Semantics and Verification of Software

Lecture 23: Dataflow Analysis VI (MOP vs. Fixpoint Solution & Wrap-Up)

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/svsw10/

Summer Semester 2010

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 (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

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

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

The MOP Solution II

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

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

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

Undecidability of the MOP Solution

Theorem (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 $u_1, \ldots, u_n, v_1, \ldots, v_n \in \Gamma^+$.
Does 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) □

Outline

1. Repetition: The MOP Solution
2. Repetition: Constant Propagation
3. MOP vs. Fixpoint Solution
4. Further Topics in Formal Semantics
5. Upcoming Courses
6. Evaluation of the Course

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

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

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

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

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

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

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

Outline

MOP vs. Fixpoint Solution I

Theorem 23.1 (MOP vs. Fixpoint Solution)

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

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

Reminder: by Definition 21.2,

$$\Phi_S : D^n \to D^n : (d_{l_1}, \ldots, d_{l_n}) \mapsto (d'_{l_1}, \ldots, d'_{l_n})$$

where $L = \{l_1, \ldots, l_n\}$ and, for each $1 \leq i \leq n$,

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

Proof.

on the board □

The next example shows that both solutions can indeed be different.

MOP vs. Fixpoint Solution II

Example 23.2 (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 $\delta = (\delta(\mathtt{x}), \delta(\mathtt{y}), \delta(\mathtt{z})) \in D$):

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

1. Fixpoint solution:

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

2. MOP solution:

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

Distributive Transfer Functions I

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

Definition 23.3 (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 23.4

1. The Available Expressions dataflow system is distributive:

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

2. The Live Variables dataflow system is distributive (similar)
3. The Constant Propagation dataflow system is not distributive:

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

Coincidence of MOP and Fixpoint Solution

Theorem 23.5 (MOP vs. Fixpoint Solution)

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

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

Proof.

- by showing that $\Phi_S(\mathsf{mop}(S)) = \mathsf{mop}(S)$ ...
  (see [Nielson/Nielson/Hankin 2005, p. 81])
- ... and using $\mathsf{mop}(S) \sqsubseteq \mathsf{fix}(\Phi_S)$ (Theorem 23.1)

□

Outline

Semantics of Functional Languages I

- Program = list of function definitions
- Simplest setting: first-order function definitions of the form
$$f(x_1, \ldots, x_n) = t$$
  - function name f
  - formal parameters $x_1, \ldots, x_n$
  - term t over (base and defined) function calls and $x_1, \ldots, x_n$
- Operational semantics (only function calls)
  - call-by-value case:
$$\frac{t_1 \to z_1 \quad \ldots \quad t_n \to z_n \quad t[x_1 \mapsto z_1, \ldots, x_n \mapsto z_n] \to z}{f(t_1, \ldots, t_n) \to z}$$
  - call-by-name case:
$$\frac{t[x_1 \mapsto t_1, \ldots, x_n \mapsto t_n] \to z}{f(t_1, \ldots, t_n) \to z}$$

Semantics of Functional Languages II

- Denotational semantics
  - program = equation system (for functions)
  - induces call-by-value and call-by-name functional
  - monotonic and continuous w.r.t. graph inclusion
  - semantics := least fixpoint (Tarski/Knaster Theorem)
  - coincides with operational semantics
- Extensions: higher-order types, data types, ...
- see [Winskel 1996, Sct. 9] and *Functional Programming* course [Giesl]

Semantics of Concurrent Languages

- **Problem:** "classical" view of sequential systems

  $$\text{Program} : \text{Input} \to \text{Output}$$

  not adequate for concurrent settings
- Missing: aspect of interaction
- Typical approach:
  - concurrency modelled by interleaving
  - interaction modelled by (explicit) communication
- Example: Milner's *Calculus of Communicating Systems (CCS)*
- Syntax: $P ::= 0 \mid \alpha.P \mid P_1 + P_2 \mid P_1 \parallel P_2 \mid \ldots$
- (Operational) Semantics: labelled transition systems defined by transition rules of the form

$$\frac{}{\alpha.P \xrightarrow{\alpha} P} \quad \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \quad \frac{P \xrightarrow{\alpha} P'}{P \parallel Q \xrightarrow{\alpha} P' \parallel Q} \quad \frac{P \xrightarrow{\alpha} P' \quad Q \xrightarrow{\bar{\alpha}} Q'}{P \parallel Q \xrightarrow{\tau} P' \parallel Q'} \quad \ldots$$

- see course on *Modelling Concurrent and Probabilistic Systems* in Summer 2009 [Katoen, Noll] and [Winskel 1996, Sct. 14]

Outline

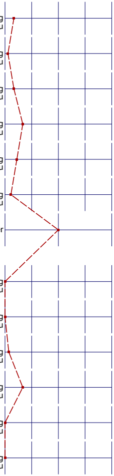
Courses Winter 2010/11

- Course Advanced Model Checking [Katoen]
- Practical Course Model Checking [Katoen, Sher, Yue]
- Course Compiler Construction [Noll] (“Hiwi” jobs available!)

