Theory of Compilation

Lecture 00

Winter 2023-2024

Who?

Shachar Itzhaky

shachari@cs.technion.ac.il

Office hours should be scheduled in advance

TAs:

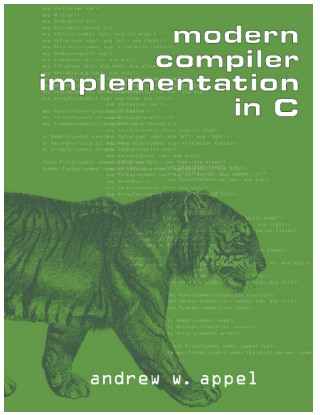
- Hila
- Andrey
- Tomer
- Guy

What?

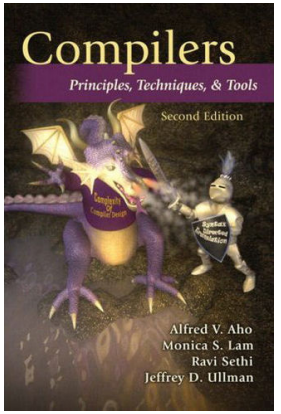
- Understand:
  - What a compiler is
  - How it works
  - Proven techniques

    (most can be re-used in other settings)

How?



- What will help us:
  - Textbooks
    - Modern Compiler Design (Grune)
    - Modern Compiler Implementation in C
    - Compilers: Principles, Techniques & Tools
  - Homework assignments
    - "Dry": deepen understanding of theory
    - "Wet": build a compiler yourself
  - Ask questions, answer questions!



piazza

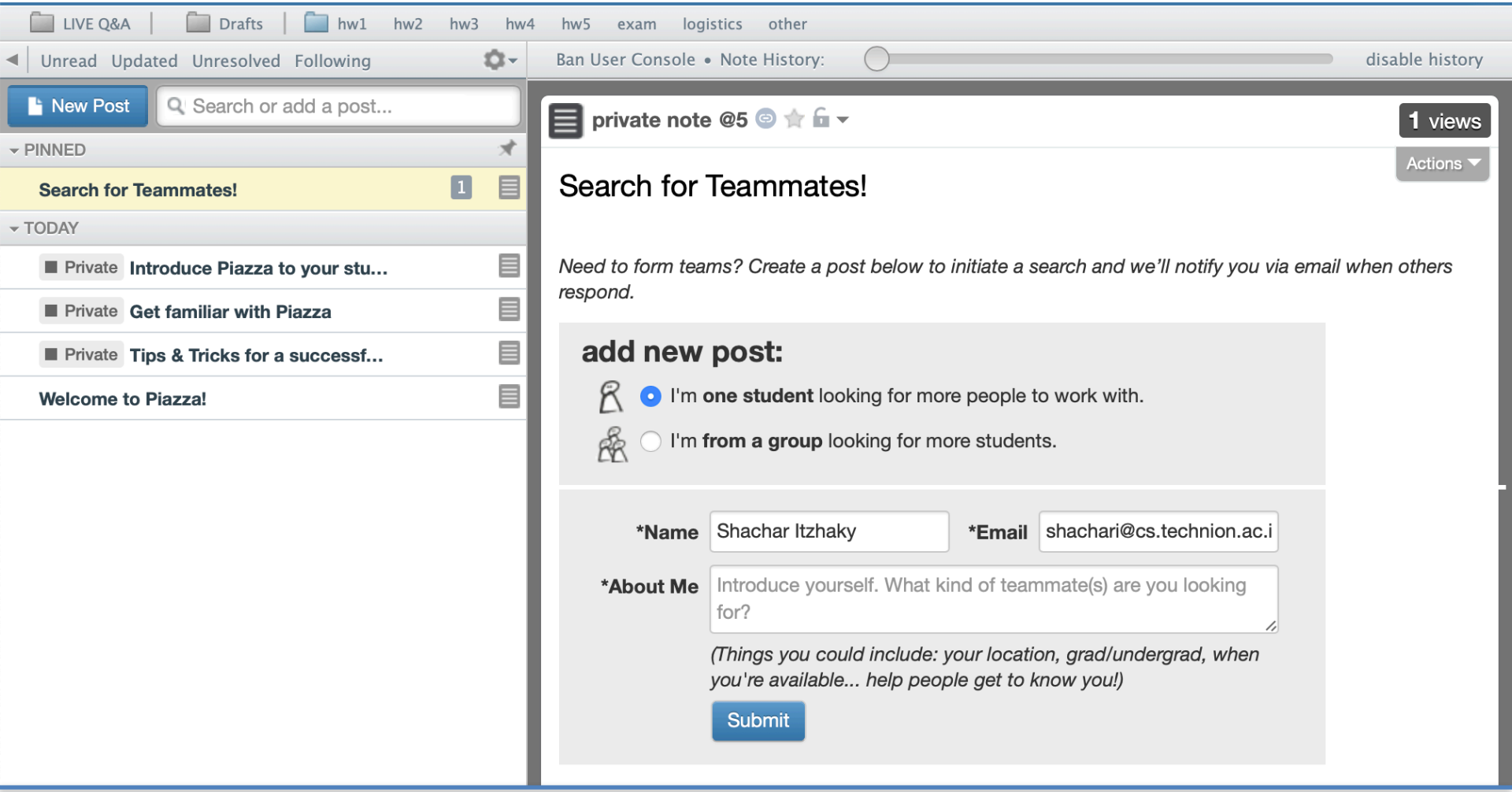
How?

piazza.com/technion.ac.il/winter2024/236360

Piazza Navona, Roma, Italia

piazza

How?


LIVE Q&A
Drafts
hw1
hw2
hw3
hw4
hw5
exam
logistics
other
Unread
Updated
Unresolved
Following
Ban User Console • Note History:
disable history
New Post
Search or add a post...
PINNED
Search for Teammates!
1
TODAY
Private
Introduce Piazza to your stu...
Private
Get familiar with Piazza
Private
Tips & Tricks for a successf...
Welcome to Piazza!
private note @5
1 views
Actions
Search for Teammates!
Need to form teams? Create a post below to initiate a search and we'll notify you via email when others respond.
add new post:
I'm one student looking for more people to work with.
I'm from a group looking for more students.
*Name
Shachar Itzhaky
*Email
shachari@cs.technion.ac.i
*About Me
Introduce yourself. What kind of teammate(s) are you looking for?
(Things you could include: your location, grad/undergrad, when you're available... help people get to know you!)
Submit

How Not?!

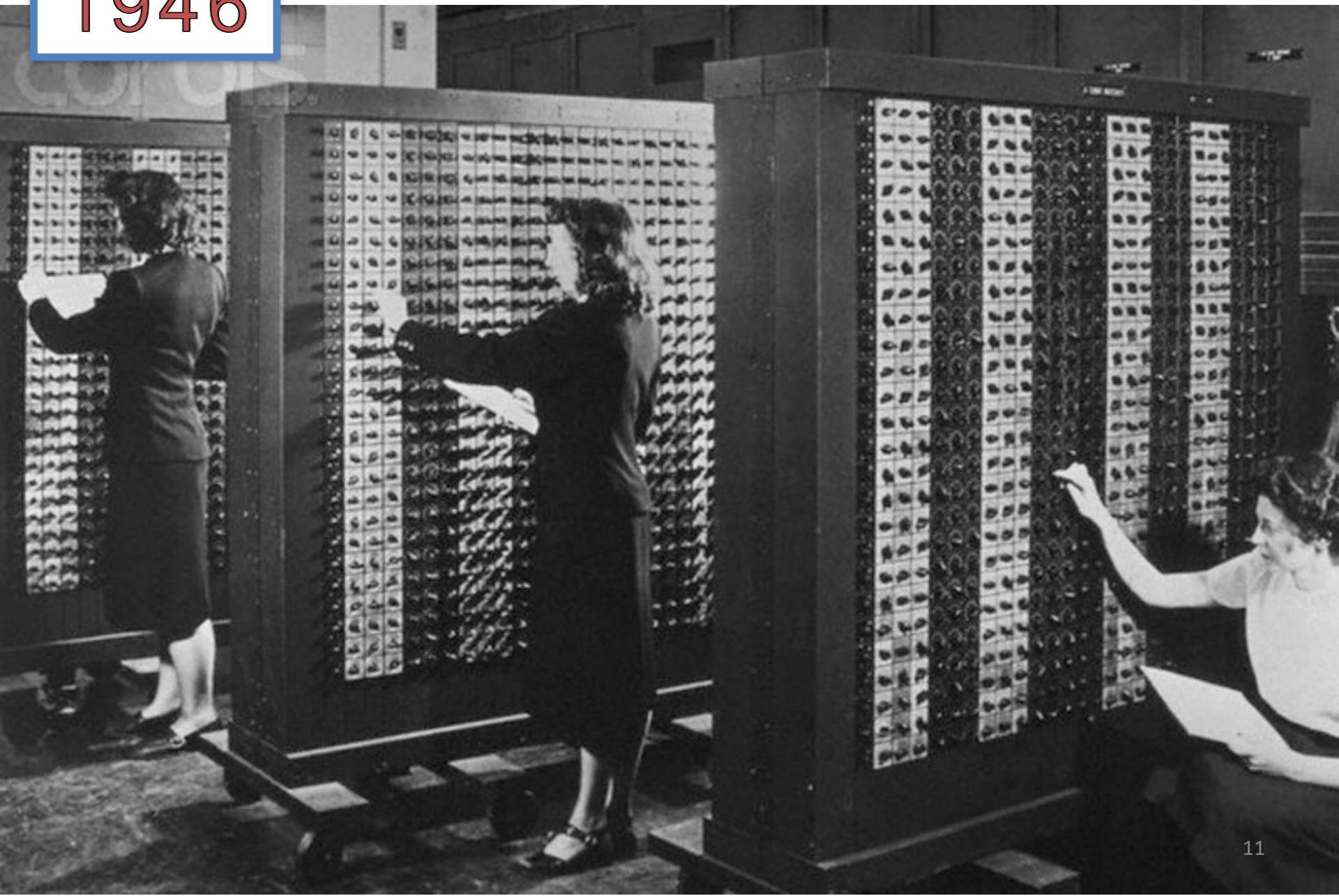
- "Your slides don't have everything you say written on them" (*colloquial*)
  - Yes, I know, this is by design
  - Presentations are a teaching aid,
    not a substitute for coming to lectures
- If you don't attend lectures or attend and don't listen, you will inevitably miss some things
- If you want slides that have all the material written on them nicely, that format is indeed **available** and **commonly known as a textbook**
- See how horrible this slide is? This is why you won't see many slides with as much text as this one for the rest of the course

