Random Input and Automated Output Generation in Submitty

Evan Maicus
Rensselaer Polytechnic Institute
maicue@rpi.edu

Drumil Patel
Indian Institue of Technology, Roorkee
drumilpatel720@gmail.com

Matthew Peveler
Rensselaer Polytechnic Institute
pevelm@rpi.edu

Barbara Cutler
Rensselaer Polytechnic Institute
cutler@cs.rpi.edu

ABSTRACT

"Fuzzing," testing a codebase against a set of randomly generated inputs, has become a promising model of testing across the industry due to its ability to reveal difficult to detect bugs. Separately, the use of randomized inputs when testing student code submissions removes the potential for student "hard-coding" behavior. Motivated by these factors, we present a solution for the automated generation of testcase inputs and expected outputs within Submitty, an open source automated grading system from Rensselaer Polytechnic Institute. We detail a new, enhanced workflow that allows instructors to provide our testing system with an assignment-specific input generation script and an assignment solution. The input generation script is run at student test-time, providing students with either entirely generated inputs, or a combination of generated and hand-crafted testcases. The instructor solution is run against the same inputs to produce expected results. This model of testcase specification carries the benefit of simple regeneration of expected output files if an assignment's specification changes after submissions open or between semesters. We present preliminary results of the use of random input generation in our large introductory programming courses, and evaluate the ability of random inputs to curb student hardcoding behavior as it relates to an "early submission incentive" system, which grants students an extension for achieving a target assignment score early in the week an assignment is due. We examine random input generation's ability to reveal bugs in student submissions from previous semesters.

CCS CONCEPTS

• **Computing Education** → Computing Education Programs; Computer Science Education.

KEYWORDS

Autograding, Testing, Fuzz Testing

ACM Reference Format:
Evan Maicus, Drumil Patel, Matthew Peveler, and Barbara Cutler. 2019. Random Input and Automated Output Generation in Submitty. In *Proceedings of ACM Technical Symposium on Computer Science Education (SIGCSE 2020).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

SIGCSE 2020, March 2020, Portland, Oregon, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 AUTOMATED GRADING CHALLENGES

Our work addresses multiple major challenges common to all automated grading systems. First, it represents a significant time savings when specifying new assignments for which a solution has been written or when amending assignment specifications. Second, the specification of testcases with random inputs allows instructors to "fuzz" student code. This carries the added benefit of halting student hard-coding behavior. Finally, our system allows instructors to capture traditionally difficult to collect expected outputs, such as outputs that result from scripted mouse and keyboard interactions.

2 RANDOM INPUT CASE STUDY

We present preliminary results of the use of random input generation in our current 300 student Data Structures course. Anecdotally the system teaches students about the use of and importance of fuzzing in automated software testing and builds confidence in the quality of their own code. Using student submissions and data from this vs. prior terms we evaluate the ability of random inputs to curb student hardcoding behavior. Specifically, we are interested in reducing the use of hardcoding to *beat the system* with our "early submission incentive" policy, which grants students a deadline extension for achieving a target assignment score early in the week an assignment is due. We also examine random input generation's ability to reveal bugs in student submissions that were not accounted for in legacy testcases from previous semesters.

3 CONCLUSIONS & FUTURE WORK

Our contribution is the integration of random test case generation and evaluation of output in an open source automated grading tool. Our system effectively halts student hard-coding behavior, makes assignment management simpler for instructors with pre-written assignment solutions, and allows "fuzzing" of student assignments.

By its nature, input fuzzing works best with many hundreds or thousands of testcases. While our system scales well to such large test-suites, the website frontend viewed by the student has been written with smaller, hand-crafted test suites in mind. As future work, we will enhance our website to better aggregate and display the results of hundreds of "fuzz" testcases.

4 WHAT IS SUBMITTY?

Submitty is an open source automated grading and course management system, developed at Rensselaer Polytechnic Institute. Submitty allows secure testing of progamming assignments with with immediate feedback to students and complementary manual grading. It also includes a integrated discussion forum, team assignments, static analysis, and many other features.

5 POSTER LAYOUT

Our poster will contain 5 columns. From left to right, these will contain:

(1) Our abstract (as above) and information about the Submitty autograding system. A high level explanation of Submitty's previous method of assignment configuration, with illustrative diagrams.
(2) An explanation of output generation using an instructor solution, including examples of generated outputs for assignments where output is difficult to otherwise capture.
(3) A detailed explanation of random input generation, including examples and an illustrative diagram. We will also analyze the performance and security implications of these extensions.
(4) The results of our case study, including information about the effectiveness of "fuzzing," and the effects of random testcase generation in supressing student hard coding behavior.
(5) Future work and additional details about Submitty and its associated resources.

6 AUTHOR EXPERTISE

Barbara Cutler, Matthew Peveler, and Evan Maicus have published a collective 12 works in the field of automated grading and course management. Drumil Patel was the primary developer for this Submitty extension for his 2019 Google Summer of Code project.

REFERENCES

[1] Ella Bounimova, Patrice Godefroid, and David Molnar. 2013. Billions and billions of constraints: whitebox fuzz testing in production. In *Proceedings of the 2013 International Conference on Software Engineering*. IEEE Press, San Francisco, CA, USA. https://dl.acm.org/citation.cfm?id=2486805

[2] Evan Maicus, Matthew Peveler, Stacy Patterson, and Barbara Cutler. 2019. Autograding Distributed Algorithms in Networked Containers. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education - SIGCSE '19*. ACM Press, Minneapolis, MN, USA, 133–138. https://doi.org/10.1145/3287324.3287505

[3] Matthew Peveler, Evan Maicus, and Barbara Cutler. 2019. Comparing Jailed Sandboxes vs Containers Within an Autograding System. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education - SIGCSE '19*. ACM Press, Minneapolis, MN, USA, 139–145. https://doi.org/10.1145/3287324.3287507

[4] Sumukh Sridhara, Brian Hou, Jeffrey Lu, and John DeNero. 2016. Fuzz Testing Projects in Massive Courses. In *Proceedings of the Third (2016) ACM Conference on Learning @ Scale - L@S '16*. ACM Press, Edinburgh, Scotland, UK, 361–367. https://doi.org/10.1145/2876034.2876050