CS3423/CS6240: Mini Assignment #1
An Introduction to the LLVM Infrastructure, AST, IR and Compiler Options
Due Sunday August 19th, 2018 at 11:59 PM

Introduction This assignment is aimed at setting up and familiarizing you with the LLVM compiler infrastructure (`http://llvm.org`), which has lot of very interesting research modules with active development from many compiler communities across the world. You will also use it for the next couple of mini-assignments.

It aims to give you some familiarity with LLVM's C-family frontend Clang (`http://clang.llvm.org/`). More specifically, this assignment asks to you to do a *study* of the AST, IR and options of the LLVM compiler.

- **Download LLVM:** Obtain source code of LLVM and Clang. Official releases are available at `http://llvm.org/releases/download.html`, however please download the latest version(trunk) using git/svn `http://llvm.org/docs/GettingStarted.html#git-mirror`. Build LLVM and Clang from source. You can refer `http://llvm.org/docs/GettingStarted.html#compiling-the-llvm-suite-source-code`.
- **LLVM and Clang:** Understand the directory hierarchy of LLVM and Clang. Read the documentation *"Getting Started with LLVM"* at `http://llvm.org/docs/GettingStarted.html#directory-layout` for LLVM and `http://clang.llvm.org/docs/IntroductionToTheClangAST.html` for Clang.
- **AST Structure:** Use the options provided with Clang to view the LLVM/Clang-AST for some non-trivial programs. Understand the design of LLVM-AST to some level. Report your findings.
- **AST Traversals:** Read the tutorial to write AST visitors at `http://clang.llvm.org/docs/RAVFrontendAction.html`. Write a short note about the visitor mechanism to traverse the AST and do semantic analysis. (The visitor pattern is design pattern, a paradigm from software engineering which you are using in your ANTLR.)
- **Error messages:** Do a study of how the error messages are handled in LLVM source code, primarily the assert mechanism of LLVM. Report on the handling of some specific error messages. You need to compile LLVM with flag -DLLVM_ENABLE_ASSERTIONS=On to enable assertions.
- **LLVM-IR:** Do a study of the LLVM-IR. Print the IR for five non-trivial programs and study them. Submit a report on your study along with the generated *.ll* files.
- **Assembly language:** Generate the assembly language codes of some simple C/C++ programs. Understand how the C/C++ compilers *mangle* the names of various entities. Report your findings.
- **Compiler toolchain and options:** For a set of programs, go all the way; print the LLVM-IR, print the assembly code. Play with the various compiler frontend/middle-end/optimizer options. Report your findings.

- **Kaleidoscope:** Read the Kaleidoscope at documentation `http://llvm.org/docs/tutorial/index.html`. Report your findings.

You do not have to limit yourself to the above list. You are welcome to explore more options of LLVM. For example, you could (i) write some AST visitors to do simple semantic checking, like print the uninitialized variables, enforce style guidelines in the source code like Compass tool of the ROSE compiler does, (ii) use the Clang Static Analyzer for static analysis of some simple programs. You can also play more with the Kaledioscope. (This section is not mandatory.)

Submission You will be evaluated on a technical report on your study. You need not submit any code, but, if you wish, you can add the code listings in a separate appendix within PDF. The report has to focus on the Frontend, LLVM-AST (its design and its traversals), LLVM-IR, the LLVM compiler options, Kaledioscope aspects(optional), and finally the ASM code generated. The report should be a PDF, preferably generated from a lyx/latex file and named as MiniAsn1_ROLLNO.pdf.

Evaluation Your report should show a *good* understanding on each of the points asked. The usual rules of plagiarism apply to your report. In particular **do not** directly copy the material from manuals/websites. Your report should professionally refer the website(s) from where you downloaded the code and the manual(s) you referred to. Very good reports will fetch bonus points.