Second Assignment

Andrea Lavino

March 2025

Contents

1 Very Busy Expressions

The search for very busy expressions can be useful for code hoisting, since very busy expressions can be moved from the place they are up to a joint point from which the flow departures.

1.1 Problem formalization

Formally an expression is said to be **very busy** when it is computed along each path that part from the point p without any redefinition of its operands. This information can be used to move the expression to a point of the code in which is computation can be used by all the paths that use the expression.

	Very Busy Expressions
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	$out[exit] = \varnothing$
Initial interior points	$out[b] = U$

Table 1: Very busy expressions summary table

1.2 Example

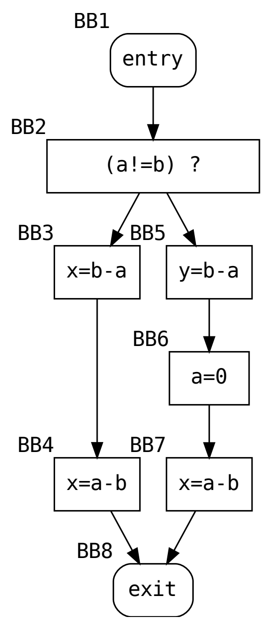

Figure 1: Very Busy Expression Example

	Iteration 1		**Iteration 2**	
	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	$\{b-a\},\{a-b\}$	$\{a-b\}$	$\{b-a\},\{a-b\}$	$\{a-b\}$
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	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

Table 2: Very Busy Expression Algorithm Execution Table

2 Dominator Analysis

Dominator analysis is fundamental to create the single static assignment form.

2.1 Problem formalization

A basic block B_1 **dominates** another block B_2 if it is encountered in every path from entry to B_2.

	Dominator Analysis
Domain	Sets of Basic Blocks
Direction	Forward
Transfer function	$f_b(x) = \{x\} \cup (\bigcap_{m \in preds(x)} f_b(m))$
Meet Operation ($\wedge$)	$\cap$
Boundary Condition	$Dom[entry] = \{entry\}$
Initial interior points	$Dom[b] = N \quad \forall b \neq entry$, with N the number of basic blocks of the CFG

Table 3: Dataflow Problem X Properties

2.2 Example

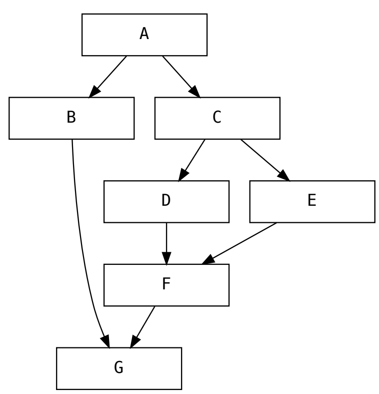

Figure 2: Dominance Analysis example

	DOM[B]
A	$\{A\}$
B	$\{A, B\}$
C	$\{A, C\}$
D	$\{A, C, D\}$
E	$\{A, C, E\}$
F	$\{A, C, F\}$
G	$\{A, G\}$

Table 4: Dominance analysis execution table

3 Constant Propagation

The constant propagation problem aims at finding what are the couples $< variable, constant\ value >$ that are availables in a certain basic block, so that the variable constant value can be propagated across the blocks.

3.1 Problem formalization

We say that a couple $< variable, constant >$ is valid at block n if it is guaranteed that the variable x gets that constant value every time that the block is reached.

	Constant Propagation
Domain	Sets of variables and their constant values
Direction	Forward $in[b] = \wedge(out[pred(b)])$ $out[b] = f_b(in[b])$
Transfer function	$f_b(x) = Gen_b \cup (x - Kill_b)$
Meet Operation ($\wedge$)	$\cap$
Boundary Condition	$out[entry] = \varnothing \quad in[entry] = \varnothing$
Initial interior points	$out[b] = \varnothing \quad in[b] = U$

