CS6383: Assignment 2
Loop Reversal
Due Friday November 15th, 2019 at 11:59 PM

Introduction In this assignment, you are required to write a transformation pass in LLVM (version 9.0) to reverse a loop in the LLVM-IR without changing its semantics. For this purpose, you have to change the usage of induction variable accordingly, in order to preserve the correctness.

For example, loop iterating from **x** to **y** should be transformed to loop from **y** to **x**

```
for (int i = 0; i < n; i++) {
  a[i] = i+n;
}
```

Listing 1: Sample Input

```
for(int i=n-1; i>=0; i--) {
  a[n-i] = n-i+n;
}
```

Listing 2: Sample Output

Assumptions

- IR is in SSA form.
- Each loop has a single induction variable.
- The induction variable is not modified inside the loop body.
- The induction variable is of integer type.
- The update statement can consist of arithmetic operations - `add, sub, mul, div` - by a *loop invariant* (constant/variable).
- The control never leaves the loop from inside the loop body - there is no break; or return; statements.
- The condition checked in the loop is a single condition on the induction variable with one of these operators : <,<=,>,>=.
- Both the lower and upper bounds are *loop invariant*, i.e., their values do not change inside the loop.

Implementation Guidelines Create a new directory *LoopReversal* in lib/Transforms/ folder of the LLVM source tree. All your implementations should be in this directory as this directory will be part of your submission. You need to put header files under include/llvm/Transforms/LoopReversal directory. Register your pass as loop-reversal and the module as `LoopReversal.so` so that it can be run using scripts.

opt -load $LLVM_BUILD/lib/LoopReversal.so -loop-reversal test.ll should work

Strictly follow the below directory structure while submitting your code. Submit new and modified files only.

```
./
├── llvm
│   ├── lib
│   │   └── Transforms
│   │       └── LoopReversal/
│   └── include
│       └── llvm
│           └── Transforms
│               └── LoopReversal/
├── test/
└── README
```

Testing You are supposed to write non-trivial test cases to test your pass. The test cases should vary in complexity from simple to more complex ones. You need to submit at least 5 test cases in C/C++.

Submission Your submission should be a tar.gz archive with name *Asgn2_ROLLNO.tar.gz* containing the source code, header files, test-cases directory and a README file in the specified format. The README should mention all the materials that you have read/used for this assignment including LLVM documentation and source files. Mention the status of your submission, in case some part of it is incomplete. You can also include feedback, like what was challenging and what was trivial. Include files that you have added or modified. Do not include the binaries in your submission.

Evaluation We will test your code using a set of 50 test cases and you will be evaluated based on the number of test cases passed. Also proper commenting, code formatting along with well structured code will be part of the evaluation and fetch you more points.
NOTE: Non-compliance to the submission guidelines will attract strict penalty.