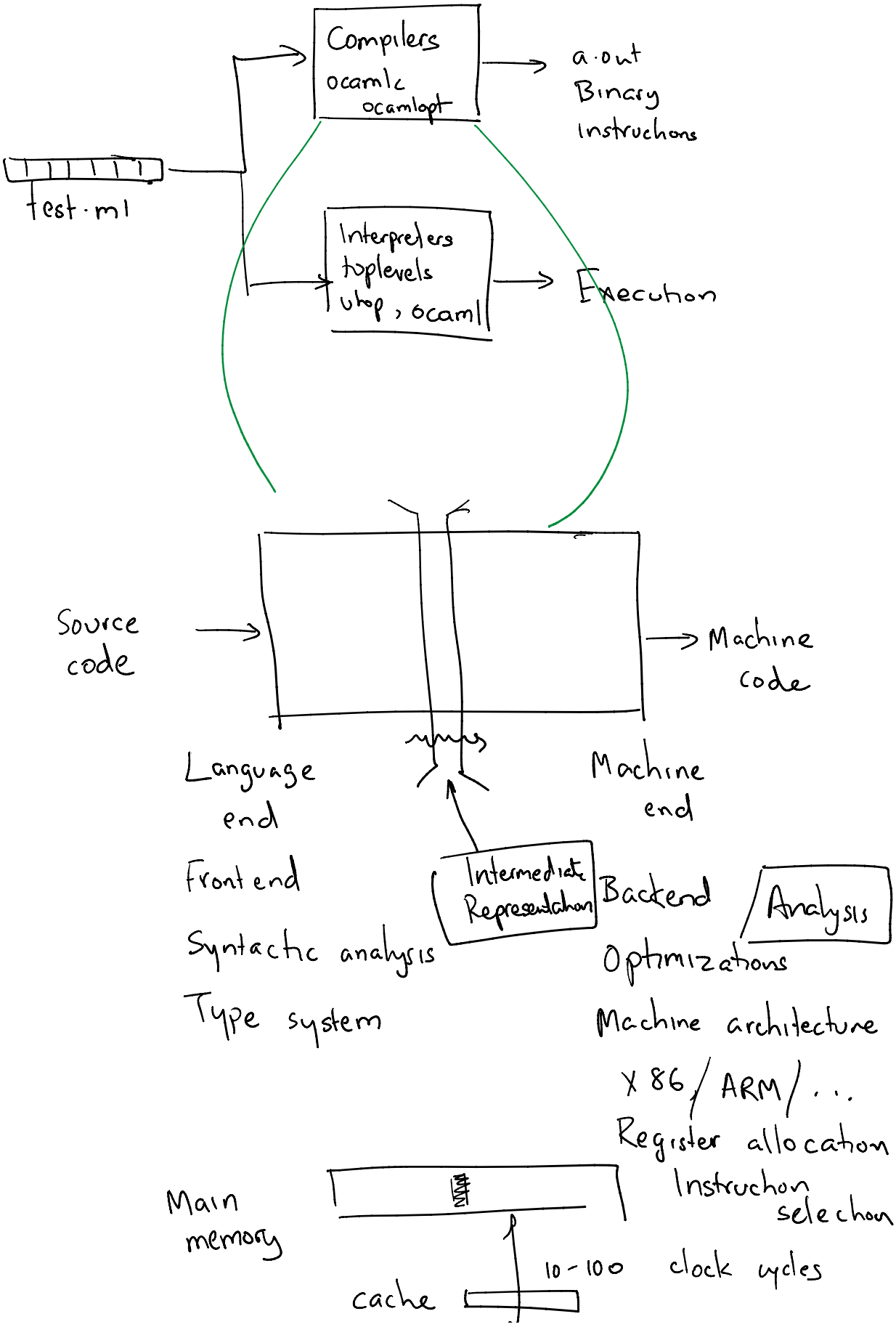
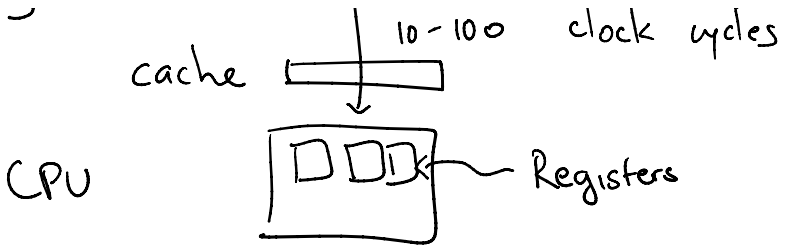
Lecture 12

