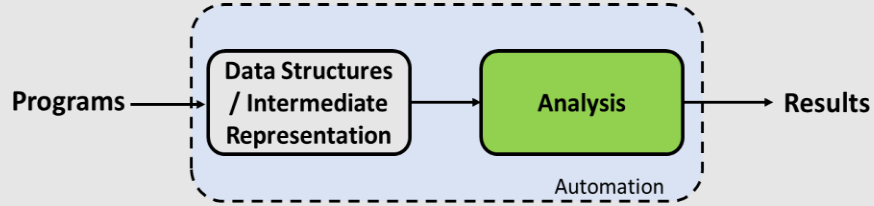
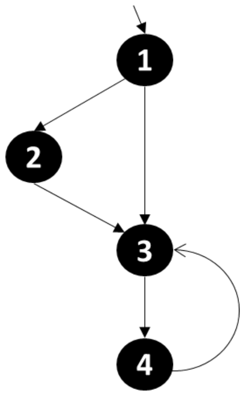




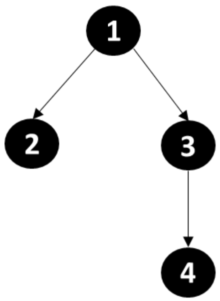
Program Analysis

Venkatesh Vinayakarao

venkateshv@cmi.ac.in

Mar – Apr, 2018

Chennai Mathematical Institute



"The theory of strings with concatenation has been widely argued as the basis of constraint solving for verifying string-manipulating programs. However, this theory is far from adequate for expressing many string constraints that are also needed in practice…"
– **Chen et al., POPL 2018.**

Quick Review – Program Representations

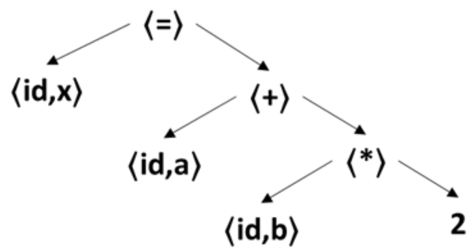
Lexemes

x = a + 2*b

Tokens

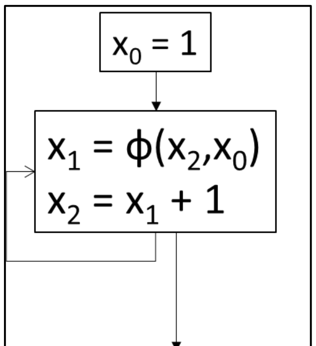
[(identifier, x), (operator, =), (identifier, a), (operator, +), (identifier, b), (operator, *), (literal, 2), (separator, ;)]

Abstract Syntax Tree

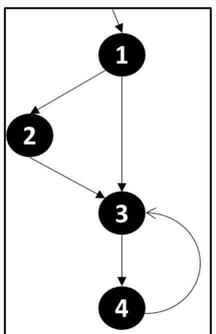


Three Address Code

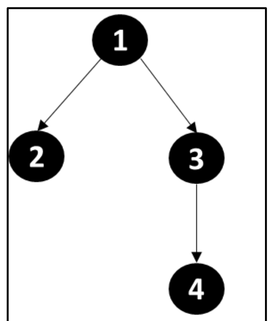
x+y*z → t1 = y * z
t2 = x + t1



Graph Representation with SSA Form



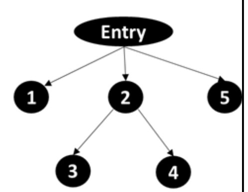
A Flow Graph

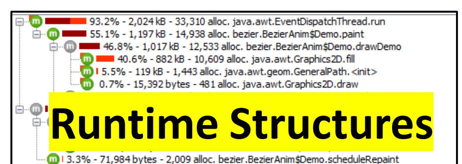


Dominator Tree

Control Dependence

$[x = 10]^1$;
if $[(x > k)]^2$
$[y = 10]^3$;
else
$[y = 1]^4$;
$[\text{print } y]^5$;

