Compiler Construction

Liveness Analysis

Goals

How to precisely catch the liveness of each variable?

If a variable is in register $1 then we can reuse this register as soon as the variable is no longer used

Scopes vs. Liveness

Scopes

- Front-end analysis
- Detect names visibility according to textual rules

Liveness

- Back-end analysis
- Focus on all generated variables (even temporaries)
- Exact computation of which variables are used at the same time

Liveness Definition

Definition

A variable is **live** if it holds a value that may be needed in the future.

Example

```
     a := 0                    1
L1:  b := a + 1                2
     c := c + b                3
     a := b * 2                4
     if a < N goto L1          5
     return c                  6
```

Question:

What are the liveness of a, b and c?

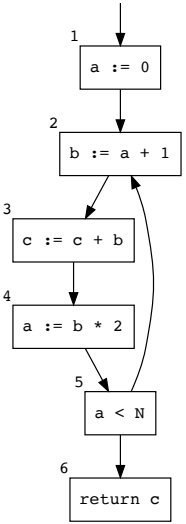
Flow Graph

Liveness computation requires an adequate data-structure.

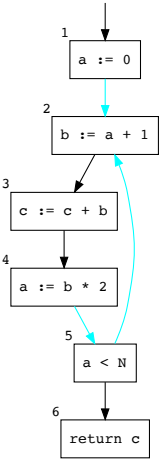
Control Flow graph (CFG):
A representation, using graph notation, of all paths that might be traversed through a program during its execution.

Remark: pred$[n]$ (resp. succ$[n]$) denotes the predecessors (resp. successors) of node n

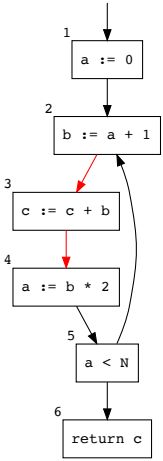
Flow Graph for the example



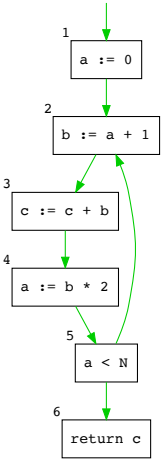
Liveness



Liveness for a



Liveness for b



Liveness for c

Terminology

- **defs**: nodes that define a variable, i.e. left (lhs) part of assignment nodes.
- **uses**: nodes that use (read) a variable (rhs).
- **live-in**: a variable is live-in of a node n, if it lives on any in-edges of that node.
- **live-out**: a variable is live-out of a node n, if it lives on any out-edges of that node.

Liveness computation

1. If a variable is in $use[n]$ then it is live-in at node n.
2. If a variable is live-in at node n then it is live-out at all nodes m in $pred[m]$.
3. If a variable is live-out at node n and not in $def[n]$, then it is live-in at node n.

Dataflow Equations for Liveness Analysis

$$
\begin{aligned}
\text{in}[n] &= \text{use}[n] \cup (\text{out}[n] \setminus \text{def}[n]) \\
\text{out}[n] &= \bigcup_{s \in \text{succ}[n]} \text{in}[s]
\end{aligned}
$$

Possible Implementation (quadratic)

```
foreach n
    in[n] ← { }
    out[n] ← { }

repeat
    foreach n
        in_t[n] ← in[n]
        out_t[n] ← out[n]
        in[n] ← use[n] ∪ (out[n] \ def[n])
        out[n] ← ⋃_{s∈succ[n]} in[s]
until in_t[n] = in[n] and out_t[n] = out[n] (∀n)
```

Liveness Calculation

n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2								
2	a	b	3								
3	bc	c	4								
4	b	a	5								
5	a		2,6								
6	c										

n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2						
2	a	b	3						
3	bc	c	4						
4	b	a	5						
5	a		2,6						
6	c								

$$\begin{aligned} \text{in}[n] &= \text{use}[n] \cup (\text{out}[n] \setminus \text{def}[n]) \\ \text{out}[n] &= \bigcup_{s \in \text{succ}[n]} \text{in}[s] \end{aligned}$$

Liveness Calculation

1st step

n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2								
2	a	b	3	a							
3	bc	c	4	bc							
4	b	a	5	b							
5	a		2,6	a	a						
6	c			c							

n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2						
2	a	b	3						
3	bc	c	4						
4	b	a	5						
5	a		2,6						
6	c								

$$\begin{aligned} \text{in}[n] &= \text{use}[n] \cup (\text{out}[n] \setminus \text{def}[n]) \\ \text{out}[n] &= \bigcup_{s \in \text{succ}[n]} \text{in}[s] \end{aligned}$$

Liveness Calculation

				1st step		2nd step					
n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2				a				
2	a	b	3	a		a	bc				
3	bc	c	4	bc		bc	b				
4	b	a	5	b		b	a				
5	a		2,6	a	a	a	ac				
6	c			c		c					

n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2						
2	a	b	3						
3	bc	c	4						
4	b	a	5						
5	a		2,6						
6	c								

$$\begin{aligned} \text{in}[n] &= \text{use}[n] \cup (\text{out}[n] \setminus \text{def}[n]) \\ \text{out}[n] &= \bigcup_{s \in \text{succ}[n]} \text{in}[s] \end{aligned}$$

