Finding Sufficient Mutation Operators via Variable Reduction

Akbar Siami Namin and James H. Andrews
Computer Science Department
The University of Western Ontario
London, Ontario, Canada
{asiamina, andrews}@csd.uwo.ca

Second Workshop on Mutation Analysis (Mutation 2006)
in conjunction with the 17th IEEE International Symposium on Software Reliability Engineering (ISSRE 2006)
November 2006, Raleigh, North Carolina, USA.
Outline

- Introduction
- Related Work
- Experimental Procedure
- Statistical Analysis
- Preliminary Results
- Discussion
- Conclusion and Future Work
Introduction

- Needs faulty versions for experiments
  - Applying mutation operators to programs
- Many mutation operators have been proposed
  - Enormous number of mutants $\Rightarrow$ Infeasible Computation
- Finding a reduced set which is sufficient
- Our approach
  - Applying variable reduction techniques
Related Work

Mutation Testing and Analysis

• Mutation Testing
  – Generating faulty versions: measuring the test suites adequacy (Offutt et al.)

• Mutants behave very similarly to real faults (Andrews et al.)

• Mutant generator tools: Mothra (Fortran), Proteum (C), and MuJava (Java)
Related Work (cont)

Sufficient Mutation Operators

- Sufficient mutation operators problem (Offutt et al.)
- Experiments to find the sufficient set for Fortran (Wong)

Variable Reduction

- Decreasing the number of variables (Jolliffe)
Experimental Procedure

Setup

- **Subject programs**: Siemens programs
  - Seven subject programs
- **Mutant generator**: Proteum, C mutant generator (Maldonado et al.)
  - 108 mutation operators
- **Code coverage tool**: ATAC
  - Block, Decision, Computation Use, Predicate Use
Experimental Procedure (cont)

Definitions of Sets and Measurements

- Mutation operators: $\mu_i$, $1 \leq i \leq 108$
- $NMNE_i$, $\#$ of non-equivalent mutants generated using $\mu_i$
- $TNMNE$, total $\#$ of mutants that are nonequivalent
- For each test suite $S$:
  - $kills_i(S)$, The set of mutants generated by $\mu_i$ and killed by $S$
  - $kills(S)$, The set of all mutants killed by $S$
  - $Am_i(S) = |kills_i(S)|/NMNE_i$, the adequacy ratio for mutants of operator $\mu_i$
  - $AM(S) = |kills(S)|/TNMNE$, the adequacy ratio for all mutants
Test Suite Groups

- Sufficient operators may depend on test suite construction
- Test suite construction methods studied:
  - $SING$, group of singleton test suites
  - $RAND_i$, $i = \{10, 20, 50, 100\}$, groups of randomly-selected test suites with size $i$
  - $BLK, DEC, CUSE, PUSE$, coverage-based test suites
Statistical Analysis

- All-Subset Regression (SUB)
- Elimination-Based Correlation Technique (EBC)
- Cluster Analysis (CA)
Statistical Analysis

All Subset Regression (SUB)

• Select a subset from a larger number of variables to predict another variable
• Finding subsets which fit well
• Number of variables in the prediction formula can be identified
• Constructing all regression models (Exhaustive)
Statistical Analysis

Elimination-Based Correlation Technique (EBC)

• Based on the correlation coefficients between variables
• constructing correlation matrix
• Eliminating one which has higher correlation with others
• High correlation: $\text{cor}(X, Y) \geq 0.90$
Statistical Analysis

Cluster Analysis (CA)

• Based on similarities of two variables
• Constructing proximity matrix
• Distances of variables in multi-dimensional space (Nearest Neighbor)
• Producing dendrogram
Preliminary Results

- Applied the techniques to tcas.c program so far.
- 300 test suites for each group
- 49 operators: no generated mutants
- Applied only to $SING, RAND_i$
- Applied each technique to each group of test suites
- Merged data for groups $\Rightarrow$ Core set of operators from each technique
## Preliminary Results (cont)