Exam

- 80% of the final grade
- Look at exams from previous years (2013–)
- Prepare *throughout* the semester…
  - If you attend lectures and finish the assignments, you should do well in the exam; if you've missed a lecture, try to keep up with the material
  - Historical evidence — attending leads to ***higher success rate*** in the final exam!

ENIAC

1946



1952

"The Education of a Computer" (Grace Hopper)

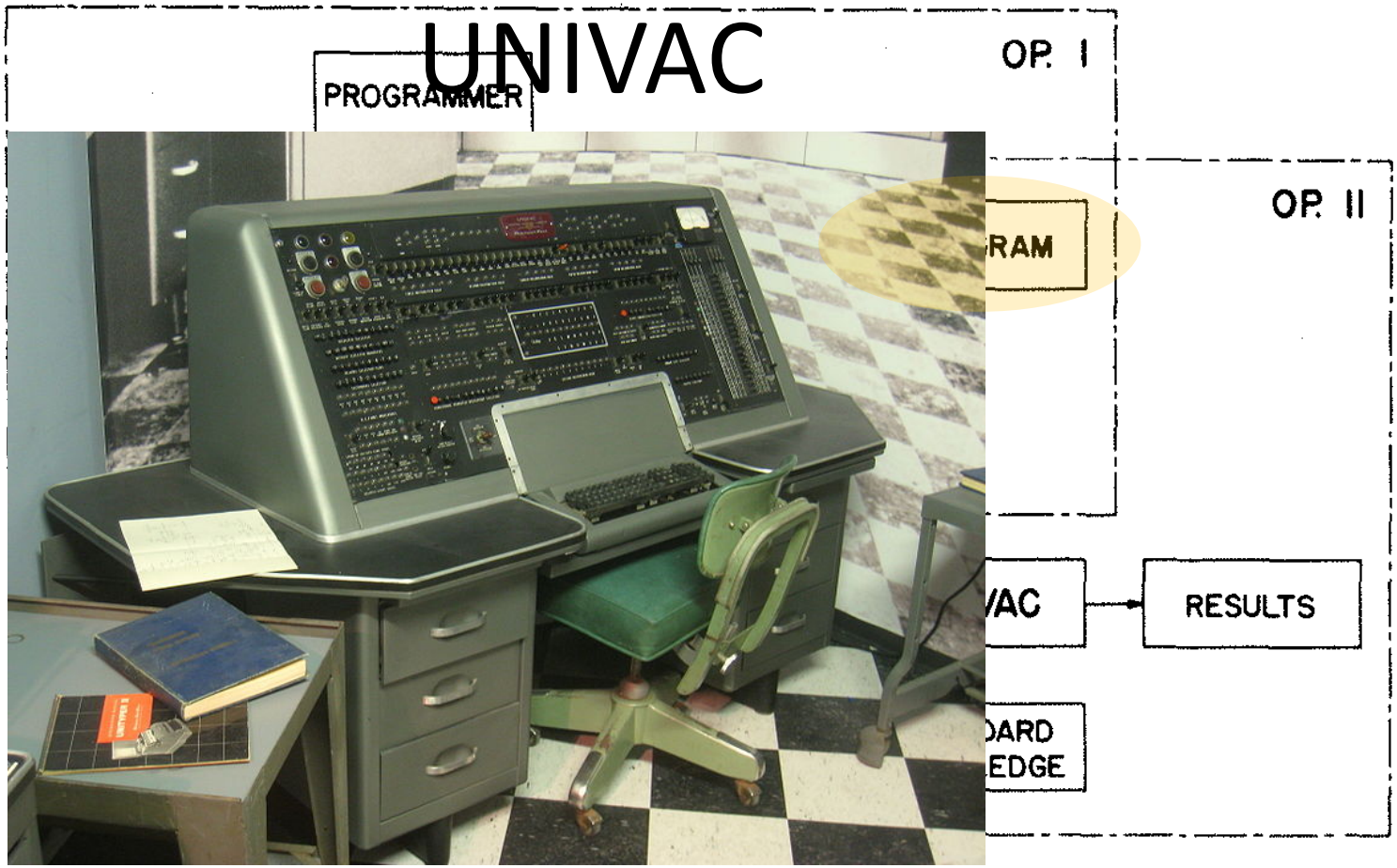
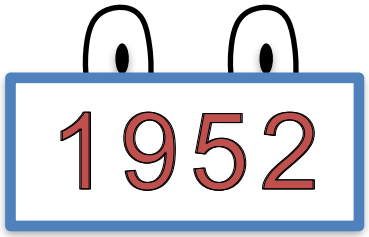


Fig. 4 - SOLUTION OF A PROBLEM



"The Education of a Computer" (Grace Hopper)

