

Static Program Analysis

Lecture 7: Dataflow Analysis VI (MOP vs. Fixpoint Solution)

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/spa11/`

Summer Semester 2011

- 1 Repetition: MOP Solution and Constant Propagation
- 2 MOP vs. Fixpoint Solution
- 3 Dataflow Analysis with Non-ACC Domains
- 4 Example: Interval Analysis

Definition (MOP solution)

Let $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$ be a dataflow system where $L = \{l_1, \dots, l_n\}$. The **MOP solution** for S is determined by

$$\text{mop}(S) := (\text{mop}(l_1), \dots, \text{mop}(l_n)) \in D^n$$

where, for every $l \in L$,

$$\text{mop}(l) := \bigsqcup \{\varphi_p(\iota) \mid p \in \text{Path}(l)\}.$$

Remark:

- $\text{Path}(l)$ is generally infinite

⇒ not clear how to compute $\text{mop}(l)$

- In fact: MOP solution generally undecidable (later)

Formalizing Constant Propagation Analysis I

The **dataflow system** $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$ is given by

- set of labels $L := L_c$,
- extremal labels $E := \{\text{init}(c)\}$ (forward problem),
- flow relation $F := \text{flow}(c)$ (forward problem),
- complete lattice $(D, \sqsubseteq)$ where
 - $D := \{\delta \mid \delta : \text{Var}_c \rightarrow \mathbb{Z} \cup \{\perp, \top\}\}$
 - $\delta(x) = z \in \mathbb{Z}$: x has **constant value** z
 - $\delta(x) = \perp$: x **undefined**
 - $\delta(x) = \top$: x **overdefined** (i.e., different possible values)
 - $\sqsubseteq \subseteq D \times D$ defined by pointwise extension of $\perp \sqsubseteq z \sqsubseteq \top$ (for every $z \in \mathbb{Z}$)

Example

$$\text{Var}_c = \{w, x, y, z\},$$

$$\delta_1 = (\underbrace{\perp}_w, \underbrace{1}_x, \underbrace{2}_y, \underbrace{\top}_z), \quad \delta_2 = (\underbrace{3}_w, \underbrace{1}_x, \underbrace{4}_y, \underbrace{\top}_z)$$

$$\Rightarrow \delta_1 \sqcup \delta_2 = (\underbrace{3}_w, \underbrace{1}_x, \underbrace{\top}_y, \underbrace{\top}_z)$$

Dataflow system $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$ (continued):

- extremal value $\iota := \delta_{\top} \in D$ where $\delta_{\top}(x) := \top$ for every $x \in \text{Var}_c$ (i.e., every x has (unknown) default value)
- transfer functions $\{\varphi_l \mid l \in L\}$ defined by

$$\varphi_l(\delta) := \begin{cases} \delta & \text{if } B^l = \text{skip or } B^l \in BExp \\ \delta[x \mapsto \text{val}_{\delta}(a)] & \text{if } B^l = (x := a) \end{cases}$$

where

$$\begin{aligned} \text{val}_{\delta}(x) &:= \delta(x) \\ \text{val}_{\delta}(z) &:= z \end{aligned} \quad \text{val}_{\delta}(a_1 \text{ op } a_2) := \begin{cases} z_1 \text{ op } z_2 & \text{if } z_1, z_2 \in \mathbb{Z} \\ \perp & \text{if } z_1 = \perp \text{ or } z_2 = \perp \\ \top & \text{otherwise} \end{cases}$$

for $z_1 := \text{val}_{\delta}(a_1)$ and $z_2 := \text{val}_{\delta}(a_2)$

- 1 Repetition: MOP Solution and Constant Propagation
- 2 MOP vs. Fixpoint Solution
- 3 Dataflow Analysis with Non-ACC Domains
- 4 Example: Interval Analysis

Example 7.1 (Constant Propagation)

