

CS 152: *Programming Language Paradigms*

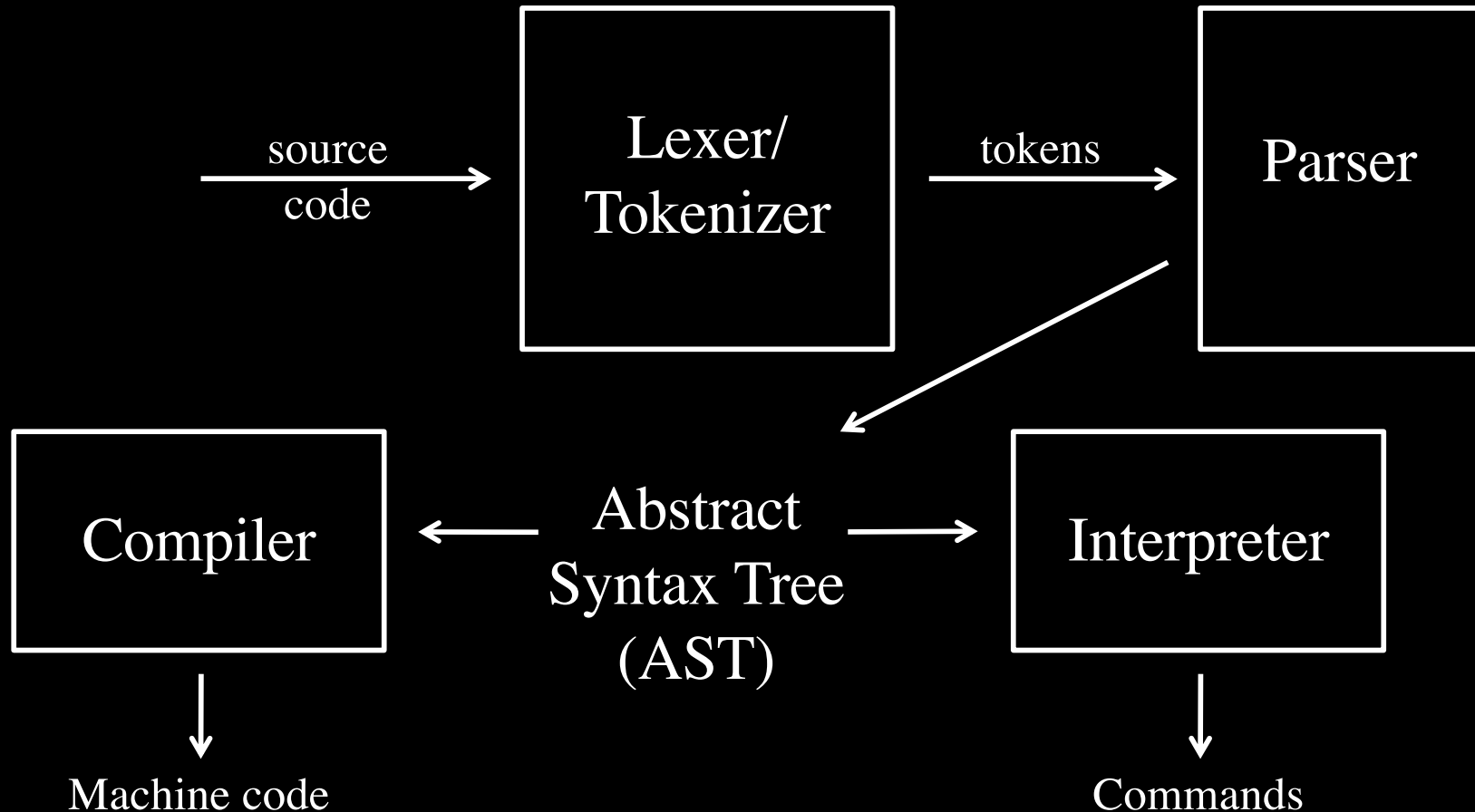


# Virtual Machine Lab

Prof. Tom Austin

San José State University

# A Review of Compilers



## Virtual Machines (VM)

- Code is compiled to *bytecode*
  - low-level
  - platform independent
- The VM interprets bytecode

## Lab: Scheme VM

In today's lab, you will implement:

- a compiler for Scheme
- a stack-based VM

## Input program

```
(println (+ 2 3 4))
```

```
(println (- 13 (* 2 4)))
```

```
(println (- 10 4 3))
```

## Supported VM Operations

- **PUSH** – adds argument to stack
- **PRINT** – pops & prints top of stack
- **ADD**
  - pops top two elements
  - adds them together
  - places result on stack
- **SUB** – subtraction
- **MUL** – multiplication

# Bytecode Output

```
PUSH 2      MUL
PUSH 3      SUB
ADD         PRINT
PUSH 4      PUSH 10
ADD         PUSH 4
PRINT      SUB
PUSH 13     PUSH 3
PUSH 2      SUB
PUSH 4      PRINT
```

## Lab – Write a Compiler and a VM

- Starter code is provided.
- `println` is functional.
- Your job: add support for the mathematical operators.

## EXTRA CREDIT

Add compiler support for

- if expressions
- boolean variables
- let expressions

Add VM support for

- labels
- Jump  
(JMP / JZ / JNZ)  
operations
- STOR / LOAD  
operations