Data-Flow Analysis

Compiler Design

CSE 504

Last modified: Sun Apr 17 2016 at 21:35:51 EDT
Version: 1.4 16:58:45 2016/01/29
Compiled at 21:37 on 2016/04/17

Program Analysis

- The compiler needs to understand properties of a program (e.g. the set of variables "live" at a program point).
- This information should be computed at *compile time*, with incomplete information on the values the program computes, and without executing the program itself!
- This information is likely to be approximate: in general, at compile time, we will not know which sequence of instructions will be executed.
- Data-Flow Analysis is a standard way to formulate *intra*-procedural program analysis.

Control Flow Graphs

When we try to deduce properties of a procedure, we first build a control flow graph (CFG).

- Nodes of a CFG are Basic Blocks.
- Edges indicate which blocks can follow which other blocks.

A Basic Block is a sequence of instructions such that:

- There are no jumps/branches in the sequence except as the last instruction.
- For all jumps/branches in the program, the target is the first instruction in some basic block.
  - In other words, no jump lands in the middle of a basic block.

Example of CFGs

```
 1.   i = 1

 2.   j = 1

 3.   t1 = 10 * i
 4.   t2 = t1 + j
 5.   t3 = 4 * t2
 6.   a[t3] = 0
 7.   j = j + 1
 8.   if j < 10 goto (3)

 9.   i = i + 1
10.   if i < 10 goto (2)

11.   i = 1

12.   t4 = 10 * i
13.   t5 = t4 + i
14.   a[t5] = 1
15.   i = i + 1
16.   if i < 10 goto (12)
```

Example of CFGs

- Branches only at the end of a block.

```
1.   i = 1

2.   j = 1

3.   t1 = 10 * i
4.   t2 = t1 + j
5.   t3 = 4 * t2
6.   a[t3] = 0
7.   j = j + 1
8.   if j < 10 goto (3)

9.   i = i + 1
10.  if i < 10 goto (2)

11.  i = 1

12.  t4 = 10 * i
13.  t5 = t4 + i
14.  a[t5] = 1
15.  i = i + 1
16.  if i < 10 goto (12)
```

Example of CFGs

- Branches only at the end of a block.

```
 1.   i = 1

 2.   j = 1

 3.   t1 = 10 * i
 4.   t2 = t1 + j
 5.   t3 = 4 * t2
 6.   a[t3] = 0
 7.   j = j + 1
 8.   if j < 10 goto (3)
--------------------------
 9.   i = i + 1
10.   if i < 10 goto (2)
--------------------------
11.   i = 1

12.   t4 = 10 * i
13.   t5 = t4 + i
14.   a[t5] = 1
15.   i = i + 1
16.   if i < 10 goto (12)
--------------------------
```

Example of CFGs

- Branches only at the end of a block.
- Branch destinations only at beginning of a block.

```
 1.   i = 1
---------------------------------
 2.   j = 1
---------------------------------
 3.   t1 = 10 * i
 4.   t2 = t1 + j
 5.   t3 = 4 * t2
 6.   a[t3] = 0
 7.   j = j + 1
 8.   if j < 10 goto (3)
---------------------------------
 9.   i = i + 1
10.   if i < 10 goto (2)
---------------------------------
11.   i = 1
---------------------------------
12.   t4 = 10 * i
13.   t5 = t4 + i
14.   a[t5] = 1
15.   i = i + 1
16.   if i < 10 goto (12)
---------------------------------
```

Example of CFGs

- Branches only at the end of a block.
- Branch destinations only at beginning of a block.

```
 1.  i = 1
---------------------------
 2.  j = 1
---------------------------
 3.  t1 = 10 * i
 4.  t2 = t1 + j
 5.  t3 = 4 * t2
 6.  a[t3] = 0
 7.  j = j + 1
 8.  if j < 10 goto (3)
---------------------------
 9.  i = i + 1
10.  if i < 10 goto (2)
---------------------------
11.  i = 1
---------------------------
12.  t4 = 10 * i
13.  t5 = t4 + i
14.  a[t5] = 1
15.  i = i + 1
16.  if i < 10 goto (12)
---------------------------
```

Example of CFGs

- Branches only at the end of a block.
- Branch destinations only at beginning of a block.

```
B1:    1.   i = 1
-----------------------------
B2:    2.   j = 1
-----------------------------
B3:    3.   t1 = 10 * i
       4.   t2 = t1 + j
       5.   t3 = 4 * t2
       6.   a[t3] = 0
       7.   j = j + 1
       8.   if j < 10 goto (3)
-----------------------------
B4:    9.   i = i + 1
      10.   if i < 10 goto (2)
-----------------------------
B5:   11.   i = 1
-----------------------------
B6:   12.   t4 = 10 * i
      13.   t5 = t4 + i
      14.   a[t5] = 1
      15.   i = i + 1
      16.   if i < 10 goto (12)
-----------------------------
Exit:
```

Live Variables

Consider the problem of finding the set of live variables at some program point.

