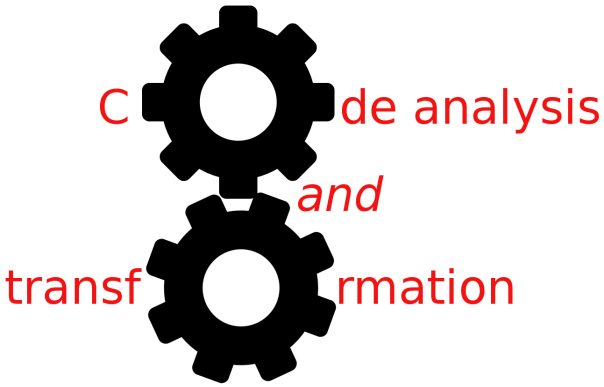

Homework H7

1 Description

Write an LLVM pass starting from the code you have developed for H6.

The goal of this assignment is to make your pass inter-procedural. In more detail, your pass is composed by a code analysis (reaching definition) and two transformations (constant propagation, constant folding). The former can become inter-procedural by propagating definitions through the call graph. The transformations can become inter-procedural by inlining and/or cloning functions whenever is necessary to unblock more propagations.

2 Testing your work

`H7.tar.bz2` includes some examples of C programs with multiple functions.

Your pass will be invoked until a fixed point is reached. In more detail, the bitcode file generated by your pass (i.e., program_optimized.bc) is checked against the input one. If they differ, then your pass will be invoked again to further modify the bitcode previously generated. This will continue until your pass does not modify the bitcode given as input.

Your pass cannot save a state. For example, you cannot create a file where you store what you have performed during the previous invocations.

3 LLVM API and Friends

For this homework you need to use some of the new APIs used in LLVM_callgraph.tar.bz2

4 What to submit

Submit via Canvas the C++ file you've implemented (CatPass.cpp).

For your information: my solution for H7 added 202 lines of C++ code to H6 (computed by `sloccount`).

Good luck with your work!