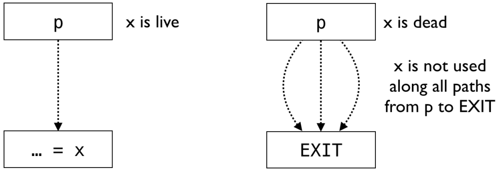
COSE312: Compilers

Lecture 16 — Data-Flow Analysis (2)

Hakjoo Oh
2015 Fall

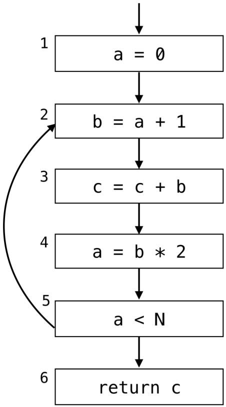
Liveness Analysis

- A variable is *live* at program point p if its value could be used in the future (along some path starting at p).



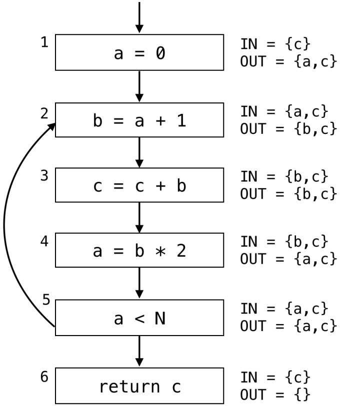
- Liveness analysis aims to compute the set of live variables for each basic block of the program.
- Applications: deadcode detection, uninitialized variable detection, register allocation, etc

Example: Liveness of Variables



- The live range of b: $\{2 \rightarrow 3, 3 \rightarrow 4\}$
- The live range of a: $\{1 \rightarrow 2, 4 \rightarrow 5 \rightarrow 2\}$
- The live range of c: the entire code

Example: Liveness of Variables



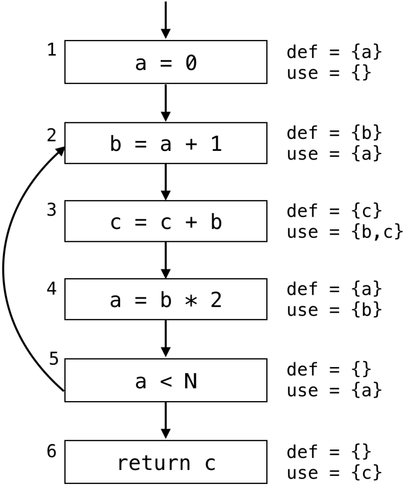
Liveness Analysis

The goal is to compute

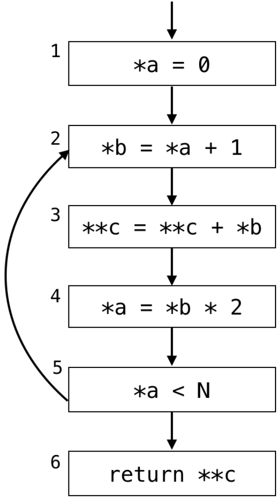
$$\begin{array}{rcl} \textbf{in} & : & Block \rightarrow \mathcal{P}(Var) \\ \textbf{out} & : & Block \rightarrow 2^{Var} \end{array}$$

1. Derive the set of data-flow equations.
2. Solve the equation by the iterative fixed point algorithm.

Def/Use Sets


1
a = 0
def = {a}
use = {}
2
b = a + 1
def = {b}
use = {a}
3
c = c + b
def = {c}
use = {b,c}
4
a = b * 2
def = {a}
use = {b}
5
a < N
def = {}
use = {a}
6
return c
def = {}
use = {c}

cf) Def/Use sets are only dynamically computable



1. Data-Flow Equations

Intuitions:

1. If a variable is in **use**(B), then it is live on entry to block B.
2. If a variable is live at the end of block B, and not in **def**(B), then the variable is also live on entry to B.
3. If a variable is live on enty to block B, then it is live at the end of predecessors of B.

Equations:

$$\begin{aligned} \textbf{in}(B) &= \\ \textbf{out}(B) &= \end{aligned}$$

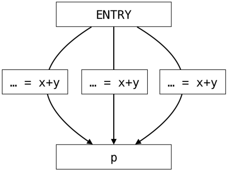
2. Fixed Point Computation

For all $i, \textbf{in}(B_i) = \textbf{out}(B_i) = \emptyset$
while (changes to any **in** and **out** occur) {
For all i, update
$\textbf{in}(B_i) = \textbf{use}(B) \cup (\textbf{out}(B) - \textbf{def}(B))$
$\textbf{out}(B_i) = \bigcup_{B \hookrightarrow S} \textbf{in}(S)$
}

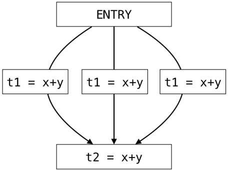
			1st		2nd		3rd	
	use	**def**	**out**	**in**	**out**	**in**	**out**	**in**
6	$\{c\}$	$\emptyset$	$\emptyset$	$\{c\}$	$\emptyset$	$\{c\}$	$\emptyset$	$\{c\}$
5	$\{a\}$	$\emptyset$	$\{c\}$	$\{a,c\}$	$\{a,c\}$	$\{a,c\}$	$\{a,c\}$	$\{a,c\}$
4	$\{b\}$	$\{a\}$	$\{a,c\}$	$\{b,c\}$	$\{a,c\}$	$\{b,c\}$	$\{a,c\}$	$\{b,c\}$
3	$\{b,c\}$	$\{c\}$	$\{b,c\}$	$\{b,c\}$	$\{b,c\}$	$\{b,c\}$	$\{b,c\}$	$\{b,c\}$
2	$\{a\}$	$\{b\}$	$\{b,c\}$	$\{a,c\}$	$\{b,c\}$	$\{a,c\}$	$\{b,c\}$	$\{a,c\}$
1	$\emptyset$	$\{a\}$	$\{a,c\}$	$\{c\}$	$\{a,c\}$	$\{c\}$	$\{a,c\}$	$\{c\}$