- A variable is live after a statement s in the program, if it is used in a statement s',
- and there is a control flow path from s to s'.

Example:

```
1.  i = 1
2.  j = 1
3.  t1 = 10 * i
4.  t2 = t1 + j
5.  t3 = 4 * t2
6.  a[t3] = 0
7.  j = j + 1
 ⋮   ⋮
```

- Variable `t3` is live after statement 5 since it is used in statement 6.
- Variable `j` is also live after statement 5 since it is used in statement 7.

Live Variable Analysis — (1)

- Let $def(s)$ be the set of all variables defined by statement s (e.g. the lhs variable in an assignment statement).
- Let $use(s)$ be the set of all variables used by statement s (e.g. the variables on the rhs of an assignment statement).
- $succ(s)$: the set of *statements* that immediately follow statement s.
- The above definitions for def, use, and $succ$ can be extended for whole blocks as well.
  - $def(B)$: set of variables defined in block b.
  - $use(B)$: set of variables used, but not defined earlier, in block b.
  - $succ(B)$: set of *blocks* that immediately succeed block B.

Live Variable Analysis — (2)

B1:	1.	i = 1
B2:	2.	j = 1
B3:	3.	t1 = 10 * i
	4.	t2 = t1 + j
	5.	t3 = 4 * t2
	6.	a[t3] = 0
	7.	j = j + 1
	8.	if j < 10 goto (3)
B4:	9.	i = i + 1
	10.	if i < 10 goto (2)
B5:	11.	i = 1
B6:	12.	t4 = 10 * i
	13.	t5 = t4 + i
	14.	a[t5] = 1
	15.	i = i + 1
	16.	if i < 10 goto (12)
Exit:		

Block	Succ	Def	Use
1	{2}	{i}	{}
2	{3}	{j}	{}
3	{3,4}	{t1, t2, t3, j}	{a,i, j}
4	{2,5}	{i}	{i}
5	{6}	{i}	{}
6	{6,Exit}	{t4,t5,i}	{a,i}

Live Variable Analysis — (3)

- $Out(s)$: the set of variables live just after statement s.
- $In(s)$: the set of variables live just before statement s.
- The above definitions for *Out* and *In* can be readily extended for blocks.
- Observe that:
  - If a variable is used by a statement, then it must be live before the statement.
  - If a variable is live immediately after a statement, then it must be live before the statement as well, unless it is defined by the statement.
  - For a statement s, if a variable is live before any of its successors, then it must be live after s.
  - From these observations, we get:

$$\begin{aligned} In(s) &= use(s) \cup (Out(s) - def(s)) \\ Out(s) &= \bigcup_{t \in succ(s)} In(t) \end{aligned}$$

Live Variable Analysis — (4)

$$
\begin{aligned}
In(s) &= use(s) \cup (Out(s) - def(s)) \\
Out(s) &= \bigcup_{t \in succ(s)} In(t)
\end{aligned}
$$

Let a be a variable that is needed after the procedure exits (e.g. it is a global variable). Then, $In(Exit) = \{a\}$.

Block	Succ	Def	Use	In	Out
1	{2}	{i}	{}	Out(1)−{i}	In(2)
2	{3}	{j}	{}	Out(2)−{j}	In(3)
3	{3,4}	{t1,t2,t3,j}	{a,i, j}	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	In(3) ∪ In(4)
4	{2,5}	{i}	{i}	{i} ∪ Out(4)−{i}	In(2) ∪ In(5)
5	{6}	{i}	{}	Out(5)−{i}	In(6)
6	{6,Exit}	{t4,t5,i}	{a,i}	{a,i} ∪ Out(6)−{t4,t5,i}	In(6) ∪ In(Exit)
Exit	{}	{}	{}	{a}	

Live Variable Analysis — (5)

- The equations for *In* and *Out* form a set of *simultaneous set equations*.
- For this analysis, we require the least solution to these equations.
- Consider the equations relating In(6), Out(6) and In(Exit):

$$\begin{aligned} In(6) &= \{a, i\} \cup Out(6) - \{t4, t5, i\} \\ Out(6) &= In(6) \cup In(Exit) \\ In(Exit) &= \{a\} \end{aligned}$$

- There are many solutions to these equations:
  1. $In(6) = Out(6) = \{a, i\}$, and $In(Exit) = \{a\}$.
  2. $In(6) = Out(6) = \{a, i, t3\}$, and $In(Exit) = \{a\}$.
  3. $\vdots$
- Of these, (1) is the least. In fact, it can be shown that every solution will contain (1).

