Oracle Strategies for Automated Testing
(Tutorial)
Douglas Hoffman

Keywords: Automated Testing, Model of Testing, Software Under Test, Test Oracles, Test Verification, Test Validation

Abstract

Software test automation is often a difficult and complex process. The most familiar aspects of test automation are organizing and running of test cases and capturing and verifying test results. A set of expected results are needed for each test case in order to check the test results. Verification of these expected results is often done using a mechanism called a test oracle. This paper describes classes of oracles the author has used for various types of automated software verification and validation. Several relevant characteristics of oracles are included with the advantages, disadvantages, and implications for test automation.

When the topic of Oracles was first presented at Quality Week 1998 it had caused ripple effect. A two day LAWST (Los Altos Workshop on Software Testing) session was dedicated to discussion of oracles and test automation and several new papers on the subject were generated and presented by other people at conferences in 1999. I have updated and expanded the original work based on the LAWST workshop and further investigation, comparison, refinement, and critical reevaluation of oracles as used in automated software testing.

Real world oracles vary widely in their characteristics. Although the mechanics of various oracles may be vastly different, a few classes can be identified which correspond with automated test approaches. These types of oracles are categorized based upon the strategy for verification using the oracle. Thus, an oracle strategy using a lookup table to generate expected results can be the same as one that uses an alternate algorithm implementation to compute them. Four types of oracle strategies (and not using any oracle) are identified and defined in Table 1. The strategies are labeled True, Heuristic, Consistency, and Self Referential.

<table>
<thead>
<tr>
<th>True Oracle</th>
<th>Heuristic Strategy</th>
<th>Consistency</th>
<th>Self Referential</th>
<th>No Oracle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition</td>
<td>-Independent generation of all expected results</td>
<td>-Verifies some values, as well as consistency of remaining values</td>
<td>-Verifies current run results with a previous run (Regression Test)</td>
<td>-Embeds answer within data in the messages</td>
</tr>
<tr>
<td>Advantages</td>
<td>-No encountered errors go undetected</td>
<td>-Faster and easier than True Oracle</td>
<td>-Verification is straightforward</td>
<td>-Allows extensive post-test analysis</td>
</tr>
<tr>
<td>Disadvantages</td>
<td>-Expensive to implement</td>
<td>-Can miss systematic errors (as in sine wave example)</td>
<td>-Original run may include undetected errors</td>
<td>-Must define answers and generate messages to contain them</td>
</tr>
</tbody>
</table>

Five Oracle Strategies

1 This work updates and extends my original presentation, “A Taxonomy of Test Oracles” from Quality Week 1988.
Douglas Hoffman is an independent consultant with Software Quality Methods, LLC. He has been in the software engineering and quality assurance fields for over 25 years and now is a management consultant in strategic and tactical planning for software quality. For five years he served as the Chairman of the Santa Clara Valley Software Quality Association (SSQA), a Task Group of the American Society for Quality (ASQ). He has been a presenter and participant at dozens of software quality conferences and has been Program Chairman for several international conferences on software quality. He is a member of the ACM and IEEE and is active in the ASQ as a Senior Member, participating in the Software Division, the Santa Clara Valley Section, and the Software Quality Task Group. He has earned a BA in Computer Science, an MS in Electrical Engineering, an MBA, a Certificate from ASQ in Software Quality Engineering, and has been a registered ISO 9000 Lead Auditor.

Douglas’ experience includes consulting, teaching, managing, and engineering across the computer systems and software industries. He has over fifteen years experience in creating and transforming software quality and development groups, and has worked as an independent consultant for over ten years. His work in corporate, quality assurance, development, manufacturing, and support organizations provides a broad technical and managerial perspective on the computer industry.