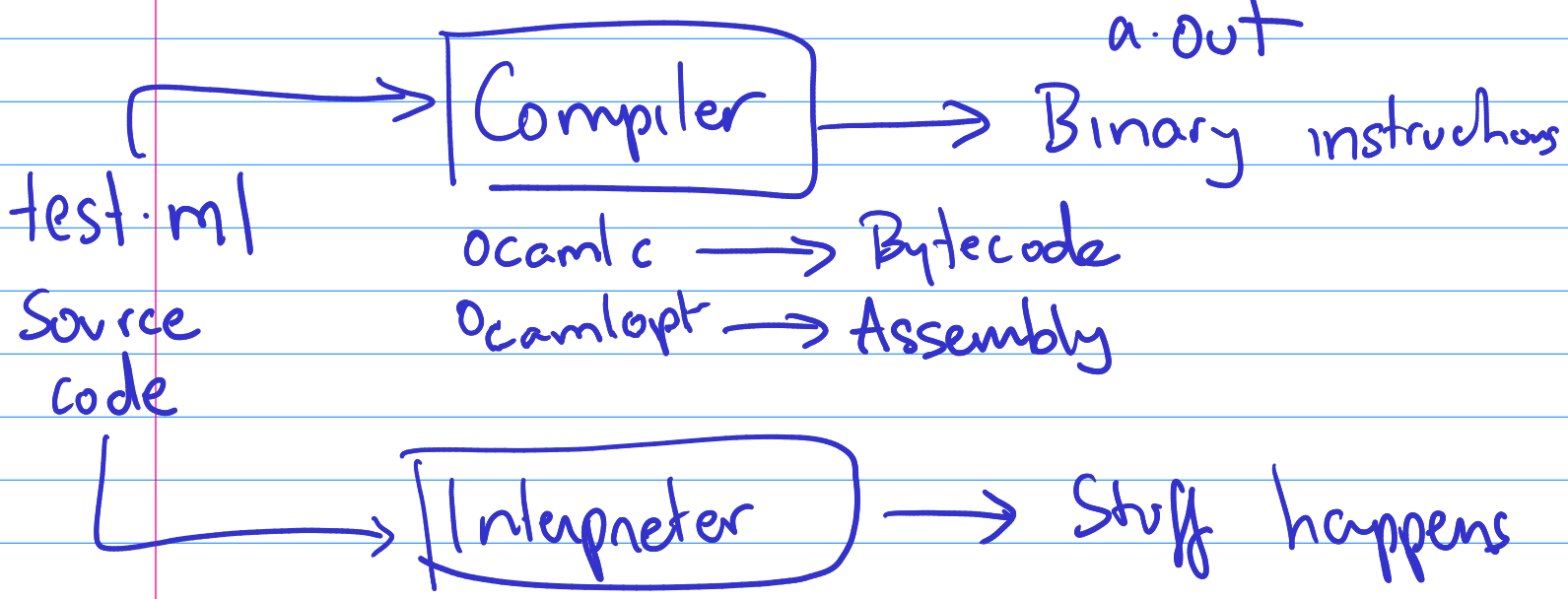
Unit 2: Implementing a Language Interpreter



"Hosted" Implementations

Python / CPython ——— C

Java / JDK (Hotspot) ——— C++

JavaScript ——— Firefox (Spidermonkey)

JavaScript ——— Chrome (V8)