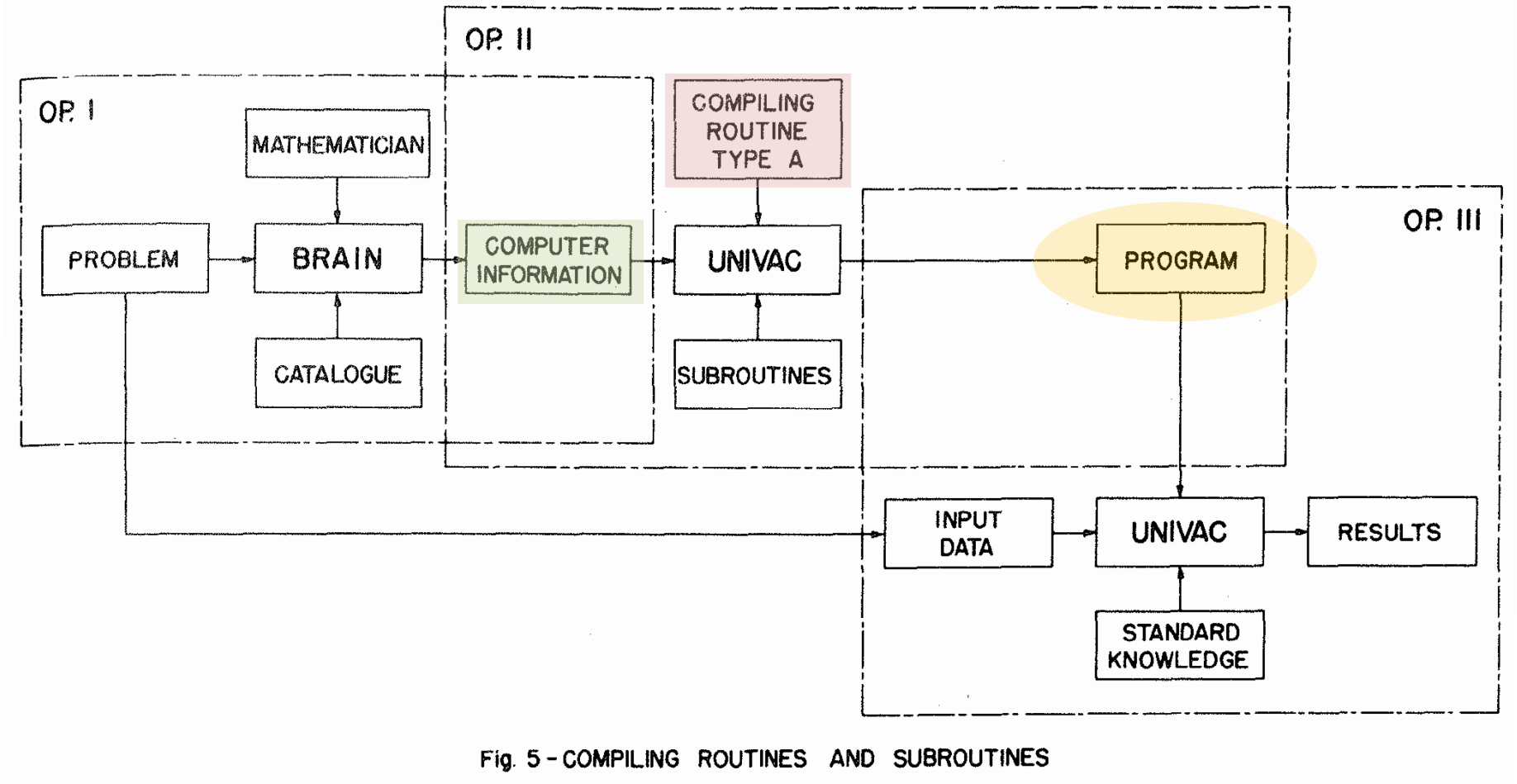
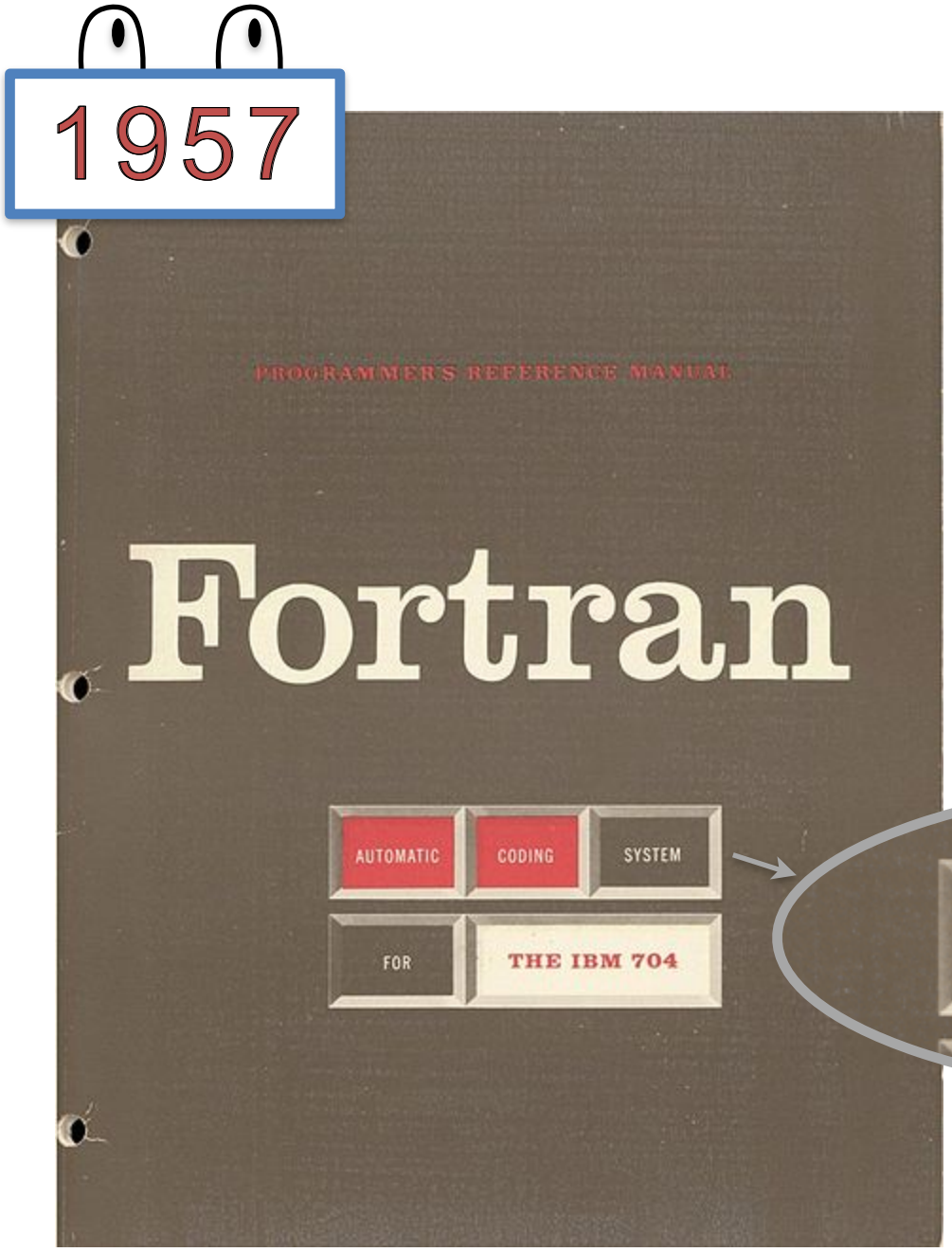
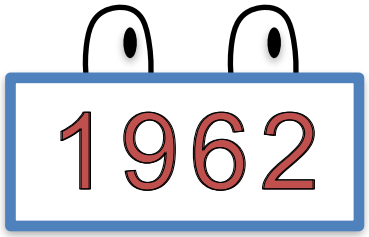


Fig. 5 - COMPILING ROUTINES AND SUBROUTINES



John Backus and team at IBM

The first complete compiler



Norwegian Computing Center, Oslo

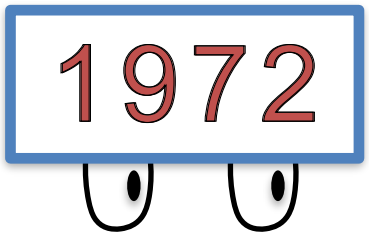
simula

Classes ● Objects ● Inheritance
Virtual Methods ● Garbage Collection

Classes are made of:

- Parameters
- Attributes
- Methods
- Life (Body)

```
Begin
   Class A(P); Integer P;
   Begin  LifeA1  Inner;  LifeA2  End;
   Part1
   A(3) Begin
     Block body
   End;
   Part2
End;
```



Xerox PARC (Palo Alto Research Center)

A

ABAP4
ABC
ACPI Source Language
ACS
Action!
Actionscript (Flash 5)
ActionScript (Flash 8)
Actionscript (Flash MX)
ActionScript 3.0
Ada
ADVPL
Agda
aheui
Algol-60
Algol-68
Alpha-Five-Xbasic
Amazon States Language
amharic
Amiga-E
AMOS
Andl
AngelScript
Anguish
Ansible
Ante
APC
APL
AppleScript
Arduino
Arena
Arendelle
Argh!
ArnoldC
AsciiDots
ASP (C#)
ASP (JavaScript)
ASP (VBE)
ASP (VBS)
ASP.NET
Assembler (6502, Apple II)
Assembler (6502, C64)
Assembler (68000, Amiga)
Assembler (68000, Atari ST)
Assembler (68008)
Assembler (8051)
Assembler (ARM, Android)
Assembler (ARM, RISC OS)
Assembler (DG-Nova)
Assembler (HLA)
Assembler (HP-85)
Assembler (IBM 370)
Assembler (Intel)
Assembler (Itanium)
Assembler (MIPS)
Assembler (MMIX)
Assembler (PA-RISC)
Assembler (PDP-11)
Assembler (PDP-8)
Assembler (PPC, Darwin)
Assembler (SPARC)
Assembler (TAS)
Assembler (VP)
Assembler (Win32)
Assembler (X1)
Assembler (X8)
Assembler (z390)
Assembler (Z80 Console)
Assembler (Z80, CP/M)
Assembler (ZX81)
Asterisk
ATS
AutoHotkey
AutoIT3
Automator
AviSynth
AWK

B

B
BAL
BASIC-PICAXE
BASIC
Bato
Batsh
bc
BCPL
BeanShell
Beef
Befunge
Beta
BibTex
Binary Lambda Calculus
BIT
BitZ
Blender
BlitzMax
BlitzPlus
BMC Remedy
Boo
Brainfuck
BrightScript
BS2000
BSP
BuddyScript
Buzz
Byte Syze

C

C*
C/AL
C#
C (Amiga Anywhere)
C (ANSI)
C (Curses)
C (GEM)
C (Intuition)
C (K&R)
C (OpenGL)
C (PresentationManager)
C (Windows)
C (X11 Athena)
C++
C++ (.NET CLI)
C++ (Epoc)
C++ (FLTK)
C++ (Gtk++)
C++ (ISO)
C++ (MFC)
C++ (Qt)
CA-Easytrieve Plus
Caché Object Script
CAML-Light
Carp
Casio BASIC
CDuce
Centura
Ceylon
ChaiScript
Chef
Chicken
CICS-COBOL
Clarion
Clean
Clipper
Clojure
CLP
CMake
COBOL
Cobra
Cocoa
Coconut
CoDScript
CoffeeScript
ColdFusion
CommandScript
Common Lisp
Console Postscript
ConTeXt
Cool
CoolBasic
cpl
Crystal
CSS
CUDA
CWEB
CYBOL

D

D
D4
Dafny
Darkbasic
Dart
Databasic
Dataflex
dBase
dc
DCL
Delphi
Di
DI
DI
Dr
DV
DV
DV

E

E
eC
E
E
E
E
ELENA 3.0
ELENA 4.0
Elixir
Elliott
Elm
Emojicode
Emoticon
EOS 2
Erlang
ERRE
Eta
Euphoria
Eve
Ezhil

