

Compiler project tasks — part 7, due Wednesday 12 December

The goal in this phase is to simplify the abstract syntax tree of the program further and generate final code.

Given input program `primes.m`, the output files should be named `primes.simplified2` and `primes.c`. These files should be generated by pretty-printing the transformed abstract syntax. Test on all programs we have used so far.

Final simplification stage

Make evaluation order in expressions explicit, by converting all complex expressions into sequences of assignment statements, with only a single primitive operation on the right-hand side. Introduce temporary variables to hold intermediate results. Temporary variables should only be of the primitive types, boolean, integer, and real, and a primitive address type for references.

Generating code

Generate C code and inspect it. Compile the resulting C program using `gcc` and test. You do not have to provide support for garbage collection—it suffices to translate each `NEW` into a `malloc`.

Optional exercises

Study C-- (<http://www.cminusminus.org/>), discover how far that project has come along, produce C-- code that can be consumed by a C-- back-end. Is there a back-end with support for garbage collection? If so, add garbage collection (talk to the instructor first).