Semantics and Verification of Software

Lecture 21: Dataflow Analysis IV (Solving Dataflow Equation Systems)

Thomas Noll

Lehrstuhl für Informatik 2
(Software Modeling and Verification)

RWTH Aachen University

noll@cs.rwth-aachen.de

http://www-i2.informatik.rwth-aachen.de/i2/svsw10/

Summer Semester 2010

Outline

Dataflow Systems I

Definition (Dataflow system)

A dataflow system $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$ consists of

- a finite set of (program) labels L (here: L_c),
- a set of extremal labels $E \subseteq L$ (here: $\{\mathsf{init}(c)\}$ or $\mathsf{final}(c)$),
- a flow relation $F \subseteq L \times L$ (here: $\mathsf{flow}(c)$ or $\mathsf{flow}^R(c)$),
- a complete lattice $(D, \sqsubseteq)$ that satisfies ACC (with LUB operator $\bigsqcup$ and least element $\bot$),
- an extremal value $\iota \in D$ (for the extremal labels), and
- a collection of monotonic transfer functions $\{\varphi_l \mid l \in L\}$ of type $\varphi_l : D \to D$.

Dataflow Systems II

Example

Problem	**Available Expressions**	**Live Variables**
E	$\{\mathsf{init}(c)\}$	$\mathsf{final}(c)$
F	$\mathsf{flow}(c)$	$\mathsf{flow}^R(c)$
D	2^{AExp_c}	2^{Var_c}
$\sqsubseteq$	$\supseteq$	$\subseteq$
$\bigsqcup$	$\bigcap$	$\bigcup$
$\bot$	$AExp_c$	$\emptyset$
ι	$\emptyset$	Var_c
φ_l	$\varphi_l(d) = (d \setminus \mathsf{kill}(B^l)) \cup \mathsf{gen}(B^l)$	

Outline

The Equation System

Definition 21.1 (Dataflow equation system)

Let $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$ be a dataflow system. S defines the following equation system over the set of variables $\{\mathsf{AI}_l \mid l \in L\}$:

$$\mathsf{AI}_l = \begin{cases} \iota & \text{if } l \in E \\ \bigsqcup\{\varphi_{l'}(\mathsf{AI}_{l'}) \mid (l', l) \in F\} & \text{otherwise} \end{cases}$$

The Functional

Just as in the denotational semantics of `while` loops, the equation system determines a functional whose fixpoints are exactly the solutions of the equation system.

Definition 21.2 (Dataflow functional)

The equation system of a dataflow system $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$ induces a functional

$$\Phi_S : D^n \to D^n : (d_{l_1}, \ldots, d_{l_n}) \mapsto (d'_{l_1}, \ldots, d'_{l_n})$$

where $L = \{l_1, \ldots, l_n\}$ and, for each $1 \leq i \leq n$,

$$d'_{l_i} := \begin{cases} \iota & \text{if } l_i \in E \\ \bigsqcup\{\varphi_{l'}(d_{l'}) \mid (l', l_i) \in F\} & \text{otherwise} \end{cases}$$

Fixpoint Iteration I

Remarks:

- $(D, \sqsubseteq)$ being a complete lattice ensures that Φ_S is well defined

Fixpoint Iteration I

Remarks:

- $(D, \sqsubseteq)$ being a complete lattice ensures that Φ_S is well defined
- $(d_1, \ldots, d_n)$ is a solution of the equation system iff it is a fixpoint of Φ_S

Fixpoint Iteration I

Remarks:

- $(D, \sqsubseteq)$ being a complete lattice ensures that Φ_S is well defined
- $(d_1, \ldots, d_n)$ is a solution of the equation system iff it is a fixpoint of Φ_S
- If $(D, \sqsubseteq)$ is a complete lattice satisfying ACC, then so is $(D^n, \sqsubseteq^n)$ (where $(d_1, \ldots, d_n) \sqsubseteq^n (d'_1, \ldots, d'_n)$ iff $d_i \sqsubseteq d'_i$ for every $1 \leq i \leq n$)

Fixpoint Iteration I

Remarks:

- $(D, \sqsubseteq)$ being a complete lattice ensures that Φ_S is well defined
- $(d_1, \ldots, d_n)$ is a solution of the equation system iff it is a fixpoint of Φ_S
- If $(D, \sqsubseteq)$ is a complete lattice satisfying ACC, then so is $(D^n, \sqsubseteq^n)$ (where $(d_1, \ldots, d_n) \sqsubseteq^n (d'_1, \ldots, d'_n)$ iff $d_i \sqsubseteq d'_i$ for every $1 \leq i \leq n$)
- Every transfer function φ_l monotonic in D
  $\implies \Phi_S$ monotonic in D^n