F

F#
F
Falcon
FALSE
Felix
Fennel
Ferite
Fetlang
Filemaker
Fjölnir
Flaming Thunder
Focal
FOCUS
Folders
Forth
Fortran
Fortran77
Fortran90
FortranIV
Fortress
FreeBASIC
Frink
Full Metal Jacket

G

G-code
Gambas
GameMonkey Script
Genero BDL
Genie
Gentee-simple
Gentee
GLBasic
GML
Go
Gofer
GoogleGadgets

ophone

HTML
Hubot
Human
HyperTalk

I

IBM-Exec
IBM-Exec2
ici
ICL SCL
Icon
IDC
IDL
Idris
Imba
Inform
Informix 4GL
Ingres ABF
InstallScript
Intercal
Io
Iptscrae

J

J
Jade
Jako
Janet
Jason
Java
Java (Mobile)
Java (Servlet)
Java (Swing)
Java Server Pages
JavaScript
JCL
Jess
Jorf
JSFuck
JudoScript
Julia

K

K
K3
K4
K5
Kitten
Kix
Kotlin
Kumir
Kylix

L

LÖVE
L

M

m4
MACRO-10
MACRO-11
Macromedia-Flex
MAD
make
Malbolge
MAMASH
Maple
Marmelade
Mathematica
MATLAB
Matrix
MAX-MSP
Maxima
MAXScript
MCSBL
MDM Zinc
MEL
MetaPost
MEX
Microtik
MiniScript
mIRC-Alias
mIRC-Commandline
mIRC-Script
MivaScript
MML-AXE10
Modula-2
Modula-3
MoHAA-Script
MOO
Mouse
MPD
MPLAB IDE
MS Small Basic
MSDOS
MSIL
MuLisp
Mumps
MySQL FUNCTION
Mythryl

N

Natural
Nemerle
newLISP
NewtonScript
Nice
Nim
Nit
Node.js
Noor
NSIS
NXC

O

Oberon.oberon
Oberon.std
Objective-C
OCaml
Occam
Octave
Omnimark
Ook
OpenVMS
OPL.dialog
OPL.simple
Oz

P

ParaSail
Parser
Pascal (Windows)
Pascal
Pawn
PBASIC
PDF
PEARL
PeopleCode
Perl
Perl 6
Pharo
PHP+GD
PHP
Picat
Piet
Pike
PILOT
PL-SQL
PL1
Plankalkül
Pocket Calculator
POP-11
Portugol
PostgreSQL
Postscript
POV-Ray
Powerbasic
Powerflex
PowerScript
Powershell
PPL
PQN-PROC
PRAAT
ProC
Processing
Profan
Prograph
Progress
Prolog
Punyforth
PureBasic (Console)
PureBasic (Messagebox)
PureBasic (Window)
Python 2
Python 3

Q

Q
Qalb
QCL
qore
QuakeC
Quartz Composer
QuickBASIC
Quorum

R

R
Racket
Rapira
ratfor
Rational Rose
React-VR
REALbasic
RealText
Reason
Rebol-view
REBOL
Redcode
REFAL-2
Refal
Regular Expression
Rexx (simple)
Rexx (window)
Rey
rhine
RPG IV v3-4
RPG IV v5
RPG IV v7.1
RPL
RSL
Ruby
Rust

S

S-Plus
SAKO
SAL
SApp
SAS
Sather
Sawzall
Scala
Scheme
Scilab
Scratch
Seed7
Self
SenseTalk
Setl2
Shakespeare
Shen
SilverBasic
SIMPLE
Simula
SinclairBasic
Skip
Smalltalk (simple)
Smalltalk (window)
Smalltalk MT
SMIL
SML
Snobol
Spin
Spiral
SPL4
Splunk SPL
Spry
SPSS
SQL (Advantage)
SQL (DB2)
SQL (Oracle)
SQL
sqlplus
SQR
Squeak
Squirrel
SSI
ST-Guide
Stata
Sucro
SuperCollider
SVG
Swift
SyMAL
Symsyn
szl

T

T-SQL
T
T9
TACL
TAL
Tcl
TECO
TeX
Texinfo
Thue
TI BASIC
TI Extended BASIC
TI-59
TI-8x
TinyFugue
Tk
Toy
Trans
troff
TSO CLIST
Turing Machine
Turing
TypeScript

U

Ubercode
UniComal
Unix Shell
unlambda
UnrealScript
Ursala

V

Vala
var'aq
Vatical
VAX Macro
VAX-11 Macro
VBA (Excel)
VBA (Word)
VBScript
Velato
Velocity
Verilog
Vexi
VHDL
Vim script
Visual Basic
Visual Basic .NET
Visual Basic 6
Visual FoxPro
Visual Prolog
VisualWorks Smalltalk
VMS-DCL
VRML
VSL
VVVV

W

wenyan
Whenever
Whirl
Whitespace
wml
Wolfram
WSH

X

X++
Xbase++
xblite
XHTML
XLogo
XPL0
XQuery
XSL-FO
XSLT

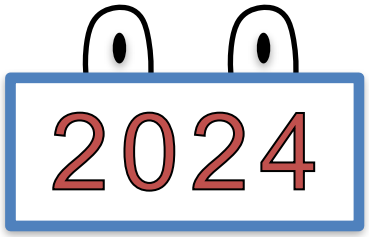
Y

Yacas

Z

Zig
ZIM

source: http://helloworldcollection.de



What is a Compiler?

"A compiler is a **computer program** that **translates** computer code written in one programming language (the **source language**) into another language (the **target language**). …primarily to a lower level language (*e.g.* assembly language, object code, or machine code) to create an **executable program**."

-- *Wikipedia*

What is a Compiler?

