Efficiently Monitoring Data-Flow Test Coverage*

Raul Santelices
Mary Jean Harrold

* Supported in part by NSF awards CCF-0541049, CCF-0429117, and CCF-0306372 to Georgia Tech and by Tata Consultancy Services, Ltd.
Problem: Cost of Monitoring

* Averages reported by Misurda et al. (ICSE 2005)
Approach: Branch-based Data-Flow Monitoring

- Infer data-flow coverage (as much as possible)
Approach: Branch-based Data-Flow Monitoring

- Infer data-flow coverage (as much as possible)
- Two alternatives for remaining data-flows:
Approach: Branch-based Data-Flow Monitoring

- Infer data-flow coverage (as much as possible)

- Two alternatives for remaining data-flows:
  1. Predict coverage
Approach: Branch-based Data-Flow Monitoring

- Infer data-flow coverage (as much as possible)
- Two alternatives for remaining data-flows:
  1. Predict coverage
  2. Directly monitor coverage (hybrid monitoring)
Previous Work

• Same-type entity subsumption
  - Agrawal (POPL ’94), Ball and Larus (TOPLAS ’94), Bertolino and Marré (TSE ’03)

• Different-type entity subsumption
  - Merlo and Antoniol (CASCON ’99), Santelices et al. (SOQUA ’06)

• Dynamic instrumentation
  - Arnold and Ryder (PLDI ’01), Tikir and Hollingsworth (ISSTA ’02), Misurda et al. (ICSE ’05)
Outline

- Background
  - Inferability analysis: what
  - Inferability analysis: how
  - Study
  - Conclusion
Program Representation

Control-Flow Graph (CFG)

En

\[ x = \ldots \]

1

\[ y = \ldots \]

\[ \ldots = x \]

2

3

T F

4

5

T F

6

y = x

7

Ex

\[ \ldots = y \]

\[ x = \ldots \quad \text{definition} \]

\[ \ldots = x \quad \text{use} \]
Program Representation

Control-Flow Graph (CFG)

\[ x = \ldots \]
\[ \ldots = x \]
\[ y = x \]

DUA \((1, 6, x)\)

\[ x = \ldots \] definition
\[ \ldots = x \] use
\[ x = \ldots \] definition-use association (DUA)
\[ \ldots = x \] data-flow entity

Aristotle Research Group

ASE 2007
November 9th
Traditional DUA Monitoring

1) At definition: record active definition
2) At use: retrieve active definition and mark DUA
Traditional DUA Monitoring

1) At definition: record active definition
2) At use: retrieve active definition and mark DUA

\[ \text{def}(y) = \]

**Diagram:}

- En
- x = ...
- y = ...
- \(\ldots = x\)
- 2
- 3
- 4
- 5
- 6
- 7
- Ex
- \(\ldots = y\)
- \(\ldots = x\)

**Tracking y**

**Definition-use association (DUA)**

**Data-flow entity**
Traditional DUA Monitoring

1) At definition: record active definition

2) At use: retrieve active definition and mark DUA

\[ \text{def}(y) = 3 \]

definition-use association (DUA)
data-flow entity
Traditional DUA Monitoring

1) At definition: record active definition
2) At use: retrieve active definition and mark DUA

\[ \text{def}(y) = 3 \]

\[ \text{def}(y) = 6 \]

\[ x = ... \]
\[ ... = x \]

14

ASE 2007
November 9th
Traditional DUA Monitoring

1) At definition: record active definition
2) At use: retrieve active definition and mark DUA

\[ \text{def}(y) = 6 \]
(6,5,y) covered!

\[ \text{def}(y) = 3 \]

1) At definition: record active definition
2) At use: retrieve active definition and mark DUA

\[ \text{def}(y) = 3 \]

\[ \text{def}(y) = 6 \]
(6,5,y) covered!