Outline

Profillinie

Teilbereich:	Informatik
Name der/des	Priv.doz. Dr.rer.nat. Thomas Noll
Titel der Lehrveranstaltung: (Name der Umfrage)	Semantik und Verifikation von Software (10ss-29423) (Vorlesung)



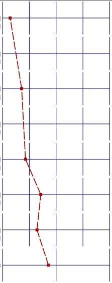
Konzept der Vorlesung			
Mir ist klar, wozu die Vorlesung gut ist.	trifft völlig zu	trifft nicht zu	n=9 mw=1.3
Die Vorlesung hat eine klar erkennbare Struktur.	trifft völlig zu	trifft nicht zu	n=9 mw=1.1
Die Vorlesung kann mit den zur Verfügung gestellten Materialien (Skript, Lehrbuch, Handouts ...) gut nachbereitet werden.	trifft völlig zu	trifft nicht zu	n=9 mw=1.3
Ich habe das nötige Vorwissen für diese Vorlesung.	trifft völlig zu	trifft nicht zu	n=9 mw=1.7
Die ausgewählten Beispiele helfen mir, die Inhalte der Vorlesung zu verstehen.	trifft völlig zu	trifft nicht zu	n=9 mw=1.4
Es werden Zusammenfassungen an sinnvollen Stellen gemacht.	trifft völlig zu	trifft nicht zu	n=9 mw=1.2
Der Schwierigkeitsgrad ist ...	zu schwer	zu leicht	n=9 mw=2
Vermittlung und Verhalten			
... kann den Stoff verständlich erklären.	trifft völlig zu	trifft nicht zu	n=9 mw=1
... geht sorgfältig auf Verständnisfragen ein.	trifft völlig zu	trifft nicht zu	n=9 mw=1
... berücksichtigt unterschiedliche Kenntnisstände der Studierenden.	trifft völlig zu	trifft nicht zu	n=7 mw=1.1
... schafft es, mich für den Vorlesungsstoff zu begeistern.	trifft völlig zu	trifft nicht zu	n=9 mw=1.7
... spricht angemessen laut und deutlich.	trifft völlig zu	trifft nicht zu	n=9 mw=1
... ist offen für Verbesserungsvorschläge.	trifft völlig zu	trifft nicht zu	n=5 mw=1

... lässt sich außerhalb der Vorlesung gut ansprechen, z.B. in Sprechstunden oder per Email.	trifft völlig zu	trifft nicht zu	n=6	mw=1
Der Einsatz von Hilfsmitteln wie Wandtafel, Overhead, Beamer und Demonstrationen ist gut.	trifft völlig zu	trifft nicht zu	n=9	mw=1.2
Schrift und Zeichnungen in der Vorlesung sind gut lesbar.	trifft völlig zu	trifft nicht zu	n=9	mw=1.2
Tafelanschrieb / Folien sind übersichtlich.	trifft völlig zu	trifft nicht zu	n=9	mw=1.7
Das Tempo ist ...	zu hoch	zu niedrig	n=9	mw=2.2

Profillinie

Teilbereich:	Informatik
Name der/des	Priv.doz. Dr.rer.nat. Thomas Noll
Titel der Lehrveranstaltung: (Name der Umfrage)	Semantik und Verifikation von Software (10ss-29422) (Übung)

Konzept der Übung				
Vorlesung und Übung sind **inhaltlich** gut aufeinander abgestimmt.	trifft völlig zu	trifft gar nicht zu	n=7	mw=2
Vorlesung und Übung sind **zeitlich** gut aufeinander abgestimmt.	trifft völlig zu	trifft gar nicht zu	n=7	mw=1.7
Mir ist klar, wozu die Übung gut ist.	trifft völlig zu	trifft gar nicht zu	n=6	mw=1.7
Der Ablauf der Übung ist gut strukturiert.	trifft völlig zu	trifft gar nicht zu	n=7	mw=1.9
Die ausgewählten Übungsaufgaben helfen mir, die Inhalte der Vorlesung zu verstehen.	trifft völlig zu	trifft gar nicht zu	n=5	mw=2
Die Übungsaufgaben sind verständlich gestellt.	trifft völlig zu	trifft gar nicht zu	n=6	mw=2.5
Die Übungsaufgaben haben einen angemessenen Umfang.	trifft völlig zu	trifft gar nicht zu	n=6	mw=2.2
Die vorgestellten Lösungswege sind nachvollziehbar.	trifft völlig zu	trifft gar nicht zu	n=7	mw=1.7
Falls Sie Ihre Lösung abgeben konnten: Wurde diese angemessen korrigiert?	trifft völlig zu	trifft gar nicht zu	n=5	mw=1
Die Übungsaufgaben sind ...	zu schwer	zu leicht	n=5	mw=1.6
Vermittlung und Verhalten				
... kann den Stoff verständlich erklären.	trifft völlig zu	trifft gar nicht zu	n=7	mw=1.7
... geht sorgfältig auf Verständnisfragen ein.	trifft völlig zu	trifft gar nicht zu	n=7	mw=1.6
... berücksichtigt unterschiedliche Kenntnisstände der Studierenden. (*)	trifft völlig zu	trifft gar nicht zu		



... spricht angemessen laut und deutlich.	trifft völlig zu	trifft gar nicht zu	n=7 mw=1.3
... ist offen für Verbesserungsvorschläge. (*)	trifft völlig zu	trifft gar nicht zu	
... ist gut vorbereitet.	trifft völlig zu	trifft gar nicht zu	n=7 mw=1.7
... lässt sich außerhalb der Übung gut ansprechen, z.B. in Sprechstunden oder per Email. (*)	trifft völlig zu	trifft gar nicht zu	
Der Einsatz von Hilfsmitteln wie Wandtafel, Overhead, Beamer und Demonstrationen ist gut.	trifft völlig zu	trifft gar nicht zu	n=7 mw=1.9
Schrift und Zeichnungen in der Übung sind gut lesbar.	trifft völlig zu	trifft gar nicht zu	n=7 mw=2.4
Tafelanschrieb / Folien sind übersichtlich.	trifft völlig zu	trifft gar nicht zu	n=7 mw=2.3
Das Tempo ist ...	zu hoch	zu niedrig	n=7 mw=1.9

(*) Hinweis: Wenn die Anzahl der Antworten auf eine Frage zu gering ist, wird für die Frage keine Auswertung angezeigt.