Solutions to Data Flow Equations

- Data flow analysis is formulated in terms of finding the least (or sometimes, the greatest) solution to a set of simultaneous equations.
- The flow equations can be written as $\overline{X} = F(\overline{X})$, where $\overline{X}$ is a vector of *In*'s and *Out*'s.
- Solutions $\overline{X}$ such that $\overline{X} = F(\overline{X})$ are *fixed points* of F.
- The smallest $\overline{X}$ such that $\overline{X} = F(\overline{X})$ is called the *least fixed point* of F.

Partial Orders

- Let $\mathcal{U}$ be a finite set, and let $\mathcal{D} = \mathcal{P}(\mathcal{U})$, i.e. the powerset of $\mathcal{U}$. Let $\mathcal{D}^n = \mathcal{D} \times \mathcal{D} \times \cdots (n \text{ times}) \cdots \times \mathcal{D}$, i.e., an n-dimensional cartesian space over $\mathcal{P}(\mathcal{U})$.
- We can define partial order among vectors of sets such that $\overline{X} \sqsubseteq \overline{X}'$ if, and only if, for all components of the vector, $X_i \subseteq X_i'$.
  - It is easy to verify that "$\sqsubseteq$" is a partial order: it is reflexive, transitive and anti-symmetric.
- Let $\perp$ be a n-vector of empty sets. Clearly, $\perp \sqsubseteq \overline{X}$ for all $\overline{X} \in \mathcal{D}^n$.
- Let $\top$ be a n-vector of $\mathcal{U}$. Observe that $\overline{X} \sqsubseteq \top$ for all $\overline{X} \in \mathcal{D}^n$.
- $(\mathcal{D}^n, \sqsubseteq)$ is a *complete lattice* with $\perp$ as the least element and $\top$ as the greatest element.
- Vectors $\overline{X}^{(0)}, \overline{X}^{(1)}, \ldots, \overline{X}^{(i)}$ is called a *chain* if $\overline{X}^{(0)} \sqsubseteq \overline{X}^{(1)} \sqsubseteq \cdots \sqsubseteq \overline{X}^{(i)}$.
- Note all chains in $(\mathcal{D}^n, \sqsubseteq)$ are finite, since $\mathcal{U}$ is finite.

Monotone Functions

- Let $F : \mathcal{D}^n \rightarrow \mathcal{D}^n$ (i.e. a function from $\mathcal{D}^n$ to $\mathcal{D}^n$).
- A function F is monotone over partial order "$\sqsubseteq$" if, for every $\overline{X}$ and $\overline{X}'$ such that $\overline{X} \sqsubseteq \overline{X}'$, we have $F(\overline{X}) \sqsubseteq F(\overline{X}')$.
  - Note the definition of monotonicity. It says the function returns smaller values if it is given smaller argument values.
  - It is not necessary that the returned values must be smaller than the argument values!
- It is easy to see that the flow equations for live variable analysis defines a monotone function.
- There is a simple way to show the existence of fixed points, and to compute the Least/Greatest Fixed Points of a monotone function.
- Tarski-Knaster Theorem: Given a complete lattice L and a monotone function $G : L \rightarrow L$, the fixed points of G form a complete lattice. Consequently, there exist both least and greatest fixed points.

Computing Least Fixed Point —(1)

Kleene's Fixed Point Theorem:

- Construct a sequence $\overline{X}^{(0)}, \overline{X}^{(1)}, \ldots, \overline{X}^{(i)}, \ldots$, where $\overline{X}^{0} = \bot$ and $\overline{X}^{(i+1)} = F(\overline{X}^{(i)})$.
- This sequence forms a chain.
  - $\overline{X}^{(0)} = \bot \sqsubseteq \overline{X}^{(1)}$.
  - If $\overline{X}^{(i)} \sqsubseteq \overline{X}^{(i+1)}$, then $\overline{X}^{(i+1)} \sqsubseteq \overline{X}^{(i+2)}$.
    - $\overline{X}^{(i+1)} = F(\overline{X}^{(i)})$
    - Since $\overline{X}^{(i)} \sqsubseteq \overline{X}^{(i+1)}$, by monotonicity of F, $F(\overline{X}^{(i)}) \sqsubseteq F(\overline{X}^{(i+1)})$.
    - $\overline{X}^{(i+2)} = F(\overline{X}^{(i+1)})$
- Since all chains over "$\sqsubseteq$" are finite, consider the last element of the chain $\overline{X}^{(n)}$.
  - $\overline{X}^{(n)} = F(\overline{X}^{(n)})$, otherwise it is not the last element.
  - So, $\overline{X}^{(n)}$ is a fixed point of F.