Available Expressions Analysis

- An expression $x + y$ is *available* at a point p if every path from the entry node to p evaluates $x + y$, and after the last such evaluation prior to reaching p, there are no subsequent assignments to x or y.



- Application: common subexpression elimination



Available Expressions Analysis

The goal is to compute

$$\begin{array}{rcl} \textbf{in} & : & Block \rightarrow 2^{Expr} \\ \textbf{out} & : & Block \rightarrow 2^{Expr} \end{array}$$

1. Derive the set of data-flow equations.
2. Solve the equation by the iterative fixed point algorithm.

Gen/Kill Sets

- **gen**(B): the set of expressions evaluated and not subsequently killed
- **kill**(B): the set of expressions whose variables can be killed

Exercises:

- What expressions are generated and killed by each of statements?

Statement s	**gen**(s)	**kill**(s)
$x = y + z$		
$x = \texttt{alloc}(n)$		
$x = y[i]$		
$x[i] = y$		

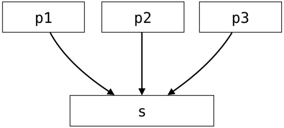
- what expressions are generated and killed by the block?

$$\boxed{\begin{array}{l} a = b + c \\ b = a - d \\ c = b + c \\ d = a - d \end{array}}$$

1. Set up a set of data-flow equations

Intuitions:

1. At the entry, no expressions are available.
2. An expression is available at the entry of a block only if it is available at the end of *all* its predecessors.



Equations:

$$\begin{aligned} \mathbf{in}(\mathit{ENTRY}) &= \emptyset \\ \\ \mathbf{out}(B) &= \mathbf{gen}(B) \cup (\mathbf{in}(B) - \mathbf{kill}(B)) \\ \mathbf{in}(B) &= \end{aligned}$$

cf) $\bigcap \emptyset = U, \bigcup \emptyset = \emptyset$

2. Solve the equations

- Trivial solution: $\textbf{in}(B_i) = \textbf{out}(B_i) = \emptyset$.
- Need to find the greatest solution (i.e., greatest fixed point) of the equation.

$\textbf{in}(ENTRY) = \emptyset$
For other $B_i, \textbf{in}(B_i) = \textbf{out}(B_i) = Expr$
while (changes to any **in** and **out** occur) {
  For all i, update
    $\textbf{in}(B_i) = \bigcap_{P \hookrightarrow B_i} \textbf{out}(P)$
    $\textbf{out}(B_i) = \textbf{gen}(B_i) \cup (\textbf{in}(B_i) - \textbf{kill}(B_i))$
}

Summary: Data-flow Analysis

	Reaching Definitions	Liveness Analysis	Available Expressions
Domain	Sets of definitions $(\mathbf{2}^{\mathit{Definitions}})$	Sets of variables $(\mathbf{2}^{\mathit{Var}})$	Sets of expressions $(\mathbf{2}^{\mathit{Expr}})$
Direction	Forwards	Backwards	Forwards
Transfer function	$\boldsymbol{gen} \cup (\boldsymbol{x} - \boldsymbol{kill})$	$\boldsymbol{use} \cup (\boldsymbol{x} - \boldsymbol{def})$	$\boldsymbol{gen} \cup (\boldsymbol{x} - \boldsymbol{kill})$
Boundary	$\mathbf{out}[\boldsymbol{ENTRY}] = \emptyset$	$\mathbf{in}[\boldsymbol{EXIT}] = \emptyset$	$\mathbf{out}[\boldsymbol{ENTRY}] = \emptyset$
Join	$\cup$	$\cup$	$\cap$
Equations	$\mathbf{out}(\boldsymbol{B}) = \boldsymbol{f_B}(\mathbf{in}(\boldsymbol{B}))$ $\mathbf{in}(\boldsymbol{B}) = \bigcup_{\boldsymbol{P} \hookrightarrow \boldsymbol{B}} \mathbf{out}(\boldsymbol{P})$	$\mathbf{in}(\boldsymbol{B}) = \boldsymbol{f_B}(\mathbf{out}(\boldsymbol{B}))$ $\mathbf{out}(\boldsymbol{B}) = \bigcup_{\boldsymbol{B} \hookrightarrow \boldsymbol{S}} \mathbf{in}(\boldsymbol{S})$	$\mathbf{out}(\boldsymbol{B}) = \boldsymbol{f_B}(\mathbf{in}(\boldsymbol{B}))$ $\mathbf{in}(\boldsymbol{B}) = \bigcup_{\boldsymbol{P} \hookrightarrow \boldsymbol{B}} \mathbf{out}(\boldsymbol{P}$
Initialize	$\mathbf{out}(\boldsymbol{B}) = \emptyset$	$\mathbf{in}(\boldsymbol{B}) = \emptyset$	$\mathbf{out}(\boldsymbol{B}) = \boldsymbol{U}$