

Semantics and Verification of Software

Lecture 17: Dataflow Analysis

Thomas Noll

Lehrstuhl für Informatik 2
RWTH Aachen University
noll@cs.rwth-aachen.de

<http://www-i2.informatik.rwth-aachen.de/i2/svsw/>

Summer semester 2007

- 1 Repetition: The Dataflow Analysis Framework
- 2 Solving Dataflow Equation Systems

Definition (Complete lattices)

A **complete lattice** is a partial order $(D, \sqsubseteq)$ such that all subsets of D have least upper as well as greatest lower bounds. In this case,

$$\begin{aligned}\perp &:= \bigsqcup \emptyset = \bigsqcap D \text{ and} \\ \top &:= \bigsqcap \emptyset = \bigsqcup D\end{aligned}$$

denote the **least** and the **greatest element** of D , respectively.

Example

- 1 (Available Expressions) $(D, \sqsubseteq) = (2^{AExp_c}, \supseteq)$ is a complete lattice with $\perp = AExp_c$ and $\top = \emptyset$
- 2 (Live Variables) $(D, \sqsubseteq) = (2^{Var_c}, \subseteq)$ is a complete lattice with $\perp = \emptyset$ and $\top = Var_c$

Chains represent the approximation of the analysis information.

Definition (Chain; repetition of Def. 5.6 and 5.8)

Let $(D, \sqsubseteq)$ be a partial order.

- 1 A subset $S \subseteq D$ is called a **chain** in D if, for every $s_1, s_2 \in S$,
$$s_1 \sqsubseteq s_2 \text{ or } s_2 \sqsubseteq s_1$$

(that is, S is a totally ordered subset of D).
- 2 $(D, \sqsubseteq)$ is called **chain complete (CCPO)** if every of its chains has a least upper bound.
- 3 $(D, \sqsubseteq)$ satisfies the **Ascending Chain Condition (ACC)** if each ascending chain $d_1 \sqsubseteq d_2 \sqsubseteq \dots$ eventually stabilizes, i.e., there exists $n \in \mathbb{N}$ such that $d_n = d_{n+1} = \dots$.

Definition (Monotonicity; repetition of Def. 6.1)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be partial orders, and let $F : D \rightarrow D'$. F is called **monotonic (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$)** if, for every $d_1, d_2 \in D$,

$$d_1 \sqsubseteq d_2 \implies F(d_1) \sqsubseteq' F(d_2).$$

Example

- ❶ (Available Expressions) $(D, \sqsubseteq) = (2^{AExp_c}, \supseteq)$
Each transfer function $\varphi_{l'}(A) := (A \setminus \text{kill}_{AE}(B^{l'})) \cup \text{gen}_{AE}(B^{l'})$ is monotonic
- ❷ (Live Variables) $(D, \sqsubseteq) = (2^{Var_c}, \subseteq)$
Each transfer function $\varphi_{l'}(V) := (V \setminus \text{kill}_{LV}(B^{l'})) \cup \text{gen}_{LV}(B^{l'})$ is monotonic

Fixpoints

Theorem (Fixpoint Theorem; repetition of Thm. 7.1)

Let $(D, \sqsubseteq)$ be a CCPO and $F : D \rightarrow D$ continuous. Then

$$\text{fix}(F) := \bigsqcup \{F^n (\bigsqcup \emptyset) \mid n \in \mathbb{N}\}$$

is the least fixpoint of F .

Definition (Continuity; repetition of Def. 6.5)

Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be CCPOs and $F : D \rightarrow D'$ monotonic. Then F is called **continuous** (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every non-empty chain $S \subseteq D$,

$$F(\bigsqcup S) = \bigsqcup F(S).$$

Corollary

Monotonic functions on partial orders that satisfy ACC are continuous.

Definition (Dataflow system)

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

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

Example

Problem	Available Expressions	Live Variables
E	$\{\text{init}(c)\}$	$\text{final}(c)$
F	$\text{flow}(c)$	$\text{flow}^R(c)$
D	2^{AExp_c}	2^{Var_c}
$\sqsubseteq$	$\supseteq$	$\subseteq$
$\sqcup$	$\cap$	$\cup$
$\perp$	$AExp_c$	$\emptyset$
ι	$\emptyset$	$\emptyset$
φ_l	$\varphi_l(d) = (d \setminus \text{kill}(B^l)) \cup \text{gen}(B^l)$	

- 1 Repetition: The Dataflow Analysis Framework
- 2 Solving Dataflow Equation Systems

Definition 17.1 (Dataflow equation system)

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

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

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

Definition 17.2 (Dataflow functional)

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

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

where $Lab = \{l_1, \dots, l_n\}$ and

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

Remarks:

- $(d_1, \dots, d_n)$ 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 $(d_1, \dots, d_n) \sqsubseteq^n (d'_1, \dots, d'_n)$ 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 **fixpoint is effectively computable** by iteration:

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

where $\perp_{D^n} = \underbrace{(\perp_D, \dots, \perp_D)}_{n \text{ times}}$

- 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 17.3 (Available Expressions; cf. Example 15.1)

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:

```
AE1 = ∅  
AE2 = AE1 ∪ {a+b}  
AE3 = (AE2 ∪ {a*b}) ∩ (AE5 ∪ {a+b})  
AE4 = AE3 ∪ {a+b}  
AE5 = AE4 \ {a+b, a*b, a+1}
```

Fixpoint iteration:

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

Fixpoint Iteration III

Example 17.4 (Live Variables; cf. Example 15.4)

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:

```
LV1 = LV2 \ {y}  
LV2 = LV3 \ {x}  
LV3 = LV4 ∪ {y}  
LV4 = ((LV5 \ {z}) ∪ {x}) ∪ ((LV6 \ {z}) ∪ {y})  
LV5 = (LV7 \ {x}) ∪ {z}  
LV6 = (LV7 \ {x}) ∪ {z}  
LV7 = ∅
```

Fixpoint iteration:

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