"Bootstrapped" implementations

Compiler is written its own language

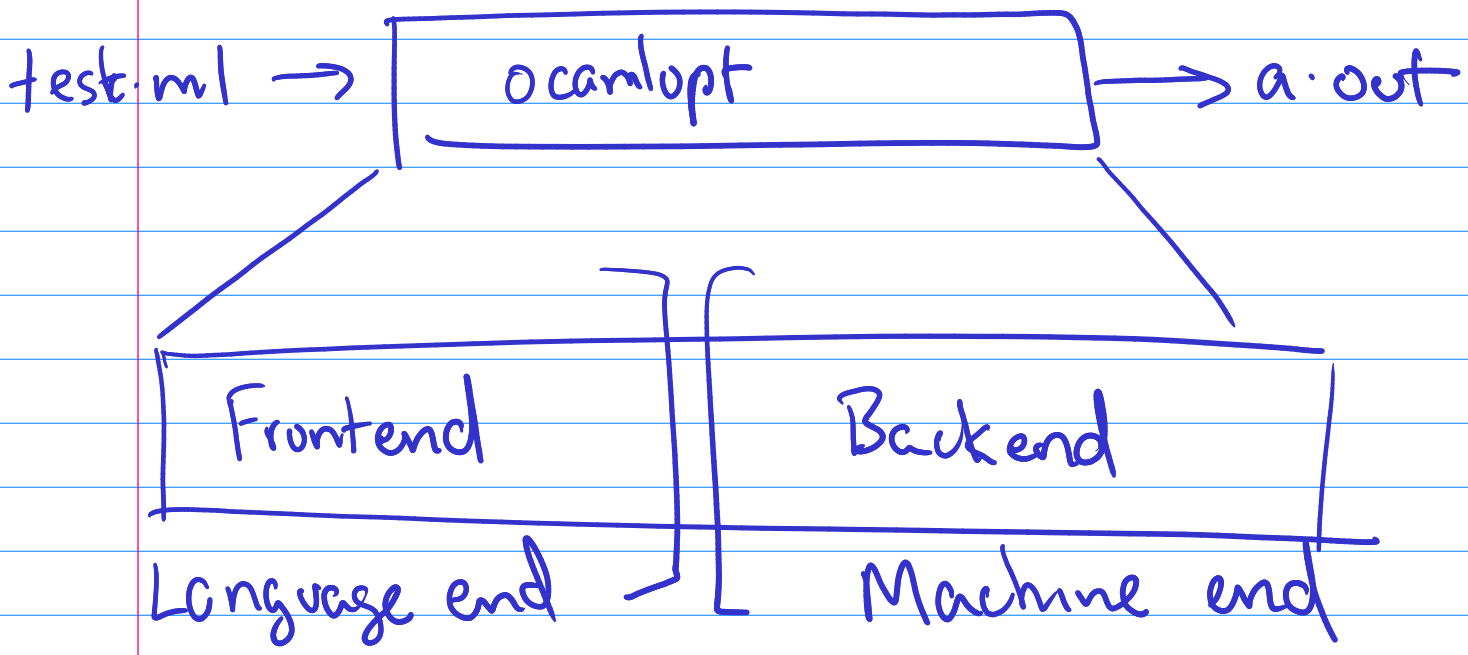
gcc ——— C++

Clang / LLVM ——— C++

Rust $\xrightarrow{\text{Rust}}$ LLVM IR $\xrightarrow[\text{C++}]{\text{LLVM}}$ Assembly

