Static Program Analysis

Lecture 6: Dataflow Analysis V (MOP vs. Fixpoint Solution)

Winter Semester 2016/17

Thomas Noll
Software Modeling and Verification Group
RWTH Aachen University

https://moves.rwth-aachen.de/teaching/ws-1617/spa/

The MOP Solution I

- Other solution method for dataflow systems
- MOP = Meet Over all Paths
- Analysis information for block B^l
  = least upper bound over all paths leading to l
  = most precise information for l (“reference solution”)

Definition (Paths)

Let $S = (Lab, E, F, (D, \sqsubseteq), \iota, \varphi)$ be a dataflow system. For every $l \in Lab$, the set of paths up to l is given by

$$Path(l) := \{[l_1, \ldots, l_{k-1}] \mid k \geq 1, l_1 \in E, (l_i, l_{i+1}) \in F \text{ for every } 1 \leq i < k, l_k = l\}.$$

For a path $\pi = [l_1, \ldots, l_{k-1}] \in Path(l)$, we define the transfer function $\varphi_\pi : D \to D$ by

$$\varphi_\pi := \varphi_{l_{k-1}} \circ \ldots \circ \varphi_{l_1} \circ \mathrm{id}_D$$

(so that $\varphi_{[]} = \mathrm{id}_D$).

The MOP Solution II

Definition (MOP solution)

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

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

where, for every $l \in Lab$,

$$\mathrm{mop}(l) := \bigsqcup\{\varphi_\pi(\iota) \mid \pi \in Path(l)\}.$$

Remark:

- $Path(l)$ is generally infinite

$\Rightarrow$ not clear how to compute $\mathrm{mop}(l)$

- In fact: MOP solution generally undecidable (later)

Recap: Constant Propagation

Goal of Constant Propagation Analysis

Constant Propagation Analysis

The goal of Constant Propagation Analysis is to determine, for each program point, whether a variable has a constant value whenever execution reaches that point.

Used for Constant Folding: replace reference to variable by constant value and evaluate constant expressions

Example (Constant Propagation Analysis)

```
[x := 1]^1;[y := 1]^2;[z := 1]^3;
while [z > 0]^4 do
  [w := x+y]^5;
  if [w = 2]^6 then
    [x := y+2]^7
  end
end
```

- y = z = 1 at labels 4–7
- w, x not constant at labels 4–7
- possible optimisations:
  `[true]^4`
  `[w := x+1]^5`
  `[x := 3]^7`

Formalising Constant Propagation Analysis I

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

- set of labels $Lab := Lab_c$,
- 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 (i.e., possible values in $\{z\}$)
    - $\delta(x) = \bot$: x undefined (i.e., possible values in $\emptyset$)
    - $\delta(x) = \top$: x overdefined (i.e., possible values in $\mathbb{Z}$)
  - $\sqsubseteq \subseteq D \times D$ defined by pointwise extension of $\bot \sqsubseteq z \sqsubseteq \top$ (for every $z \in \mathbb{Z}$)

Example

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

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

Formalising Constant Propagation Analysis II

Dataflow system $S = (Lab, E, F, (D, \sqsubseteq), \iota, \varphi)$ (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 $\{\varphi_l \mid l \in Lab\}$ defined by

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

where

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

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

Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

Formalising Constant Propagation Analysis III

Example

If $\delta = (\underbrace{\bot}_{\texttt{w}}, \underbrace{1}_{\texttt{x}}, \underbrace{2}_{\texttt{y}}, \underbrace{\top}_{\texttt{z}})$, then

$$\varphi_l(\delta) = \begin{cases} (\underbrace{0}_{\texttt{w}}, \underbrace{1}_{\texttt{x}}, \underbrace{2}_{\texttt{y}}, \underbrace{\top}_{\texttt{z}}) & \text{if } B^l = (\texttt{w := 0}) \\ (\underbrace{3}_{\texttt{w}}, \underbrace{1}_{\texttt{x}}, \underbrace{2}_{\texttt{y}}, \underbrace{\top}_{\texttt{z}}) & \text{if } B^l = (\texttt{w := y+1}) \\ (\underbrace{\bot}_{\texttt{w}}, \underbrace{1}_{\texttt{x}}, \underbrace{2}_{\texttt{y}}, \underbrace{\top}_{\texttt{z}}) & \text{if } B^l = (\texttt{w := w+x}) \\ (\underbrace{\top}_{\texttt{w}}, \underbrace{1}_{\texttt{x}}, \underbrace{2}_{\texttt{y}}, \underbrace{\top}_{\texttt{z}}) & \text{if } B^l = (\texttt{w := z+2}) \end{cases}$$