Fixpoint Iteration I

Remarks:

- $(D, \sqsubseteq)$ being a complete lattice ensures that Φ_S is well defined
- $(d_1, \ldots, d_n)$ is a solution of the equation system iff it is a fixpoint of Φ_S
- If $(D, \sqsubseteq)$ is a complete lattice satisfying ACC, then so is $(D^n, \sqsubseteq^n)$ (where $(d_1, \ldots, d_n) \sqsubseteq^n (d'_1, \ldots, d'_n)$ iff $d_i \sqsubseteq d'_i$ for every $1 \leq i \leq n$)
- Every transfer function φ_l monotonic in D
  $\implies \Phi_S$ monotonic in D^n
- Thus the (least) fixpoint is effectively computable by iteration:

$$\mathsf{fix}(\Phi_S) = \bigsqcup \{\Phi_S^i(\bot_{D^n}) \mid i \in \mathbb{N}\}$$

where $\bot_{D^n} = \underbrace{(\bot_D, \ldots, \bot_D)}_{n \text{ times}}$

Fixpoint Iteration I

Remarks:

- $(D, \sqsubseteq)$ being a complete lattice ensures that Φ_S is well defined
- $(d_1, \ldots, d_n)$ is a solution of the equation system iff it is a fixpoint of Φ_S
- If $(D, \sqsubseteq)$ is a complete lattice satisfying ACC, then so is $(D^n, \sqsubseteq^n)$ (where $(d_1, \ldots, d_n) \sqsubseteq^n (d'_1, \ldots, d'_n)$ iff $d_i \sqsubseteq d'_i$ for every $1 \leq i \leq n$)
- Every transfer function φ_l monotonic in D
  $\implies$ Φ_S monotonic in D^n
- Thus the (least) fixpoint is effectively computable by iteration:

$$\mathsf{fix}(\Phi_S) = \bigsqcup \{\Phi_S^i(\bot_{D^n}) \mid i \in \mathbb{N}\}$$

  where $\bot_{D^n} = \underbrace{(\bot_D, \ldots, \bot_D)}_{n \text{ times}}$
- If maximal length of chains in D is m
  $\implies$ maximal length of chains in D^n is $m \cdot n$
  $\implies$ fixpoint iteration requires at most $m \cdot n$ steps

Fixpoint Iteration II

Example 21.3 (Available Expressions; cf. Example 18.9)

Program:

$c = $ [x := a+b]^1;
    [y := a*b]^2;
    while [y > a+b]^3 do
      [a := a+1]^4;
      [x := a+b]^5

Fixpoint Iteration II

Example 21.3 (Available Expressions; cf. Example 18.9)

Program:

$c = [\texttt{x := a+b}]^1;$
$\quad [\texttt{y := a*b}]^2;$
$\quad \texttt{while } [\texttt{y > a+b}]^3 \texttt{ do}$
$\qquad [\texttt{a := a+1}]^4;$
$\qquad [\texttt{x := a+b}]^5$

Equation system:

$$
\begin{aligned}
\mathsf{AE}_1 &= \emptyset \\
\mathsf{AE}_2 &= \mathsf{AE}_1 \cup \{\texttt{a+b}\} \\
\mathsf{AE}_3 &= (\mathsf{AE}_2 \cup \{\texttt{a*b}\}) \cap (\mathsf{AE}_5 \cup \{\texttt{a+b}\}) \\
\mathsf{AE}_4 &= \mathsf{AE}_3 \cup \{\texttt{a+b}\} \\
\mathsf{AE}_5 &= \mathsf{AE}_4 \setminus \{\texttt{a+b}, \texttt{a*b}, \texttt{a+1}\}
\end{aligned}
$$

Fixpoint Iteration II

Example 21.3 (Available Expressions; cf. Example 18.9)

Program:

$c =$ [x := a+b]^1;
    [y := a*b]^2;
    while [y > a+b]^3 do
        [a := a+1]^4;
        [x := a+b]^5

