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

Outline

The MOP Solution

Definition (MOP solution)

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

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

where, for every $l \in L$,

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

Remark:

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

$\Longrightarrow$ 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
  - $D := \{\delta \mid \delta : Var_c \to \mathbb{Z} \cup \{\bot, \top\}\}$
    - $\delta(x) = z \in \mathbb{Z}$: x has constant value z
    - $\delta(x) = \bot$: x undefined
    - $\delta(x) = \top$: x overdefined (i.e., different possible values)
  - $\sqsubseteq \subseteq D \times D$ defined by pointwise extension of $\bot \sqsubseteq z \sqsubseteq \top$ (for every $z \in \mathbb{Z}$)

Example

$Var_c = \{\texttt{w}, \texttt{x}, \texttt{y}, \texttt{z}\}$,

$\delta_1 = (\underbrace{\bot}_{\texttt{w}}, \underbrace{1}_{\texttt{x}}, \underbrace{2}_{\texttt{y}}, \underbrace{\top}_{\texttt{z}}),\ \delta_2 = (\underbrace{3}_{\texttt{w}}, \underbrace{1}_{\texttt{x}}, \underbrace{4}_{\texttt{y}}, \underbrace{\top}_{\texttt{z}})$

$\Longrightarrow\ \delta_1 \sqcup \delta_2 = (\underbrace{3}_{\texttt{w}}, \underbrace{1}_{\texttt{x}}, \underbrace{\top}_{\texttt{y}}, \underbrace{\top}_{\texttt{z}})$

Formalizing Constant Propagation Analysis II

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 $x \in Var_c$ (i.e., every x has (unknown) default value)
- transfer functions $\{\varphi_l \mid l \in L\}$ defined by

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

where

$$val_\delta(x) := \delta(x) \qquad val_\delta(z) := z$$

$$val_\delta(a_1\ op\ a_2) := \begin{cases} z_1\ op\ z_2 & \text{if } z_1, z_2 \in \mathbb{Z} \\ \bot & \text{if } z_1 = \bot \text{ or } z_2 = \bot \\ \top & \text{otherwise} \end{cases}$$