An Example

Example 6.1

Constant Propagation Analysis for

```
c := [x := 1]^1; [y := 1]^2; [z := 1]^3;
     while [z > 0]^4 do
       [w := x+y]^5;
       if [w = 2]^6 then
         [x := y+2]^7
       end
     end
```

$$\varphi_1(a,b,c,d) = (a,1,c,d)$$
$$\varphi_2(a,b,c,d) = (a,b,1,d)$$
$$\varphi_3(a,b,c,d) = (a,b,c,1)$$
$$\varphi_4(a,b,c,d) = (a,b,c,d)$$
$$\varphi_5(a,b,c,d) = (b+c,b,c,d)$$
$$\varphi_6(a,b,c,d) = (a,b,c,d)$$
$$\varphi_7(a,b,c,d) = (a,c+2,c,d)$$

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

1. Fixpoint solution (on the board)
2. MOP solution (on the board)

MOP vs. Fixpoint Solution I

Example 6.2 (Constant Propagation)

$c :=$ `if [z > 0]`^{1} `then`
`[x := 2]`^{2}; `[y := 3]`^{3}
`else`
`[x := 3]`^{4}; `[y := 2]`^{5}
`end;`
`[z := x+y]`^{6}; $[\ldots]^7$

Transfer functions

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

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

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:

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

MOP vs. Fixpoint Solution II

Theorem 6.3 (MOP vs. Fixpoint Solution)

Let $S = (Lab, E, F, (D, \sqsubseteq), \iota, \varphi)$ *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 $Lab = \{1, \ldots, n\}$ and, for each $l \in Lab$,

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

Proof.

on the board □

Remark: as Example 6.2 shows, $\text{mop}(S) \neq \text{fix}(\Phi_S)$ is possible

Distributivity of Transfer Functions I

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

Definition 6.4 (Distributivity)

- Let $(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ be complete lattices. Function $F : D \to D'$ 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 $S = (Lab, E, F, (D, \sqsubseteq), \iota, \varphi)$ is called distributive if every $\varphi_l : D \to D$ $(l \in Lab)$ is so.

Distributivity of Transfer Functions II

Example 6.5

1. The Available Expressions dataflow system is distributive:

$$\begin{aligned}\varphi_l(A_1 \sqcup A_2) &= ((A_1 \cap A_2) \setminus \mathsf{kill}_{\mathsf{AE}}(B^l)) \cup \mathsf{gen}_{\mathsf{AE}}(B^l) \\ &= ((A_1 \setminus \mathsf{kill}_{\mathsf{AE}}(B^l)) \cup \mathsf{gen}_{\mathsf{AE}}(B^l)) \cap \\ & ((A_2 \setminus \mathsf{kill}_{\mathsf{AE}}(B^l)) \cup \mathsf{gen}_{\mathsf{AE}}(B^l)) \\ &= \varphi_l(A_1) \sqcup \varphi_l(A_2)\end{aligned}$$

2. The Live Variables dataflow system is distributive: similarly
3. The Constant Propagation dataflow system is not distributive (cf. Example 6.2):

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

Coincidence of MOP and Fixpoint Solution

Theorem 6.6 (MOP vs. Fixpoint Solution)

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

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

Proof.

- $\mathrm{mop}(S) \sqsubseteq \mathrm{fix}(\Phi_S)$: Theorem 6.3
- $\mathrm{fix}(\Phi_S) \sqsubseteq \mathrm{mop}(S)$: as $\mathrm{fix}(\Phi_S)$ is the *least* fixpoint of Φ_S, it suffices to show that $\Phi_S(\mathrm{mop}(S)) = \mathrm{mop}(S)$ (on the board)

□

Undecidability of the MOP Solution

Theorem 6.7 (Undecidability of MOP solution)

The MOP solution for Constant Propagation is undecidable.

Proof.

Based on undecidability of Modified Post Correspondence Problem:
Let Γ be some alphabet, $n \in \mathbb{N}$, and $u_1, \ldots, u_n, v_1, \ldots, v_n \in \Gamma^+$.
Do there exist $i_1, \ldots, i_m \in \{1, \ldots, n\}$ with $m \geq 1$ and $i_1 = 1$ such that $u_{i_1} u_{i_2} \ldots u_{i_m} = v_{i_1} v_{i_2} \ldots v_{i_m}$?

(on the board) □