Equation system:

$$\mathsf{AE}_1 = \emptyset$$
$$\mathsf{AE}_2 = \mathsf{AE}_1 \cup \{\texttt{a+b}\}$$
$$\mathsf{AE}_3 = (\mathsf{AE}_2 \cup \{\texttt{a*b}\}) \cap (\mathsf{AE}_5 \cup \{\texttt{a+b}\})$$
$$\mathsf{AE}_4 = \mathsf{AE}_3 \cup \{\texttt{a+b}\}$$
$$\mathsf{AE}_5 = \mathsf{AE}_4 \setminus \{\texttt{a+b}, \texttt{a*b}, \texttt{a+1}\}$$

Fixpoint iteration:

i	1	2	3	4	5
0	$AExp_c$	$AExp_c$	$AExp_c$	$AExp_c$	$AExp_c$

Fixpoint Iteration II

Example 21.3 (Available Expressions; cf. Example 18.9)

Program:

$$c = [\texttt{x := a+b}]^1;\ [\texttt{y := a*b}]^2;\ \texttt{while}\ [\texttt{y > a+b}]^3\ \texttt{do}\ [\texttt{a := a+1}]^4;\ [\texttt{x := a+b}]^5$$

Equation system:

$$\begin{aligned}
\mathsf{AE}_1 &= \emptyset \\
\mathsf{AE}_2 &= \mathsf{AE}_1 \cup \{\texttt{a+b}\} \\
\mathsf{AE}_3 &= (\mathsf{AE}_2 \cup \{\texttt{a*b}\}) \cap (\mathsf{AE}_5 \cup \{\texttt{a+b}\}) \\
\mathsf{AE}_4 &= \mathsf{AE}_3 \cup \{\texttt{a+b}\} \\
\mathsf{AE}_5 &= \mathsf{AE}_4 \setminus \{\texttt{a+b}, \texttt{a*b}, \texttt{a+1}\}
\end{aligned}$$

Fixpoint iteration:

i	1	2	3	4	5
0	$AExp_c$	$AExp_c$	$AExp_c$	$AExp_c$	$AExp_c$
1	$\emptyset$	$AExp_c$	$AExp_c$	$AExp_c$	$\emptyset$

Fixpoint Iteration II

Example 21.3 (Available Expressions; cf. Example 18.9)

Program:

$$c = [\texttt{x := a+b}]^1;\\ [\texttt{y := a*b}]^2;\\ \texttt{while } [\texttt{y > a+b}]^3 \texttt{ do}\\ [\texttt{a := a+1}]^4;\\ [\texttt{x := a+b}]^5$$

Equation system:

$$\begin{aligned} \mathsf{AE}_1 &= \emptyset \\ \mathsf{AE}_2 &= \mathsf{AE}_1 \cup \{\texttt{a+b}\} \\ \mathsf{AE}_3 &= (\mathsf{AE}_2 \cup \{\texttt{a*b}\}) \cap (\mathsf{AE}_5 \cup \{\texttt{a+b}\}) \\ \mathsf{AE}_4 &= \mathsf{AE}_3 \cup \{\texttt{a+b}\} \\ \mathsf{AE}_5 &= \mathsf{AE}_4 \setminus \{\texttt{a+b}, \texttt{a*b}, \texttt{a+1}\} \end{aligned}$$

Fixpoint iteration:

i	1	2	3	4	5
0	$AExp_c$	$AExp_c$	$AExp_c$	$AExp_c$	$AExp_c$
1	$\emptyset$	$AExp_c$	$AExp_c$	$AExp_c$	$\emptyset$
2	$\emptyset$	$\{\texttt{a+b}\}$	$\{\texttt{a+b}\}$	$AExp_c$	$\emptyset$

Fixpoint Iteration II

Example 21.3 (Available Expressions; cf. Example 18.9)

Program:

$$
\begin{aligned}
c = {} & [\texttt{x := a+b}]^1; \\
& [\texttt{y := a*b}]^2; \\
& \texttt{while } [\texttt{y > a+b}]^3 \texttt{ do} \\
& \quad [\texttt{a := a+1}]^4; \\
& \quad [\texttt{x := a+b}]^5
\end{aligned}
$$

Equation system:

$$
\begin{aligned}
\mathsf{AE}_1 &= \emptyset \\
\mathsf{AE}_2 &= \mathsf{AE}_1 \cup \{\texttt{a+b}\} \\
\mathsf{AE}_3 &= (\mathsf{AE}_2 \cup \{\texttt{a*b}\}) \cap (\mathsf{AE}_5 \cup \{\texttt{a+b}\}) \\
\mathsf{AE}_4 &= \mathsf{AE}_3 \cup \{\texttt{a+b}\} \\
\mathsf{AE}_5 &= \mathsf{AE}_4 \setminus \{\texttt{a+b}, \texttt{a*b}, \texttt{a+1}\}
\end{aligned}
$$

