Automatic Test Input Generation

Wei Le

February 21, 2021

Outline

- terminologies
- black box test input generation: random, combinatorial testing
- white box test input generation: symbolic execution, conclic testing
- grey/black box test input generation: fuzzing

Terminologies

two types of test input generation:

- test input generation for a function: input for a function
- test case/input generation for a class: a program with a combination of methods in the class

unit testing, *test harness*, *mocking*, (create objects or calls that simulate the behavior of real objects and calls), *test oracle* (used to determine if the output is correct), *testing criteria/coverage criteria*

Blackbox testing

- testing criteria focus on inputs
- random: for all data types – integers, floats, strings, chars, Boolean, pointers, structures
- combinatorial testing: a program/function has more than one input
  pairwise testing: test all possible discrete combinations for every two input parameters

  See an example here

Whitebox testing

- testing criteria focus on code: covering statements, branches, paths, testing def-use pairs
- *symbolic execution*, *concolic testing*
- Tools: KLEE (open source, C), Pex, Sage (Microsoft), Java Pathfinder (open source, NASA), CUTE (Berkeley, open source, C), S2E (open source, distributed)

Greybox testing

- start testing based on the random seed and use whitebox testing criteria to guide the test input generation
- Tools: afl, libfuzz, Ramdasa Cluster fuzzing by Google

Reference and further reading

- The Fuzzing Project
- Whitepaper for AFL
- Evaluating Fuzzing Testing