Computing Least Fixed Point —(2)

- Consider the sequence $\overline{X}^{(0)}, \overline{X}^{(1)}, \ldots, \overline{X}^{(i)}, \ldots, \overline{X}^{(n)}$, where $\overline{X}^{0} = \bot$ and $\overline{X}^{(i+1)} = F(\overline{X}^{(i)})$.
- $\overline{X}^{(n)}$ is the least fixed point of F.
  - We already know that $\overline{X}^{(n)}$ is a fixed point of F.
  - Let $\overline{Y}$ be any fixed point of F.
  - Clearly, $\overline{X}^{(0)} = \bot \sqsubseteq \overline{Y}$.
  - If $\overline{X}^{(i)} \sqsubseteq \overline{Y}$, since F is monotone, $\overline{X}^{(i+1)} = F(\overline{X}^{(i)}) \sqsubseteq F(\overline{Y}) = \overline{Y}$ (since $\overline{Y}$ is a fixed point).
  - Hence, by induction, for all elements of the chain $\overline{X}^{(i)} \sqsubseteq \overline{Y}$.
  - In particular, $\overline{X}^{(n)} \sqsubseteq \overline{Y}$, is at least as small as any fixed point $\overline{Y}$ of F, and hence is the least fixed point.

Computing the Greatest Fixed Point

- Consider the sequence $\overline{X}^{(0)}, \overline{X}^{(1)}, \ldots, \overline{X}^{(i)}, \ldots, \overline{X}^{(n)}$, where $\overline{X}^{0} = \top$ and $\overline{X}^{(i+1)} = F(\overline{X}^{(i)})$.
- Note the starting point of this sequence: the greatest element in the lattice.
- By an argument similar to the one we used for the least fixed point, $\overline{X}^{(n)}$ can be shown to be the greatest fixed point of F.

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}			
Out(1)	In(2)	{}			
In(2)	Out(2)−{j}	{}			
Out(2)	In(3)	{}			
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}			
Out(3)	In(3) ∪ In(4)	{}			
In(4)	{i} ∪ Out(4)−{i}	{}			
Out(4)	In(2) ∪ In(5)	{}			
In(5)	Out(5)−{i}	{}			
Out(5)	In(6)	{}			
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}			
Out(6)	In(6) ∪ In(Exit)	{}			
In(Exit)	{a}	{}			

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}			
Out(1)	In(2)	{}			
In(2)	Out(2)−{j}	{}			
Out(2)	In(3)	{}			
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}			
Out(3)	In(3) ∪ In(4)	{}			
In(4)	{i} ∪ Out(4)−{i}	{}			
Out(4)	In(2) ∪ In(5)	{}			
In(5)	Out(5)−{i}	{}			
Out(5)	In(6)	{}			
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}			
Out(6)	In(6) ∪ In(Exit)	{}			
In(Exit)	{a}	{}	{a}		

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}			
Out(1)	In(2)	{}			
In(2)	Out(2)−{j}	{}			
Out(2)	In(3)	{}			
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}			
Out(3)	In(3) ∪ In(4)	{}			
In(4)	{i} ∪ Out(4)−{i}	{}			
Out(4)	In(2) ∪ In(5)	{}			
In(5)	Out(5)−{i}	{}			
Out(5)	In(6)	{}			
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}			
Out(6)	In(6) ∪ In(Exit)	{}	{a}		
In(Exit)	{a}	{}	{a}		

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}			
Out(1)	In(2)	{}			
In(2)	Out(2)−{j}	{}			
Out(2)	In(3)	{}			
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}			
Out(3)	In(3) ∪ In(4)	{}			
In(4)	{i} ∪ Out(4)−{i}	{}			
Out(4)	In(2) ∪ In(5)	{}			
In(5)	Out(5)−{i}	{}			
Out(5)	In(6)	{}			
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}		
Out(6)	In(6) ∪ In(Exit)	{}	{a}		
In(Exit)	{a}	{}	{a}		

Live Variable Analysis Revisited

Set	**Eqn**	**0**	**1**	**2**	**3**
In(1)	Out(1)−{i}	{}			
Out(1)	In(2)	{}			
In(2)	Out(2)−{j}	{}			
Out(2)	In(3)	{}			
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}			
Out(3)	In(3) ∪ In(4)	{}			
In(4)	{i} ∪ Out(4)−{i}	{}			
Out(4)	In(2) ∪ In(5)	{}			
In(5)	Out(5)−{i}	{}			
Out(5)	In(6)	{}	{a,i}		
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}		
Out(6)	In(6) ∪ In(Exit)	{}	{a}		
In(Exit)	{a}	{}	{a}		