Fixpoint iteration:

i	1	2	3	4	5
0	$AExp_c$	$AExp_c$	$AExp_c$	$AExp_c$	$AExp_c$
1	$\emptyset$	$AExp_c$	$AExp_c$	$AExp_c$	$\emptyset$
2	$\emptyset$	$\{\texttt{a+b}\}$	$\{\texttt{a+b}\}$	$AExp_c$	$\emptyset$
3	$\emptyset$	$\{\texttt{a+b}\}$	$\{\texttt{a+b}\}$	$\{\texttt{a+b}\}$	$\emptyset$

Fixpoint Iteration II

Example 21.3 (Available Expressions; cf. Example 18.9)

Program:

c = [x := a+b]^1;
[y := a*b]^2;
while [y > a+b]^3 do
[a := a+1]^4;
[x := a+b]^5

Equation system:

$$\begin{aligned}
\mathsf{AE}_1 &= \emptyset \\
\mathsf{AE}_2 &= \mathsf{AE}_1 \cup \{\texttt{a+b}\} \\
\mathsf{AE}_3 &= (\mathsf{AE}_2 \cup \{\texttt{a*b}\}) \cap (\mathsf{AE}_5 \cup \{\texttt{a+b}\}) \\
\mathsf{AE}_4 &= \mathsf{AE}_3 \cup \{\texttt{a+b}\} \\
\mathsf{AE}_5 &= \mathsf{AE}_4 \setminus \{\texttt{a+b}, \texttt{a*b}, \texttt{a+1}\}
\end{aligned}$$

Fixpoint iteration:

i	1	2	3	4	5
0	$AExp_c$	$AExp_c$	$AExp_c$	$AExp_c$	$AExp_c$
1	$\emptyset$	$AExp_c$	$AExp_c$	$AExp_c$	$\emptyset$
2	$\emptyset$	{a+b}	{a+b}	$AExp_c$	$\emptyset$
3	$\emptyset$	{a+b}	{a+b}	{a+b}	$\emptyset$
4	$\emptyset$	{a+b}	{a+b}	{a+b}	$\emptyset$

Fixpoint Iteration III

Example 21.4 (Live Variables; cf. Example 19.3)

Program:

```
[x := 2]^1;[y := 4]^2;
[x := 1]^3;
if [y > 0]^4 then
  [z := x]^5
else
  [z := y*y]^6;
[x := z]^7
```

Fixpoint Iteration III

Example 21.4 (Live Variables; cf. Example 19.3)

Program:

```
[x := 2]^1;[y := 4]^2;
[x := 1]^3;
if [y > 0]^4 then
  [z := x]^5
else
  [z := y*y]^6;
[x := z]^7
```

Equation system:

$$\begin{aligned}
\mathsf{LV}_1 &= \mathsf{LV}_2 \setminus \{\mathtt{y}\} \\
\mathsf{LV}_2 &= \mathsf{LV}_3 \setminus \{\mathtt{x}\} \\
\mathsf{LV}_3 &= \mathsf{LV}_4 \cup \{\mathtt{y}\} \\
\mathsf{LV}_4 &= ((\mathsf{LV}_5 \setminus \{\mathtt{z}\}) \cup \{\mathtt{x}\}) \cup ((\mathsf{LV}_6 \setminus \{\mathtt{z}\}) \cup \{\mathtt{y}\}) \\
\mathsf{LV}_5 &= (\mathsf{LV}_7 \setminus \{\mathtt{x}\}) \cup \{\mathtt{z}\} \\
\mathsf{LV}_6 &= (\mathsf{LV}_7 \setminus \{\mathtt{x}\}) \cup \{\mathtt{z}\} \\
\mathsf{LV}_7 &= \{\mathtt{x}, \mathtt{y}, \mathtt{z}\}
\end{aligned}$$

Fixpoint Iteration III

Example 21.4 (Live Variables; cf. Example 19.3)

Program:

```
[x := 2]^1;[y := 4]^2;
[x := 1]^3;
if [y > 0]^4 then
  [z := x]^5
else
  [z := y*y]^6;
[x := z]^7
```

