UTBot Simplifies Auto Test Generation

Samat Gaynutdinov, Saveliy Grigoryev, Pavel Iatchenii, Elena Ilina, Dmitry Ivanov, Vladislav Kalugin, Aleksei Pleshakov, Pavel Ponomarev, Konstantin Rybkin, Svetlana Shmidt, Vadim Volodin, Alexey Utkin
KLEE for testing C code

To create a test case for a function you would need to:

- Configure a project
- Introduce KLEE entry point with symbolic variables
- Build the project in LLVM IR
- Run KLEE
- Parse KLEE output for generated test cases
- Write the test code based on parsed KLEE output
- Run the test cases

The process requires a lot of time!
It is hard to generate tests for real-world projects with KLEE
Solution: UTBot for C

- Test generation process automation
- Multiple test generation scenarios:
  - for a whole project;
  - for a folder/file/function;
  - for a line (i.e., generate a test that executes the specific line);
  - for an assert (i.e., generate test that fails given assert);
  - with an expected return value.
- Support for C
  - including floats, data types, function pointers and recursive types
- Key features
  - Context definition, stubs generation, running tests, coverage calculation
- Basic C++ support
How UTBot works
Prepare a project

<table>
<thead>
<tr>
<th>Project Type</th>
<th>Project preparation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMake project</td>
<td>UTBot runs <strong>patched CMake</strong> to generate both <code>compile_commands.json</code> and <code>link_commands.json</code></td>
</tr>
<tr>
<td>Make project</td>
<td>UTBot runs <strong>patched Bear</strong> to generate both <code>compile_commands.json</code> and <code>link_commands.json</code></td>
</tr>
<tr>
<td>Other</td>
<td>UTBot asks user to run <code>bear $BUILD_COMMAND</code></td>
</tr>
</tbody>
</table>

- **`compile_commands.json`** — compilation database, compilation commands for libraries and/or executables
- **`link_commands.json`** — linkage database, linking commands for libraries and executables (specific to UTBot)
Prepare KLEE run

```c
int klee_entry__main_abs__wrapped(int utbot_argc, char ** utbot_argv, char ** utbot_envp) {
    int val;
    klee_make_symbolic(&val, sizeof(val), "val");
    klee_prefer_cex(&val, val >= -10 & val <= 10);
    //////////////////////////////////////////////////
    int utbot_result;
    klee_make_symbolic(&utbot_result, sizeof(utbot_result), "utbot_result");
    int utbot_tmp = abs(val);
    klee_assume(utbot_tmp == utbot_result);
    return 0;
}
```
Prepare bitcode and run KLEE

KLEE patches

- **Speed**
  - Pruning The Recursive States*
  - Weakest Precondition in Symbolic Execution

- **Code coverage**
  - Floating-point Support
  - Complex Test Input Generation*
  - Detection of Undefined Behavior*

* presented at KLEE Workshop 2022
Google Test generation

1  v TEST(regression, abs_test_1) {
2      // Construct input
3      int val = 0;
4      // Expected output
5      int expected = 0;
6      // Trigger the function
7      int actual = abs(val);
8      // Check results
9      EXPECT_EQ(expected, actual);
10     }

Test generation: Problems & Solutions

```cpp
#include "lib.c"

int abs_lib_c(int x) {
    return abs(x);
}

namespace UTBot {

    /*
     * Types definitons
     */

    extern "C"
    int abs_lib_c(int x);

    static int abs(int x) {
        return abs_lib_c(x);
    }
}

namespace UTBot {

    TEST(regression, abs_test_1) {
        int actual = abs(-10);
        EXPECT_EQ(10, actual);
    }

    TEST(regression, abs_test_2) {
        int actual = abs(2);
        EXPECT_EQ(2, actual);
    }

    TEST(error, abs_test_3) {
        abs(-2147483648);
    }
}
```

Wrapper abs_wrapper.c

Test header tests/lib.h

Test file tests/lib.cpp
Compile and run tests
Conclusion

Results

● Built UTBot for C — tool for auto tests generation
● Complemented KLEE with a user-friendly interface
● Improved KLEE functionality: speed & coverage patches

Future work

● Full C++ support
● CLion integration
● CI integration
References

- UTBot (UnitTestBot) project website
  https://www.utbot.org

- UTBot C/C++ project page on GitHub
  https://github.com/UnitTestBot/UTBotCpp

- UTBot C/C++ documentation
  https://www.utbot.org/docs/cpp/general/home