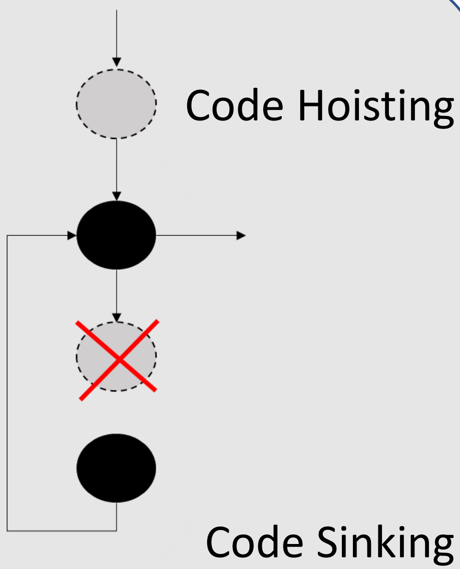




Tools

Eclipse JDT
Soot
JProfiler
Z3

Quick Review – Popular Compiler Optimizations



Loop Invariant Code Motion

```
int a = 30;
int b = 9 - (a / 5);
int c;

c = b * 4;
if (c > 10) {
   c = c - 10;
}
return c * (60 / a);
```

=

```
return 4;
```

Constant Folding and Propagation

$$\text{prog}: I_{fixed} \times I_{dynamic} \rightarrow O$$

$$\downarrow$$

$$\text{prog}^*: I_{dynamic} \rightarrow O$$

Partial Evaluation

Before	After
y = x / 8	y = x >> 3
y = x * 64	y = x << 6
y = x * 2	y = x << 1

Strength Reduction

Peephole Optimization

```
float x,y,a,b,c,d;
…
x = (a/b)*c;
y = (a/b)*d;
```

Common Subexpressions

Quick Review

```
[y = x]^1;
[z = 1]^2;
while [(y > 0)]^3 {
    [z = z * y]^4;
    [y = y – 1]^5;
}
[y = 0]^6
```

Input Program

Data Flow Analysis

F(RD)

$RD_{entry}(1) = \{(x,?),(y,?),(z,?)\}$
$RD_{entry}(2) = RD_{exit}(1)$
$RD_{entry}(3) = RD_{exit}(2) \cup RD_{exit}(5)$
$RD_{entry}(4) = RD_{exit}(3)$
$RD_{entry}(5) = RD_{exit}(4)$
$RD_{entry}(6) = RD_{exit}(3)$

$RD_{exit}(1) = (RD_{entry}(1) \setminus \{ (y,\ell) \mid \ell \in \textbf{Lab} \}) \cup \{ (y,1) \}$
$RD_{exit}(2) = (RD_{entry}(2) \setminus \{ (z,\ell) \mid \ell \in \textbf{Lab} \}) \cup \{ (z,2) \}$
$RD_{exit}(3) = RD_{entry}(3)$
$RD_{exit}(4) = (RD_{entry}(4) \setminus \{ (z,\ell) \mid \ell \in \textbf{Lab} \}) \cup \{ (z,4) \}$
$RD_{exit}(5) = (RD_{entry}(5) \setminus \{ (y,\ell) \mid \ell \in \textbf{Lab} \}) \cup \{ (y,5) \}$
$RD_{exit}(6) = (RD_{entry}(6) \setminus \{ (y,\ell) \mid \ell \in \textbf{Lab} \}) \cup \{ (y,6) \}$

F(RD)

ℓ	$RD_{entry}(\ell)$	$RD_{exit}(\ell)$
1	(x,?),(y,?),(z,?)	(x,?),(y,1),(z,?)
2	(x,?),(y,1),(z,?)	(x,?),(y,1),(z,2)
3	(x,?),(y,1),(z,2)(z,4),(y,5)	(x,?),(y,1),(z,2),(z,4),(y,5)
4	(x,?),(y,1),(z,2)(z,4),(y,5)	(x,?),(y,1),(z,4),(y,5)
5	(x,?),(y,1),(z,4),(y,5)	(x,?),(y,5),(z,4)
6	(x,?),(y,1),(z,2),(z,4),(y,5)	(x,?),(y,6),(z,2),(z,4)

ℓ	$RD_{entry}(\ell)$	$RD_{exit}(\ell)$
1	(x,?),(y,?),(z,?)	(x,?),(y,1),(z,?)
2	(x,?),(y,1),(z,?)	(x,?),(y,1),(z,2)
3	(x,?),(y,1),(z,2)(z,4),(y,5)	(x,?),(y,1),(z,2),(z,4),(y,5)
4	(x,?),(y,1),(z,2)(z,4),(y,5)	(x,?),(y,1),(z,4),(y,5)
5	(x,?),(y,1),(z,4),(y,5)	(x,?),(y,5),(z,4)
6	(x,?),(y,1),(z,2),(z,4),(y,5)	(x,?),(y,6),(z,2),(z,4)

