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/
Recap: The MOP Solution

Outline of Lecture 6

Recap: The MOP Solution

Recap: Constant Propagation

Example of Constant Propagation Analysis

MOP vs. Fixpoint Solution

Coincidence of MOP and Fixpoint Solution

Undecidability of the MOP Solution
Recap: The MOP Solution

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

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

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

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

(so that $\varphi[] = \text{id}_D$).
Recap: The MOP Solution

The MOP Solution II

Definition (MOP solution)

Let $S = (\mathit{Lab}, E, F, (D, \sqsubseteq), \iota, \varphi)$ be a dataflow system where $\mathit{Lab} = \{l_1, \ldots, l_n\}$. 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 $l \in \mathit{Lab}$,

$$\text{mop}(l) := \bigcup \{\varphi_\pi(l) \mid \pi \in \text{Path}(l)\}.$$  

Remark:

- $\text{Path}(l)$ is generally infinite
- $\Rightarrow$ not clear how to compute $\text{mop}(l)$
- In fact: MOP solution generally undecidable (later)
Recap: Constant Propagation

Outline of Lecture 6

Recap: The MOP Solution

Recap: Constant Propagation

Example of Constant Propagation Analysis

MOP vs. Fixpoint Solution

Coincidence of MOP and Fixpoint Solution

Undecidability of the MOP Solution
Recap: Constant Propagation

Goal of 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.

Example (Constant Propagation Analysis)

\[
\begin{align*}
x &:= 1 \\
y &:= 2 \\
z &:= 3 \\
\text{while}(z > 0) &\text{ do} \\
\quad w &:= x + y \\
\quad \text{if}(w = 2) &\text{ then} \\
\text{end} \\
\text{end}
\end{align*}
\]

- \( y = z = 1 \) at labels 4–7
- \( w, x \) not constant at labels 4–7
- possible optimisations:
  \[
  \begin{align*}
  \text{true} &\text{ at 4} \\
  w &:= x + 1 \\
  x &:= 3
  \end{align*}
  \]
Recap: Constant Propagation

Goal of 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
Recap: Constant Propagation

Goal of 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)

\[ x := 1; y := 1; z := 1; \]

while \( z > 0 \) do
\[ w := x+y; \]
if \( w = 2 \) then
\[ x := y+2 \]
end
end
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)

\[
\begin{align*}
      x &:= 1^1; \ y &:= 1^2; \ z &:= 1^3; \\
    \text{while} \ [z > 0]^4 \ \text{do} \\
    \quad [w &:= x+y]^5; \\
    \quad \text{if} \ [w = 2]^6 \ \text{then} \\
    \quad \quad [x &:= y+2]^7 \\
    \text{end} \\
    \text{end}
\end{align*}
\]

\bullet y = z = 1 at labels 4–7
Recap: Constant Propagation

Goal of 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)**

\[
\begin{align*}
  x &:= 1^1; y := 1^2; z := 1^3; \\
  \text{while } [z > 0]^4 \text{ do} \\
  &\quad w := x+y^5; \\
  &\quad \text{if } [w = 2]^6 \text{ then} \\
  &\quad \quad x := y+2^7 \\
  &\quad \text{end} \\
  &\text{end}
\end{align*}
\]

- \( y = z = 1 \) at labels 4–7
- \( w, x \) not constant at labels 4–7
Recap: Constant Propagation

Goal of 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)

\[
\begin{align*}
\text{x := 1}^1 ; \text{y := 1}^2 ; \text{z := 1}^3 ; \\
\text{while } [\text{z > 0}]^4 \text{ do} \\
\quad \text{w := x+y}^5 ; \\
\quad \text{if } [\text{w = 2}]^6 \text{ then} \\
\quad \quad \text{x := y+2}^7 \\
\quad \text{end} \\
\text{end}
\end{align*}
\]

- \( y = z = 1 \) at labels 4–7
- \( w, x \) not constant at labels 4–7
- possible optimisations:
  - \([\text{true}]^4\)
  - \([\text{w := x+1}]^5\)
  - \([\text{x := 3}]^7\)
Recap: Constant Propagation