source language

C
C++
Pascal
Java

txt

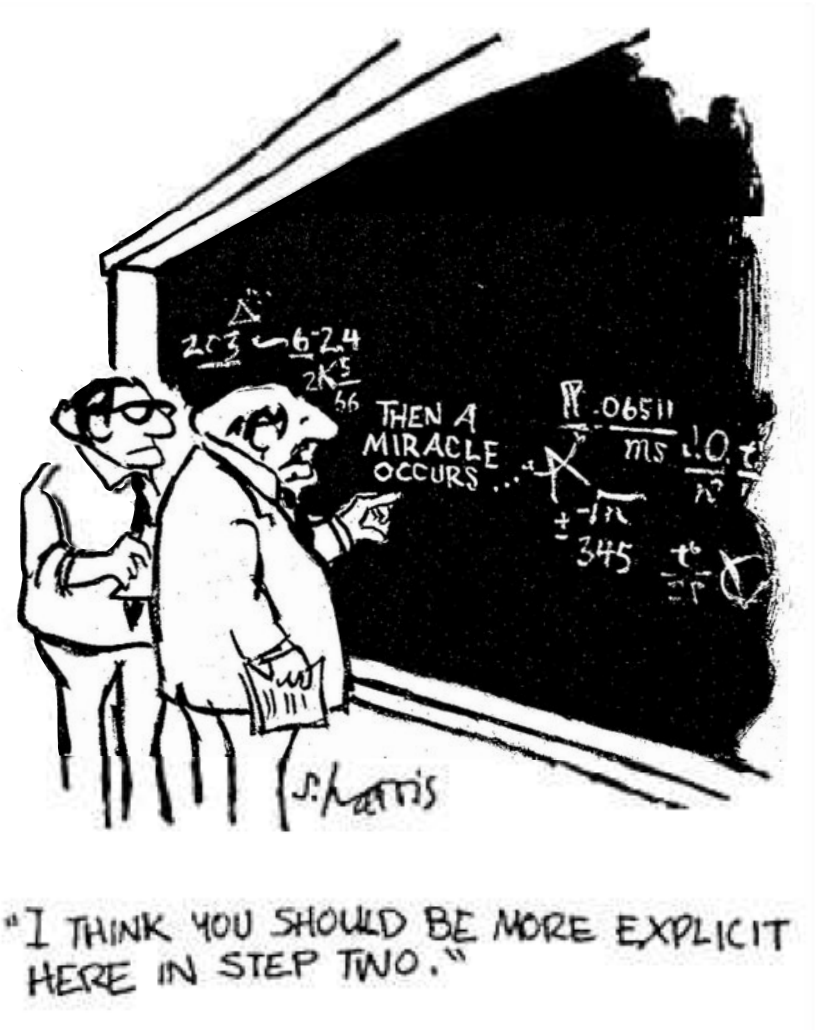
Perl
JavaScript
Python
Ruby

Source text

Lisp
Scheme
ML
OCaml

Prolog

Postscript
TeX



target language

IA32
IA64
ARM
SPARC

exe

Java Bytecode

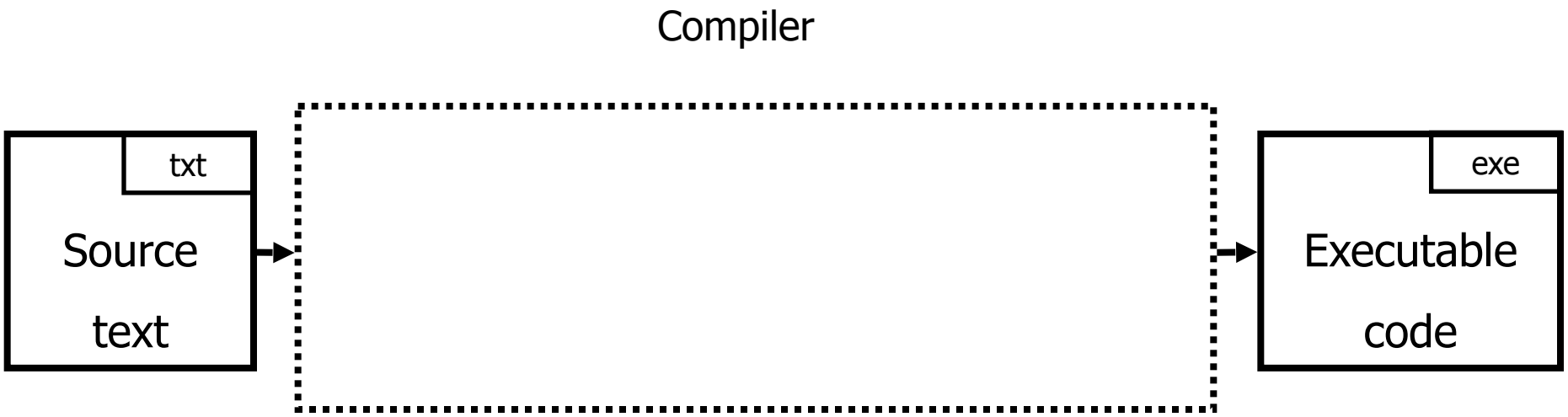
Executable code

C
C++
Pascal
Java

JavaScript

PDF
Bitmap
…

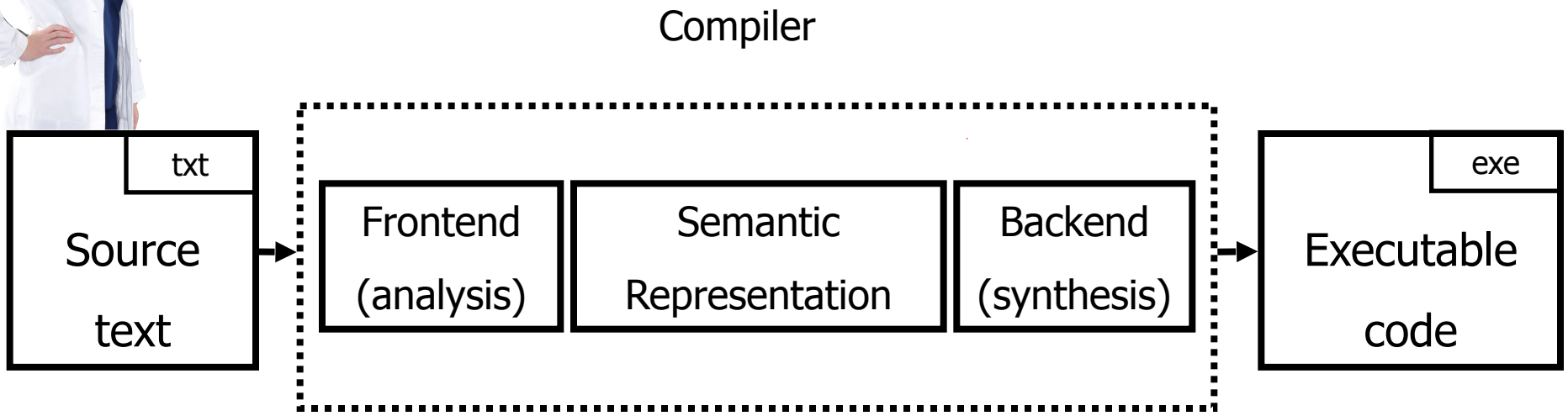
What is a Compiler?




```
int a, b;
a = 2;
b = a*2 + 1;
```

```
MOV R1,2
SAL R1
INC R1
MOV R2,R1
```

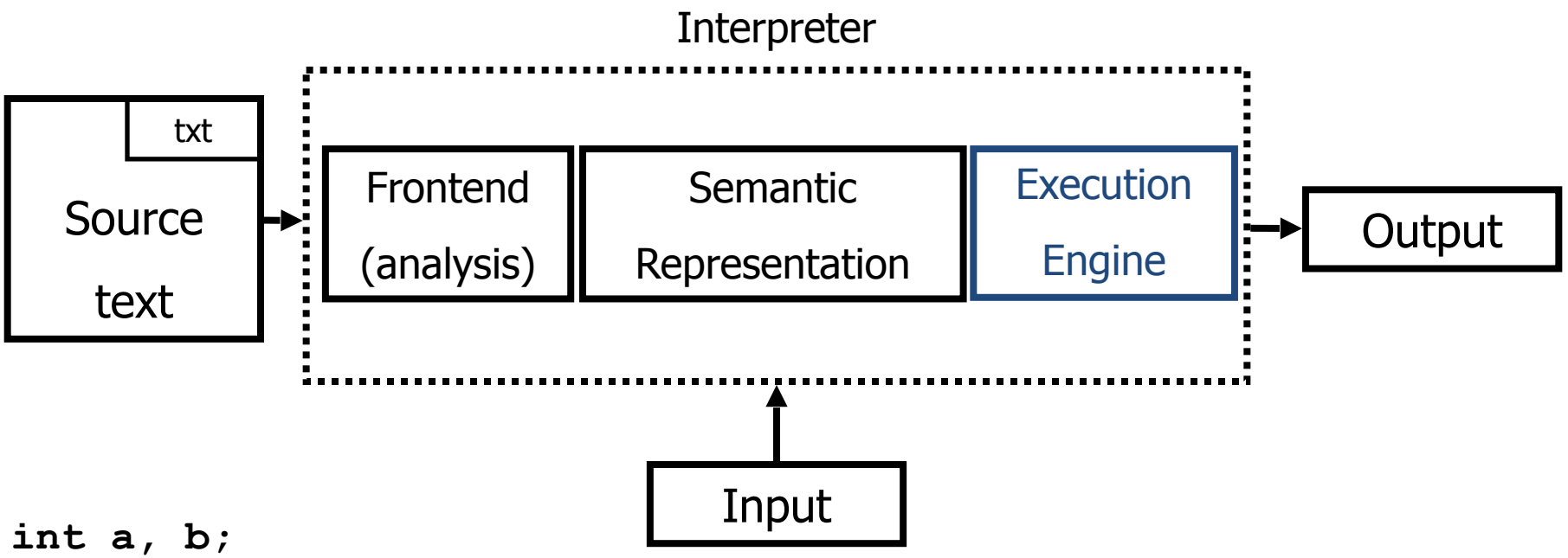
Anatomy of a Compiler




```
int a, b;
a = 2;
b = a*2 + 1;
```

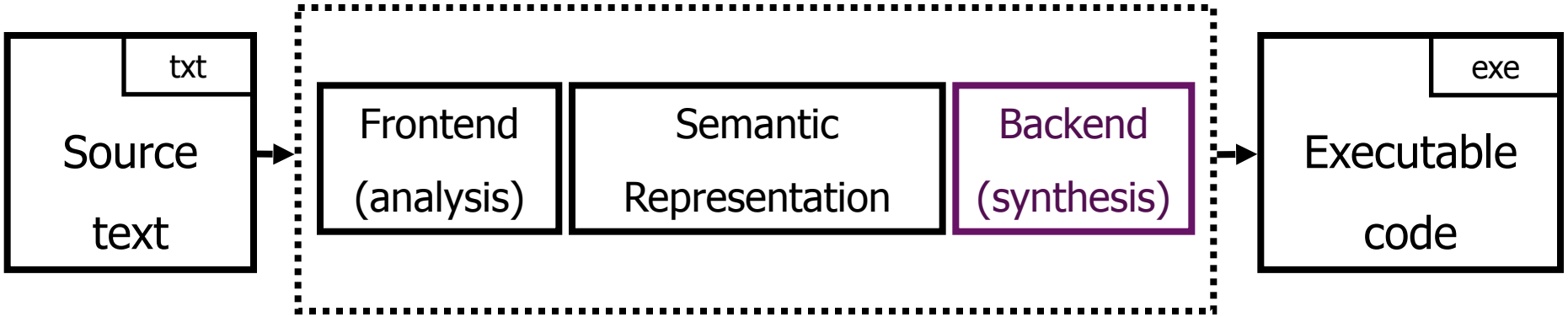
```
mov eax, 2
sal eax
inc eax
mov ebx, eax
```