Bootstrapped Interpreter

Pypy RikesJVM



Syntax analysis

Type system

Optimization

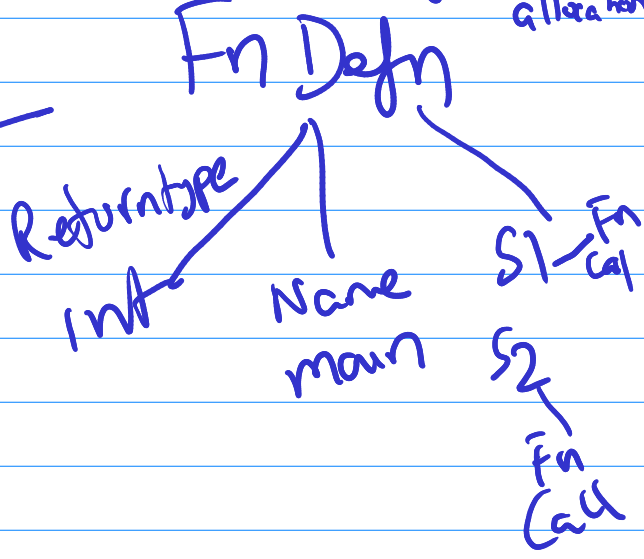
What assembly instructions to emit

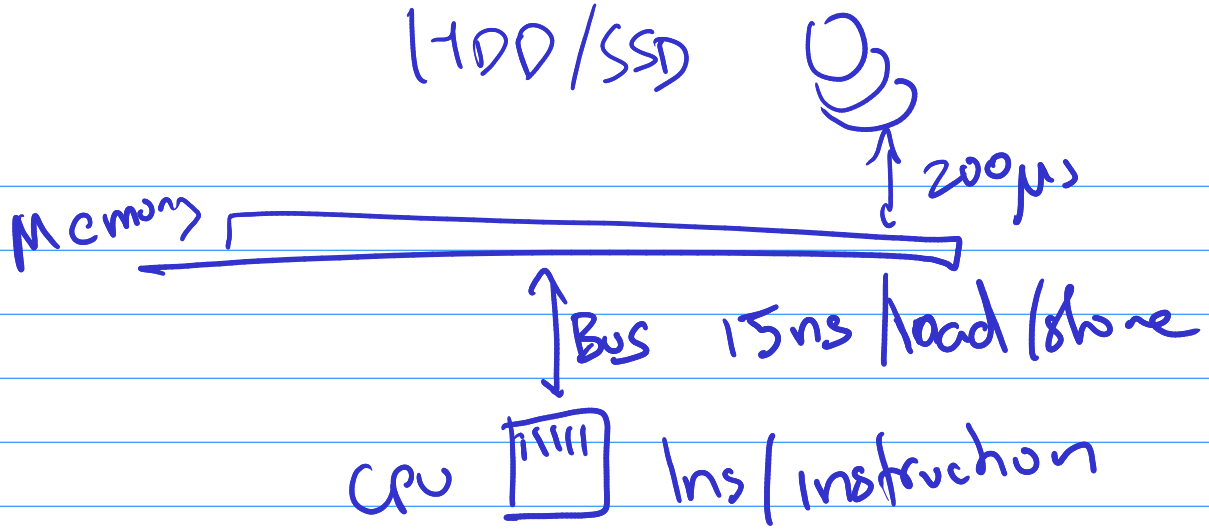
Optimization

Instruction selection

Register allocation

```
int main ()
    print Hello
    print World
```





add 43 rax

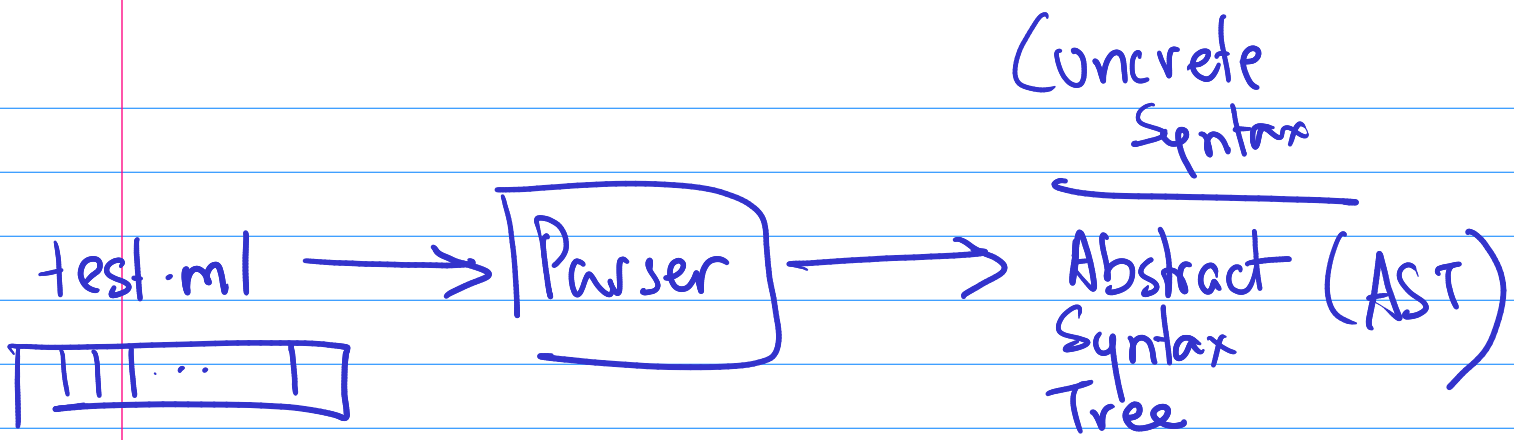
store rax 0x4827

Runtime

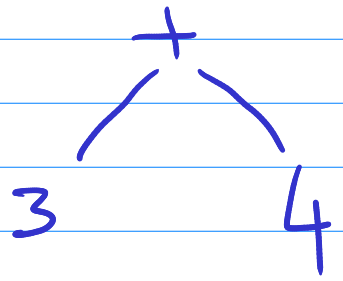
Libraries

System calls