**Formalising Constant Propagation Analysis I**

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

- set of labels \( \text{Lab} := \text{Lab}_c \),
- extremal labels \( E := \{ \text{init}(c) \} \) (forward problem)
- flow relation \( F := \text{flow}(c) \) (forward problem)
- complete lattice \( (D, \sqsubseteq) \) where
  - \( D := \{ \delta \mid \delta : \text{Var}_c \to \mathbb{Z} \cup \{ \bot, \top \} \} \)
    - \( \delta(x) = z \in \mathbb{Z} : x \) 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 \( \bot \subseteq z \subseteq \top \) (for every \( z \in \mathbb{Z} \))

**Example**

\( \text{Var}_c = \{ w, x, y, z \} \), \( \delta_1 = (\bot, 1, 2, \top) \), \( \delta_2 = (3, 1, 4, \top) \)

\[ \Rightarrow \delta_1 \sqcup \delta_2 = (3, 1, \top, \top) \]
Recap: Constant Propagation

Formalising Constant Propagation Analysis II

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

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

where

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

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

Formalising Constant Propagation Analysis III

Example

If $\delta = (\bot, 1, 2, \top)$, then

$$\varphi_l(\delta) = \begin{cases} 
(0, 1, 2, \top) & \text{if } B^l = (w := 0) \\
(3, 1, 2, \top) & \text{if } B^l = (w := y+1) \\
(\bot, 1, 2, \top) & \text{if } B^l = (w := w+x) \\
(\top, 1, 2, \top) & \text{if } B^l = (w := z+2) 
\end{cases}$$
Example of Constant Propagation Analysis

Outline of Lecture 6

Recap: The MOP Solution

Recap: Constant Propagation

Example of Constant Propagation Analysis

MOP vs. Fixpoint Solution

Coincidence of MOP and Fixpoint Solution

Undecidability of the MOP Solution
Example of Constant Propagation Analysis

An Example

Example 6.1

Constant Propagation Analysis for

\[
\begin{align*}
c &:= [x := 1];[y := 1];[z := 1]; \\
\text{while } [z > 0] &\text{ do} \\
&w := x+y; \\
\text{if } [w = 2] &\text{ then} \\
&[x := y+2] \\
\text{end} \\
\text{end}
\end{align*}
\]

\[
\begin{align*}
\phi_1(a, b, c, d) &= (a, 1, c, d) \\
\phi_2(a, b, c, d) &= (a, b, 1, d) \\
\phi_3(a, b, c, d) &= (a, b, c, 1) \\
\phi_4(a, b, c, d) &= (a, b, c, d) \\
\phi_5(a, b, c, d) &= (b + c, b, c, d) \\
\phi_6(a, b, c, d) &= (a, b, c, d) \\
\phi_7(a, b, c, d) &= (a, c + 2, c, d)
\end{align*}
\]

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

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

Outline of Lecture 6

Recap: The MOP Solution

Recap: Constant Propagation

Example of Constant Propagation Analysis

MOP vs. Fixpoint Solution

Coincidence of MOP and Fixpoint Solution

Undecidability of the MOP Solution
Example 6.2 (Constant Propagation)

\[
c := \text{if } [z > 0] \text{ then } [x := 2] ; [y := 3] \text{ else } [x := 3] ; [y := 2] \text{ end; } 
[z := x+y] ; [...]
\]
Example 6.2 (Constant Propagation)

\[ c := \text{if } [z > 0] \text{ then } \\
[ x := 2 ]; [ y := 3 ] \text{ else } \\
[ x := 3 ]; [ y := 2 ] \text{ end; } \\
[ z := x+y ] \] \[ \ldots \]