Liveness Calculation

				1st step		2nd step		3rd step			
n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2				a		a		
2	a	b	3	a		a	bc	ac	bc		
3	bc	c	4	bc		bc	b	bc	b		
4	b	a	5	b		b	a	b	a		
5	a		2,6	a	a	a	ac	ac	ac		
6	c			c		c		c			

n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2						
2	a	b	3						
3	bc	c	4						
4	b	a	5						
5	a		2,6						
6	c								

$$\begin{aligned} \text{in}[n] &= \text{use}[n] \cup (\text{out}[n] \setminus \text{def}[n]) \\ \text{out}[n] &= \bigcup_{s \in \text{succ}[n]} \text{in}[s] \end{aligned}$$

Liveness Calculation

				1st step		2nd step		3rd step		4th step	
n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2				a		a		ac
2	a	b	3	a		a	bc	ac	bc	ac	bc
3	bc	c	4	bc		bc	b	bc	b	bc	b
4	b	a	5	b		b	a	b	a	b	ac
5	a		2,6	a	a	a	ac	ac	ac	ac	ac
6	c			c		c		c		c	

n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2						
2	a	b	3						
3	bc	c	4						
4	b	a	5						
5	a		2,6						
6	c								

$$\begin{aligned} \text{in}[n] &= \text{use}[n] \cup (\text{out}[n] \setminus \text{def}[n]) \\ \text{out}[n] &= \bigcup_{s \in \text{succ}[n]} \text{in}[s] \end{aligned}$$

Liveness Calculation

				1st step		2nd step		3rd step		4th step	
n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2				a		a		ac
2	a	b	3	a		a	bc	ac	bc	ac	bc
3	bc	c	4	bc		bc	b	bc	b	bc	b
4	b	a	5	b		b	a	b	a	b	ac
5	a		2,6	a	a	a	ac	ac	ac	ac	ac
6	c			c		c		c		c	

				5th step					
n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2	c	ac				
2	a	b	3	ac	bc				
3	bc	c	4	bc	b				
4	b	a	5	bc	ac				
5	a		2,6	ac	ac				
6	c			c					

$$\begin{aligned} \text{in}[n] &= \text{use}[n] \cup (\text{out}[n] \setminus \text{def}[n]) \\ \text{out}[n] &= \bigcup_{s \in \text{succ}[n]} \text{in}[s] \end{aligned}$$

Liveness Calculation

				1st step		2nd step		3rd step		4th step	
n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2				a		a		ac
2	a	b	3	a		a	bc	ac	bc	ac	bc
3	bc	c	4	bc		bc	b	bc	b	bc	b
4	b	a	5	b		b	a	b	a	b	ac
5	a		2,6	a	a	a	ac	ac	ac	ac	ac
6	c			c		c		c		c	

				5th step		6th step			
n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2	c	ac	c	ac		
2	a	b	3	ac	bc	ac	bc		
3	bc	c	4	bc	b	bc	bc		
4	b	a	5	bc	ac	bc	ac		
5	a		2,6	ac	ac	ac	ac		
6	c			c		c			

$$\text{in}[n] = \text{use}[n] \cup (\text{out}[n] \setminus \text{def}[n])$$

$$\text{out}[n] = \bigcup_{s \in \text{succ}[n]} \text{in}[s]$$

Liveness Calculation

				1st step		2nd step		3rd step		4th step	
n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2				a		a		ac
2	a	b	3	a		a	bc	ac	bc	ac	bc
3	bc	c	4	bc		bc	b	bc	b	bc	b
4	b	a	5	b		b	a	b	a	b	ac
5	a		2,6	a	a	a	ac	ac	ac	ac	ac
6	c			c		c		c		c	

				5th step		6th step		7th step	
n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2	c	ac	c	ac	c	ac
2	a	b	3	ac	bc	ac	bc	ac	bc
3	bc	c	4	bc	b	bc	bc	bc	bc
4	b	a	5	bc	ac	bc	ac	bc	ac
5	a		2,6	ac	ac	ac	ac	ac	ac
6	c			c		c		c	

$$\begin{aligned} \text{in}[n] &= \text{use}[n] \cup (\text{out}[n] \setminus \text{def}[n]) \\ \text{out}[n] &= \bigcup_{s \in \text{succ}[n]} \text{in}[s] \end{aligned}$$

Liveness Calculation (Forward)

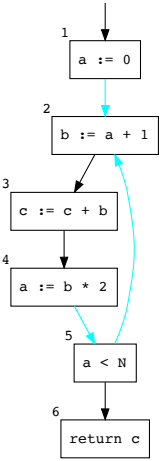
				1st step		2nd step		3rd step		4th step	
n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2				a		a		ac
2	a	b	3	a		a	bc	ac	bc	ac	bc
3	bc	c	4	bc		bc	b	bc	b	bc	b
4	b	a	5	b		b	a	b	a	b	ac
5	a		2,6	a	a	a	ac	ac	ac	ac	ac
6	c			c		c		c		c	