Equation system:

$$
\begin{aligned}
\mathsf{LV}_1 &= \mathsf{LV}_2 \setminus \{\mathtt{y}\} \\
\mathsf{LV}_2 &= \mathsf{LV}_3 \setminus \{\mathtt{x}\} \\
\mathsf{LV}_3 &= \mathsf{LV}_4 \cup \{\mathtt{y}\} \\
\mathsf{LV}_4 &= ((\mathsf{LV}_5 \setminus \{\mathtt{z}\}) \cup \{\mathtt{x}\}) \cup ((\mathsf{LV}_6 \setminus \{\mathtt{z}\}) \cup \{\mathtt{y}\}) \\
\mathsf{LV}_5 &= (\mathsf{LV}_7 \setminus \{\mathtt{x}\}) \cup \{\mathtt{z}\} \\
\mathsf{LV}_6 &= (\mathsf{LV}_7 \setminus \{\mathtt{x}\}) \cup \{\mathtt{z}\} \\
\mathsf{LV}_7 &= \{\mathtt{x}, \mathtt{y}, \mathtt{z}\}
\end{aligned}
$$

Fixpoint iteration:

i	1	2	3	4	5	6	7
0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Fixpoint Iteration III

Example 21.4 (Live Variables; cf. Example 19.3)

Program:

```
[x := 2]^1;[y := 4]^2;
[x := 1]^3;
if [y > 0]^4 then
  [z := x]^5
else
  [z := y*y]^6;
[x := z]^7
```

Equation system:

$$
\begin{aligned}
LV_1 &= LV_2 \setminus \{y\} \\
LV_2 &= LV_3 \setminus \{x\} \\
LV_3 &= LV_4 \cup \{y\} \\
LV_4 &= ((LV_5 \setminus \{z\}) \cup \{x\}) \cup ((LV_6 \setminus \{z\}) \cup \{y\}) \\
LV_5 &= (LV_7 \setminus \{x\}) \cup \{z\} \\
LV_6 &= (LV_7 \setminus \{x\}) \cup \{z\} \\
LV_7 &= \{x, y, z\}
\end{aligned}
$$

Fixpoint iteration:

i	1	2	3	4	5	6	7
0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
1	$\emptyset$	$\emptyset$	$\{y\}$	$\{x, y\}$	$\{z\}$	$\{z\}$	$\{x, y, z\}$

Fixpoint Iteration III

Example 21.4 (Live Variables; cf. Example 19.3)

Program:

```
[x := 2]^1;[y := 4]^2;
[x := 1]^3;
if [y > 0]^4 then
  [z := x]^5
else
  [z := y*y]^6;
[x := z]^7
```

Equation system:

$$
\begin{aligned}
\mathsf{LV}_1 &= \mathsf{LV}_2 \setminus \{\mathtt{y}\} \\
\mathsf{LV}_2 &= \mathsf{LV}_3 \setminus \{\mathtt{x}\} \\
\mathsf{LV}_3 &= \mathsf{LV}_4 \cup \{\mathtt{y}\} \\
\mathsf{LV}_4 &= ((\mathsf{LV}_5 \setminus \{\mathtt{z}\}) \cup \{\mathtt{x}\}) \cup ((\mathsf{LV}_6 \setminus \{\mathtt{z}\}) \cup \{\mathtt{y}\}) \\
\mathsf{LV}_5 &= (\mathsf{LV}_7 \setminus \{\mathtt{x}\}) \cup \{\mathtt{z}\} \\
\mathsf{LV}_6 &= (\mathsf{LV}_7 \setminus \{\mathtt{x}\}) \cup \{\mathtt{z}\} \\
\mathsf{LV}_7 &= \{\mathtt{x}, \mathtt{y}, \mathtt{z}\}
\end{aligned}
$$

Fixpoint iteration:

i	1	2	3	4	5	6	7
0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
1	$\emptyset$	$\emptyset$	$\{y\}$	$\{x, y\}$	$\{z\}$	$\{z\}$	$\{x, y, z\}$
2	$\emptyset$	$\{y\}$	$\{x, y\}$	$\{x, y\}$	$\{y, z\}$	$\{y, z\}$	$\{x, y, z\}$

Fixpoint Iteration III

Example 21.4 (Live Variables; cf. Example 19.3)

Program:

```
[x := 2]^1;[y := 4]^2;
[x := 1]^3;
if [y > 0]^4 then
  [z := x]^5
else
  [z := y*y]^6;
[x := z]^7
```

