CSE 504: Assignment 1
LLVM IR and Pass Construction
Due date: Thursday, September 27

In this assignment, you will get familiar with the LLVM intermediate representation (IR), and write a basic LLVM pass. This assignment is intended to help you understand the IR and give you some basic ideas about how code optimization works in the LLVM framework, in order to be prepared for the assignments coming later.

To complete this assignment, you will perform the work described below. The report and the files are to be made available to the TA for evaluation by midnight on the due date. Do **not** delete interim versions of your files or anything else that documents how you have performed your assignment tasks.

Here is what you should do:

For this work you will use your account on Seawulf. You should have already installed LLVM/Clang and tested it with CG before beginning this work.

The things you need to accomplish are listed as follows.

1) Use Clang to compile CG into the LLVM IR form, and study the output IR. The LLVM IR plays a central role in the LLVM code analysis and optimization, so it is important to have a good idea about it in order to complete the remaining assignment work. You should be able to relate the output IR to the C source code well as a result.

Also, compare the output IR under different optimization levels. Document the difference that you observe in the assignment report.

2) Study the concept of an LLVM pass, which is the means to analyze and optimize the LLVM IR. A detailed tutorial is available at https://www.cs.cornell.edu/~asampson/blog/llvm.html. Please read through it carefully. As a result, you should know how to write an LLVM pass.

3) Write a new LLVM pass to analyze the IR of programs to collect information as follows about a program. The pass should be able to print out the number of memory access instructions, branch instructions, and arithmetic instructions, in each function in the source code. Use CG as a case study to test your pass. You are encouraged to use additional programs as test cases.

Along with the LLVM pass you write, please also submit a report for your work. Your report should describe the IR difference under different optimization levels, the implementation of the instruction counting pass, the output results of your pass for all test cases used. It should be no more than 3 letter-size pages in length. **Upload your report**

in pdf or word format to Blackboard by midnight on the due date. We might ask you to upload additional files. If so, it will be announced on Blackboard.

Grading:

Grades from A through F will be assigned. Grades will be based upon the correctness of your work, completeness of your responses, and the overall quality of your work. Please do your own work: do NOT copy code or text from other class participants or from any other source.