Analysis Result

Data Flow Analysis

Classic Four Analyses

Classical Data Flow Analyses

- Reaching Definitions
- Available Expressions
- Very Busy Expressions
- Live Variables

The Setup: Some Preliminaries

- *The initial label of a statement*.

init: Stmt -> Lab

$\text{init}([x:=a]^\ell) = \ell$

$\text{init}([S_1;S_2]^\ell) = \text{init}(S_1)$

$\text{init}([\text{skip}]^\ell) = \ell$

$\text{init}(\text{if } [b]^\ell \text{ then } S_1 \text{ else } S_2) = \ell$

$\text{init}(\text{while } [b]^\ell \text{ do } S) = \ell$

Setup

- **Final Labels**

final: Stmt -> P(Lab)

$\text{final}([x{:=}a]^\ell) = \{\ell\}$
$\text{final}([S_1;S_2]^\ell) = \text{final}(S_2)$
$\text{final}([\text{skip}]^\ell) = \{\ell\}$
$\text{final}(\text{if } [b]^\ell \text{ then } S_1 \text{ else } S_2) = \text{final}(S_1) \cup \text{final}(S_2)$
$\text{final}(\text{while } [b]^\ell \text{ do } S) = \{\ell\}$

Setup

- We use Blocks to refer to set of statements

blocks: Stmt -> P(Blocks)

$\text{blocks}([x:=a]^\ell) = \{[x:=a]^\ell\}$
$\text{blocks}([S_1;S_2]^\ell) = \text{blocks}(S_1) \cup \text{blocks}(S_2)$
$\text{blocks}([\text{skip}]^\ell) = \{[\text{skip}]^\ell\}$
$\text{blocks}(\text{if } [b]^\ell \text{ then } S_1 \text{ else } S_2) = \{[b]^\ell\} \cup \text{blocks}(S_1) \cup \text{blocks}(S_2)$
$\text{blocks}(\text{while } [b]^\ell \text{ do } S) = \{[b]^\ell\} \cup \text{blocks}(S)$

Setup

- We refer to a statement with a label

labels: Stmt -> P(Lab)

labels(S) = { ℓ | $[B]^{\ell}$ ∊ blocks(S) }
init(S) ∊ labels(S)
final(S) ⊆ labels(S)

Setup

- The edges of our flow graphs are captured using a flow function.

flow: Stmt -> P(Lab x Lab)

$\text{flow}([x{:=}a]^\ell) = \emptyset$

$\text{flow}(S_1;S_2) = \text{flow}(S_1) \cup \text{flow}(S_2) \cup \{(\ell, \text{init}(S_2)) \mid \ell \in \text{final}(S_1)\}$

$\text{flow}([\text{skip}]^\ell) = \emptyset$

$\text{flow}(\text{if } [b]^\ell \text{ then } S_1 \text{ else } S_2) =$

$\quad \text{flow}(S_1) \cup \text{flow}(S_2) \cup \{(\ell, \text{init}(S_1)), (\ell, \text{init}(S_2))\}$

$\text{flow}(\text{while } [b]^\ell \text{ do } S) = \text{flow}(S) \cup \{(\ell, \text{init}(S))\} \cup \{(\ell', \ell) \mid \ell' \in \text{final}(S)\}$

---

flow denotes forward flow here.

Example

- Program power is given below:

$$[z:=1]^1; \text{ while } [x>0]^2 \text{ do } ([z:=z*y]^3; [x:=x-1]^4)$$

What are $\text{init}(\text{power})$, $\text{final}(\text{power})$, $\text{labels}(\text{power})$ **and** $\text{flow}(\text{power})$**?**

Example

- Program power is given below:

$$[z{:=}1]^1;\ \text{while } [x>0]^2 \text{ do } ([z{:=}z*y]^3;\ [x{:=}x-1]^4)$$

What are $init(power)$, $final(power)$, $labels(power)$ and $flow(power)$?

$$init(power) = 1$$

$$final(power) = \{2\}$$

$$labels(power) = \{1,2,3,4\}$$

$$flow(power) = \{(1,2),(2,3),(3,4),(4,2)\}$$

Label Consistency Assumption

- All blocks are uniquely labeled.