Equation system:

$$\begin{aligned}
\mathsf{LV}_1 &= \mathsf{LV}_2 \setminus \{y\}\\
\mathsf{LV}_2 &= \mathsf{LV}_3 \setminus \{x\}\\
\mathsf{LV}_3 &= \mathsf{LV}_4 \cup \{y\}\\
\mathsf{LV}_4 &= ((\mathsf{LV}_5 \setminus \{z\}) \cup \{x\}) \cup ((\mathsf{LV}_6 \setminus \{z\}) \cup \{y\})\\
\mathsf{LV}_5 &= (\mathsf{LV}_7 \setminus \{x\}) \cup \{z\}\\
\mathsf{LV}_6 &= (\mathsf{LV}_7 \setminus \{x\}) \cup \{z\}\\
\mathsf{LV}_7 &= \{x, y, z\}
\end{aligned}$$

Fixpoint iteration:

i	1	2	3	4	5	6	7
0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
1	$\emptyset$	$\emptyset$	$\{y\}$	$\{x, y\}$	$\{z\}$	$\{z\}$	$\{x, y, z\}$
2	$\emptyset$	$\{y\}$	$\{x, y\}$	$\{x, y\}$	$\{y, z\}$	$\{y, z\}$	$\{x, y, z\}$
3	$\emptyset$	$\{y\}$	$\{x, y\}$	$\{x, y\}$	$\{y, z\}$	$\{y, z\}$	$\{x, y, z\}$

Outline

Uniqueness of Solutions

Just as in the denotational semantics of `while` loops, solutions of dataflow equation systems are not unique.

Uniqueness of Solutions

Just as in the denotational semantics of `while` loops, solutions of dataflow equation systems are not unique.

Example 21.5

1. Available Expressions: see Exercise 12.1

Uniqueness of Solutions

Just as in the denotational semantics of `while` loops, solutions of dataflow equation systems are not unique.

Example 21.5

1. Available Expressions: see Exercise 12.1
2. Live Variables: consider

   while $[\texttt{x>1}]^1$ do
     $[\texttt{skip}]^2$;
   $[\texttt{x := x+1}]^3$;
   $[\texttt{y := 0}]^4$

Uniqueness of Solutions

Just as in the denotational semantics of `while` loops, solutions of dataflow equation systems are not unique.

Example 21.5

1. Available Expressions: see Exercise 12.1
2. Live Variables: consider

$$\begin{array}{ll} \texttt{while } [\texttt{x>1}]^1 \texttt{ do} & \Longrightarrow \; \mathsf{LV}_1 = \mathsf{LV}_2 \cup (\mathsf{LV}_3 \cup \{\texttt{x}\}) \\ \quad [\texttt{skip}]^2; & \; \mathsf{LV}_2 = \mathsf{LV}_1 \cup \{\texttt{x}\} \\ [\texttt{x := x+1}]^3; & \; \mathsf{LV}_3 = \mathsf{LV}_4 \setminus \{\texttt{y}\} \\ [\texttt{y := 0}]^4 & \; \mathsf{LV}_4 = \{\texttt{x}, \texttt{y}\} \end{array}$$

Uniqueness of Solutions

Just as in the denotational semantics of `while` loops, solutions of dataflow equation systems are not unique.

Example 21.5

1. Available Expressions: see Exercise 12.1
2. Live Variables: consider

```
while [x>1]^1 do
  [skip]^2;
[x := x+1]^3;
[y := 0]^4
```

$$\Longrightarrow \begin{aligned} LV_1 &= LV_2 \cup (LV_3 \cup \{x\}) \\ LV_2 &= LV_1 \cup \{x\} \\ LV_3 &= LV_4 \setminus \{y\} \\ LV_4 &= \{x, y\} \end{aligned}$$

$$\Longrightarrow LV_3 = \{x\}$$

Uniqueness of Solutions

Just as in the denotational semantics of `while` loops, solutions of dataflow equation systems are not unique.

Example 21.5

1. Available Expressions: see Exercise 12.1
2. Live Variables: consider

```
while [x>1]^1 do
  [skip]^2;
[x := x+1]^3;
[y := 0]^4
```

$$\Longrightarrow \begin{aligned} \mathsf{LV}_1 &= \mathsf{LV}_2 \cup (\mathsf{LV}_3 \cup \{x\}) \\ \mathsf{LV}_2 &= \mathsf{LV}_1 \cup \{x\} \\ \mathsf{LV}_3 &= \mathsf{LV}_4 \setminus \{y\} \\ \mathsf{LV}_4 &= \{x, y\} \end{aligned}$$

$$\Longrightarrow \mathsf{LV}_3 = \{x\}$$

$$\Longrightarrow \begin{aligned} \mathsf{LV}_1 &= \mathsf{LV}_2 \cup \{x\} \\ &= \mathsf{LV}_1 \cup \{x\} \end{aligned}$$

Uniqueness of Solutions

Just as in the denotational semantics of `while` loops, solutions of dataflow equation systems are not unique.