				5th step		6th step		7th step	
n	*use*	*def*	*succ*	*in*	*out*	*in*	*out*	*in*	*out*
1		a	2	c	ac	c	ac	c	ac
2	a	b	3	ac	bc	ac	bc	ac	bc
3	bc	c	4	bc	b	bc	bc	bc	bc
4	b	a	5	bc	ac	bc	ac	bc	ac
5	a		2,6	ac	ac	ac	ac	ac	ac
6	c			c		c		c	

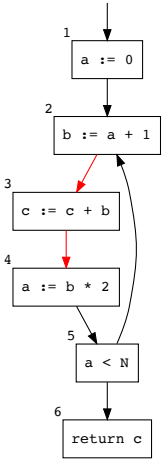
$$\text{in}[n] = \text{use}[n] \cup (\text{out}[n] \setminus \text{def}[n])$$

$$\text{out}[n] = \bigcup_{s \in \text{succ}[n]} \text{in}[s]$$

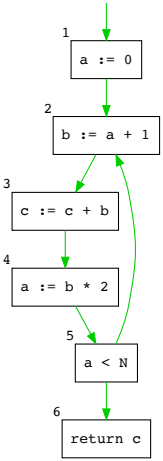
Liveness



Liveness for a



Liveness for b



Liveness for c

Liveness Calculation (Backward)

n	*use*	*def*	*succ*	*out*	*in*	*out*	*in*	*out*	*in*
6	c								
5	a		2,6						
4	b	a	5						
3	bc	c	4						
2	a	b	3						
1		a	2						

$$\begin{aligned} \text{in}[n] &= \text{use}[n] \cup (\text{out}[n] \setminus \text{def}[n]) \\ \text{out}[n] &= \bigcup_{s \in \text{succ}[n]} \text{in}[s] \end{aligned}$$

Calculation done following *reverse* control-flow edges.

Liveness Calculation (Backward)

				1st step					
n	*use*	*def*	*succ*	*out*	*in*	*out*	*in*	*out*	*in*
6	c				c				
5	a		2,6	c	ac				
4	b	a	5	ac	bc				
3	bc	c	4	bc	bc				
2	a	b	3	bc	ac				
1		a	2	ac	c				

$$\begin{aligned} \text{in}[n] &= \text{use}[n] \cup (\text{out}[n] \setminus \text{def}[n]) \\ \text{out}[n] &= \bigcup_{s \in \text{succ}[n]} \text{in}[s] \end{aligned}$$

Calculation done following *reverse* control-flow edges.

Liveness Calculation (Backward)

				1st step		2nd step			
n	*use*	*def*	*succ*	*out*	*in*	*out*	*in*	*out*	*in*
6	c				c		c		
5	a		2,6	c	ac	ac	ac		
4	b	a	5	ac	bc	ac	bc		
3	bc	c	4	bc	bc	bc	bc		
2	a	b	3	bc	ac	bc	ac		
1		a	2	ac	c	ac	c		

$$\begin{aligned} \text{in}[n] &= \text{use}[n] \cup (\text{out}[n] \setminus \text{def}[n]) \\ \text{out}[n] &= \bigcup_{s \in \text{succ}[n]} \text{in}[s] \end{aligned}$$

Calculation done following *reverse* control-flow edges.

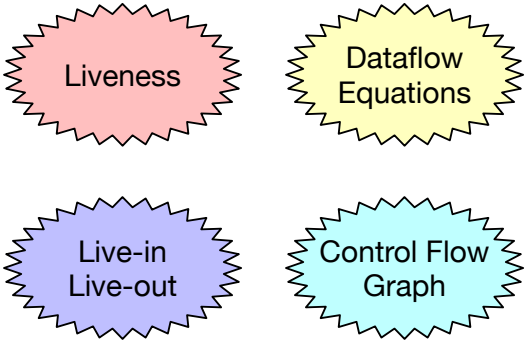
Liveness Calculation (Backward)

				1st step		2nd step		3rd step	
n	*use*	*def*	*succ*	*out*	*in*	*out*	*in*	*out*	*in*
6	c				c		c		c
5	a		2,6	c	ac	ac	ac	ac	ac
4	b	a	5	ac	bc	ac	bc	ac	bc
3	bc	c	4	bc	bc	bc	bc	bc	bc
2	a	b	3	bc	ac	bc	ac	bc	ac
1		a	2	ac	c	ac	c	ac	c

$$\begin{aligned} \text{in}[n] &= \text{use}[n] \cup (\text{out}[n] \setminus \text{def}[n]) \\ \text{out}[n] &= \bigcup_{s \in \text{succ}[n]} \text{in}[s] \end{aligned}$$

Calculation done following *reverse* control-flow edges.

Summary


Liveness
Dataflow Equations
Live-in Live-out
Control Flow Graph