Semantics and Verification of Software

Lecture 19: Dataflow Analysis VI (The MOP 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/svsw08/

Winter semester 2008/09

Online Registration for Seminars and Practical Courses (Praktika) in Summer Term 2009

Who?

Students of:
- Hauptstudium Informatik
- Master Courses
- Bachelor Informatik (~~Pro~~Seminar!)

Where?

www.graphics.rwth-aachen.de/apse

When?

05.01.2009 - 18.01.2009

Seminar: Applying Formal Verification Methods to Embedded Systems

- Joint weekly Seminar with Embedded Software Laboratory
- Theoretical and Practical CS
- Topics:
  - Static program analysis
  - Abstract interpretation
  - Software model checking (of assembly and source code)
  - Analysis of timed behavior
  - Resource awareness
- Requirements:
  - Vordiplom/Bachelor
  - In particular: Automata Theory and Formal Languages
  - Helpful: basic knowledge in
    - this course
    - (Formal Methods for) Embedded Systems
    - Model Checking Technology

Outline

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

Definition 19.1 (Paths)

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

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

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

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

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

Definition 19.2 (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

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

where, for every $l \in L$,

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

Remark:

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

$\Longrightarrow$ not clear how to compute $\mathsf{mop}(l)$

- In fact: MOP solution generally undecidable (later)

The MOP Solution III

Example 19.3 (Live Variables; cf. Examples 15.3 and 17.4)

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

$\Longrightarrow \mathit{Path}(1) = \{[7,5,4,3,2], [7,6,4,3,2]\}$

$$\begin{aligned}
\Longrightarrow \mathsf{mop}(1) &= \varphi_{[7,5,4,3,2]}(\iota) \sqcup \varphi_{[7,6,4,3,2]}(\iota) \\
&= \varphi_2(\varphi_3(\varphi_4(\varphi_5(\varphi_7(\{\mathtt{x},\mathtt{y},\mathtt{z}\}))))) \sqcup \\
&\quad\ \varphi_2(\varphi_3(\varphi_4(\varphi_6(\varphi_7(\{\mathtt{x},\mathtt{y},\mathtt{z}\}))))) \\
&= \varphi_2(\varphi_3(\varphi_4(\varphi_5(\{\mathtt{y},\mathtt{z}\})))) \sqcup \\
&\quad\ \varphi_2(\varphi_3(\varphi_4(\varphi_6(\{\mathtt{y},\mathtt{z}\})))) \\
&= \varphi_2(\varphi_3(\varphi_4(\{\mathtt{x},\mathtt{y}\}))) \sqcup \\
&\quad\ \varphi_2(\varphi_3(\varphi_4(\{\mathtt{y}\}))) \\
&= \varphi_2(\varphi_3(\{\mathtt{x},\mathtt{y}\})) \sqcup \varphi_2(\varphi_3(\{\mathtt{y}\})) \\
&= \varphi_2(\{\mathtt{y}\}) \sqcup \varphi_2(\{\mathtt{y}\}) \\
&= \emptyset \sqcup \emptyset \\
&= \emptyset
\end{aligned}$$

Outline

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

Example 19.4 (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
```

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

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 $E := \{\mathsf{init}(c)\}$ (forward problem),
- flow relation $F := \mathsf{flow}(c)$ (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 19.5

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

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

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

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 $x \in Var_c$,
- transfer functions $\{\varphi_l \mid l \in L\}$ defined by

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

where

$$\begin{aligned} \mathfrak{A}[\![x]\!]\delta &:= \delta(x) \\ \mathfrak{A}[\![z]\!]\delta &:= z \end{aligned} \qquad \mathfrak{A}[\![a_1 \ op \ a_2]\!]\delta := \begin{cases} z_1 \ op \ z_2 & \text{if } z_1, z_2 \in \mathbb{Z} \\ \bot & \text{if } z_1 = \bot \text{ or } z_2 = \bot \\ \top & \text{otherwise} \end{cases}$$

if $z_1 := \mathfrak{A}[\![a_1]\!]\delta$ and $z_2 := \mathfrak{A}[\![a_2]\!]\delta$

Example 19.6

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

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

Example 19.7

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
```

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

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