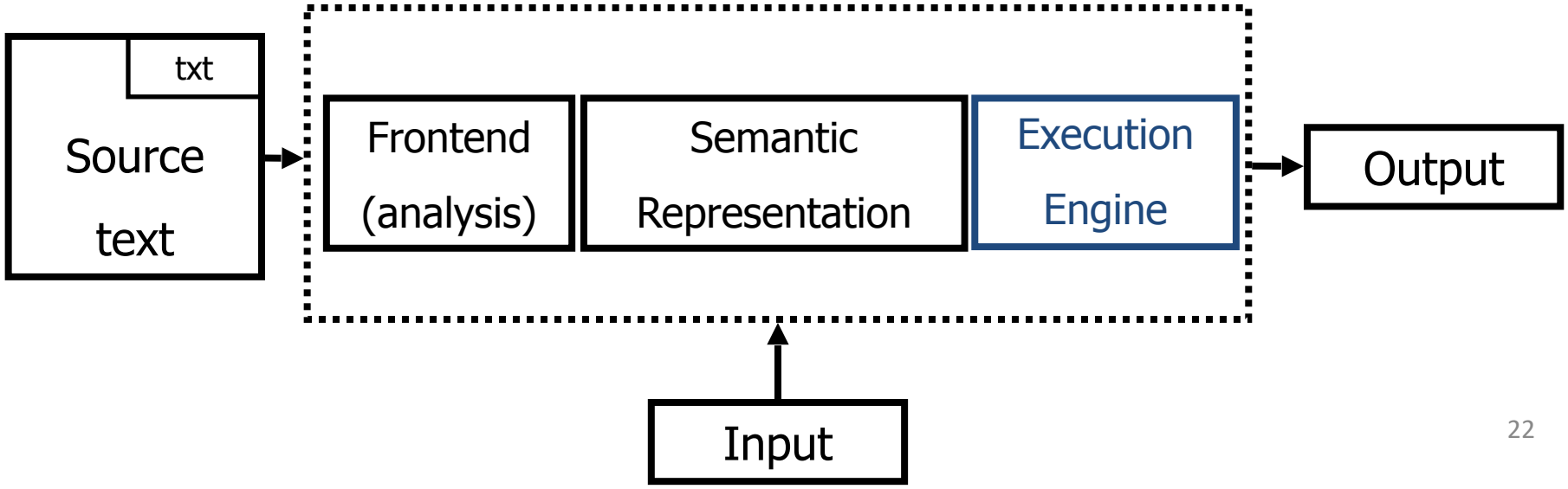
Interpreter



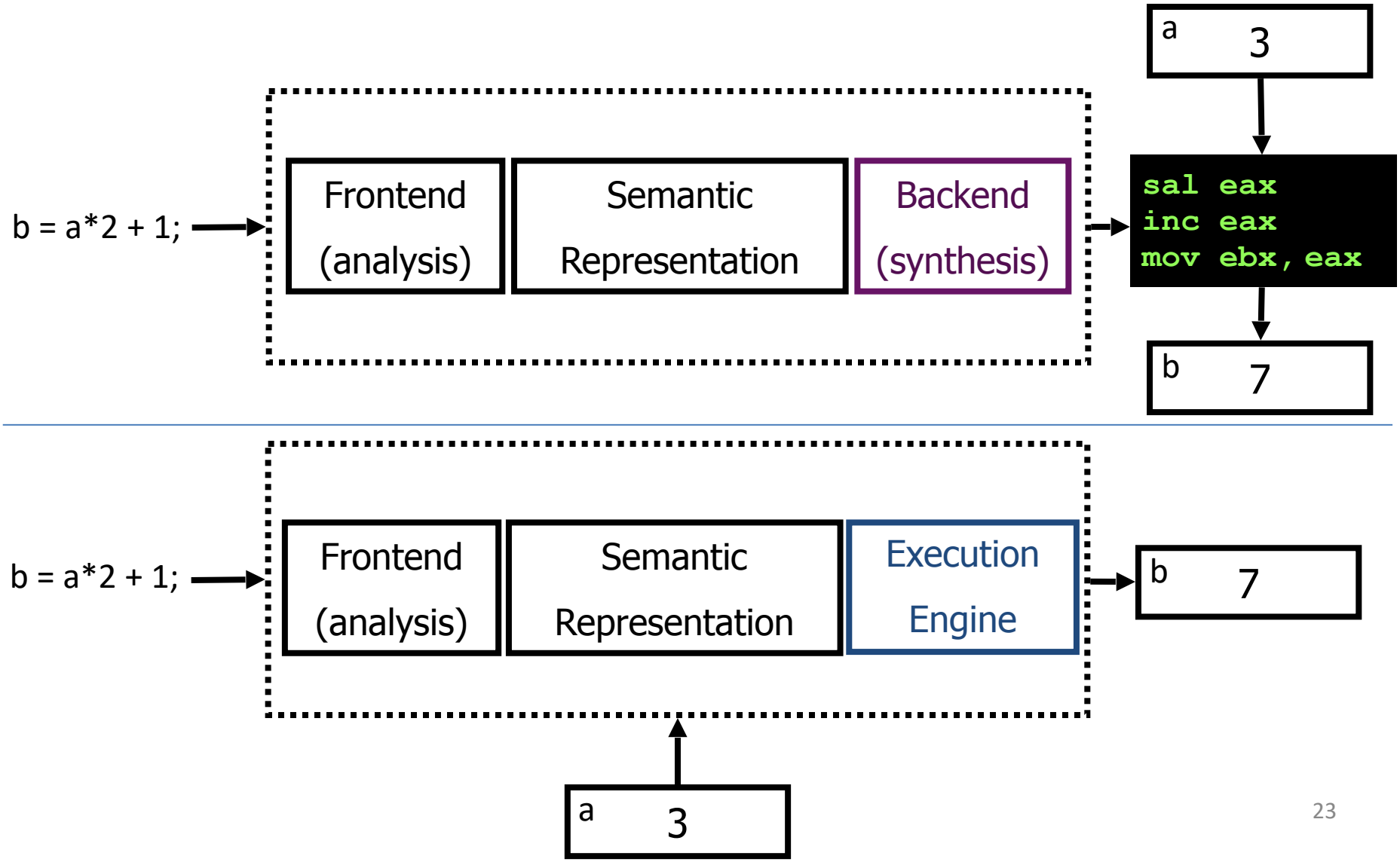

```
int a, b;
a = 2;
b = a*2 + 1;
```

Compiler vs. Interpreter

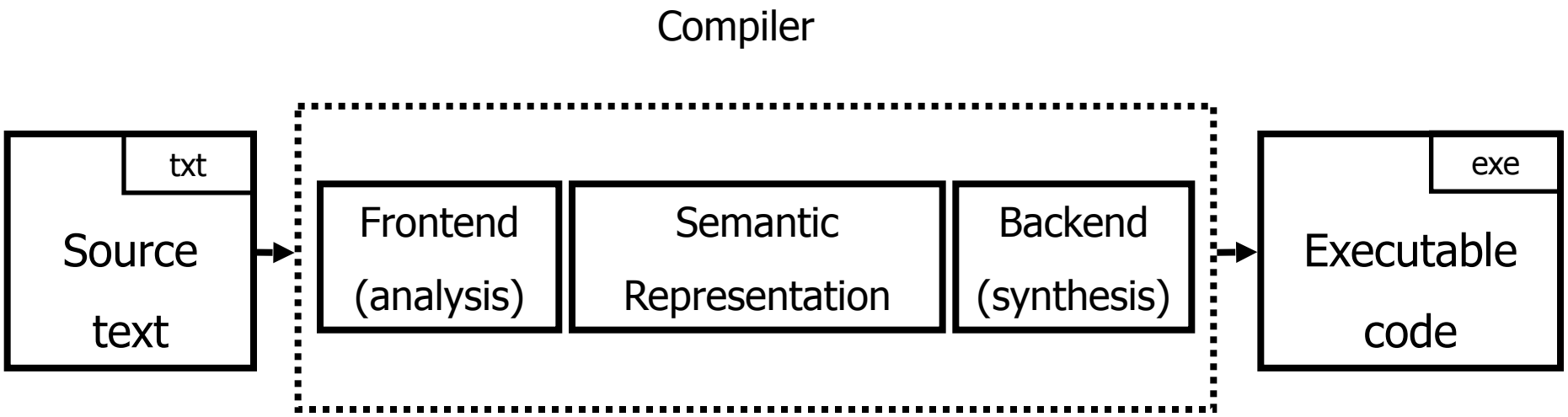




Compiler vs. Interpreter



Anatomy of a Compiler: Why?



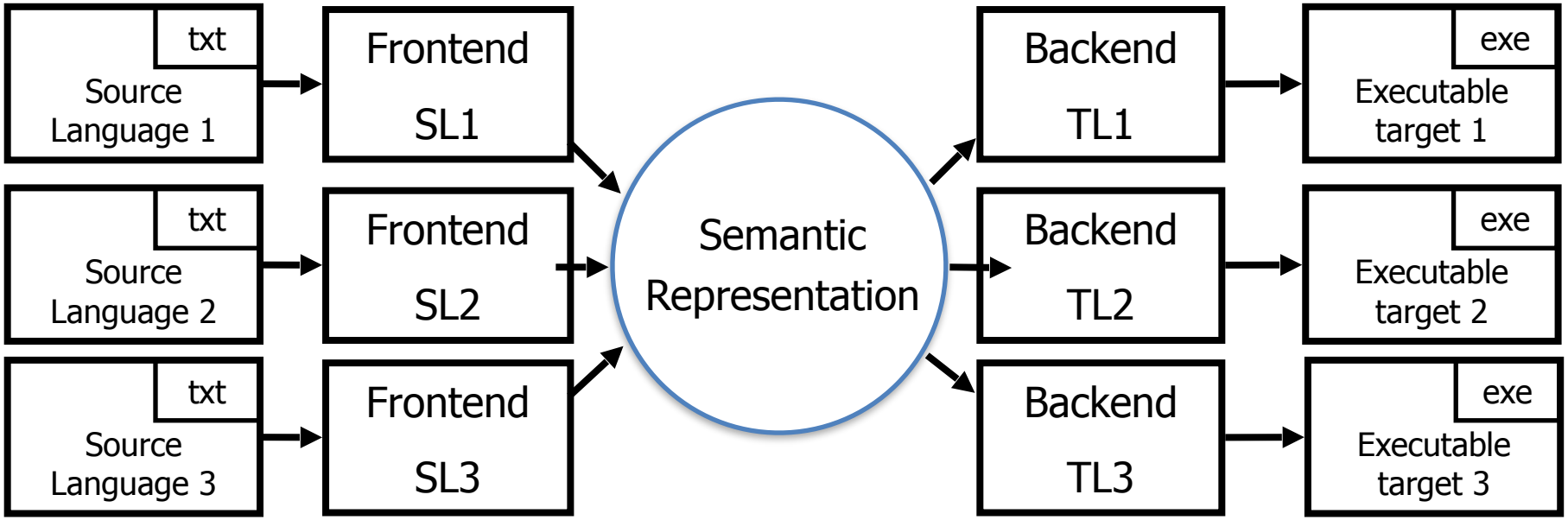

```
int a, b;
a = 2;
b = a*2 + 1;
```

```
mov eax, 2
sal eax
inc eax
mov ebx, eax
```