Thursday, September 30, 2021 1:57 PM

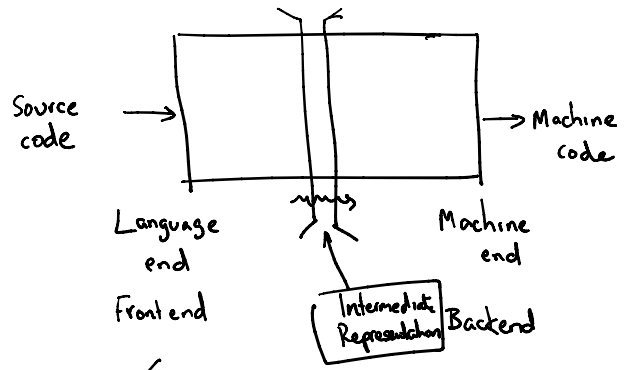


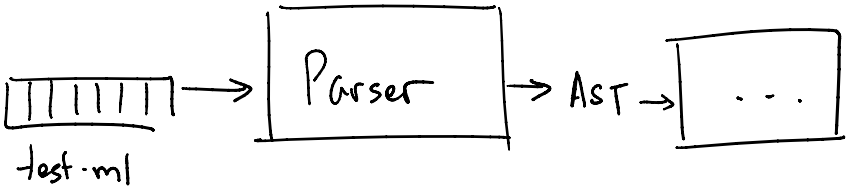


add 3 iax

mov iax ibx

load 0x8432 ibx

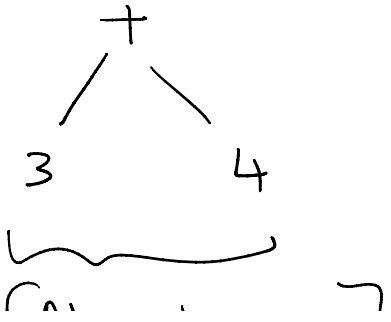




"3 + 4"

char list

string



string

Abstract Syntax Tree

First Part of this Unit : Parsing

Tool-building: ocamllex, menhir

flex — bison, yacc — C

Regular expressions

CYK Parsing

Fancy Board games

automata theory

Language Specification

What does a C program look like?

"Shape" / "Syntax"

- Backus Naur form

What does a C program mean?

"Semantics"