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

Transfer functions

(for $\delta = (\delta(x), \delta(y), \delta(z)) \in D$):

$$\varphi_1((a, b, c)) = (a, b, c)$$

$$\varphi_2((a, b, c)) = (2, b, c)$$

$$\varphi_3((a, b, c)) = (a, 3, c)$$

$$\varphi_4((a, b, c)) = (3, b, c)$$

$$\varphi_5((a, b, c)) = (a, 2, c)$$

$$\varphi_6((a, b, c)) = (a, b, a + b)$$

1 Fixpoint solution:

$$CP_1 = \iota = (\top, \top, \top)$$

$$CP_2 = \varphi_1(CP_1) = (\top, \top, \top)$$

$$CP_3 = \varphi_2(CP_2) = (2, \top, \top)$$

$$CP_4 = \varphi_1(CP_1) = (\top, \top, \top)$$

$$CP_5 = \varphi_4(CP_4) = (3, \top, \top)$$

$$CP_6 = \varphi_3(CP_3) \sqcup \varphi_5(CP_5)$$

$$= (2, 3, \top) \sqcup (3, 2, \top) = (\top, \top, \top)$$

$$CP_7 = \varphi_6(CP_6) = (\top, \top, \top)$$

2 MOP solution:

$$\text{mop}(7) = \varphi_{[1,2,3,6]}(\top, \top, \top) \sqcup$$

$$\varphi_{[1,4,5,6]}(\top, \top, \top)$$

$$= (2, 3, 5) \sqcup (3, 2, 5)$$

$$= (\top, \top, 5)$$

Theorem 7.2 (MOP vs. Fixpoint Solution)

Let $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$ be a dataflow system. Then

$$\text{mop}(S) \sqsubseteq \text{fix}(\Phi_S)$$

Reminder: by Definition 4.9,

$$\Phi_S : D^n \rightarrow D^n : (d_1, \dots, d_n) \mapsto (d'_1, \dots, d'_n)$$

where $L = \{1, \dots, n\}$ and, for each $1 \leq l \leq n$,

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

Proof.

on the board □

Remark: as Example 7.1 shows, $\text{mop}(S) \neq \text{fix}(\Phi_S)$ is possible

A sufficient criterion for the coincidence of MOP and Fixpoint Solution is the distributivity of the transfer functions.

Definition 7.3 (Distributivity)

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

$$F(d_1 \sqcup_D d_2) = F(d_1) \sqcup_{D'} F(d_2).$$

- A dataflow system $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$ is called **distributive** if every $\varphi_I : D \rightarrow D$ ($I \in L$) is so.

Example 7.4

- ① The Available Expressions dataflow system is distributive: see Exercise 2.3
- ② The Live Variables dataflow system is distributive: see Exercise 2.3
- ③ The Constant Propagation dataflow system is not distributive:

$$\begin{aligned}(\top, \top, \top) &= \varphi_{z:=x+y}((2, 3, \top) \sqcup (3, 2, \top)) \\ &\neq \varphi_{z:=x+y}((2, 3, \top)) \sqcup \varphi_{z:=x+y}((3, 2, \top)) \\ &= (\top, \top, 5)\end{aligned}$$

Theorem 7.5 (MOP vs. Fixpoint Solution)

Let $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$ be a distributive dataflow system. Then

$$\text{mop}(S) = \text{fix}(\Phi_S)$$

Proof.

- $\Phi_S(\text{mop}(S)) = \text{mop}(S)$: on the board
- $\text{mop}(S) \sqsubseteq \text{fix}(\Phi_S)$: Theorem 7.2

$\Rightarrow$ claim



- 1 Repetition: MOP Solution and Constant Propagation
- 2 MOP vs. Fixpoint Solution
- 3 Dataflow Analysis with Non-ACC Domains**
- 4 Example: Interval Analysis

- **Reminder:** $(D, \sqsubseteq)$ satisfies **ACC** if each ascending chain $d_1 \sqsubseteq d_2 \sqsubseteq \dots$ eventually stabilizes, i.e., there exists $n \in \mathbb{N}$ such that $d_n = d_{n+1} = \dots$
- If **height** (= maximal chain length) of $(D, \sqsubseteq)$ is m , then fixpoint computation terminates after $\leq |L| \cdot m$ iterations
- **But:** if $(D, \sqsubseteq)$ has **infinite ascending chains**
 $\implies$ algorithm may not terminate
- **Solution:** use **widening operators** to enforce termination