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

\[ \varphi_1(a, b, c) = (a, b, c) \]
\[ \varphi_2(a, b, c) = (2, b, c) \]
\[ \varphi_3(a, b, c) = (a, 3, c) \]
\[ \varphi_4(a, b, c) = (3, b, c) \]
\[ \varphi_5(a, b, c) = (a, 2, c) \]
\[ \varphi_6(a, b, c) = (a, b, a + b) \]
Example 6.2 (Constant Propagation)


c := if \([z > 0]\)^1 then
\[
    \begin{align*}
    [x := 2]^2; [y := 3]^3 \\
    \text{else} \\
    [x := 3]^4; [y := 2]^5
    \end{align*}
\]
end;

\([z := x+y]^6; [...]^7\)

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

\[
\begin{align*}
\varphi_1(a, b, c) &= (a, b, c) \\
\varphi_2(a, b, c) &= (2, b, c) \\
\varphi_3(a, b, c) &= (a, 3, c) \\
\varphi_4(a, b, c) &= (3, b, c) \\
\varphi_5(a, b, c) &= (a, 2, c) \\
\varphi_6(a, b, c) &= (a, b, a + b)
\end{align*}
\]

1. Fixpoint solution:

\[
\begin{align*}
CP_1 &= \iota &= (T, T, T) \\
CP_2 &= \varphi_1(CP_1) &= (T, T, T) \\
CP_3 &= \varphi_2(CP_2) &= (2, T, T) \\
CP_4 &= \varphi_1(CP_1) &= (T, T, T) \\
CP_5 &= \varphi_4(CP_4) &= (3, T, T) \\
CP_6 &= \varphi_3(CP_3) \sqcup \varphi_5(CP_5) &= (2, 3, T) \sqcup (3, 2, T) &= (T, T, T) \\
CP_7 &= \varphi_6(CP_6) &= (T, T, T)
\end{align*}
\]
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]\(^6\); [...]\(^7\)

1. Fixpoint solution:

   \begin{align*}
   CP_1 &= \eta = (T, T, T) \\
   CP_2 &= \varphi_1(CP_1) = (T, T, T) \\
   CP_3 &= \varphi_2(CP_2) = (T, T, T) \\
   CP_4 &= \varphi_1(CP_1) = (T, T, T) \\
   CP_5 &= \varphi_4(CP_4) = (T, T, T) \\
   CP_6 &= \varphi_3(CP_3) \sqcup \varphi_5(CP_5) = (T, T, T) \\
   CP_7 &= \varphi_6(CP_6) = (T, T, T)
   \end{align*}

2. MOP solution:

   \begin{align*}
   \text{mop}(7) &= \varphi_{[1,2,3,6]}(T, T, T) \sqcup \varphi_{[1,4,5,6]}(T, T, T) \\
                 &= (2, 3, 5) \sqcup (3, 2, 5) \\
                 &= (T, T, 5)
   \end{align*}
Theorem 6.3 (MOP vs. Fixpoint Solution)

Let $S = (\text{Lab}, E, F, (D, ⊑), ι, ϕ)$ be a dataflow system. Then

$mop(S) \sqsubseteq fix(Φ_S)$

Reminder: by Definition 4.9,

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

where $\text{Lab} = \{1, \ldots, n\}$ and, for each $l \in \text{Lab},$

$d'_l := \begin{cases} ι & \text{if } l \in E \\ \bigsqcup \{ϕ_{l'}(d_{l'}) \mid (l', l) \in F\} & \text{otherwise} \end{cases}$
Theorem 6.3 (MOP vs. Fixpoint Solution)

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

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

**Proof:**

on the board

**Remark:** as Example 6.2 shows, $\text{mop}(S) \neq \text{fix}(\Phi_S)$ is possible
Coincidence of MOP and Fixpoint Solution

Outline of Lecture 6

Recap: The MOP Solution

Recap: Constant Propagation

Example of Constant Propagation Analysis

MOP vs. Fixpoint Solution

Coincidence of MOP and Fixpoint Solution

Undecidability of the MOP Solution
Coincidence of MOP and Fixpoint Solution

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 \(d_1, d_2 \in D\),

\[
F(d_1 \sqcup_D d_2) = F(d_1) \sqcup_{D'} F(d_2).
\]
Coincidence of MOP and Fixpoint Solution