- Operational semantics

- Backus Naur form
- Context free grammars

- Operational semantics
- Axiomatic semantics
- Denotational semantics

Tony Hoare

Dana Scott

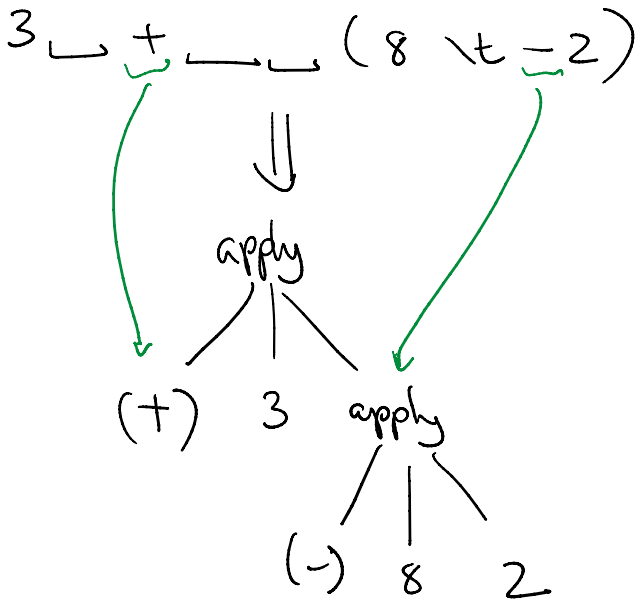
```
void foo() {
    int x=3;
}
```

Well-formed programs

```
void foo() {
    int int = 5;
}
```

```
void >> () {
    int * =3
}
```

Ill formed programs



Expr ::= Integer Literal ⇝ 38 | 42 | 361 ...

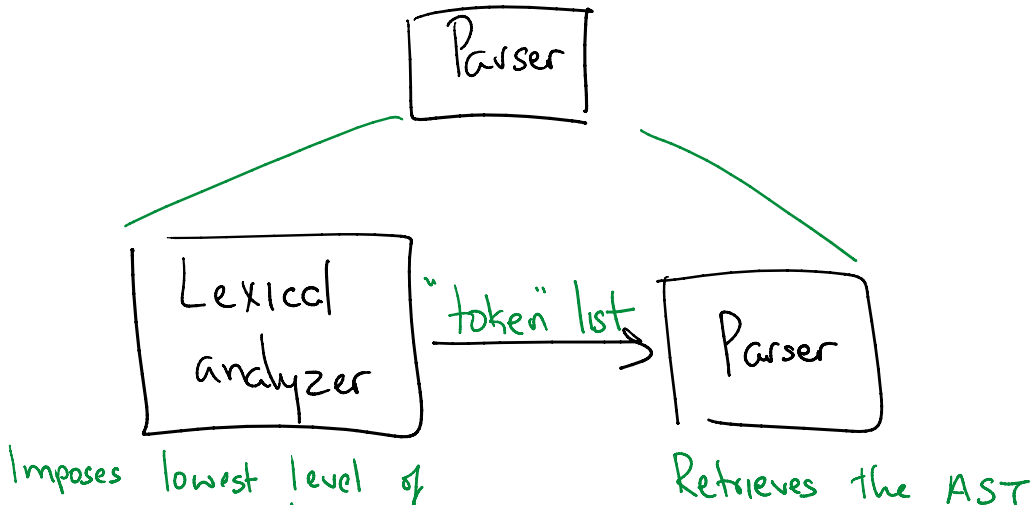
[0-9]+

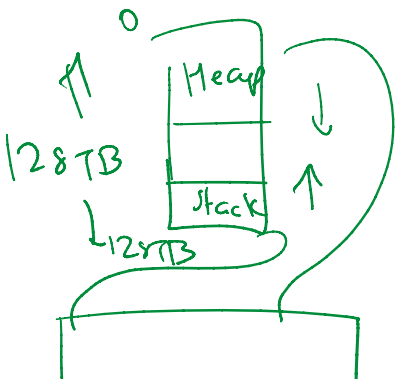
| Expr '+' Expr ⇝ Two expressions with a plus sign in between

| Expr '-' Expr ⟶ Two expressions with a minus sign.

| '(' Expr ')' ⇝ Expression within parentheses

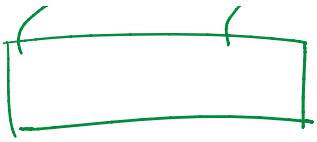
Question: How does one go from sequences of characters to a parse tree?





Imposes lowest level of syntactic structure

Retrieves the AST



38 ␣ + ␣ - (423 ␣ ␣ - 32)

⇓ Tokenize

IntLiteral(38); PLUS; MINUS;

LPAREN; IntLit(423); MINUS; RPAREN