

Make sure to include testing and benchmarking code as well as the sample results, measurements, plots and conclusions into the report.

TODO

REQUIREMENTS

- ☒ The library should overload operators to support the reaction rule typesetting directly in C++ code.
- ☒ Provide pretty-printing of the reaction network in
 - ☒ a) human readable format
 - ☒ b) network graph (e.g. Fig. 4).
- ☒ Implement a generic symbol table to store and lookup objects of user-defined key and value types. Support failure cases when a) the table does not contain a looked up symbol, b) the table already contains a symbol that is being added. Demonstrate the usage of the symbol table with the reactants (names and initial counts).
- ☐ Implement the stochastic simulation (Alg. 1) of the system using the reaction rules.
- ☒ Demonstrate the application of the library on the three examples (shown in Fig. 1, 2, 3).
- ☐ Display simulation trajectories of how the amounts change. External tools/libraries can be used to visualize.
- ☐ Implement a generic support for (any) user-supplied state observer function object or provide a lazy trajectory generation interface (coroutine). The observer itself should be part by the user/test program and not part of the library. To demonstrate the generic support, estimate the peak of hospitalized agents in Covid-19 example without storing an entire trajectory. Record the peak hospitalization values for population sizes of NNJ and NDK.
- ☐ Implement support for multiple CPU cores by parallelizing the computation of several simulations at the same time. Estimate the likely (average) value of the hospitalized peak over 100 simulations.
 - [-] Implement unit tests (e.g. test symbol table methods, their failure cases, pretty-printing reaction rules, etc).
- ☐ Benchmark and compare the stochastic simulation performance (e.g. the time it takes to compute 100 simulations a single core, multiple cores, or improved implementation). Record the timings and make your conclusions.
- ☐ The source code in original form (without binary/object/debug/release files). The code is evaluated by the detailed requirements implemented and thus should be commented with references to them (e.g. // R7).
- ☐ The source code listing in PDF report. The report should start with student name and a short documentation about what compiler, C++ standard and compilation options are used to build the project. A brief command line, a Makefile or a CMakeLists.txt is enough.
- ☐ Include experiment data into the report: tables, plots, figures, screenshots and conclusions!