Modularity

```
var
  x, y : integer;
begin
  x := 2;
  y := x*2 + 1;
end.
```

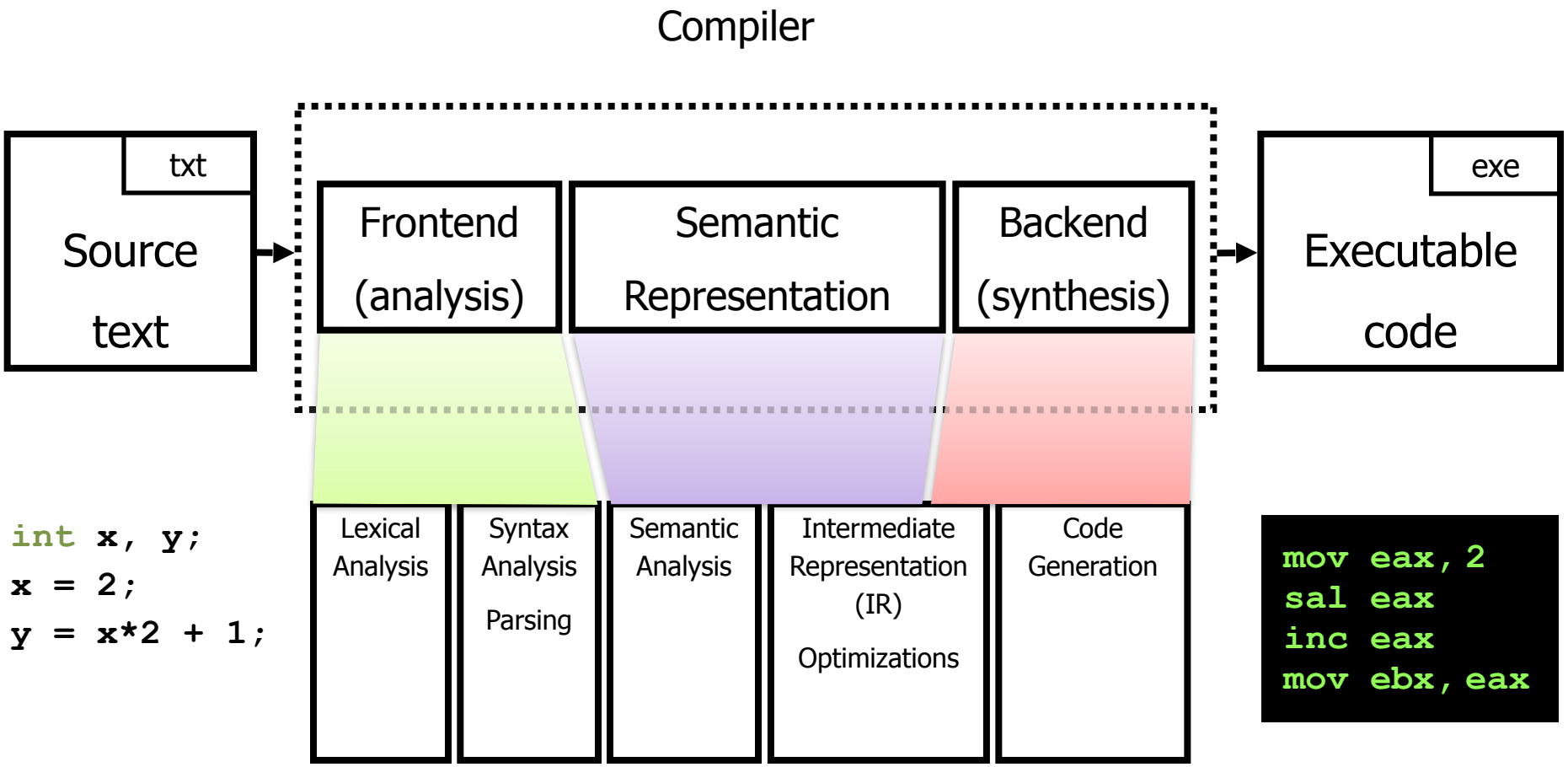
```
iconst 2
iconst 1
ishl
istore_2
iinc 2, 1
```




```
int x, y;
x = 2;
y = x*2 + 1;
```

```
mov eax, 2
sal eax
inc eax
mov ebx, eax
```

Anatomy of a Compiler



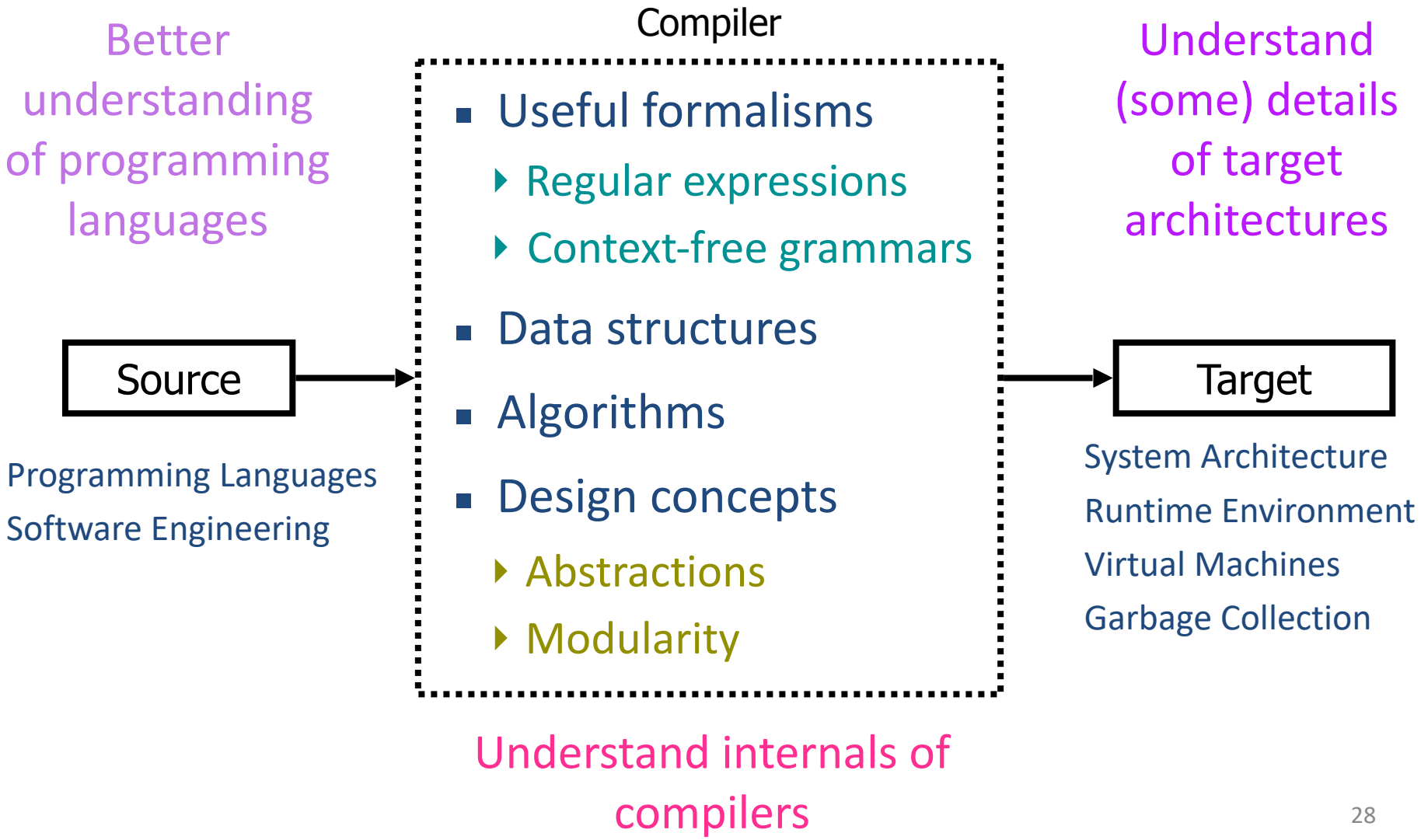

```
int x, y;
x = 2;
y = x*2 + 1;
```

```
mov eax, 2
sal eax
inc eax
mov ebx, eax
```