$[B_1]^\ell, [B_2]^\ell \in \text{blocks}(S) \Rightarrow B_1 = B_2$

Generalizing Data Flow Equations

Recall, RD Equations Were…

$RD_{entry}(1) = \{(x,?),(y,?),(z,?)\}$
$RD_{entry}(2) = RD_{exit}(1)$
$RD_{entry}(3) = RD_{exit}(2) \cup RD_{exit}(5)$
$RD_{entry}(4) = RD_{exit}(3)$
$RD_{entry}(5) = RD_{exit}(4)$
$RD_{entry}(6) = RD_{exit}(3)$

$RD_{exit}(1) = (RD_{entry}(1) \setminus \{ (y,\ell) \mid \ell \in \mathbf{Lab}\}) \cup \{ (y,1) \}$
$RD_{exit}(2) = (RD_{entry}(2) \setminus \{ (z,\ell) \mid \ell \in \mathbf{Lab}\}) \cup \{ (z,2) \}$
$RD_{exit}(3) = RD_{entry}(3)$
$RD_{exit}(4) = (RD_{entry}(4) \setminus \{ (z,\ell) \mid \ell \in \mathbf{Lab}\}) \cup \{ (z,4) \}$
$RD_{exit}(5) = (RD_{entry}(5) \setminus \{ (y,\ell) \mid \ell \in \mathbf{Lab}\}) \cup \{ (y,5) \}$
$RD_{exit}(6) = (RD_{entry}(6) \setminus \{ (y,\ell) \mid \ell \in \mathbf{Lab}\}) \cup \{ (y,6) \}$

Generalizing the Entry and Exit

$$RD_{entry}(\ell) = \begin{cases} \{ (x,?) \mid x \in Var_* \} & \text{if } \ell = init(S_*) \\ \bigcup \{RD_{exit}(\ell') \mid (\ell', \ell) \in flow(S_*)\} & \text{otherwise} \end{cases}$$

$$RD_{exit}(\ell) = (RD_{entry}(\ell) \setminus kill_{RD}(B^\ell)) \cup gen_{RD}(B^\ell)$$

$$\text{where } B^\ell \in blocks(S_*)$$

May Analysis

Forward Analysis

Least Solution Desired

The kill and gen Functions

$kill_{RD}([x:=a]^{\ell}) = \{(x,?)\} \cup \{(x, \ell') \mid B^{\ell'} \text{ is an assignment to x in } S^*\}$
$kill_{RD}([skip]^{\ell}) = \emptyset$
$kill_{RD}([b]^{\ell}) = \emptyset$

$gen_{RD}([x:=a]^{\ell}) = \{(x, \ell)\}$
$gen_{RD}([skip]^{\ell}) = \emptyset$
$gen_{RD}([b]^{\ell}) = \emptyset$

Graded Quiz

- Can you write the kill and gen sets for the following example?

```
x = 5;
y = 1;
while (x > 1) {
    y = x* y;
    x = x – 1;
}
```

- In our notation,

$[x:=5]^1$; $[y:=1]^2$; while $[x>1]^3$ do (....

The Kill and gen Sets

ℓ	$kill_{RD}(\ell)$	$gen_{RD}(\ell)$
1	{(x,?),(x,1),(x,5)}	{(x,1)}
2		
3		
4		
5		

$kill_{RD}([x:=a]^{\ell}) =$
{(x,?)} U
{(x, ℓ') | $B^{\ell'}$ is an assignment to x in S*}
$gen_{RD}([x:=a]^{\ell}) = \{(x, \ell)\}$

Labeled
Input Program

```
[x = 5]^1;
[y = 1]^2;
while [(x > 1)]^3 {
    [y = x * y]^4;
    [x = x – 1]^5;
}
```

The Kill and gen Sets

ℓ	$kill_{RD}(\ell)$	$gen_{RD}(\ell)$
1	{(x,?),(x,1),(x,5)}	{(x,1)}
2	{(y,?),(y,2),(y,4)}	{(y,2)}
3	∅	∅
4	{(y,?),(y,2),(y,4)}	{(y,4)}
5	{(x,?),(x,1),(x,5)}	{(x,5)}

Let us now write the flow equations and solve them to find the reaching definitions.