Table 5: Constant Propagation Problem Summary Table

3.2 Example

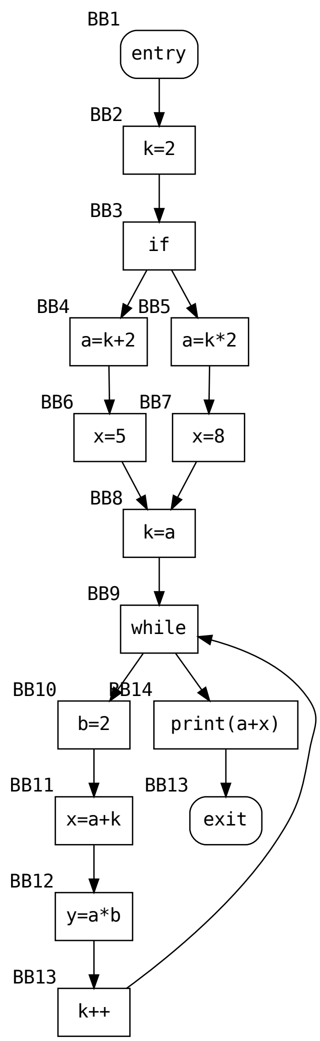

Figure 3: Constant Propagation example

	Iteration 1	
	IN[B]	**OUT[B]**
BB1	$\varnothing$	$< k, 2 >$
BB2	$\varnothing$	$< k, 2 >$
BB3	$< k, 2 >$	$< k, 2 >$
BB4	$< k, 2 >$	$< k, 2 >, < a, 4 >$
BB5	$< k, 2 >$	$< k, 2 >, < a, 4 >$
BB6	$< k, 2 >, < a, 4 >$	$< k, 2 >, < a, 4 >, < x, 5 >$
BB7	$< k, 2 >, < a, 4 >$	$< k, 2 >, < a, 4 >, < x, 8 >$
BB8	$< k, 2 >, < a, 4 >$	$< k, 4 >, < a, 4 >$
BB9	$< k, 2 >, < a, 4 >$	$< k, 2 >, < a, 4 >$
BB10	$< k, 4 >, < a, 4 >$	$< k, 2 >, < a, 4 >, < b, 2 >$
BB11	$< k, 4 >, < a, 4 >, < b, 2 >$	$< k, 4 >, < a, 4 >, < b, 2 >, < x, 8 >$
BB12	$< k, 4 >, < a, 4 >,$ $< b, 2 >, < x, 8 >$	$< k, 2 >, < a, 4 >,$ $< b, 2 >, < x, 8 >, < y, 8 >$
BB13	$< k, 4 >, < a, 4 >,$ $< b, 2 >, < x, 8 >, < y, 8 >$	$< k, 5 >, < a, 4 >,$ $< b, 2 >, < x, 8 >, < y, 8 >$
BB14	$< k, 4 >, < a, 4 >$	$< k, 4 >, < a, 4 >$
BB15	$< k, 4 >, < a, 4 >$	$< k, 4 >, < a, 4 >$

Table 6: Constant Propagation Algorithm Execution Table (Iteration 1)

	Iteration 2	
	IN[B]	**OUT[B]**
BB1	$\varnothing$	$< k, 2 >$
BB2	$\varnothing$	$< k, 2 >$
BB3	$< k, 2 >$	$< k, 2 >$
BB4	$< k, 2 >$	$< k, 2 >, < a, 4 >$
BB5	$< k, 2 >$	$< k, 2 >, < a, 4 >$
BB6	$< k, 2 >, < a, 4 >$	$< k, 2 >, < a, 4 >, < x, 5 >$
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	$< a, 4 >, < b, 2 >$	$< k, 8 >, < a, 4 >, < b, 2 >$
BB12	$< a, 4 >, < b, 2 >$	$< a, 4 >, < b, 2 >, < y, 8 >$
BB13	$< a, 4 >, < b, 2 >, < y, 8 >$	$< k, 5 >, < a, 4 >, < b, 2 >, < y, 8 >$
BB14	$< a, 4 >$	$< a, 4 >$
BB15	$< a, 4 >$	$< a, 4 >$

Table 7: Constant Propagation Algorithm Execution Table (Iteration 2)