

# Lambda Calculus Constructions

Greyson Knippel

CPSC 326-01 — Gonzaga University, compiled with L<sup>A</sup>T<sub>E</sub>X

Professor Johnson

February 24, 2026

**to start off:**

## 1. von neumann diagram

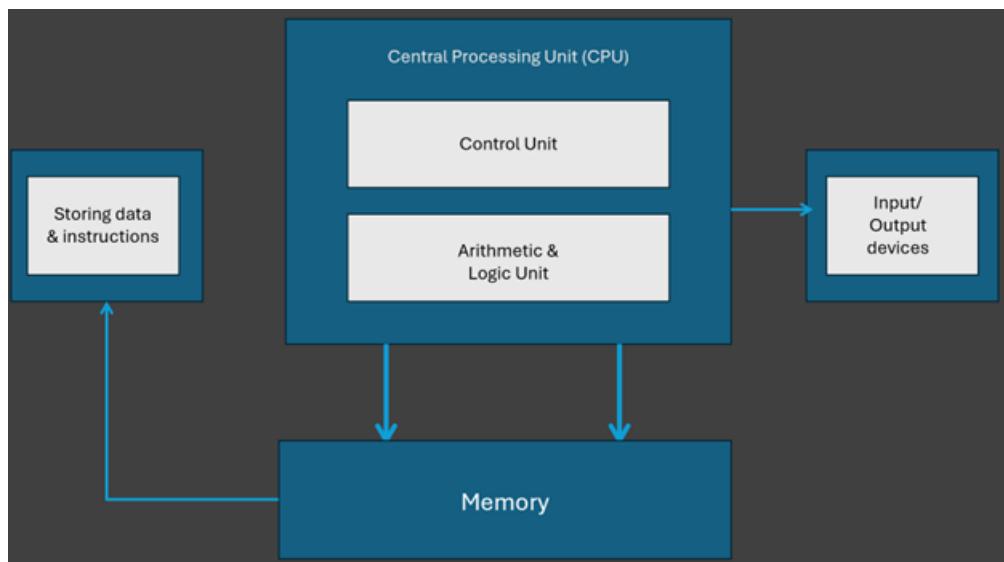


Figure 1: von neumann diagram

## 2. two reasons to study programming languages and compilers

first, understanding compilers is very useful for learning how many different programming languages are implemented. it gets us more familiar with their own paradigms. second, learning programming languages helps us realize how most other programming languages work. knowing how different programming languages work can help you figure out a confusing thing in another language. for example, learning Lean has helped me understand what some rust functions do.

### 3. compiler drawing

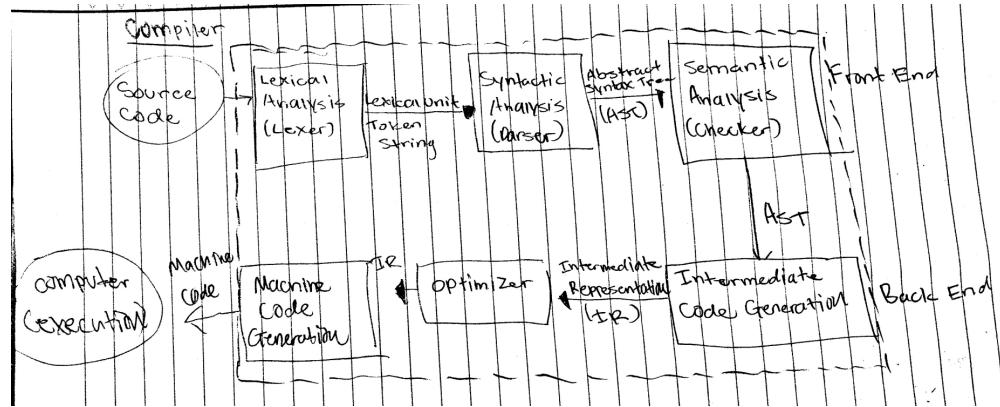


Figure 2: compiler drawing

### 4. compiler vs. hybrid interpreter

the difference between a compiler and a hybrid interpreter is that the compiler goes through the entire process of flowing the source code into the lexer, then performing syntactical analysis, using an abstract syntax tree to go into semantic analysis, and another AST to get intermediate code generation. but instead of going to the optimizer and generating machine code, the hybrid interpreter just executes the intermediate representation directly. so it does the same front-end operations, but compiles and interprets the IR rather than generating native machine code.

### 5. hybrid interpreter vs. tree-walk interpreter

the difference between a hybrid interpreter and a tree-walk interpreter is that the tree-walk interpreter goes from the lexer, to the parser, and then the checker, but then immediately uses the abstract syntax tree to reach the interpreter. it does not do any intermediate code generation like the hybrid interpreter does.

### 6. tree-Walk Interpreter vs. pure interpreter

the difference between a tree-walk interpreter and a pure interpreter is that the pure interpreter goes straight from source code and interprets it, reading and executing directly. it does not do any lexing, parsing, or checking, it does not use an abstract syntax tree, it just reads the source code and directly interprets it.