Labeled Input Program

```
[x = 5]^1;
[y = 1]^2;
while [(x > 1)]^3 {
    [y = x * y]^4;
    [x = x – 1]^5;
}
```

Flow Equations

$RD_{entry}(1) = \{(x,?),(y,?)\}$
$RD_{entry}(2) = RD_{exit}(1)$
$RD_{entry}(3) = RD_{exit}(2)\ U\ RD_{exit}(5)$
$RD_{entry}(4) = RD_{exit}(3)$
$RD_{entry}(5) = RD_{exit}(4)$

$RD_{exit}(1) = (RD_{entry}(1) \setminus \{(x,?),(x,1),(x,5)\})\ U\ \{ (x,1) \}$
$RD_{exit}(2) = (RD_{entry}(2) \setminus \{(y,?),(y,2),(y,4)\})\ U\ \{ (y,2) \}$
$RD_{exit}(3) = RD_{entry}(3)$
$RD_{exit}(4) = (RD_{entry}(4) \setminus \{(y,?),(y,2),(y,4)\})\ U\ \{ (y,4) \}$
$RD_{exit}(5) = (RD_{entry}(5) \setminus \{(x,?),(x,1),(x,5)\})\ U\ \{ (x,5) \}$

Solution after using Simple Iteration Method

ℓ	$RD_{entry}(\ell)$	$RD_{exit}(\ell)$
1	{(x,?),(y,?)}	{(y,?),(x,1)}
2	{(y,?),(x,1)}	{(x,1),(y,2)}
3	{(x,1),(y,2),(y,4),(x,5)}	{(x,1),(y,2),(y,4),(x,5)}
4	{(x,1),(y,2),(y,4),(x,5)}	{(x,1),(y,4),(x,5)}
5	{(x,1),(y,4),(x,5)}	{(y,4),(x,5)}

Labeled
Input Program

```
[x = 5]^1;
[y = 1]^2;
while [(x > 1)]^3 {
    [y = x * y]^4;
    [x = x – 1]^5;
}
```

Summary

$[x = 5]^1$;
$[y = 1]^2$;
while $[(x > 1)]^3$ {
    $[y = x * y]^4$;
    $[x = x - 1]^5$;
}

Input Program

Data Flow Analysis

ℓ	$\mathbf{kill_{RD}(\ell)}$	$\mathbf{gen_{RD}(\ell)}$
1	{(x,?),(x,1),(x,5)}	{(x,1)}
2	{(y,?),(y,2),(y,4)}	{(y,2)}
3	Ø	Ø
4	{(y,?),(y,2),(y,4)}	{(y,4)}
5	{(x,?),(x,1),(x,5)}	{(x,5)}

$RD_{entry}(1) = \{(x,?),(y,?)\}$
$RD_{entry}(2) = RD_{exit}(1)$
$RD_{entry}(3) = RD_{exit}(2) \cup RD_{exit}(5)$
$RD_{entry}(4) = RD_{exit}(3)$
$RD_{entry}(5) = RD_{exit}(4)$

$RD_{exit}(1) = (RD_{entry}(1) \setminus \{(x,?),(x,1),(x,5)\}) \cup \{ (x,1) \}$
$RD_{exit}(2) = (RD_{entry}(2) \setminus \{(y,?),(y,2),(y,4)\}) \cup \{ (y,2) \}$
$RD_{exit}(3) = RD_{entry}(3)$
$RD_{exit}(4) = (RD_{entry}(4) \setminus \{(y,?),(y,2),(y,4)\}) \cup \{ (y,4) \}$
$RD_{exit}(5) = (RD_{entry}(5) \setminus \{(x,?),(x,1),(x,5)\}) \cup \{ (x,5) \}$

ℓ	$RD_{entry}(\ell)$	$RD_{exit}(\ell)$
1	{(x,?),(y,?)}	{(y,?),(x,1)}
2	{(y,?),(x,1)}	{(x,1),(y,2)}
3	{(x,1),(y,2),(y,4),(x,5)}	{(x,1),(y,2),(y,4),(x,5)}
4	{(x,1),(y,2),(y,4),(x,5)}	{(x,1),(y,4),(x,5)}
5	{(x,1),(y,4),(x,5)}	{(y,4),(x,5)}

Analysis Result

Available Expressions Analysis

- For each program point,
  - determine the expressions that must have already been computed,
  - and not later modified,
  - on all paths to the program point.

```
x = a + b;
y = a * b;
while (y > a+b) {
    a = a+1;
    x = a + b;
}
```

(a+b) is available for the while loop every time. No need to recompute.