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 \(F : D \rightarrow D'\) 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 \(S = (\text{Lab}, E, F, (D, \sqsubseteq), \iota, \varphi)\) is called distributive if every \(\varphi_I : D \rightarrow D\) (\(I \in \text{Lab}\)) is so.
Coincidence of MOP and Fixpoint Solution

Distributivity of Transfer Functions II

Example 6.5

1. The Available Expressions dataflow system is distributive:

\[ \varphi_l(A_1 \sqcup A_2) = ((A_1 \cap A_2) \setminus \text{kill}_{AE}(B')) \cup \text{gen}_{AE}(B') \]
\[ = ((A_1 \setminus \text{kill}_{AE}(B')) \cup \text{gen}_{AE}(B')) \cap \]
\[ ((A_2 \setminus \text{kill}_{AE}(B')) \cup \text{gen}_{AE}(B')) \]
\[ = \varphi_l(A_1) \sqcup \varphi_l(A_2) \]
Coincidence of MOP and Fixpoint Solution

Distributivity of Transfer Functions II

Example 6.5

1. The Available Expressions dataflow system is distributive:
   \[ \varphi_l(A_1 \sqcup A_2) = ((A_1 \cap A_2) \setminus \text{kill}_{AE}(B')) \cup \text{gen}_{AE}(B') \]
   \[ = ((A_1 \setminus \text{kill}_{AE}(B')) \cup \text{gen}_{AE}(B')) \cap ((A_2 \setminus \text{kill}_{AE}(B')) \cup \text{gen}_{AE}(B')) \]
   \[ = \varphi_l(A_1) \sqcup \varphi_l(A_2) \]

2. The Live Variables dataflow system is distributive: similarly
Coincidence of MOP and Fixpoint Solution

Distributivity of Transfer Functions II

Example 6.5

1. The Available Expressions dataflow system is distributive:

\[ \varphi_I(A_1 \sqcup A_2) = ((A_1 \cap A_2) \setminus \text{kill}_{AE}(B')) \cup \text{gen}_{AE}(B') \]
\[ = ((A_1 \setminus \text{kill}_{AE}(B')) \cup \text{gen}_{AE}(B')) \cap \\
((A_2 \setminus \text{kill}_{AE}(B')) \cup \text{gen}_{AE}(B')) \]
\[ = \varphi_I(A_1) \sqcup \varphi_I(A_2) \]

2. The Live Variables dataflow system is distributive: similarly

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

\[ (T, T, T) = \varphi_{z:=x+y}((2, 3, T) \sqcup (3, 2, T)) \]
\[ \neq \varphi_{z:=x+y}(2, 3, T) \sqcup \varphi_{z:=x+y}(3, 2, T) \]
\[ = (T, T, 5) \]
Theorem 6.6 (MOP vs. Fixpoint Solution)

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

$$\text{mop}(S) = \text{fix}(\Phi_S)$$
Theorem 6.6 (MOP vs. Fixpoint Solution)

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

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

Proof.

- $\text{mop}(S) \subseteq \text{fix}(\Phi_S)$: Theorem 6.3
- $\text{fix}(\Phi_S) \subseteq \text{mop}(S)$: as $\text{fix}(\Phi_S)$ is the least fixpoint of $\Phi_S$, it suffices to show that $\Phi_S(\text{mop}(S)) = \text{mop}(S)$ (on the board)
Undecidability of the MOP Solution

Outline of Lecture 6

Recap: The MOP Solution
Recap: Constant Propagation
Example of Constant Propagation Analysis
MOP vs. Fixpoint Solution
Coincidence of MOP and Fixpoint Solution
Undecidability of the MOP Solution
Theorem 6.7 (Undecidability of MOP solution)

The MOP solution for Constant Propagation is undecidable.
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 \( \Gamma \) 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)