\[ \text{def}(y) = 3 \]
Outline

• Background
  ▶ Inferability analysis: what
  • Inferability analysis: how
  • Study
  • Conclusion
The Big Picture

pre-execution

Instrument Branches

P

execution

execute Test

P_{INST}

covered branches

post-execution

Report Coverage: Branches

branch
- covered
- not covered
The Big Picture

pre-execution

Instrument Branches

P

P_{\text{Inst}}

Execution Test

covered branches

Report Coverage: Branches + DUAs

branch
- covered
- not covered

DUA
- covered
- not covered
- possibly covered

imprecise

Inferability Analysis

branch requirements for DUAs

post-execution

P

ASE 2007
November 9th
The Big Picture

pre-execution

Instrument Branches + DUAs

execution

Execute Test

covered branches
covered DUAs

post-execution

Report Coverage: Branches + DUAs

branch
- covered
- not covered

DUA
- covered
- not covered
- possibly covered

precise!

Inferability Analysis

branch requirements for DUAs

remaining DUAs

precise!
Relating DUAs to Branches

DUA \((1,3,x)\)

\[ x = \ldots \]

\[ y = \ldots \]

\[ \ldots = x \]

\[ \text{1F } \Rightarrow (1,3,x) \text{ is covered} \]
Relating DUAs to Branches

DUA (3,5,y)

1F ⇒ (1,3,x) is covered

1F,4T ⇒ (3,5,y) is covered
- if y is not killed (redefined) at node 6
Relating DUAs to Branches

\[ x = \ldots \]
\[ y = \ldots \]
\[ \ldots = x \]

\[ x = \ldots \]
\[ \ldots = y \]

\[ y = x \]

necessary and:
- sufficient
- not sufficient

\[ B_{REQ} \]

\[ B_{KILL} \]

DUA

- must kill
- may kill

default if \[ B_{KILL} = \emptyset \]
Relating DUAs to Branches

- DUA \((1, 3, x)\)
  - En
  - \(x = \ldots\)
  - 1
    - y = \ldots
    - \ldots = x
  - 2
    - F
    - 3
      - F
      - Ex
    - T
    - 4
      - T
      - 5
        - y = x
      - 6
    - T
    - 7
    - Ex

- \(B_{REQ}\)
  - \{1F\}
  - sufficient
  - necessary and:
    - sufficient
    - not sufficient

- \(B_{KILL}\)
  - must kill
  - \(\emptyset\)

- DUA
  - \(B_{KILL}\)
  - • must kill
  - • may kill

- default if \(B_{KILL} = \emptyset\)
Relating DUAs to Branches

DUA (3, 5, y)

\[ x = \ldots \]
\[ y = \ldots \]
\[ \ldots = x \]

\[ x = \ldots \]
\[ y = x \]

\[ \ldots = y \]

\[ B_{\text{REQ}} \]

\{1F, 4T\}
sufficient

necessary and:
- sufficient
- not sufficient

\[ B_{\text{KILL}} \]

may kill
\{4F\}

- must kill
- may kill

default if \[ B_{\text{KILL}} = \emptyset \]
Relating DUAs to Branches

DUA \((6,5,y)\)

\[ x = \ldots \]
\[ y = \ldots \]
\[ \ldots = x \]

\[ y = x \]
\[ y = x \]
\[ y = x \]

\(B_{\text{REQ}}\) necessary and:
- sufficient
- not sufficient

\(B_{\text{KILL}}\) must kill
- must kill
- may kill

default if \(B_{\text{KILL}} = \emptyset\)
Deciding Coverage of a DUA

1. $B_{REQ}$ covered?
   - yes
     - yes
     - $B_{KILL}$ covered?
       - yes
         - yes
         - $B_{KILL}$ must kill?
           - yes
             - DUA possibly covered
           - no
             - DUA not covered
       - no
         - no
         - DUA definitely covered
   - no
     - no
     - DUA definitely covered

2. $B_{REQ}$ sufficient?
   - yes
     - yes
     - DUA definitely covered
   - no
     - no
     - DUA possibly covered

3. $B_{KILL}$ covered?
   - yes
     - yes
     - $B_{KILL}$ must kill?
       - yes
         - DUA definitely covered
       - no
         - no
         - DUA not covered
   - no
     - no
     - DUA possibly covered

