EEL4837 – Programming for Electrical Engineers II

Excursion 1: Circuit Analysis Tool

Due Date: November 2, 2023, 4:00am

Submission Instructions: Submission must be through Canvas. You need to submit a single zip file which expands to a single folder containing he following: (1) source code for your tool, (2) a Readme file (named "Readme") explaining how to compile, execute, and run your code, (3) a set of netlist files that can be used as test inputs, and (4) any additional documentation that you wish to provide on the tool.

You will be graded on (1) correctness, i.e., printing correct output values given "legal" inputs, 60%; (2) robustness, i.e., ability to tolerate different corner cases or unexpected inputs, 15%; (3) efficiency with respect to time complexity and storage requirement, 10%; (4) elegance, 10%; and (5) documentation 5%.

Please note the following points:

1. Your tool should be written in C++.
2. Programs that do not compile on an ECE Linux server (e.g., linux.ece.ufl.edu) generally receive zero credit. However, if there is a specific reason for your file to not work on the server (such as a library mismatch) we will consider those situations and handle them on a case-by-case basis if you let us know the problem in advance.
3. You should not use stdlib or use any functions from C++ STL. If for some reason you want to use a specific library (e.g., for visualization or debugging) you must get written approval from the instructor in advance.

The goal of this excursion is to implement an elementary interactive circuit analysis tool (like SPICE) that reads a circuit and computes the current and voltage across the different wires. For the purpose of this excursion, it will be okay to assume that the circuit contains only voltage sources and resistors (e.g., no capacitor and no inductor). The tool, when run, should initially print out a message like the following:

Select one of the following options:

A. Read a new netlist
B. Compute current values for the current netlist
C. Compute voltage values for the current netlist
D. Exit

On entering the appropriate menu option, the tool should provide a sub-menu for that specific option. For example, for reading a new netlist, your tool should provide an option prompting for a netlist file.

At a minimum, your tool should support the following features:

1. Netlist filenames can be provided either as a relative path from the current directory or as a full path.
2. For computing currents, the user can give a single net name or a list of them as comma-separated list. For the circuit shown in class, the user can (for example) ask you to find between R1 and R2 in a single command by giving a list like this: [1, 2], [2, 0].

3. Your tool should enable calculation of the voltage at any node or the voltage drop between a pair of nodes. For example, the user may provide a list like this: 1, [1, 2] for the circuit shown in class.
4. Your tool should at the minimum do something reasonable in situations where (1) the netlist filename is wrong; (2) the netlist file is malformed; and (3) the user asks the tool to compute voltage or current across the wrong set of junctions (for example, a junction value that doesn't exist). What is "something reasonable" will be up to you, as long as it seems reasonable to us. For example, the tool crashing under such circumstances is not reasonable.

If you can handle large circuits, e.g., through sparse matrix representations, you will get bonus points on efficiency (but you get efficiency bonus points only if your program is both correct and robust).