Assignment III: Pseudocode Emulation

Nothing to hand in.

Implement an emulator for your pseudocode formal syntax.

An *emulator* of a program is a different program, usually in a different language, that does the same thing as the target program. Emulators are often built for hardware: a software module performs the same functionality as the target hardware, but in software. Software emulation is usually much slower. Another example is programs which simulate, say, a Windows environment on a Mac OS; these programs emulate Windows on a Mac.

In Assignment II, you designed a language fragment and formalized it with BNF or another structuring tool. In this assignment, you will implement the syntax of your language. (You may elect to use a different fragment, or a completely different formal specification.)

The assignment is simple if there is a one-to-one correspondence between your specification and some existing language. For example, if you specify a WHILE construct, then the specification language can translate directly to WHILE is some existing language like C. What gets tricky is verifying that the correspondence holds for all cases and for all implementation strategies.

You can view the assignment as one of *metaprogramming*. You will be writing a program in say language A. This program takes another program in language B (the BNF spec for example) as input and translates it into a third program as output which is in language A but does the functionality of the input in language B.

A good emulator will include a lexical scanner and a syntactic parser to assure structural properties of the output.