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}			
Out(1)	In(2)	{}			
In(2)	Out(2)−{j}	{}			
Out(2)	In(3)	{}			
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}			
Out(3)	In(3) ∪ In(4)	{}			
In(4)	{i} ∪ Out(4)−{i}	{}			
Out(4)	In(2) ∪ In(5)	{}			
In(5)	Out(5)−{i}	{}	{a}		
Out(5)	In(6)	{}	{a,i}		
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}		
Out(6)	In(6) ∪ In(Exit)	{}	{a}		
In(Exit)	{a}	{}	{a}		

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}			
Out(1)	In(2)	{}			
In(2)	Out(2)−{j}	{}			
Out(2)	In(3)	{}			
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}			
Out(3)	In(3) ∪ In(4)	{}			
In(4)	{i} ∪ Out(4)−{i}	{}			
Out(4)	In(2) ∪ In(5)	{}	{a}		
In(5)	Out(5)−{i}	{}	{a}		
Out(5)	In(6)	{}	{a,i}		
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}		
Out(6)	In(6) ∪ In(Exit)	{}	{a}		
In(Exit)	{a}	{}	{a}		

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}			
Out(1)	In(2)	{}			
In(2)	Out(2)−{j}	{}			
Out(2)	In(3)	{}			
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}			
Out(3)	In(3) ∪ In(4)	{}			
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}		
Out(4)	In(2) ∪ In(5)	{}	{a}		
In(5)	Out(5)−{i}	{}	{a}		
Out(5)	In(6)	{}	{a,i}		
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}		
Out(6)	In(6) ∪ In(Exit)	{}	{a}		
In(Exit)	{a}	{}	{a}		

Live Variable Analysis Revisited

Set	**Eqn**	**0**	**1**	**2**	**3**
In(1)	Out(1)−{i}	{}			
Out(1)	In(2)	{}			
In(2)	Out(2)−{j}	{}			
Out(2)	In(3)	{}			
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}			
Out(3)	In(3) ∪ In(4)	{}	{a,i}		
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}		
Out(4)	In(2) ∪ In(5)	{}	{a}		
In(5)	Out(5)−{i}	{}	{a}		
Out(5)	In(6)	{}	{a,i}		
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}		
Out(6)	In(6) ∪ In(Exit)	{}	{a}		
In(Exit)	{a}	{}	{a}		

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}			
Out(1)	In(2)	{}			
In(2)	Out(2)−{j}	{}			
Out(2)	In(3)	{}			
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}		
Out(3)	In(3) ∪ In(4)	{}	{a,i}		
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}		
Out(4)	In(2) ∪ In(5)	{}	{a}		
In(5)	Out(5)−{i}	{}	{a}		
Out(5)	In(6)	{}	{a,i}		
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}		
Out(6)	In(6) ∪ In(Exit)	{}	{a}		
In(Exit)	{a}	{}	{a}		

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}			
Out(1)	In(2)	{}			
In(2)	Out(2)−{j}	{}			
Out(2)	In(3)	{}	{a,i,j}		
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}		
Out(3)	In(3) ∪ In(4)	{}	{a,i}		
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}		
Out(4)	In(2) ∪ In(5)	{}	{a}		
In(5)	Out(5)−{i}	{}	{a}		
Out(5)	In(6)	{}	{a,i}		
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}		
Out(6)	In(6) ∪ In(Exit)	{}	{a}		
In(Exit)	{a}	{}	{a}		

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}			
Out(1)	In(2)	{}			
In(2)	Out(2)−{j}	{}	{a,i}		
Out(2)	In(3)	{}	{a,i,j}		
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}		
Out(3)	In(3) ∪ In(4)	{}	{a,i}		
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}		
Out(4)	In(2) ∪ In(5)	{}	{a}		
In(5)	Out(5)−{i}	{}	{a}		
Out(5)	In(6)	{}	{a,i}		
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}		
Out(6)	In(6) ∪ In(Exit)	{}	{a}		
In(Exit)	{a}	{}	{a}		

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}			
Out(1)	In(2)	{}	{a,i}		
In(2)	Out(2)−{j}	{}	{a,i}		
Out(2)	In(3)	{}	{a,i,j}		
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}		
Out(3)	In(3) ∪ In(4)	{}	{a,i}		
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}		
Out(4)	In(2) ∪ In(5)	{}	{a}		
In(5)	Out(5)−{i}	{}	{a}		
Out(5)	In(6)	{}	{a,i}		
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}		
Out(6)	In(6) ∪ In(Exit)	{}	{a}		
In(Exit)	{a}	{}	{a}		

