

CENG513 Midterm Exam - Spring 2024

Due date: 19.04.2024 23:00

Q.1 (40 Points) Consider the C function given in *prime_number.c* file, which checks if the given number is prime or not.

a) Attempt to generate hand-written LLVM IR for the function with no SSA restriction. You should find and write the IR instructions as we did for the *iterative factorial* function in the class (You can refer to [this video](#), which is already given in your lecture slides and the corresponding presentation if you want to recall the example). You can also refer to the [LLVM Language Reference Manual](#) for the specific IR syntax. Use [opt tool](#) to check the correctness of your LLVM IR. You will receive an error about SSA restrictions. As part of your submission, you must submit your IR with the name **is_prime_hand_v1.ll** and the screenshots of your opt command with the error message.

b) Fix the error (s) by modifying your LLVM IR: **1)** Using phi functions, **2)** Using alloca functions. Use opt tool to check the correctness of those versions. You will not receive any error. As part of your submission, you must submit your IRs with the names **is_prime_hand_phi.ll** and **is_prime_hand_alloca.ll**, and the screenshots of your opt command with no error message.

c) For the same function, generate LLVM IR with clang -emit-llvm flag. You are required to have two versions **1)** -O0 option, **2)** -O1 option. As part of your submission, you must submit generated IRs with names **is_prime_gen_O0.ll** and **is_prime_gen_O1.ll**.

d) For each LLVM IR (two hand-written and two generated IRs), generate the control flow graphs using opt tool ([dot-cfg flag](#)). As part of your submission, you must submit generated CFGs with names **is_prime_hand_phi.svg**, **is_prime_hand_alloca.svg**, **is_prime_gen_O0.svg** and **is_prime_gen_O1.svg**.

Compare IRs and CFGs by specifying/explaining the main similarities and differences.

Your answer to this question should include all the commands you used, outputs of your executions, and your explanations. You can either put the IR and CFG also in your answer or simply reference them.

Q.2 (40 Points) The given chapter from the book [lex & yacc](#) (*Ch1_lexandyacc.pdf*) provides an introduction to the lex and yacc tools. For the grammar given in Example 1-8, answer the following questions:

a) Construct a Context Free Grammar in a format (four tuples) we discussed in our lectures. Based on the grammar, and the corresponding Lexer given in the previous examples in the chapter, give successful derivations for one simple sentence (*simple_sentence*) and one compound sentence (*compound_sentence*). Show the parse trees for the derivations (you can draw it by hand or any visual tool if you prefer).

b) There is a left recursion for *compound_sentence* definition in the grammar. Show a sample derivation that becomes a problem for top-down parsers. Eliminate all left recursions in the grammar.

For this question, you do not need to understand all the syntax and the details of the tools. You also do not need to build or run the tools. You just need to understand the language given in terms of yacc/lex syntax.

Q.3 (20 Points) Watch the following talk: [Understanding the LLVM build](#), given at the 2023 LLVM Developer Meeting. Answer the following questions based on the video content:

- Give examples of LLVM projects and runtimes mentioned in the talk. What is the main difference to build them? Specify the problem given in the talk to build runtimes, and give a solution proposed in the talk.
- What are the ways to build LLVM faster? Explain each one briefly.
- What points do you think it was better to watch the video before building llvm in your first assignment?
- Which part of the talk is clear and unclear for you?

What to Submit:

- ✓ IR and CFG files for the first question in a zip file (q1_StudentID.zip).
- ✓ One pdf file including the answer for each question (q1_StudentID.pdf, q2_StudentID.pdf, q3_StudentID.pdf).

Important Note: You need to use **your own** words in the answer of the questions. Text-based answers will be checked for plagiarism using iThenticate software (<http://www.ithenticate.com/>). If you have any reference (paper, video, web site, friend discussion etc.) used in your answers, you must explicitly cite them in your submission.