4. necessary and:
   - sufficient
   - not sufficient

5. $B_{KILL}$
   - must kill
   - may kill

6. default if $B_{KILL} = \emptyset$
Classifying DUAs

- $B_{REQ}$ covered?
  - no: DUA not covered
  - yes: $B_{KILL}$ covered?
    - yes: $B_{KILL}$ must kill?
      - yes: inferable $B_{REQ}$ sufficient $B_{KILL}$ must kill
      - no: $B_{REQ}$ sufficient
    - no: DUA possibly covered
- yes: DUA definitely covered
Classifying DUAs

- $B_{REQ}$ covered?  
  - yes
  - no

- $B_{KILL}$ covered?  
  - yes
  - no

- $B_{REQUEST}$ sufficient?  
  - yes
  - no

- $B_{KILL}$ must kill?  
  - yes
  - no

inferable
- $B_{REQUEST}$ sufficient
- $B_{KILL}$ must kill

remaining DUAs:
conditionally-inferable
- $B_{REQUEST}$ sufficient
- $B_{KILL}$ contains a may-kill
Classifying DUAs

- \( B_{\text{REQ}} \) covered?
  - yes
    - \( B_{\text{REQ}} \) sufficient
    - \( B_{\text{KILL}} \) must kill
    - DUA definitely covered
  - no
    - \( B_{\text{KILL}} \) covered?
      - yes
        - \( B_{\text{KILL}} \) must kill?
          - yes
            - 
          - no
            - \( B_{\text{REQ}} \) sufficient?
              - no
                - DUA not covered
              - yes
                - DUA possibly covered
      - no
        - \( B_{\text{REQ}} \) covered?
          - yes
            - DUA not covered
          - no
            - remaining DUAs:
              - conditionally-inferable
                - \( B_{\text{REQ}} \) sufficient
                - \( B_{\text{KILL}} \) contains a may-kill
              - non-inferable
                - \( B_{\text{REQ}} \) not sufficient

inferable
Outline

• Background
• Inferability analysis: what
  ▶ Inferability analysis: how
• Study
• Conclusion
1. Computing $B_{\text{REQ}}$ and $B_{\text{KILL}}$

- Control-dependence:
  node $N$ is control-dependent on branch $B$ if taking $B$ implies $N$ covered

- For DUA $(d, u, v)$:
  $B_{\text{REQ}} = \text{union of control-dependences of } d \text{ and } u$
  $B_{\text{KILL}} = \text{union of control-dependences of all } \text{kills} \text{ of } (d, u, v)$

$B_{\text{REQ}} = \{1F\} \cup \{4T\}$
$B_{\text{KILL}} = \{4F\}$
2. Sufficiency of $B_{REQ}$

- Nodes $d$ and $u$ are in **node order** if:
  1. there exists no path from $u$ to $d$; or
  2. $d$ dominates $u$; or
  3. $u$ postdominates $d$

- $B_{REQ}$ is **sufficient** if definition and use are in node order
2. Sufficiency of $B_{REQ}$

$(1,3,x)$: $B_{REQ}$ is sufficient
- 1 is not reachable from 3
- 1 dominates 3
2. Sufficiency of $B_{REQ}$

(1,3,x): $B_{REQ}$ is sufficient
- 1 is not reachable from 3
- 1 dominates 3

(6,5,y): $B_{REQ}$ not sufficient
- there is a path from 5 to 6
- 6 does not dominate 5
- 5 does not postdominate 6
3. Must vs. May $B_{\text{KILL}}$

- Kill $k$ is aligned with DUA $(d,u,v)$ if:
  - $(d,k)$ and $(k,u)$ are in node order, and
  - $d$ is not reachable from $u$

- $B_{\text{KILL}}$ must kill if all kills aligned

DUA $(3,5,y)$

$\begin{array}{c}
\text{En} \\
\text{1} \\
\text{2} \\
\text{3} \\
\text{4} \\
\text{5} \\
\text{6} \\
\text{7} \\
\text{Ex}
\end{array}$