Definition 7.6 (Widening operator)

Let $(D, \sqsubseteq)$ be a complete lattice. A mapping $\nabla : D \times D \rightarrow D$ is called **widening operator** if

- for every $d_1, d_2 \in D$,

$$d_1 \sqcup d_2 \sqsubseteq d_1 \nabla d_2$$

and

- for all ascending chains $d_0 \sqsubseteq d_1 \sqsubseteq \dots$, the ascending chain $d_0^\nabla \sqsubseteq d_1^\nabla \sqsubseteq \dots$ eventually stabilizes where
$$d_0^\nabla := d_0 \text{ and } d_{i+1}^\nabla := d_i^\nabla \nabla d_{i+1} \text{ for } i \in \mathbb{N}$$

Remarks:

- $(d_i^\nabla)_{i \in \mathbb{N}}$ is clearly an ascending chain as
$$d_{i+1}^\nabla = d_i^\nabla \nabla d_{i+1} \sqsupseteq d_i^\nabla \sqcup d_{i+1} \sqsupseteq d_i^\nabla$$
- In contrast to $\sqcup$, ∇ does not have to be commutative, associative, monotonic, nor absorptive ($d \nabla d = d$)
- The requirement $d_1 \sqcup d_2 \sqsubseteq d_1 \nabla d_2$ guarantees **soundness** of widening

- 1 Repetition: MOP Solution and Constant Propagation
- 2 MOP vs. Fixpoint Solution
- 3 Dataflow Analysis with Non-ACC Domains
- 4 Example: Interval Analysis

Example: Interval Analysis

Interval Analysis

The goal of **Interval Analysis** is to determine, for each (interesting) program point, a safe interval for the values of the (interesting) program variables.

Interval analysis is actually a generalization of constant propagation ($\approx$ interval analysis with 1-element intervals)

Example 7.7 (Interval Analysis)

```
var a[100]: int;
...
i := 0;
while i <= 42 do
  if i >= 0  $\wedge$  i < 100 then  $\leftarrow$ 
    a[i] := i;
  i := i + 1;
```

Here, ~~redundant array bounds check~~ can be removed

The Domain of Interval Analysis

- The domain $(Int, \subseteq)$ of **intervals over $\mathbb{Z}$** is defined by

$$Int := \{[z_1, z_2] \mid z_1 \in \mathbb{Z} \cup \{-\infty\}, z_2 \in \mathbb{Z} \cup \{+\infty\}, z_1 \leq z_2\} \cup \{\emptyset\}$$

where

- $-\infty \leq z, z \leq +\infty$, and $-\infty \leq +\infty$ (for all $z \in \mathbb{Z}$)
 - $\emptyset \subseteq I$ (for all $I \in Int$)
 - $[y_1, y_2] \subseteq [z_1, z_2]$ iff $z_1 \leq y_1$ and $y_2 \leq z_2$
- $(Int, \subseteq)$ is a **complete lattice** with (for every $\mathcal{I} \subseteq Int$)

$$\bigsqcup \mathcal{I} = \begin{cases} \emptyset & \text{if } \mathcal{I} = \emptyset \text{ or } \mathcal{I} = \{\emptyset\} \\ [Z_1, Z_2] & \text{otherwise} \end{cases}$$

where

$$Z_1 := \bigsqcap_{\mathbb{Z} \cup \{-\infty\}} \{z_1 \mid [z_1, z_2] \in \mathcal{I}\}$$
$$Z_2 := \bigsqcap_{\mathbb{Z} \cup \{+\infty\}} \{z_2 \mid [z_1, z_2] \in \mathcal{I}\}$$

(and thus $\perp = \emptyset$, $\top = [-\infty, +\infty]$)

- Clearly $(Int, \subseteq)$ has **infinite ascending chains**, such as

$$\emptyset \subseteq [1, 1] \subseteq [1, 2] \subseteq [1, 3] \subseteq \dots$$

The Complete Lattice of Interval Analysis

$[-\infty, +\infty]$