Example 21.5

1. Available Expressions: see Exercise 12.1
2. Live Variables: consider

```
while [x>1]^1 do
  [skip]^2;
[x := x+1]^3;
[y := 0]^4
```

$$\Longrightarrow \begin{aligned} \mathsf{LV}_1 &= \mathsf{LV}_2 \cup (\mathsf{LV}_3 \cup \{x\}) \\ \mathsf{LV}_2 &= \mathsf{LV}_1 \cup \{x\} \\ \mathsf{LV}_3 &= \mathsf{LV}_4 \setminus \{y\} \\ \mathsf{LV}_4 &= \{x, y\} \end{aligned}$$

$$\Longrightarrow \mathsf{LV}_3 = \{x\}$$

$$\Longrightarrow \begin{aligned} \mathsf{LV}_1 &= \mathsf{LV}_2 \cup \{x\} \\ &= \mathsf{LV}_1 \cup \{x\} \end{aligned}$$

$\Longrightarrow$ Solutions: $\mathsf{LV}_1 = \mathsf{LV}_2 = (\{x\} \text{ or } \{x, y\})$, $\mathsf{LV}_3 = \mathsf{LV}_4 = \emptyset$

Uniqueness of Solutions

Just as in the denotational semantics of `while` loops, solutions of dataflow equation systems are not unique.

Example 21.5

1. Available Expressions: see Exercise 12.1
2. Live Variables: consider

```
while [x>1]^1 do
  [skip]^2;
[x := x+1]^3;
[y := 0]^4
```

$$\Longrightarrow \begin{aligned} \mathsf{LV}_1 &= \mathsf{LV}_2 \cup (\mathsf{LV}_3 \cup \{x\}) \\ \mathsf{LV}_2 &= \mathsf{LV}_1 \cup \{x\} \\ \mathsf{LV}_3 &= \mathsf{LV}_4 \setminus \{y\} \\ \mathsf{LV}_4 &= \{x, y\} \end{aligned}$$

$$\Longrightarrow \mathsf{LV}_3 = \{x\}$$

$$\Longrightarrow \begin{aligned} \mathsf{LV}_1 &= \mathsf{LV}_2 \cup \{x\} \\ &= \mathsf{LV}_1 \cup \{x\} \end{aligned}$$

$\Longrightarrow$ Solutions: $\mathsf{LV}_1 = \mathsf{LV}_2 = (\{x\} \text{ or } \{x, y\})$, $\mathsf{LV}_3 = \mathsf{LV}_4 = \emptyset$

Here: least solution $\{x\}$ (maximal potential for optimization)

Outline

A Worklist Algorithm I

Observation: fixpoint iteration re-computes every AI_l in every step

A Worklist Algorithm I

Observation: fixpoint iteration re-computes every AI_l in every step
$\Longrightarrow$ redundant if $\mathsf{AI}_{l'}$ at no F-predecessor l' changed

A Worklist Algorithm I

Observation: fixpoint iteration re-computes every AI_l in every step

$\Longrightarrow$ redundant if $\mathsf{AI}_{l'}$ at no F-predecessor l' changed

$\Longrightarrow$ optimization by worklist

A Worklist Algorithm I

Observation: fixpoint iteration re-computes every AI_l in every step
$\Longrightarrow$ redundant if $\mathsf{AI}_{l'}$ at no F-predecessor l' changed
$\Longrightarrow$ optimization by worklist

Algorithm 21.6 (Worklist algorithm)

Input: *dataflow system* $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$

A Worklist Algorithm I

Observation: fixpoint iteration re-computes every AI_l in every step

$\Longrightarrow$ redundant if $\mathsf{AI}_{l'}$ at no F-predecessor l' changed

$\Longrightarrow$ optimization by worklist

Algorithm 21.6 (Worklist algorithm)

Input: *dataflow system* $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$

Variables: $W \in (L \times L)^*$, $\{\mathsf{AI}_l \in D \mid l \in L\}$

A Worklist Algorithm I

Observation: fixpoint iteration re-computes every AI_l in every step

$\Longrightarrow$ redundant if $\mathsf{AI}_{l'}$ at no F-predecessor l' changed

$\Longrightarrow$ optimization by worklist

Algorithm 21.6 (Worklist algorithm)