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1) Out(1)	Out(1)−{i} In(2)	{} {}	{a} {a,i}		
In(2) Out(2)	Out(2)−{j} In(3)	{} {}	{a,i} {a,i,j}		
In(3) Out(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j} In(3) ∪ In(4)	{} {}	{a,i,j} {a,i}		
In(4) Out(4)	{i} ∪ Out(4)−{i} In(2) ∪ In(5)	{} {}	{a,i} {a}		
In(5) Out(5)	Out(5)−{i} In(6)	{} {}	{a} {a,i}		
In(6) Out(6)	{a,i} ∪ Out(6)−{t4,t5,i} In(6) ∪ In(Exit)	{} {}	{a,i} {a}		
In(Exit)	{a}	{}	{a}		

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}	{a}		
Out(1)	In(2)	{}	{a,i}		
In(2)	Out(2)−{j}	{}	{a,i}		
Out(2)	In(3)	{}	{a,i,j}		
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}		
Out(3)	In(3) ∪ In(4)	{}	{a,i}		
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}		
Out(4)	In(2) ∪ In(5)	{}	{a}		
In(5)	Out(5)−{i}	{}	{a}		
Out(5)	In(6)	{}	{a,i}		
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}		
Out(6)	In(6) ∪ In(Exit)	{}	{a}		
In(Exit)	{a}	{}	{a}	{a}	

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}	{a}		
Out(1)	In(2)	{}	{a,i}		
In(2)	Out(2)−{j}	{}	{a,i}		
Out(2)	In(3)	{}	{a,i,j}		
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}		
Out(3)	In(3) ∪ In(4)	{}	{a,i}		
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}		
Out(4)	In(2) ∪ In(5)	{}	{a}		
In(5)	Out(5)−{i}	{}	{a}		
Out(5)	In(6)	{}	{a,i}		
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}		
Out(6)	In(6) ∪ In(Exit)	{}	{a}	{a,i}	
In(Exit)	{a}	{}	{a}	{a}	

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}	{a}		
Out(1)	In(2)	{}	{a,i}		
In(2)	Out(2)−{j}	{}	{a,i}		
Out(2)	In(3)	{}	{a,i,j}		
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}		
Out(3)	In(3) ∪ In(4)	{}	{a,i}		
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}		
Out(4)	In(2) ∪ In(5)	{}	{a}		
In(5)	Out(5)−{i}	{}	{a}		
Out(5)	In(6)	{}	{a,i}		
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}	{a,i}	
Out(6)	In(6) ∪ In(Exit)	{}	{a}	{a,i}	
In(Exit)	{a}	{}	{a}	{a}	

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}	{a}		
Out(1)	In(2)	{}	{a,i}		
In(2)	Out(2)−{j}	{}	{a,i}		
Out(2)	In(3)	{}	{a,i,j}		
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}		
Out(3)	In(3) ∪ In(4)	{}	{a,i}		
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}		
Out(4)	In(2) ∪ In(5)	{}	{a}		
In(5)	Out(5)−{i}	{}	{a}		
Out(5)	In(6)	{}	{a,i}	{a,i}	
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}	{a,i}	
Out(6)	In(6) ∪ In(Exit)	{}	{a}	{a,i}	
In(Exit)	{a}	{}	{a}	{a}	

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}	{a}		
Out(1)	In(2)	{}	{a,i}		
In(2)	Out(2)−{j}	{}	{a,i}		
Out(2)	In(3)	{}	{a,i,j}		
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}		
Out(3)	In(3) ∪ In(4)	{}	{a,i}		
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}		
Out(4)	In(2) ∪ In(5)	{}	{a}		
In(5)	Out(5)−{i}	{}	{a}	{a}	
Out(5)	In(6)	{}	{a,i}	{a,i}	
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}	{a,i}	
Out(6)	In(6) ∪ In(Exit)	{}	{a}	{a,i}	
In(Exit)	{a}	{}	{a}	{a}	

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}	{a}		
Out(1)	In(2)	{}	{a,i}		
In(2)	Out(2)−{j}	{}	{a,i}		
Out(2)	In(3)	{}	{a,i,j}		
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}		
Out(3)	In(3) ∪ In(4)	{}	{a,i}		
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}		
Out(4)	In(2) ∪ In(5)	{}	{a}	{a,i}	
In(5)	Out(5)−{i}	{}	{a}	{a}	
Out(5)	In(6)	{}	{a,i}	{a,i}	
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}	{a,i}	
Out(6)	In(6) ∪ In(Exit)	{}	{a}	{a,i}	
In(Exit)	{a}	{}	{a}	{a}	

