\}$	$\{(k,2),(a,4)\}$
BB5	$\{(k,2),(a,4)\}$	$\{(k,2),(a,4),(x,5)\}$	$\{(k,2),(a,4)\}$	$\{(k,2),(a,4),(x,5)\}$
BB6	$\{(k,2)\}$	$\{(k,2),(a,4)\}$	$\{(k,2)\}$	$\{(k,2),(a,4)\}$
BB7	$\{(k,2),(a,4)\}$	$\{(k,2),(a,4),(x,8)\}$	$\{(k,2),(a,4)\}$	$\{(k,2),(a,4),(x,8)\}$
BB8	$\{(k,2),(a,4)\}$	$\{(k,4),(a,4)\}$	$\{(k,2),(a,4)\}$	$\{(k,4),(a,4)\}$
BB9	$\{(k,4),(a,4)\}$	$\{(k,4),(a,4)\}$	$\{(a,4)\}$	$\{(a,4)\}$
BB10	$\{(k,4),(a,4)\}$	$\{(k,4),(a,4),(b,2)\}$	$\{(a,4)\}$	$\{(a,4),(b,2)\}$
BB11	$\{(k,4),(a,4),(b,2)\}$	$\{(k,4),(a,4),(b,2),(x,8)\}$	$\{(a,4),(b,2)\}$	$\{(a,4),(b,2)\}$
BB12	$\{(k,4),(a,4),(b,2),(x,8)\}$	$\{(k,4),(a,4),(b,2),(x,8),(y,8)\}$	$\{(a,4),(b,2)\}$	$\{(a,4),(b,2),(y,8)\}$
BB13	$\{(k,4),(a,4),(b,2),(x,8),(y,8)\}$	$\{(k,5),(a,4),(b,2),(x,8),(y,8)\}$	$\{(a,4),(b,2),(y,8)\}$	$\{(a,4),(b,2),(y,8)\}$
BB14	$\{(k,4),(a,4)\}$	$\{(k,4),(a,4)\}$	$\{(a,4)\}$	$\{(a,4)\}$
BB15	$\{(k,4),(a,4)\}$	$\{(k,4),(a,4)\}$	$\{(a,4)\}$	$\{(a,4)\}$

Constant Propagation - Iterations		
	3° Iteration	
BB1	$\{\varnothing\}$	$\{\varnothing\}$
BB2	$\{\varnothing\}$	$\{(k,2)\}$
BB3	$\{(k,2)\}$	$\{(k,2)\}$
BB4	$\{(k,2)\}$	$\{(k,2),(a,4)\}$
BB5	$\{(k,2),(a,4)\}$	$\{(k,2),(a,4),(x,5)\}$
BB6	$\{(k,2)\}$	$\{(k,2),(a,4)\}$
BB7	$\{(k,2),(a,4)\}$	$\{(k,2),(a,4),(x,8)\}$
BB8	$\{(k,2),(a,4)\}$	$\{(k,4),(a,4)\}$
BB9	$\{(a,4)\}$	$\{(a,4)\}$
BB10	$\{(a,4)\}$	$\{(a,4),(b,2)\}$
BB11	$\{(a,4),(b,2)\}$	$\{(a,4),(b,2)\}$
BB12	$\{(a,4),(b,2)\}$	$\{(a,4),(b,2),(y,8)\}$
BB13	$\{(a,4),(b,2),(y,8)\}$	$\{(a,4),(b,2),(y,8)\}$
BB14	$\{(a,4)\}$	$\{(a,4)\}$
BB15	$\{(a,4)\}$	$\{(a,4)\}$

The algorithm stops after the third iteration because the output set from each basic block doesn't change between the second and third iterations.