E0255 (Jan-Apr 2023)

ASSIGNMENT 1

[Will be updated in place. Please check this file for the latest.]
https://www.csa.iisc.ac.in/~udayb/e0255/2023/asst-1.txt

Deadline: Wednesday Mar 1, 2022 5pm.

Design and implement an LLVM pass that computes "anticipated expressions" using dataflow analysis and performs code hoisting using the computed information to reduce code size and repeated execution of expressions. The pass will be tested by checking if anticipated expressions have been eliminated from the IR.

Use the official LLVM git repository at commit id f08d86fc7f4479d5f44d75c720201334682075b8 (Feb 5, 2023) as the base to implement your pass.

The dataflow analysis computation needed for anticipated expressions should be implemented as part of the assignment as opposed to being reused from elsewhere.

The pass should be named `hoist-anticipated-expressions`. The pass should not emit any debug output on normal runs. The `IN` and `OUT` sets for anticipated expression computations should be emitted under LLVM_DEBUG; these will however not be checked and matched for correctness; only the pass's IR output will be tested.

Reference test cases (*.ll) will be provided. Your assignment will be evaluated on these reference test cases as well as additional similar test cases. You are free to generate test cases on your own for additional testing during development by using clang on simple C snippets to generate LLVM IR.

Reference test cases can be found here:
https://www.csa.iisc.ac.in/~udayb/e0255/2023/hoist-anticipated-expressions.ll

The test case file can be run with lit.

We recommend using CSA's clserv (can be accessed remotely) for all assignment work.

WHAT TO SUBMIT AND HOW

Please submit your assignment as: (1) a single git commit patch file that should apply on the base version mentioned above, and (2) a README.txt. Once you have created a single commit to contain your implementation, such a patch can be generated using:

```
$ git format-patch HEAD~
```

The README.txt can include any short notes or observations you made or to document any limitations of your implementation that make it fall short of the assignment requirements.

Please make sure you use the same coding style and standards as the rest of LLVM. The code should be formatted using clang-format as follows:

```
# After creating your commit (i.e., your commit is HEAD), format it as follows.
$ git clang-format HEAD~
```

Whitespace errors on the commit can be checked with: `git diff --check HEAD~`.

Submissions that aren't formatted per LLVM guidelines will require a resubmission.

The git patch and README.txt should be emailed to udayb@iisc.ac.in by the deadline mentioned on the top with the subject "E0255 Asst-1 submission".

REFERENCES

1) LLVM documentation: https://llvm.org
2) LLVM coding style and standards: https://llvm.org/docs/CodingStandards.html
3) https://llvm.org/docs/ProgrammersManual.html (Important reference on the data structures to use)