Static Program Analysis

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

Summer Semester 2018

Thomas Noll
Software Modeling and Verification Group
RWTH Aachen University

https://moves.rwth-aachen.de/teaching/ss-18/spa/

Software Modeling and Verification Chair

RWTH AACHEN UNIVERSITY

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

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

(and thus $\varphi_{[]} = \mathrm{id}_D$).

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)

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

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

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

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

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

where

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

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

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

MOP vs. Fixpoint Solution I

Example 6.1 (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; [...]^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:

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

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 6.2 (MOP vs. Fixpoint Solution)

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

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

Reminder: by Definition 4.3,

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

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 $\square$

Remark: as Example 6.1 shows, $\mathsf{mop}(S) \neq \mathsf{fix}(\Phi_S)$ 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 6.3 (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 6.4

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 6.1):

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

Coincidence of MOP and Fixpoint Solution

Theorem 6.5 (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 6.2
- $\mathrm{fix}(\Phi_S) \sqsubseteq \mathrm{mop}(S)$: 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)

$\square$

Undecidability of the MOP Solution I

Theorem 6.6 (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 $u_{i_1} u_{i_2} \ldots u_{i_m} = v_{i_1} v_{i_2} \ldots v_{i_m}$?

Given a MPCP, we construct a WHILE program (with strings and Booleans) whose MOP analysis detects a constant property iff the MPCP has no solution (see next slide). $\square$

Undecidability of the MOP Solution II

Proof (continued).

```
x := u1;y := v1;
while ... do
  if ... then
    x := x ++ u1;
    y := y ++ v1
  else if ... then
    ⋮
  else
    x := x ++ un;
    y := y ++ vn
  end ... end
end;
z := (x = y);
[skip]^l
```

Then: $\mathrm{mop}(l)(\mathtt{z}) = \mathrm{false}$

$\iff$ $\mathtt{x} \neq \mathtt{y}$ at the end of every path to l

$\iff$ the MPCP has no solution

□

Dataflow Analysis with Non-ACC Domains

- **Reminder:** $(D, \sqsubseteq)$ satisfies ACC if each ascending chain $d_0 \sqsubseteq d_1 \sqsubseteq \ldots$ eventually stabilises, i.e., there exists $n \in \mathbb{N}$ such that $d_n = d_{n+1} = \ldots$
- If height (= maximal chain size) of $(D, \sqsubseteq)$ is m, then fixpoint computation terminates after at most $|Lab| \cdot m$ iterations
- **But:** if $(D, \sqsubseteq)$ has non-stabilising ascending chains
  $\implies$ algorithm may not terminate
- **Solution:** use widening operators to enforce termination

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 generalisation of constant propagation
($\approx$ interval analysis with one-element intervals)

Example 6.7 (Interval Analysis)

```
var a[100]:  int;
i := 0;
while i <= 42 do
  if i >= 0 ∧ i < 100 then      ⟸ redundant array bounds check
    a[i] := i
  end;
  i := i + 1;
end;
```

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
  - $-\infty \leq z \leq +\infty$ (for all $z \in \mathbb{Z}$)
  - $\emptyset \subseteq J$ (for all $J \in Int$)
  - $[y_1, y_2] \subseteq [z_1, z_2]$ 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

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

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

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

Dataflow Analysis with Non-ACC Domains

The Complete Lattice of Interval Analysis