CSE 504: Assignment 3
Nested Loop Interchange
Due date: Monday, December 10

In this assignment, you will implement an LLVM pass to interchange the order of perfectly (or tightly) nested loops. Loop interchange is a very useful loop optimization which can be used to enhance data locality, help task scheduling, etc. This assignment is intended to give you some insight about real world compiler optimization.

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 may use the results of your 2nd assignment.

The following assumptions can be made for this assignment: 1) you should not use any optimization flags while compiling your test codes, 2) only C for loops are considered.

The things you need to accomplish are listed as follows. See further below for full instructions on what to output.

1) Enumerate the loops in your input code. Identify perfectly nested loops (with more than one loop level) in input code using a function pass as in your previous assignment..

2) For each pair of nested loops identified, analyze whether the loop index variables depend on each other. Loops with dependent index variables are not interchangeable. Output the result, indicating the pairs of perfectly nested loops you have found and whether their loop variables are dependent. For example:
Nested loop pair 1 and 2: index variables are dependent
Nested loop pair 4 and 5: index variables are independent

Note that your pass must be conservative in the sense that it should not identify a pair of loop nests as having independent variables when they are, in fact, dependent.

3) For all pairs of nested loops that you have **not** identified as being dependent: Interchange the loops and output the IR for the modified version of a given source code. Note that this will not always result in correct code, as you have not performed any dependence tests. LLVM has a polyhedral analyzer to test for dependences, but that is out of scope for this assignment.

Generally you will need to modify basic blocks and instructions to implement loop interchange. For example:
for(i=0; i<10; i++) { for(j=1; j<3; j++){ a++; } }, is converted to
for(j=1; j<3; j++) { for(i=0; i<10; i++){ a++; } }

4) Write several test programs which includes at least 3 pefrectly nested loop pairs with different kinds of index variables, as well as some loop nests that are not perfect. Test the implemented pass with these programs.

5) How might you go about determining the profitability of loop interchange on LLVM IR (assuming you have available the results of dependence tests)? You do not need to write any code to do so, but provide your thoughts in your assignment report.

You will only need to compile your test program to the IR (i.e., using -c -S -emit-llvm), and thus a complete program is not required (i.e., the main function is not necessary). Do **NOT** use existing LLVM passes that can directly change loop order. You are allowed to use the results of other LLVM passes.

As always, do **not wait** until the due date to do your work: otherwise the machine may become very full around the due date.

You will need to submit your pass code, intermediate files generated and test programs, as well as an assignment report, as a tarball,. Please also leave appropriately named files in an appropriately named directory in your course account in case we need to inspect them. Your documentation should provide any explanations needed. Please upload the files on blackboard by midnight on the due date.

Your report should serve as a design document which includes both high level and medium level description of your pass implementation, as well as a brief description of your test programs and test results. It should be no more than 4 letter-size pages in length. Do not write more than a half page for 5). Submit your report in pdf or word format. Upload your results as requested by the TAs.

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.