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}	{a}		
Out(1)	In(2)	{}	{a,i}		
In(2)	Out(2)−{j}	{}	{a,i}		
Out(2)	In(3)	{}	{a,i,j}		
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}		
Out(3)	In(3) ∪ In(4)	{}	{a,i}		
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}	{a,i}	
Out(4)	In(2) ∪ In(5)	{}	{a}	{a,i}	
In(5)	Out(5)−{i}	{}	{a}	{a}	
Out(5)	In(6)	{}	{a,i}	{a,i}	
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}	{a,i}	
Out(6)	In(6) ∪ In(Exit)	{}	{a}	{a,i}	
In(Exit)	{a}	{}	{a}	{a}	

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}	{a}		
Out(1)	In(2)	{}	{a,i}		
In(2)	Out(2)−{j}	{}	{a,i}		
Out(2)	In(3)	{}	{a,i,j}		
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}		
Out(3)	In(3) ∪ In(4)	{}	{a,i}	{a,i,j}	
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}	{a,i}	
Out(4)	In(2) ∪ In(5)	{}	{a}	{a,i}	
In(5)	Out(5)−{i}	{}	{a}	{a}	
Out(5)	In(6)	{}	{a,i}	{a,i}	
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}	{a,i}	
Out(6)	In(6) ∪ In(Exit)	{}	{a}	{a,i}	
In(Exit)	{a}	{}	{a}	{a}	

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}	{a}		
Out(1)	In(2)	{}	{a,i}		
In(2)	Out(2)−{j}	{}	{a,i}		
Out(2)	In(3)	{}	{a,i,j}		
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}	{a,i,j}	
Out(3)	In(3) ∪ In(4)	{}	{a,i}	{a,i,j}	
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}	{a,i}	
Out(4)	In(2) ∪ In(5)	{}	{a}	{a,i}	
In(5)	Out(5)−{i}	{}	{a}	{a}	
Out(5)	In(6)	{}	{a,i}	{a,i}	
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}	{a,i}	
Out(6)	In(6) ∪ In(Exit)	{}	{a}	{a,i}	
In(Exit)	{a}	{}	{a}	{a}	

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}	{a}		
Out(1)	In(2)	{}	{a,i}		
In(2)	Out(2)−{j}	{}	{a,i}		
Out(2)	In(3)	{}	{a,i,j}	{a,i,j}	
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}	{a,i,j}	
Out(3)	In(3) ∪ In(4)	{}	{a,i}	{a,i,j}	
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}	{a,i}	
Out(4)	In(2) ∪ In(5)	{}	{a}	{a,i}	
In(5)	Out(5)−{i}	{}	{a}	{a}	
Out(5)	In(6)	{}	{a,i}	{a,i}	
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}	{a,i}	
Out(6)	In(6) ∪ In(Exit)	{}	{a}	{a,i}	
In(Exit)	{a}	{}	{a}	{a}	

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}	{a}		
Out(1)	In(2)	{}	{a,i}		
In(2)	Out(2)−{j}	{}	{a,i}	{a,i}	
Out(2)	In(3)	{}	{a,i,j}	{a,i,j}	
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}	{a,i,j}	
Out(3)	In(3) ∪ In(4)	{}	{a,i}	{a,i,j}	
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}	{a,i}	
Out(4)	In(2) ∪ In(5)	{}	{a}	{a,i}	
In(5)	Out(5)−{i}	{}	{a}	{a}	
Out(5)	In(6)	{}	{a,i}	{a,i}	
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}	{a,i}	
Out(6)	In(6) ∪ In(Exit)	{}	{a}	{a,i}	
In(Exit)	{a}	{}	{a}	{a}	

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}	{a}		
Out(1)	In(2)	{}	{a,i}	{a,i}	
In(2)	Out(2)−{j}	{}	{a,i}	{a,i}	
Out(2)	In(3)	{}	{a,i,j}	{a,i,j}	
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}	{a,i,j}	
Out(3)	In(3) ∪ In(4)	{}	{a,i}	{a,i,j}	
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}	{a,i}	
Out(4)	In(2) ∪ In(5)	{}	{a}	{a,i}	
In(5)	Out(5)−{i}	{}	{a}	{a}	
Out(5)	In(6)	{}	{a,i}	{a,i}	
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}	{a,i}	
Out(6)	In(6) ∪ In(Exit)	{}	{a}	{a,i}	
In(Exit)	{a}	{}	{a}	{a}	

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}	{a}	{a}	
Out(1)	In(2)	{}	{a,i}	{a,i}	
In(2)	Out(2)−{j}	{}	{a,i}	{a,i}	
Out(2)	In(3)	{}	{a,i,j}	{a,i,j}	
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}	{a,i,j}	
Out(3)	In(3) ∪ In(4)	{}	{a,i}	{a,i,j}	
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}	{a,i}	
Out(4)	In(2) ∪ In(5)	{}	{a}	{a,i}	
In(5)	Out(5)−{i}	{}	{a}	{a}	
Out(5)	In(6)	{}	{a,i}	{a,i}	
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}	{a,i}	
Out(6)	In(6) ∪ In(Exit)	{}	{a}	{a,i}	
In(Exit)	{a}	{}	{a}	{a}	

