COM S 413/513: Homework 2 [written] - Constructing Control Flow Graphs

September 5, 2023

Learning Objectives:

In this homework, students will

1. exercise and understand terminologies related to control flow graph and paths
2. * gain hands-on experience on generating a CFG using real-world program analysis tool LLVM Compiler Infrastructure (Extra Credit)

Instructions:

1. Total points: 15 + 10 (Extra credit) pt
2. Early Deadline: Sep 6 (Wed) 11:59PM
3. Deadline: Sep 8 (Fri) 11:59PM
4. How to submit:
   - Submission **with** extra credit question: Create a zip containing the following files and upload it to Canvas:
     - PDF with answers to Q1.
     - Source code for Q2. Any build/run instructions should be placed in a "README" at the root of the source code folder.
     - CFG graph as a ".png" file produced using your code. Please put this in a folder called "Output" within your source code folder.
   - Submission **without** extra credit question: Create a single PDF with answers to Q1 and upload it to Canvas.

Questions:

1. (15 pt) Perform control flow analysis (manually) and contruct an ICFG for the program below:

```
1  bool check_limit(int marks) {
2    if (marks >= 0 && marks <= 100)
3      return true;
4    return false;
5  }
6
7  bool check_pass(int marks) {
8    if (!check_limit(marks)) {
9      return false;
10   }
11   if (marks >= 80) {
12     return true;
```

```
13   } else if (marks < 80) {
14     return false;
15   } else {
16     return false;
17   }
18 }
19
20 int main() {
21   int marks = 30;
22   while (marks < 90) {
23     check_pass(marks);
24     ++marks;
25   }
26   marks += 50;
27   check_pass(marks);
28 }
```

(a) (3 pt) Draw a call graph

(b) (4 pt) Draw an ICFG

(c) (4 pt) Report two realizable paths

(d) (4 pt) Report one infeasible path, and explain why it is infeasible

2. **Extra Credit** (10 pt) Automatically generate a CFG for the program below:

```
1  int base_power(int base, int power) {
2    if (power != 0) {
3      int tmp = base_power(base, power - 1);
4      int tmp2 = base;
5      for (int i = 1; i < tmp; ++i) {
6        tmp2 = base + tmp2;
7      }
8      return tmp2;
9    } else
10     return 1;
11 }
```

(a) (6 pt) Code for generating a **source code level** CFG using LLVM

(b) (4 pt) CFG output as a ".png" file for the above program

You are welcome to use APIs, like `viewCFG`, provided by LLVM/Clang to generate the CFG graph.

Appendix

Useful links for getting started with LLVM

- Setting up Clang tooling: Needed for generating source level CFG
- Tutorial for standalone Clang tools
- Recurvise AST Vistor Clang Tutorial
- CFG Class Reference

CFG Example

For the code below:

```
1 int sum() {
2   int n = 5, sum = 0;
3   for (int i = 1; i <= n; ++i) {
4     sum += i;
5   }
6   return sum;
7 }
```

The source code level CFG, using built-in APIs would look as shown below