Let Us Do It Manually

ℓ	$AE_{entry}(\ell)$	$AE_{exit}(\ell)$
1	∅	{a+b}
2	{a+b}	{a+b, a*b}
3	{a+b}	{a+b}
4	{a+b}	∅
5	∅	{a+b}

Labeled Input Program

```
[x = a + b]^1;
[y = a * b]^2;
while ([y > a+b]^3) {
    [a = a + 1]^4;
    [x = a + b]^5;
}
```

Deriving Equations

Must Analysis

$AE_{entry}(1) = \emptyset$
$AE_{entry}(2) = AE_{exit}(1)$
$AE_{entry}(3) = AE_{exit}(2) \cap AE_{exit}(5)$
$AE_{entry}(4) = AE_{exit}(3)$
$AE_{entry}(5) = AE_{exit}(4)$

$$AE_{entry}(\ell) = \begin{cases} \emptyset & \text{if } \ell = init(S_*) \\ \cap\{AE_{exit}(\ell') \mid (\ell', \ell) \in flow(S_*)\} & \text{otherwise} \end{cases}$$

Labeled Input Program

```
[x = a + b]^1;
[y = a * b]^2;
while ([y > a+b]^3) {
    [a = a + 1]^4;
    [x = a + b]^5;
}
```

The Equations…

$AE_{exit}(1) = AE_{entry}(1) \cup \{a+b\}$
$AE_{exit}(2) = AE_{entry}(2) \cup \{a*b\}$
$AE_{exit}(3) = AE_{entry}(3) \cup \{a+b\}$
$AE_{exit}(4) = AE_{entry}(4) \setminus \{ a+b, a*b, a+1\}$
$AE_{exit}(5) = AE_{entry}(5) \cup \{a+b\}$

$$AE_{exit}(\ell) = (AE_{entry}(\ell) \setminus kill_{AE}(B^\ell)) \cup gen_{AE}(B^\ell)$$
where $B^\ell \in blocks(S_*)$

Labeled Input Program

```
[x = a + b]1;
[y = a * b]2;
while ([y > a+b]3) {
    [a = a + 1]4;
    [x = a + b]5;
}
```

The Kill and gen Sets

$kill_{AE}([x{:=}a]^{\ell}) = \{a' \in AExp_{*} \mid x \in FV(a')\}$
$kill_{AE}([skip]^{\ell}) = \emptyset$
$kill_{AE}([b]^{\ell}) = \emptyset$

$gen_{AE}([x{:=}a]^{\ell}) = \{a' \in AExp(a) \mid x \notin FV(a')\}$
$gen_{AE}([skip]^{\ell}) = \emptyset$
$gen_{AE}([b]^{\ell}) = AExp(b)$

where **FV(a')** is a set of variables used in a'

a' denotes arithmetic expressions

Labeled Input Program

```
[x = a + b]1;
[y = a * b]2;
while ([y > a+b]3) {
    [a = a + 1]4;
    [x = a + b]5;
}
```

The Kill and gen Sets

ℓ	$kill_{AE}(\ell)$	$gen_{AE}(\ell)$
1	∅	{a+b}
2	∅	{a*b}
3	∅	{a+b}
4	{a+b, a*b, a+1}	∅
5	∅	{a+b}

Labeled Input Program

```
[x = a + b]^1;
[y = a * b]^2;
while ([y > a+b]^3) {
    [a = a + 1]^4;
    [x = a + b]^5;
}
```

Summary

```
[x = a + b]1;
[y = a * b]2;
while ([y > a+b]3) {
      [a = a + 1]4;
      [x = a + b]5;
}
```

Input Program

Data Flow Analysis

ℓ	$kill_{AE}(\ell)$	$gen_{AE}(\ell)$
1	∅	{a+b}
2	∅	{a*b}
3	∅	{a+b}
4	{a+b, a*b, a+1}	∅
5	∅	{a+b}

$AE_{entry}(1) = \emptyset$
$AE_{entry}(2) = AE_{exit}(1)$
$AE_{entry}(3) = AE_{exit}(2) \cap AE_{exit}(5)$
$AE_{entry}(4) = AE_{exit}(3)$
$AE_{entry}(5) = AE_{exit}(4)$
$AE_{exit}(1) = AE_{entry}(1) \cup \{a+b\}$
$AE_{exit}(2) = AE_{entry}(2) \cup \{a*b\}$
$AE_{exit}(3) = AE_{entry}(3) \cup \{a+b\}$
$AE_{exit}(4) = AE_{entry}(4) \setminus \{ a+b,a*b,a+1\}$
$AE_{exit}(5) = AE_{entry}(5) \cup \{a+b\}$

ℓ	$AE_{entry}(\ell)$	$AE_{exit}(\ell)$
1	∅	{a+b}
2	{a+b}	{a+b, a*b}
3	{a+b}	{a+b}
4	{a+b}	∅
5	∅	{a+b}

Analysis Result

Very Busy Expressions Analysis

- No matter what path is taken from a given label, the expression must always be used before any of the variables occurring in it are redefined.

```
if (a>b) {
   x = b – a;
   y = a – b;
} else {
    y = b – a;
    x = a – b;
}
```

b-a **and** a-b **are very busy!**

Can be hoisted to the beginning.

