**RESEARCH PROBLEM**

- Test dependence causes inconsistent test result when tests are run in different execution orders.
- There are two serious consequences.
  - False positives: Test fails even when the software is correct.
  - False negatives: Test passes even when a bug exists.

**APPROACH**

- Automatically applies various refactoring to the code under test and test code as they are compiled.
- Most common root cause of test dependence is side-effecting access to shared global variables.
- Our prototype addresses side-effecting tests in two phases.
  1. Determine the initial value of every global variable.
  2. For every test, reassign all occurrences of global variables whose value is potentially read during the test’s execution.

**RESULTS**

- Subject programs were chosen from our previous study.
- Had no knowledge of whether the programs actually contained dependent tests.
- Measure the number of dependent tests exposed by applying test prioritization with and without our prototype.
- With a lower-bound number from a previous study, our prototype eliminates
  - 10.8% of human-written dependent tests
  - 12.5% of the automatically-generated dependent tests

<table>
<thead>
<tr>
<th>Program</th>
<th>LOC</th>
<th>CUT</th>
<th>Tests</th>
<th>Human</th>
<th>Auto</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crystal</td>
<td>4573</td>
<td>1302</td>
<td>78</td>
<td>3198</td>
<td>1.02</td>
<td>20111015</td>
</tr>
<tr>
<td>JFreechart</td>
<td>92255</td>
<td>49942</td>
<td>2234</td>
<td>2438</td>
<td>1.015</td>
<td>b069d7d66d</td>
</tr>
<tr>
<td>Joda-Time</td>
<td>27183</td>
<td>51492</td>
<td>3875</td>
<td>2234</td>
<td>1.015</td>
<td>d5ea6bf3157e</td>
</tr>
<tr>
<td>Synoptic</td>
<td>5317</td>
<td>2758</td>
<td>118</td>
<td>2467</td>
<td>1.04</td>
<td></td>
</tr>
<tr>
<td>XML Security</td>
<td>18255</td>
<td>3807</td>
<td>108</td>
<td>665</td>
<td>1.04</td>
<td></td>
</tr>
</tbody>
</table>

**RELATED & FUTURE WORK**

**Related work**

- Tests could be required to run in a given order.
  - Prevents the use of test selection and prioritization.
- Execute each test in a separate virtual machine.
  - Significantly increases test execution time.
- Other related work only detects test dependence.
  - Our approach repairs test dependence.
  - Repaired tests yield consistent results on a single, standard JVM.

**Future work**

- Our prototype fixes test dependence for only variables that are of primitive or String types.
  - Improve the prototype to support variables of any type.
- Expand our approach to handle additional causes of test dependence.
  - Such as access to a database or file system and concurrent programs.
- By covering additional causes of test dependence, our approach can repair all cases of test dependence.

This work is supported by the US National Science Foundation under grant no. CNS-1434582.