for $z_1 := val_\delta(a_1)$ and $z_2 := val_\delta(a_2)$

Outline

MOP vs. Fixpoint Solution I

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
```

MOP vs. Fixpoint Solution I

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}
$[\ldots]^7$

Transfer functions
(for $\delta = (\delta(\mathtt{x}), \delta(\mathtt{y}), \delta(\mathtt{z})) \in D$):

$\varphi_1((a, b, c)) = (a, b, c)$
$\varphi_2((a, b, c)) = (2, b, c)$
$\varphi_3((a, b, c)) = (a, 3, c)$
$\varphi_4((a, b, c)) = (3, b, c)$
$\varphi_5((a, b, c)) = (a, 2, c)$
$\varphi_6((a, b, c)) = (a, b, a + b)$

MOP vs. Fixpoint Solution I

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:

$$\begin{aligned}
\mathrm{CP}_1 &= \iota && = (\top,\top,\top)\\
\mathrm{CP}_2 &= \varphi_1(\mathrm{CP}_1) && = (\top,\top,\top)\\
\mathrm{CP}_3 &= \varphi_2(\mathrm{CP}_2) && = (2,\top,\top)\\
\mathrm{CP}_4 &= \varphi_1(\mathrm{CP}_1) && = (\top,\top,\top)\\
\mathrm{CP}_5 &= \varphi_4(\mathrm{CP}_4) && = (3,\top,\top)\\
\mathrm{CP}_6 &= \varphi_3(\mathrm{CP}_3) \sqcup \varphi_5(\mathrm{CP}_5) &&\\
&= (2,3,\top) \sqcup (3,2,\top) && = (\top,\top,\top)\\
\mathrm{CP}_7 &= \varphi_6(\mathrm{CP}_6) && = (\top,\top,\top)
\end{aligned}$$

MOP vs. Fixpoint Solution I

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
  $[\ldots]^7$

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

$$\begin{aligned}
\varphi_1((a,b,c)) &= (a,b,c)\\
\varphi_2((a,b,c)) &= (2,b,c)\\
\varphi_3((a,b,c)) &= (a,3,c)\\
\varphi_4((a,b,c)) &= (3,b,c)\\
\varphi_5((a,b,c)) &= (a,2,c)\\
\varphi_6((a,b,c)) &= (a,b,a+b)
\end{aligned}$$

1. Fixpoint solution:

$$\begin{aligned}
CP_1 &= \iota && = (\top,\top,\top)\\
CP_2 &= \varphi_1(CP_1) && = (\top,\top,\top)\\
CP_3 &= \varphi_2(CP_2) && = (2,\top,\top)\\
CP_4 &= \varphi_1(CP_1) && = (\top,\top,\top)\\
CP_5 &= \varphi_4(CP_4) && = (3,\top,\top)\\
CP_6 &= \varphi_3(CP_3) \sqcup \varphi_5(CP_5)\\
&= (2,3,\top) \sqcup (3,2,\top) && = (\top,\top,\top)\\
CP_7 &= \varphi_6(CP_6) && = (\top,\top,\top)
\end{aligned}$$

2. MOP solution:

$$\begin{aligned}
\mathrm{mop}(7) &= \varphi_{[1,2,3,6]}(\top,\top,\top) \sqcup\\
&\quad\ \varphi_{[1,4,5,6]}(\top,\top,\top)\\
&= (2,3,5) \sqcup (3,2,5)\\
&= (\top,\top,5)
\end{aligned}$$

MOP vs. Fixpoint Solution II

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,

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

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

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

MOP vs. Fixpoint Solution II

Theorem 7.2 (MOP vs. Fixpoint Solution)

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

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

Reminder: by Definition 4.9,

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

where $L = \{1, \ldots, n\}$ 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, $\mathrm{mop}(S) \neq \mathrm{fix}(\Phi_S)$ is possible

Distributive Transfer Functions I

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 $F : D \to D'$. F is called distributive (w.r.t. $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$) if, for every $d_1, d_2 \in D$,
$$F(d_1 \sqcup_D d_2) = F(d_1) \sqcup_{D'} F(d_2).$$

Distributive Transfer Functions I

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 $F : D \to D'$. 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 $\varphi_l : D \to D$ $(l \in L)$ is so.

Distributive Transfer Functions II

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:

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

Coincidence of MOP and Fixpoint Solution

Theorem 7.5 (MOP vs. Fixpoint Solution)

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

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

Theorem 7.5 (MOP vs. Fixpoint Solution)

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

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

Proof.

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

$\Rightarrow$ claim

□

Outline

Dataflow Analysis with Non-ACC Domains

- **Reminder:** $(D, \sqsubseteq)$ satisfies ACC if each ascending chain $d_1 \sqsubseteq d_2 \sqsubseteq \ldots$ eventually stabilizes, i.e., there exists $n \in \mathbb{N}$ such that $d_n = d_{n+1} = \ldots$
- If height (= maximal chain length) of $(D, \sqsubseteq)$ is m, then fixpoint computation terminates after $\leq |L| \cdot m$ iterations

Dataflow Analysis with Non-ACC Domains

- **Reminder:** $(D, \sqsubseteq)$ satisfies ACC if each ascending chain $d_1 \sqsubseteq d_2 \sqsubseteq \ldots$ eventually stabilizes, i.e., there exists $n \in \mathbb{N}$ such that $d_n = d_{n+1} = \ldots$
- 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
  $\Longrightarrow$ algorithm may not terminate

Dataflow Analysis with Non-ACC Domains

- **Reminder:** $(D, \sqsubseteq)$ satisfies ACC if each ascending chain $d_1 \sqsubseteq d_2 \sqsubseteq \ldots$ eventually stabilizes, i.e., there exists $n \in \mathbb{N}$ such that $d_n = d_{n+1} = \ldots$
- 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
  $\Longrightarrow$ algorithm may not terminate
- **Solution:** use widening operators to enforce termination

Widening Operators

Definition 7.6 (Widening operator)

Let $(D, \sqsubseteq)$ be a complete lattice. A mapping $\nabla : D \times D \to D$ 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 $d_0 \sqsubseteq d_1 \sqsubseteq \ldots$, the ascending chain $d_0^\nabla \sqsubseteq d_1^\nabla \sqsubseteq \ldots$ 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

Outline

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: 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 ∧ i < 100 then
    a[i] := i;
  i := i + 1;
```

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 ∧ i < 100 then     ⟸
    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 $y_2 \leq z_2$

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

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

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

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

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

  (and thus $\bot = \emptyset$, $\top = [-\infty, +\infty]$)
- Clearly $(Int, \subseteq)$ has infinite ascending chains, such as

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

The Complete Lattice of Interval Analysis