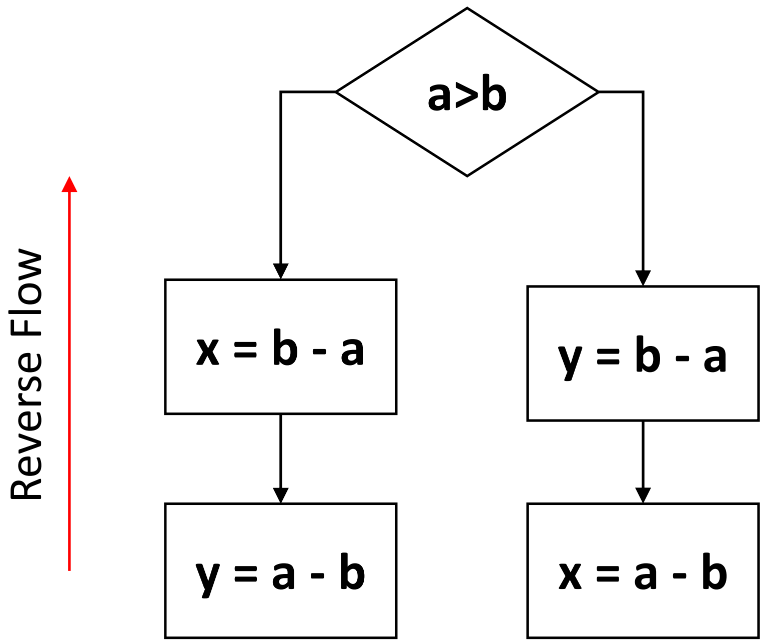
The Analysis

ℓ	$VB_{entry}(\ell)$	$VB_{exit}(\ell)$
1	{a – b, b – a}	{a – b, b – a}
2	{a – b, b – a}	{a – b}
3	{a – b}	Ø
4	{a – b, b – a}	{a – b}
5	{a – b}	Ø

Labeled Input Program

```
if [a>b]1 {
    [x = b - a]2;
    [y = a - b]3;
} else {
    [y = b - a]4;
    [x = a - b]5;
}
```

The CFG



The Equations…

Must Analysis

$VB_{exit}(1) = VB_{entry}(2) \cap VB_{entry}(4)$
$VB_{exit}(2) = VB_{entry}(3)$
$VB_{exit}(3) = \emptyset$
$VB_{exit}(4) = VB_{entry}(5)$
$VB_{exit}(5) = \emptyset$

$$VB_{exit}(\ell) = \begin{cases} \emptyset & \text{if } \ell \in final(S_*) \\ \cap \{VB_{entry}(\ell') \mid (\ell', \ell) \in flow^R(S_*)\} & \text{otherwise} \end{cases}$$

Labeled
Input Program

```
if [a>b]¹ {
    [x = b - a]²;
    [y = a - b]³;
} else {
    [y = b - a]⁴;
    [x = a - b]⁵;
}
```

Recall…

- The edges of our flow graphs are captured using a flow function.

flow: Stmt -> P(Lab x Lab)

flow($[x:=a]^\ell$) = Ø
flow($[S_1;S_2]^\ell$) = blocks(S_1) U blocks(S_2)
flow($[skip]^\ell$) = Ø
flow(if $[b]^\ell$ then S_1 else S_2) = flow(S_1) U flow(S_2) U {ℓ, init(S_2)) | ℓ ∈ final(S_1)}
flow(while $[b]^\ell$ do S) = flow(S) U {(ℓ, init(S))} U {(ℓ, ℓ')| ℓ' ∈ final(S)}

flow denotes forward flow here.

