AN INTELLIGENT METHOD FOR C++ TEST CASE SYNTHESIS BASED ON A Q-LEARNING AGENT

PhD Applicant Mykhailo Hulevych, National Technical University "Kharkiv Polytechnic Institute", Kharkiv

Ensuring the quality of a software library requires an effective test suite (TS). Works [1 – 4] review approaches and techniques for automating the formation and optimization of test cases (TCs) and substantiate the feasibility of employing reinforcement learning for TC construction and optimization.

We propose a novel intelligent method for synthesizing test cases for C++ libraries based on a Q-learning agent [3]. The TC construction process is formalized as a Markov Decision Process (MDP) combined with Q-learning. The agent operates over an action space built from API function calls of the target library, obtained via instrumentation and execution tracing of the original TCs. The agent state is represented compactly as a suffix of the function calls history. During training, the agent receives a combined reward that encourages increases in branch coverage while penalizing redundant actions. Experience is accumulated in a Q-table, enabling knowledge transfer within a given test suite. After training, a synthesis algorithm constructs new TCs according to the learned action-selection policy. The method does not require API specifications of the tested library.

A compression coefficient for original TC_{orig} and output TC_{out} test cases is defined as follows:

$$k = 1 - Len(TC_{out})/Len(TC_{orig}).$$

The method was evaluated on two C++ libraries of different complexity. The results show that our method achieves $k \approx 0.67$, while preserving branch coverage, confirming competitiveness with classical greedy minimization techniques [4], which on average achieve $k \approx 0.70$. The approach is therefore suitable for minimizing test suites in resource-constrained or specifications-limited environments.

References: 1. Hulevych, M. CIDER: ASSISTED AUTOMATION TOOL FOR C++ LIBRARIES TESTING. *Systems of Control, Nav. and Communication*. 2024;2(76):74–77.https://doi.org/10.26906/sunz.2024.2.074. **2.** Hulevych M., Kolomiitsev O. AUTOMATED TESTGENERATION TECHNIQUES FOR C++ SOFTWARE. *Systems of Control, Nav. and Communication*. 2025; 2(80): 102–107. https://doi.org/10.26906/SUNZ.2025.2.102. **3.** Semenov S., Kolomiitsev O., Hulevych M., Mazurek P., Chernyk O. An Intelligent Method for C++ Test Case Synthesis Based on a Q-Learning Agent. Applied Sciences. 2025; 15: 8596. https://doi.org/10.3390/app15158596. **4.** Jehan, S.; Wotawa, F. An Empirical Study of Greedy Test Suite Minimization Techniques Using Mutation Coverage. IEEE Access 2023, 11, 65427–65442. https://doi.org/10.1109/ACCESS.2023.3289073.