Garbage collection



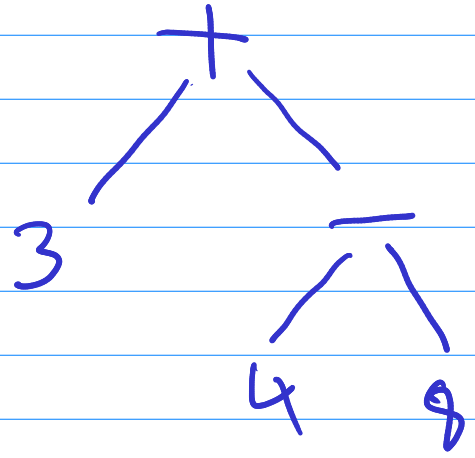
"3 + 4" ⟶



3 + (4 − 8)

char list

string



Parsing

Language specification?

~~Grammar~~

"What do well formed programs look like?"

Well-formed	Ill-formed
3 + 4 - 5	3 + - 4
3 + (4 - 5)	3 - + 4
3 - (4 + 5)	
⋮	

Expr ::= Integer Literal

| Expr '+' Expr

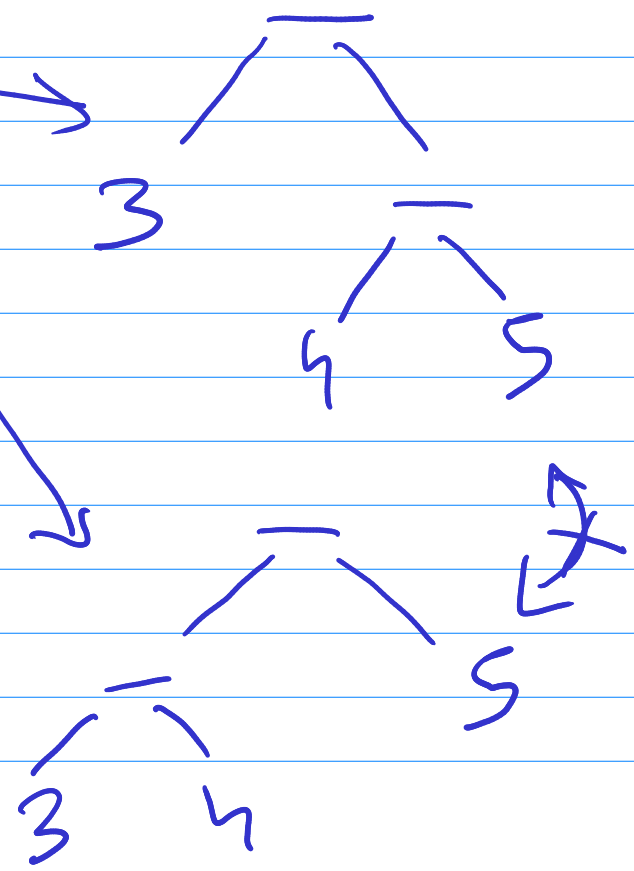
| Expr '-' Expr

| '(' Expr ')'

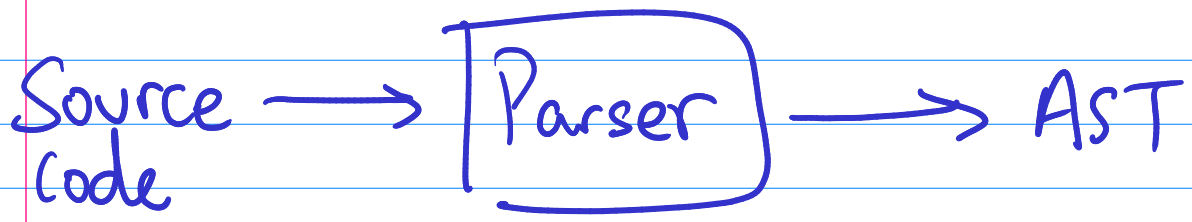
"3-4-5"

3-(4-5)