Recall…

- Program power is given below:

$[z{:=}1]^1$; while $[x{>}0]^2$ do ($[z{:=}z*y]^3$; $[x{:=}x{-}1]^4$)

What are $\mathit{init}(power)$, $\mathit{final}(power)$, $\mathit{labels}(power)$ **and** $\mathit{flow}(power)$**?**

$$\mathit{init}(power) = 1$$
$$\mathit{final}(power) = \{2\}$$
$$\mathit{labels}(power) = \{1,2,3,4\}$$
$$\mathit{flow}(power) = \{(1,2),(2,3),(3,4),(4,2)\}$$

Setup

- Reverse Flow

flow^R: Stmt -> P(Lab x Lab)

$$\text{flow}^R(S) = \{(\ell, \ell') \mid (\ell', \ell) \in \text{flow}(S)\}$$

Example

- Program power is given below:

$$[z:=1]^1;\ \text{while } [x>0]^2 \text{ do } ([z:=z*y]^3;\ [x:=x-1]^4)$$

What are $flow(power)$ and $flow^R(power)$ **?**

$$flow(power) = \{(1,2),(2,3),(3,4),(4,2)\}$$

$$flow^R(power) = \{(2,4),(4,3),(3,2),(2,1)\}$$

The Equations

$VB_{entry}(1) = VB_{exit}(1)$
$VB_{entry}(2) = VB_{exit}(2) \cup \{b - a\}$
$VB_{entry}(3) = \{a - b\}$
$VB_{entry}(4) = VB_{exit}(4) \cup \{b - a\}$
$VB_{entry}(5) = \{a - b\}$

$VB_{entry}(\ell) = (VB_{exit}(\ell) \setminus kill_{VB}(B^{\ell})) \cup gen_{VB}(B^{\ell})$
where $B^{\ell} \in blocks(S_*)$

Labeled Input Program

```
if [a>b]¹ {
    [x = b - a]²;
    [y = a - b]³;
} else {
    [y = b - a]⁴;
    [x = a - b]⁵;
}
```

The Kill and gen Sets

$kill_{VB}([x{:=}a]^\ell) = \{a' \in AExp_* \mid x \in FV(a')\}$
$kill_{VB}([skip]^\ell) = \emptyset$
$kill_{VB}([b]^\ell) = \emptyset$

$gen_{VB}([x{:=}a]^\ell) = AExp(a)$
$gen_{VB}([skip]^\ell) = \emptyset$
$gen_{VB}([b]^\ell) = AExp(b)$

Labeled Input Program

```
if [a>b]^1 {
    [x = b - a]^2;
    [y = a - b]^3;
} else {
    [y = b - a]^4;
    [x = a - b]^5;
}
```

The Kill and gen Sets

ℓ	$\mathbf{kill_{VB}(\ell)}$	$\mathbf{gen_{VB}(\ell)}$
1	∅	∅
2	∅	{b-a}
3	∅	{a-b}
4	∅	{b-a}
5	∅	{a-b}

Labeled
Input Program

```
if [a>b]1 {
    [x = b - a]2;
    [y = a - b]3;
} else {
    [y = b - a]4;
    [x = a - b]5;
}
```

Summary

Data Flow Analysis

```
if [a>b]^1 {
    [x = b - a]^2;
    [y = a - b]^3;
} else {
    [y = b - a]^4;
    [y = a - b]^5;
}
```

Input Program

ℓ	$kill_{VB}(\ell)$	$gen_{VB}(\ell)$
1	∅	∅
2	∅	{b-a}
3	∅	{a-b}
4	∅	{b-a}
5	∅	{a-b}

$VB_{entry}(1) = VB_{exit}(1)$
$VB_{entry}(2) = VB_{exit}(2) \cup \{b - a\}$
$VB_{entry}(3) = \{a - b\}$
$VB_{entry}(4) = VB_{exit}(4) \cup \{b - a\}$
$VB_{entry}(5) = \{a - b\}$

$VB_{exit}(1) = VB_{entry}(2) \cap VB_{entry}(4)$
$VB_{exit}(2) = VB_{entry}(3)$
$VB_{exit}(3) = \emptyset$
$VB_{exit}(4) = VB_{entry}(5)$
$VB_{exit}(5) = \emptyset$

ℓ	$VB_{entry}(\ell)$	$VB_{exit}(\ell)$
1	{a – b, b – a}	{a – b, b – a}
2	{a – b, b – a}	{a – b}
3	{a – b}	∅
4	{a – b, b – a}	{a – b}
5	{a – b}	∅

Analysis Result