$\begin{array}{c}
x = \ldots \\
y = \ldots \\
\ldots = x \\
\text{kill} \\
\text{kill} \\
y = x \\
\text{not aligned...}
\end{array}$
Outline

• Background
• Inferability analysis: what
• Inferability analysis: how

Study

• Conclusion
Study and Tool

• Implemented tool: **DUA-Forensics**
  - Java, interprocedural, local variables

• Subjects:

<table>
<thead>
<tr>
<th>Subject</th>
<th>Lines of Code</th>
<th># DUAs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tcas</td>
<td>150</td>
<td>136</td>
</tr>
<tr>
<td>Compress</td>
<td>587</td>
<td>445</td>
</tr>
<tr>
<td>Db</td>
<td>663</td>
<td>526</td>
</tr>
<tr>
<td>Mtrt</td>
<td>1242</td>
<td>101</td>
</tr>
<tr>
<td>Scimark2</td>
<td>1805</td>
<td>1054</td>
</tr>
<tr>
<td>Jess</td>
<td>6365</td>
<td>12927</td>
</tr>
</tbody>
</table>
Study 1: Distribution of DUA Inferability Types

Subject program

<table>
<thead>
<tr>
<th>Program</th>
<th>Inferable</th>
<th>Conditionally Inferable</th>
<th>Non-inferable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tcas</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Compress</td>
<td>50</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>Db</td>
<td>40</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Mtrt</td>
<td>30</td>
<td>10</td>
<td>60</td>
</tr>
<tr>
<td>Scimark2</td>
<td>20</td>
<td>10</td>
<td>70</td>
</tr>
<tr>
<td>Jess</td>
<td>10</td>
<td>0</td>
<td>90</td>
</tr>
</tbody>
</table>
Study 2: Precision of Branch-based DUA Coverage Report

Subject program

- Compress
- Db
- Mtrt
- Scimark2
- Jess

total reported as covered

% DUAs
Study 2: Precision of Branch-based DUA Coverage Report

% DUA

- Compress
- Db
- Mtrt
- Scimark2
- Jess

Subject program

- total reported as covered
- definitely covered

ASE 2007
November 9th
Study 2: Precision of Branch-based DUA Coverage Report

% DUAs

- total reported as covered
- definitely covered
- predicted as covered

Subject program

Compress
Db
Mtrt
Scimark2
Jess

ASE 2007
November 9th
Study 2: Precision of Branch-based DUA Coverage Report

% DUA:
- total reported as covered
- definitely covered
- predicted as covered
- false positives

Precision:
- range: 76-100%
- average: 85%

Subject program

Compress  Db  Mtrt  Scimark2  Jess

ASE 2007
November 9th
Study 3: Branch and DUA Monitoring Overheads

<table>
<thead>
<tr>
<th>Type</th>
<th>min</th>
<th>max</th>
<th>avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>branch</td>
<td>1.3%</td>
<td>32.4%</td>
<td>9.0%</td>
</tr>
<tr>
<td>hybrid</td>
<td>1.7%</td>
<td>110.2%</td>
<td>47.0%</td>
</tr>
<tr>
<td>DUA</td>
<td>1.3%</td>
<td>160.4%</td>
<td>65.9%</td>
</tr>
</tbody>
</table>

imprecise, but cheap

fully precise, more expensive

Subject program

ASE 2007
November 9th
Outline

- Background
- Inferability analysis: what
- Inferability analysis: how
- Study
- Conclusion
Conclusion

- Inferability analysis and studies:
  - Monitor DUAs using only branches, with good estimate accuracy
  - Hybrid mix for precise DUA monitoring, more efficient than traditional approach
  - Support decision on branch-based vs. precise DUA monitoring
Future Work

• Study complications of aliasing
• Infer DUAs from acyclic paths instead of branches
• Infer DUAs or other flows for debugging, security
• Inferability types distribution as a data-flow complexity metric
Questions?