### Core sets - No operators common to all

<table>
<thead>
<tr>
<th>core-SUB</th>
<th>Mutants</th>
<th>core-EBC</th>
<th>Mutants</th>
<th>core-CA</th>
<th>Mutants</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-DirVarAriNeg</td>
<td>44</td>
<td>I-IndVarAriNeg</td>
<td>19</td>
<td>I-CovAllEdg</td>
<td>12</td>
</tr>
<tr>
<td>I-DirVarRepReq</td>
<td>220</td>
<td>I-IndVarLogNeg</td>
<td>19</td>
<td>I-DirVarRepCon</td>
<td>38</td>
</tr>
<tr>
<td>I-IndVarLogNeg</td>
<td>19</td>
<td>II-ArgRepReq</td>
<td>5</td>
<td>I-IndVarAriNeg</td>
<td>19</td>
</tr>
<tr>
<td>I-IndVarRepReq</td>
<td>91</td>
<td>u-OALN</td>
<td>2</td>
<td>I-RetStaDel</td>
<td>17</td>
</tr>
<tr>
<td>u-OABN</td>
<td>3</td>
<td>u-OASN</td>
<td>2</td>
<td>II-ArgIncDec</td>
<td>54</td>
</tr>
<tr>
<td>u-OLSN</td>
<td>34</td>
<td>u-OCNG</td>
<td>8</td>
<td>u-OABN</td>
<td>3</td>
</tr>
<tr>
<td>u-VDTR</td>
<td>111</td>
<td>u-OLLN</td>
<td>17</td>
<td>u-OALN</td>
<td>2</td>
</tr>
<tr>
<td>u-VGSR</td>
<td>794</td>
<td></td>
<td></td>
<td>u-OASN</td>
<td>2</td>
</tr>
<tr>
<td>u-VTWD</td>
<td>74</td>
<td></td>
<td></td>
<td>u-OCNG</td>
<td>8</td>
</tr>
<tr>
<td>II-ArgLogNeg</td>
<td>3</td>
<td></td>
<td></td>
<td>u-OLNG</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong> 1393</td>
<td></td>
<td><strong>Total</strong> 72</td>
<td></td>
<td><strong>Total</strong> 366</td>
</tr>
</tbody>
</table>
Comparison of core sets

<table>
<thead>
<tr>
<th></th>
<th>Core-SUB</th>
<th>Core-EBC</th>
<th>Core-CA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutants</td>
<td>1393</td>
<td>72</td>
<td>366</td>
</tr>
<tr>
<td>%Total</td>
<td>28.22%</td>
<td>1.45%</td>
<td>7.41%</td>
</tr>
<tr>
<td>%Saving</td>
<td>71.78%</td>
<td>98.55%</td>
<td>92.59%</td>
</tr>
<tr>
<td>Operators</td>
<td>10</td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>%Total</td>
<td>9.25%</td>
<td>6.48%</td>
<td>12.03%</td>
</tr>
<tr>
<td>%Saving</td>
<td>90.75%</td>
<td>93.52%</td>
<td>87.97%</td>
</tr>
</tbody>
</table>
Preliminary Results (cont)

Predicted vs. Actual AM for Core–SUB

Predicted AM – SUB

Actual AM

Predicted vs. Actual AM for Core–SUB

Predicted AM – SUB

Actual AM
Preliminary Results (cont)

Predicted vs. Actual AM for Core–EBC
Preliminary Results (cont)
Similarity of Sufficient Mutation Operators for tcas-100
Discussion

Threats to Validity

• Analysis for tcas.c so far
• The correctness of the mutant generation and scripting processes
  – Rely on Proteum and our shell scripts
• Use of C programs

Data Combination and Analysis

• The order of activities:
  1. Combining Data from subject program
  2. Statistical Analysis
  3. Combine data from test suite groups
  4. Fit linear models
Conclusion and Future Work

Conclusion

• Applying variable reduction techniques to address sufficient mutation operators problem
  – Subset selection analysis considers AM as target
  – Correlation and cluster analyses try to identify the most distinct set of operators
• Our experiment reveals valuable information

Future Work

• Extend to all seven Siemens programs
• Study of possibilities of finding non-linear relationships among mutation operators
• Identifying one or several sets of mutation operators as sufficient
Thanks

Questions
and
Answers