

LLVM Project - Nature Loop Detector

Feb 21st, 2018

1 Description

For this assignment, you need to use data flow analysis to find loops in the LLVM IR code.

2 Implementation

For implementation, three steps described in the slides are needed as follows. Each step generates an output which will be used for grading.

1. **Dominance relation for basic blocks:** Provide a dump function to output the dominance relation. (Do NOT use the dominator analysis in the LLVM compiler.)
 - To calculate Dominance, you need to traverse CFG. Take a look at the class reference for functions and basic blocks to see how to find the entry block and how to find the successors or predecessors for a given basic block;
 - To find the name of basic blocks, look at the APIs such as *hasName()*, *getValueName()*, *setValueName()* provided in value class. **NOTE:** The function that gives the name for each BasicBlock is already given in the code template, all you need is to find a proper place to call this function.
 - You also need to know about some C++ data structures, such as vectors, map, pair, stack, queue. That will simplify your implementation.
2. **Back edges:** Provide a dump function to output all back edges.
3. **Natural Loops:** Provide a dump function to dump the set of basic blocks in the loop.

The output should first give the number of loops, the dominance relation and all back edges. For each loop, output a set of basic blocks in the loop. For example:

=====

Number of loops: 1

Dominance relation: BB1 → BB2, BB1 → BB3, BB2 → BB3

Back edge: BB3 → BB1

Basic blocks in the loop 1: BB1, BB2, BB3

=====

Write at least two test programs with at least one loop in each program.

3 Report

You are also required to write a report for this assignment. This report should include the title, your name, a brief description of the design and limitations, including what types of loops can be detected and what types of loops can not, and your test results. The report should be in PDF format and **NO MORE THAN 3** pages.

4 Submission Guideline

This assignment is **DUE ON midnight of Mar.9(Fri), 2018**. For submission, you have to prepare following files: NatLoop.cpp, your own test program(with Makefile is preferred) and YOURNetID-report.pdf. Put these files in a directory name YOURNetID-cs255-llvm and then archive it using the following command line, and submit on Blackboard.

```
tar -czvf YOURNetID-cs255-llvm.tar.gz YOURNetID-cs255-llvm
```

5% of the grade depends on the right submission.(correct name and correct files)

5 Late Submission Policy

Remember that you have a **OVERALL 2 DAY EXTENSIONS**. And after you run out of that, you will get **10%** off per day for late submission penalty.

6 Useful Documentations

1. **LLVM Programmer's Guide:** instruction on how to write a hello-world LLVM pass step by step. <http://llvm.org/docs/WritingAnLLVMPass.html>
2. **LLVM Programers Manual:** highlights of some important classes and interfaces available in the LLVM source-base (For example, how to iterate over basic blocks, how to iterate over instructions inside a basic block or function). <http://llvm.org/docs/ProgrammersManual.html>
3. **LLVM Language Reference Manual:** reference manual for the LLVM assembly language. <http://llvm.org/docs/LangRef.html>
4. **LLVM class reference:** reference for the interfaces of the classes needed. (For example instructions, basic block, functions) <http://llvm.org/doxygen/namespacellvm.html>
5. **Definition for a loop in the flow graph:** *Compilers: Principles, Techniques, and Tools (2nd edition)*, Chapter 8.4.5 P531.
6. **The loop identification approach:** http://www.cs.cmu.edu/afs/cs/academic/class/15745-f03/public/lectures/L7_handouts.pdf.