CS320 – Focused on building a compiler and certain necessary programs to make the whole system of programs run appropriately.

This course focused on teaching us the different programs needed to process programming languages. Most of the course was done in python. Each assignment tackled a different concept. The following files are included throughout the assignments: a compiler file, an interpreter file, a machine file (in which we worked with our own version of assembly language), a parser, a validation program, and a tokenizing file (hw1.py).

The tokenizing (hw1.py) program separates a defined grammar by white space before forwarding the result to the interpreter.

The interpreter program would then read in the appropriate terms, numbers, and values and output the appropriate syntax for operations to be passed through to the compiler. The grammar was split into “numbers,” “variables,” “terms,” “formulas,” and “programs.” Some terms and formulas included are: +, -, xor, not, <, >, =, (, ), true, and false. The interpreter also handles “programs.” The produced syntax for these allow for assigning values to variables, printing results, performing if operations, and while operations.

The compiler program is fed the syntax that was defined in the interpreter and then translates the syntax to our defined assembly code. The compiler can compile and produce assembly for the defined expressions, formulas, and programs.

The machine program takes the defined assembly language from the compiler and performs the actual defined operations.

The validate program helps to ensure the output of the compiler is of correct format. (I.E. a formula is of the form defined in the previous programs.)

The midterm is a simplified and complete version of the above code. It includes a compiler, interpreter, a parser, a validation program, and an addition program called analyze. The analyze program handles type-checking.