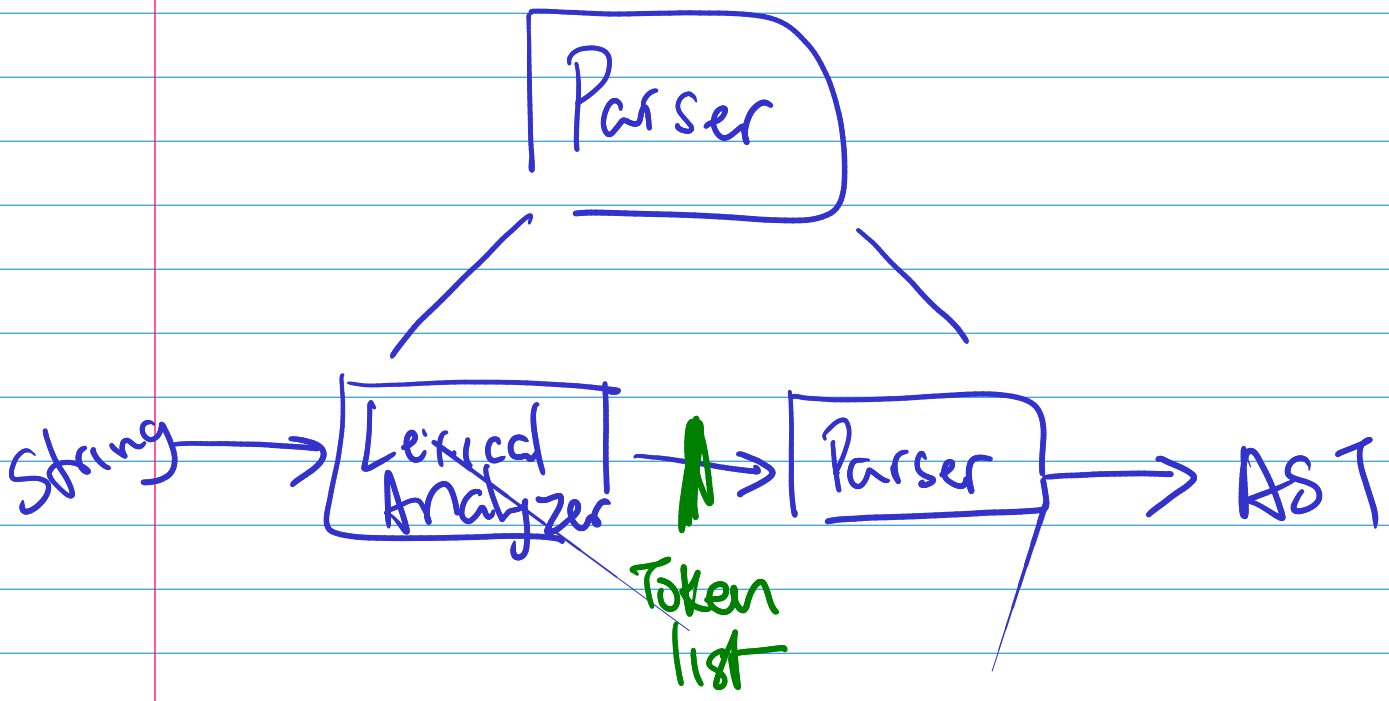
(3-4)-5

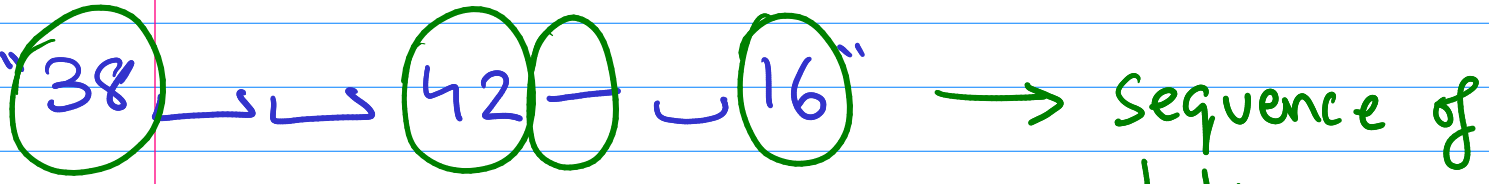


"Ambiguity"



Question How to go from source code to AST, if it exists?





IntLit(38); IntLit(42); MINUS; IntLit(16)