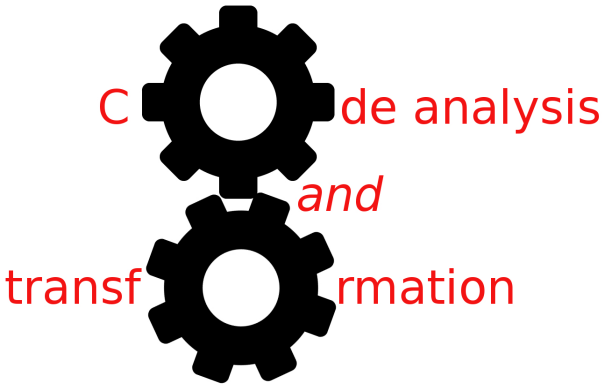

Homework H0

1 Description

Write an LLVM pass starting from the template available in Canvas (CAT-c.tar.bz2). The goal of your first pass is to print every function being compiled. Specifically, for each function compiled, you have to

1. print to standard error the name of the function being compiled
2. print to standard error the body of the function being compiled.

2 Develop Your Compiler

Develop To develop your compiler, extend `CAT-c/catpass/CatPass.cpp`.

Build and Run Follow the instructions in `CAT-c/README` to build and run your compiler.

Test To test your compiler has been properly installed:

```
$ cat-c --version
```

The output needs to include

```
clang version 8.0.0
```

Now you are ready to test your work.

3 Testing Your Work

Run all tests Go to `H0/tests` and run `make` to test your work.
The following output means you passed all tests:

```
SUMMARY: 2 tests passed out of 2
```

If you didn't pass a test, then the output will include the names of all tests that have failed.

Run a test You probably want to figure out why you failed a test. To do so, go to such test (let's assume test0 failed):

```
$ cd H0/tests/test0
```

Now, compile a program

```
$ make clean ; make
```

Check the output generated by your pass against the oracle output:

```
$ make check
```

and follow the instructions printed in the output to debug your work.

A good advice `H0.tar.bz2` includes a few programs you can use to test your work as well as their Makefile that shows how to invoke `cat-c` to generate the output file `compiler_output`. Read these makefiles to become familiar with `cat-c` and `llvm` tools.

The correct output of a test is stored in its subdirectories that start with the name `output`. Because different platforms might generate different compilations, in an `output` directory there are multiple files, one per platform. You need to match at least one of them. When you will debug a test, read these output files to understand what you should generate and compare it with your current output.

4 LLVM API

This section describes the set of LLVM APIs I have used in my H0 solution. You can choose whether or not using these APIs.

- Method `getName` of the class `Function`
- Method `print` of the class `Function`

5 What to submit

Submit via Canvas only your CatPass.cpp

6 Homework due

4/10 at 2pm

Good luck with your work!