## Report Documentation Page

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>2. REPORT TYPE</th>
<th>3. DATES COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>28 OCT 2014</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. TITLE AND SUBTITLE</th>
<th>5a. CONTRACT NUMBER</th>
<th>5b. GRANT NUMBER</th>
<th>5c. PROGRAM ELEMENT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verifying Evolving Software</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. AUTHOR(S)</th>
<th>5d. PROJECT NUMBER</th>
<th>5e. TASK NUMBER</th>
<th>5f. WORK UNIT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gurfinkel / Arie</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</th>
<th>8. PERFORMING ORGANIZATION REPORT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</th>
<th>10. SPONSOR/MONITOR’S ACRONYM(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>11. SPONSOR/MONITOR’S REPORT NUMBER(S)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>12. DISTRIBUTION/AVAILABILITY STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approved for public release, distribution unlimited.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>13. SUPPLEMENTARY NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>The original document contains color images.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>14. ABSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>15. SUBJECT TERMS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16. SECURITY CLASSIFICATION OF:</th>
<th>17. LIMITATION OF ABSTRACT</th>
<th>18. NUMBER OF PAGES</th>
<th>19a. NAME OF RESPONSIBLE PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. REPORT</td>
<td>unclassified</td>
<td>SAR</td>
<td>13</td>
</tr>
<tr>
<td>b. ABSTRACT</td>
<td>unclassified</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. THIS PAGE</td>
<td>unclassified</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Standard Form 298 (Rev. 8-98)
Proscribed by ANSI Std Z39-18
Team: Verifying Evolving Software

SEI team members

• Dr. Arie Gurfinkel
• Dr. Sagar Chaki

Collaborators

• Dr. Anton Belov (Synopsys)
• Dr. Nikolaj Bjorner (Microsoft Research)
• Grigory Fedyukovich (Univ. of Lugano)
• Dr. Pierre-Loic Garoche (Onera)
• Dr. Alexander Ivrii (IBM)
• Dr. Temesghen Kahsai (NASA Ames)
• Prof. Natasha Sharygina (University of Lugano)
• Prof. Ofer Strichman (Technion)
Overview

Problem: Scalable verification of evolving software
- reduce re-verification effort
- close semantic gap between compiler and verifier
- enable safe use of compiler optimizations in safety-critical code

Related Work: Current solutions are limited by
- effectiveness (syntactic slicing, regression verification)
- high-maintenance cost (translation validation)
- narrow applicability (upgrade checking)

Key Idea: Propagate verification certificates across evolution boundaries
- generate verification certificates using proof-based verification techniques
- iteratively guess the mapping between original and evolved program
- propagate certificates and strengthen using incremental inductive verification
  - IIV is a new verification technique co-developed by us
Research Tasks

Verifying instcomine and simplifycfg optimizations of LLVM
  • with Prof. Natasha Sharygina and Grigory Fedyukovich (Univ. of Lugano)

Closing the semantic gap between Compiler and Verifier
  • with Dr. Anton Belov (Synopsys) and J. Marques-Silva (UCD)

Minimizing verification certificates
  • with Dr. Anton Belov (Synopsys) and Dr. Alexander Ivrii (IBM)

Certifying compiler for Luster
  • with Dr. Temesghen Kahsai (NASA Ames) and Dr. PL. Garoche (Onera)

Polyhedral Verification Certificates
  • with Dr. Nikolaj Bjorner (Microsoft Research)
Our Approach

1. Compute a verification certificate $C_1$ for program $P_1$

2. Evolve program $P_1$ to a program $P_2$
   - $P_2$ is obtained by compiler optimization, user change, semantics change, etc.

3. Adapt $C_1$ to certificate $C_2$ for $P_2$

4. Strengthen $C_2$ if necessary

Enabled by our recent breakthroughs in Inductive Incremental Verification that produces and uses verification certificates
Research Tasks

Verifying instcomine and simplifycfg optimizations of LLVM
  • with Prof. Natasha Sharygina and Grigory Fedyukovich (Univ. of Lugano)

Closing the semantic gap between Compiler and Verifier
  • with Dr. Anton Belov (Synopsys) and J. Marques-Silva (UCD)

Minimizing verification certificates
  • with Dr. Anton Belov (Synopsys) and Dr. Alexander Ivrii (IBM)

Certifying compiler for Luster
  • with Dr. Temesghen Kahsai (NASA Ames) and Dr. PL. Garoche (Onera)

Polyhedral Verification Certificates
  • with Dr. Nikolaj Bjorner (Microsoft Research)
Compiler and Verifier Semantic Gap

Source Code

```
int x, a;
```
```
x = 2 * a;
```

Machine Code

```
% x = mul i32 2, % a
```
```
(= x_0 (mul 2 a_0))
```

Compiler

Verifier

Arithmetic Constraints

semantic gap

Compiler and Verifier Semantic Gap

Source Code

```
int x, a;
```
```
x = 2 * a;
```

Machine Code

```
% x = mul i32 2, % a
```
```
(= x_0 (mul 2 a_0))
```

Compiler

Verifier

Arithmetic Constraints

semantic gap
**MISPER: Synthesizing Safe Bit-Precise Invariants**

Source code:

```c
int x, a;
x = 2 * a;
```

Machine code:

```assembly
...%x = mul i32 2, %a...
```

Constraints:

```plaintext
... (= x₀ (mul 2 a₀))...
```

Precise certificate

Precise Invariant

Scalable Verifier

Arithmetic certificate

Adapt Certificate

Fall 2014 SEI Research Review
Gurfinkel, October 28, 2014
© 2014 Carnegie Mellon University
FrankenBit: Bit-Precise Verification w/ Many Bits

MISPER to synthesize bit-precise invariants
LLBMC to search for counterexamples
Silver and Bronze medals at SV-COMP 2014

ControlFlow

DeviceDrivers64

Outcomes

Tools

• FrankenBit – bit-precise verifier for C
• Niagara – validator for LLVM compiler optimizations
• Zuster – verifier for Luster programs

Publications

• Synthesizing Safe Bit-Precise Invariants. TACAS 2014
• FrankenBit: Bit-Precise Verification with Many Bits (Tool paper). TACAS 2014
• Incremental Verification of Compiler Optimizations. NASA FM 2014
• Synthesizing Modular Invariants for Synchronous Code. HCVS 2014
• Small Inductive Safe Invariants. FMCAD 2014
• Property Directed Polyhedral Abstraction. VMCAI 2015
• Automated Discovery of Simulation Between Programs. Submitted to TACAS 2015
## Contact Information

<table>
<thead>
<tr>
<th>Arie Gurfinkel</th>
<th>U.S. Mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sr. Researcher</td>
<td>Software Engineering Institute</td>
</tr>
<tr>
<td>SSD</td>
<td>Customer Relations</td>
</tr>
<tr>
<td>Telephone: +1 412-268-5800</td>
<td>4500 Fifth Avenue</td>
</tr>
<tr>
<td>Email: <a href="mailto:arie@sei.cmu.edu">arie@sei.cmu.edu</a></td>
<td>Pittsburgh, PA 15213-2612</td>
</tr>
<tr>
<td></td>
<td>USA</td>
</tr>
</tbody>
</table>

### Web
- www.sei.cmu.edu/staff/arie
- www.sei.cmu.edu/contact.cfm

### Customer Relations
- Email: info@sei.cmu.edu
- Telephone: +1 412-268-5800
- SEI Phone: +1 412-268-5800
- SEI Fax: +1 412-268-6257