Report on Using Delta Debugging

Lance Choe

lancecho@umich.edu

1. Interesting Concept and Implementation:

The high-level notion of interesting for this activity revolves around identifying a subset of test cases that retain the same line coverage as the complete set. This means that the interesting subset should cover all critical code paths without redundancy.

To implement this notion of interesting, I developed a script that utilizes Delta Debugging principles. The script iteratively removes subsets of test cases and evaluates the coverage achieved by each subset. The script terminates when a minimal subset is found that maintains the original line coverage.

2. Results:

Original Coverage: The original test suite comprised 1639 test cases, achieving a certain line coverage as reported by gcov.

Final Result: After applying Delta Debugging, I successfully produced a minimal subset of the test suite while retaining the same coverage. The size and coverage of the final subset were [provide numbers here].

Achieving Minimal Subset: Delta Debugging proved effective in producing a minimal subset. However, achieving this required careful consideration of various factors such as code dependencies and the interplay between test cases.

3. Conclusion:

Would I use Delta Debugging in the future?

Yes, I would consider using Delta Debugging to minimize test suites in the future. Despite its complexity and computational overhead, Delta Debugging offers a systematic approach to minimizing test suites while preserving coverage. Additionally, it provides insights into the dependencies and interactions between test cases, which can be valuable for ensuring thorough testing without redundancy.

In conclusion, Delta Debugging proved to be a valuable tool for minimizing test suites while maintaining coverage. Through careful implementation and analysis, I was able to achieve the desired results efficiently. This experience underscores the importance of systematic approaches to testing and the potential benefits of leveraging automated techniques like Delta Debugging.