Input: *dataflow system* $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$

Variables: $W \in (L \times L)^*$, $\{\mathsf{AI}_l \in D \mid l \in L\}$

Procedure: $W := \varepsilon$; **for** $(l, l') \in F$ **do** $W := (l, l') \cdot W$; % *Initialize W*

for $l \in L$ **do** % *Initialize* AI

  if $l \in E$ **then** $\mathsf{AI}_l := \iota$ **else** $\mathsf{AI}_l := \bot_D$;

A Worklist Algorithm I

Observation: fixpoint iteration re-computes every AI_l in every step

$\Longrightarrow$ redundant if $\mathsf{AI}_{l'}$ at no F-predecessor l' changed

$\Longrightarrow$ optimization by worklist

Algorithm 21.6 (Worklist algorithm)

Input: *dataflow system* $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$

Variables: $W \in (L \times L)^*$, $\{\mathsf{AI}_l \in D \mid l \in L\}$

Procedure: $W := \varepsilon$; **for** $(l, l') \in F$ **do** $W := (l, l') \cdot W$; % *Initialize* W
for $l \in L$ **do** % *Initialize* AI
  if $l \in E$ **then** $\mathsf{AI}_l := \iota$ **else** $\mathsf{AI}_l := \bot_D$;
while $W \neq \varepsilon$ **do**
  $(l, l') :=$ **head**(W); $W :=$ **tail**(W);
  if $\varphi_l(\mathsf{AI}_l) \not\sqsubseteq \mathsf{AI}_{l'}$ **then** % *Fixpoint not yet reached*
    $\mathsf{AI}_{l'} := \mathsf{AI}_{l'} \sqcup \varphi_l(\mathsf{AI}_l)$;
    for $(l', l'') \in F$ **do** $W := (l', l'') \cdot W$;

A Worklist Algorithm I

Observation: fixpoint iteration re-computes every AI_l in every step
$\Longrightarrow$ redundant if $\mathsf{AI}_{l'}$ at no F-predecessor l' changed
$\Longrightarrow$ optimization by worklist

Algorithm 21.6 (Worklist algorithm)

Input: *dataflow system* $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$

Variables: $W \in (L \times L)^*$, $\{\mathsf{AI}_l \in D \mid l \in L\}$

Procedure: $W := \varepsilon$; **for** $(l, l') \in F$ **do** $W := (l, l') \cdot W$; *% Initialize W*
for $l \in L$ **do** *% Initialize* AI
  if $l \in E$ **then** $\mathsf{AI}_l := \iota$ **else** $\mathsf{AI}_l := \bot_D$;
while $W \neq \varepsilon$ **do**
  $(l, l') :=$ **head**(W); $W :=$ **tail**(W);
  if $\varphi_l(\mathsf{AI}_l) \not\sqsubseteq \mathsf{AI}_{l'}$ **then** *% Fixpoint not yet reached*
    $\mathsf{AI}_{l'} := \mathsf{AI}_{l'} \sqcup \varphi_l(\mathsf{AI}_l)$;
    for $(l', l'') \in F$ **do** $W := (l', l'') \cdot W$;

Output: $\{\mathsf{AI}_l \mid l \in L\}$

A Worklist Algorithm II

Example 21.7 (Worklist algorithm)

Available Expression analysis for $c =$ [x := a+b][1];
[y := a*b][2];
while [y > a+b][3] do
  [a := a+1][4];
  [x := a+b][5]

(cf. Examples 18.9 and 21.3)

Transfer functions:

$$\begin{aligned}
\varphi_1(A) &= A \cup \{\texttt{a+b}\} \\
\varphi_2(A) &= A \cup \{\texttt{a*b}\} \\
\varphi_3(A) &= A \cup \{\texttt{a+b}\} \\
\varphi_4(A) &= A \setminus \{\texttt{a+b}, \texttt{a*b}, \texttt{a+1}\} \\
\varphi_5(A) &= A \cup \{\texttt{a+b}\}
\end{aligned}$$

Computation protocol: on the board

Properties of the algorithm:

Theorem 21.8 (Correctness of worklist algorithm)

Given a dataflow system $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$*, Algorithm 21.6 always terminates and computes* $\mathsf{fix}(\Phi_S)$.

A Worklist Algorithm III

Properties of the algorithm:

Theorem 21.8 (Correctness of worklist algorithm)

Given a dataflow system $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$, *Algorithm 21.6 always terminates and computes* $\mathsf{fix}(\Phi_S)$.

Proof.

see [Nielson/Nielson/Hankin 2005, p. 75 ff] □