A Replication and Reproduction of icse20-main-171 "Efficient Generation of Error-Inducing Floating-Point Inputs via Symbolic Execution"

Hui Guo[1] **and Cindy Rubio-González**[2]

[1]**higuo@ucdavis.edu, github ID: HGuo15, the corresponding author**
[2]**crubio@ucdavis.edu, github ID: crubiog**

ABSTRACT

Link to original paper: https://www.dropbox.com/s/j499nc0383hkncd/icse20-paper171.pdf?dl=0

WHAT This work is to replicate and reproduce the results of FPGen that implements the algorithm proposed in the accepted paper icse20-main-171 "Efficient Generation of Error-Inducing Floating-Point Inputs via Symbolic Execution". It provides the artifact and instructions to replicate/reproduce the results of FPGen and all 3 baseline tools on all of the 27 benchmarks described in the paper.

WHY Floating point is widely used in software applications to emulate arithmetic over reals. Unfortunately, floating point generates rounding errors that propagate and accumulate during execution. Generating inputs to *maximize* the numerical error is critical when evaluating the accuracy of floating-point code. To the best of our knowledge, FPGen is the first tool that enables symbolic execution to generate high error-inducing inputs for floating-point code, and our evaluation shows that FPGen greatly improves the state of the art.

HOW The original results are produced on a workstation with Intel(R) Xeon(R) Gold 6238 CPU (8 cores, 2.10GHz) and 32GB RAM. To reproduce the results, a machine with similar CPUs($\sim$2.10GHz and at least 8 cores) and 32GB or larger of RAM is required.

- Pull the docker image of FPGen artifact and create and run the FPGen container.
  $docker pull ucdavisplse/fpgen-artifact:icse20
  $docker run -ti --name=FPGen --cpus=8 --memory=32g ucdavisplse/fpgen-artifact:icse20
- In the home directory of FPGen container, clone the source of FPGen from GitHub, and rename it,
  $cd; git clone https://github.com/ucd-plse/FPGen.git; mv FPGen FPTesting
  then follow the instructions here:
  https://github.com/ucd-plse/FPGen.git
  to run FPGen or baseline tools (RANDOM, S3FP, KLEE-FLOAT) on selected benchmarks or all benchmarks.

WHERE The replication/reproduction includes the results of FPGen and all 3 baseline tools on all the 27 benchmarks described in the paper, more specifically, it includes all the results shown in Table 3 on Page 9 "Accuracy testing results for numerical library routines" in the paper. There are no other tables to reproduce.

DISCUSSION FPGen starts with a random search that provides the base values of the input data for the algorithm to further improve. The random search, however, is performed with a time threshold which we found is unstable across machines even with similar hardware specifications. To increase the chance of replication/reproduction, we provide the base values in the artifact and skip this step, which is observed to be helpful in replicating/reproducing the FPGen results in other machines. Note that the instructions to replicate the FPGen results from scratch (i.e., also create base values) are also provided for full reference.