Semantics and Verification of Software

Lecture 18: Dataflow Analysis V
(Efficient Fixpoint Computation)

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

Outline

Dataflow Systems

Definition (Dataflow system)

A dataflow system $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$ consists of

- a finite set of (program) labels L (here: L_c),
- a set of extremal labels $E \subseteq L$ (here: $\{\mathsf{init}(c)\}$ or $\mathsf{final}(c)$),
- a flow relation $F \subseteq L \times L$ (here: $\mathsf{flow}(c)$ or $\mathsf{flow}^R(c)$),
- a complete lattice $(D, \sqsubseteq)$ that satisfies ACC (with LUB operator $\bigsqcup$ and least element $\bot$),
- an extremal value $\iota \in D$ (for the extremal labels), and
- a collection of monotonic transfer functions $\{\varphi_l \mid l \in L\}$ of type $\varphi_l : D \to D$.

The Equation System

Definition (Dataflow equation system)

Let $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$ be a dataflow system. S defines the following equation system over the set of variables $\{\mathsf{AI}_l \mid l \in L\}$:

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

The Functional and Its Fixpoint

Definition (Dataflow functional)

The equation system of a dataflow system $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$ induces a functional

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

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

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

Corollary

The least fixpoint of Φ_S is effectively computable by iteration:

$$\mathsf{fix}(\Phi_S) = \bigsqcup\{\Phi_S^i(\bot_{D^n}) \mid i \in \mathbb{N}\}$$

where $\bot_{D^n} = (\underbrace{\bot_D, \ldots, \bot_D}_{n \text{ times}})$

Uniqueness of Solutions

Just as in the denotational semantics of `while` loops, solutions of dataflow equation systems are not unique.

Example

1. Available Expressions: consider

```
[z := x+y]^1;
while [true]^2 do
  [skip]^3;
```

$$\Longrightarrow \begin{aligned} AE_1 &= \emptyset \\ AE_2 &= (AE_1 \cup \{\texttt{x+y}\}) \cap AE_3 \\ AE_3 &= AE_2 \end{aligned}$$

$$\Longrightarrow \begin{aligned} AE_1 &= \emptyset \\ AE_2 &= \{\texttt{x+y}\} \cap AE_3 \\ AE_3 &= AE_2 \end{aligned}$$

$\Longrightarrow$ Solutions: $AE_1 = AE_2 = AE_3 = \emptyset$ or
$AE_1 = \emptyset, AE_2 = AE_3 = \{\texttt{x+y}\}$

Here: greatest solution $\{\texttt{x+y}\}$ (maximal potential for optimization)

2. Live Variables: see Exercise 9.3

Outline

A Worklist Algorithm I

Observation: fixpoint iteration re-computes every AI_l in every step

$\Longrightarrow$ redundant if $\mathsf{AI}_{l'}$ at no F-predecessor l' changed

$\Longrightarrow$ optimization by worklist

Algorithm 18.1 (Worklist algorithm)

Input: *dataflow system* $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$

Variables: $W \in (L \times L)^*$, $\{\mathsf{AI}_l \in D \mid l \in L\}$

Procedure: $W := \varepsilon;$ **for** $(l, l') \in F$ **do** $W := (l, l') \cdot W;$ *% Initialize W*
for $l \in L$ **do** *% Initialize* AI
  if $l \in E$ **then** $\mathsf{AI}_l := \iota$ **else** $\mathsf{AI}_l := \bot_D;$
while $W \neq \varepsilon$ **do**
  $(l, l') :=$ **head**$(W); W :=$ **tail**$(W);$
  if $\varphi_l(\mathsf{AI}_l) \not\sqsubseteq \mathsf{AI}_{l'}$ **then** *% Fixpoint not yet reached*
    $\mathsf{AI}_{l'} := \mathsf{AI}_{l'} \sqcup \varphi_l(\mathsf{AI}_l);$
    for $(l', l'') \in F$ **do** $W := (l', l'') \cdot W;$

Output: $\{\mathsf{AI}_l \mid l \in L\}$

A Worklist Algorithm II

Example 18.2 (Worklist algorithm)

Available Expression analysis for $c =$

```
[x := a+b]^1;
[y := a*b]^2;
while [y > a+b]^3 do
  [a := a+1]^4;
  [x := a+b]^5
```

(cf. Examples 14.9 and 17.3)

Transfer functions:

$$
\begin{aligned}
\varphi_1(A) &= A \cup \{\texttt{a+b}\}\\
\varphi_2(A) &= A \cup \{\texttt{a*b}\}\\
\varphi_3(A) &= A \cup \{\texttt{a+b}\}\\
\varphi_4(A) &= A \setminus \{\texttt{a+b}, \texttt{a*b}, \texttt{a+1}\}\\
\varphi_5(A) &= A \cup \{\texttt{a+b}\}
\end{aligned}
$$

Computation protocol: on the board

A Worklist Algorithm III

Properties of the algorithm:

Theorem 18.3 (Correctness of worklist algorithm)

Given a dataflow system $S = (L, E, F, (D, \sqsubseteq), \iota, \varphi)$*, Algorithm 18.1 always terminates and computes* $\mathsf{fix}(\Phi_S)$.

Proof.

see [Nielson/Nielson/Hankin 2005, p. 75 ff] □