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}	{a}	{a}	
Out(1)	In(2)	{}	{a,i}	{a,i}	
In(2)	Out(2)−{j}	{}	{a,i}	{a,i}	
Out(2)	In(3)	{}	{a,i,j}	{a,i,j}	
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}	{a,i,j}	
Out(3)	In(3) ∪ In(4)	{}	{a,i}	{a,i,j}	{a,i,j}
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}	{a,i}	{a,i}
Out(4)	In(2) ∪ In(5)	{}	{a}	{a,i}	{a,i}
In(5)	Out(5)−{i}	{}	{a}	{a}	{a}
Out(5)	In(6)	{}	{a,i}	{a,i}	{a,i}
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}	{a,i}	{a,i}
Out(6)	In(6) ∪ In(Exit)	{}	{a}	{a,i}	{a,i}
In(Exit)	{a}	{}	{a}	{a}	{a}

Live Variable Analysis Revisited

Set	Eqn	0	1	2	3
In(1)	Out(1)−{i}	{}	{a}	{a}	{a}
Out(1)	In(2)	{}	{a,i}	{a,i}	{a,i}
In(2)	Out(2)−{j}	{}	{a,i}	{a,i}	{a,i}
Out(2)	In(3)	{}	{a,i,j}	{a,i,j}	{a,i,j}
In(3)	{a,i,j} ∪ Out(3)−{t1,t2,t3,j}	{}	{a,i,j}	{a,i,j}	{a,i,j}
Out(3)	In(3) ∪ In(4)	{}	{a,i}	{a,i,j}	{a,i,j}
In(4)	{i} ∪ Out(4)−{i}	{}	{a,i}	{a,i}	{a,i}
Out(4)	In(2) ∪ In(5)	{}	{a}	{a,i}	{a,i}
In(5)	Out(5)−{i}	{}	{a}	{a}	{a}
Out(5)	In(6)	{}	{a,i}	{a,i}	{a,i}
In(6)	{a,i} ∪ Out(6)−{t4,t5,i}	{}	{a,i}	{a,i}	{a,i}
Out(6)	In(6) ∪ In(Exit)	{}	{a}	{a,i}	{a,i}
In(Exit)	{a}	{}	{a}	{a}	{a}

Reaching Definitions

- An assignment of the form $x = e$ for some expression e is said to define x.
- A definition at statement s_1 *reaches* another statement s_2 if:
  - there is some control flow path from s_1 to s_2, such that
  - there is no other definition of x on the path from s_1 to s_2.
- Let $In(s)$ be the set of all definitions that reach s.
- Let $Out(s)$ be the set of all definitions that reach all the immediate successors of s.
- Then $Out(s) = gen(s) \cup (In(s) - kill(s))$, where
  - $gen(s)$ is the set of definitions generated by s, and
  - $kill(s)$ is the set of definitions with the same lhs variables as those in s.
- $In(s) = \bigcup_{t \in pred(s)} Out(t)$

Reaching Definitions vs. Live Variables

- **Live Variables:** *In* and *Out* are the smallest sets such that

$$\begin{aligned} In(s) &= use(s) \cup (Out(s) - def(s)) \\ Out(s) &= \bigcup_{t \in succ(s)} In(t) \end{aligned}$$

- **Reaching Definitions:** *In* and *Out* are the smallest sets such that

$$\begin{aligned} In(s) &= \bigcup_{t \in pred(s)} Out(t) \\ Out(s) &= gen(s) \cup (In(s) - kill(s)) \end{aligned}$$

- The form of equations is identical, and they can be computed using the same procedure, except:
  - Live Variables are best computed backwards through the flow graph (information goes from successors to predecessors).
  - Reaching Definitions are best computed forwards through the flow graph (information goes from predecessors to successors).

Available Expressions

- An expression e is *available* at statement s if, for *every path* that reaches s_1, there is *some* statement s' where e is evaluated.
- Let $In(s)$ be the set of all expressions available immediately before s is evaluated.
- Let $Out(s)$ be the set of all expressions available immediately after s is evaluated.
- Then $Out(s) = gen(s) \cup (In(s) - kill(s))$, where
  - $gen(s)$ is the set of all expressions evaluated in s, and
  - $kill(s)$ is the set of all expressions that use the lhs variables defined in s.
- $In(s) = \bigcap_{t \in pred(s)} Out(t)$
- In and Out are the greatest sets that satisfy the above equations.