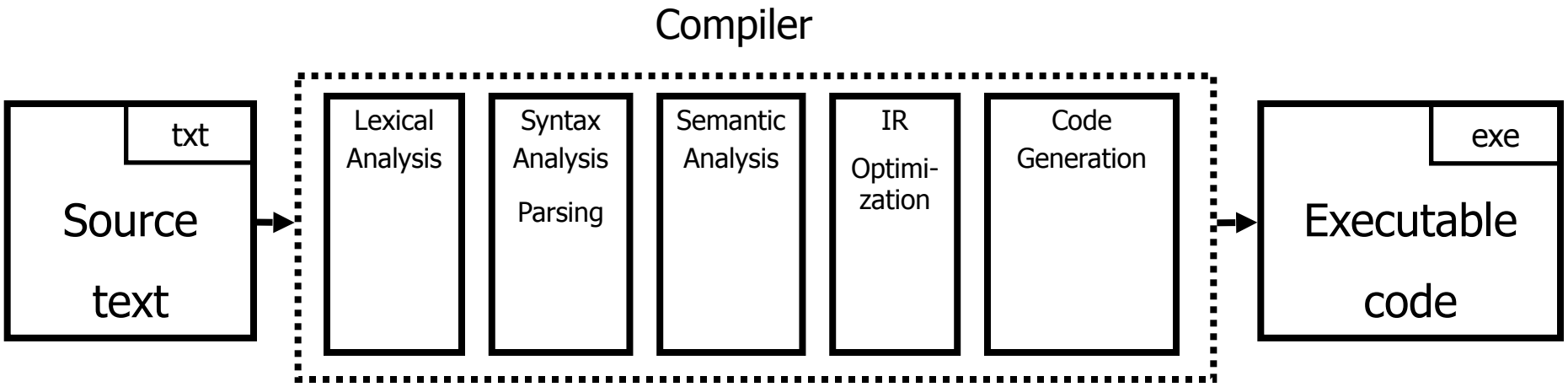
Why should you care?

- Every person in this class will build a “compiler” some day
  - Or wish they knew how to build one…
- But that is not all!

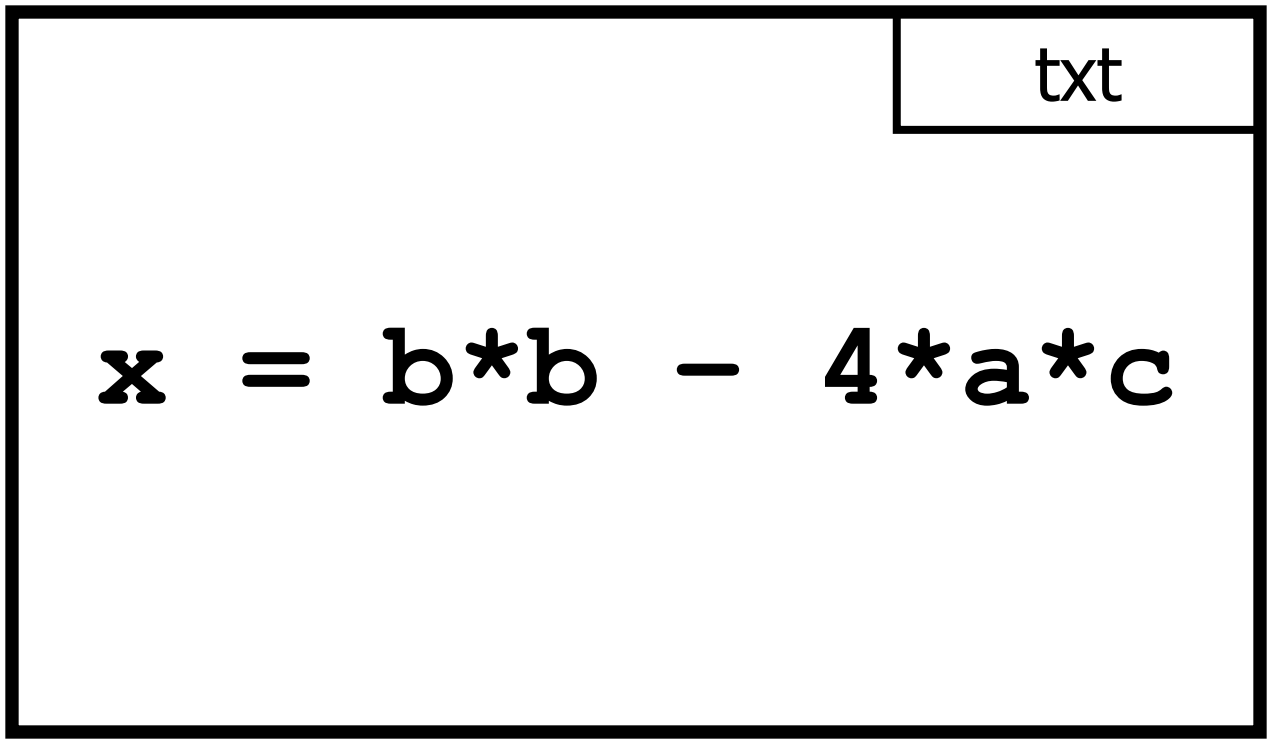
Why should you care?



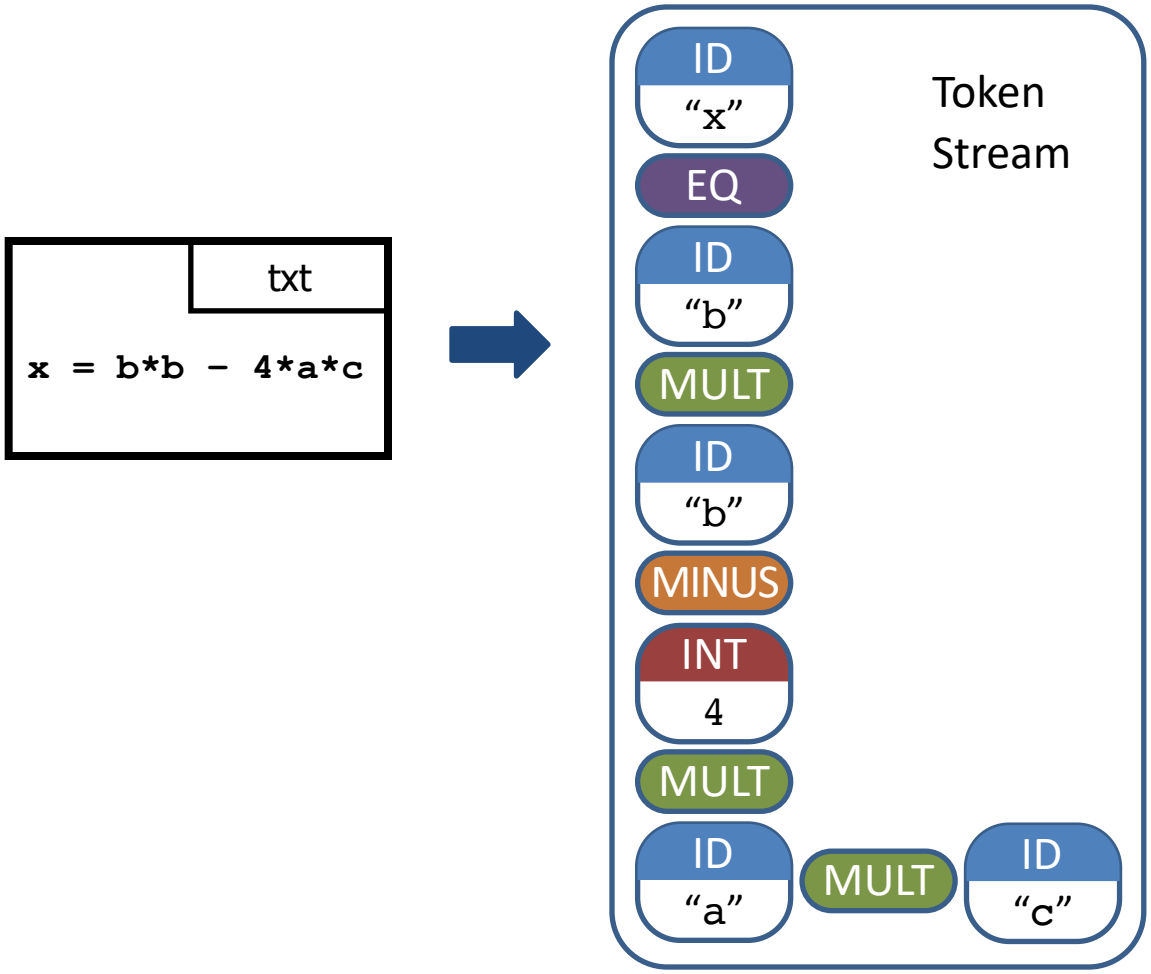
Course Overview



Journey inside a compiler

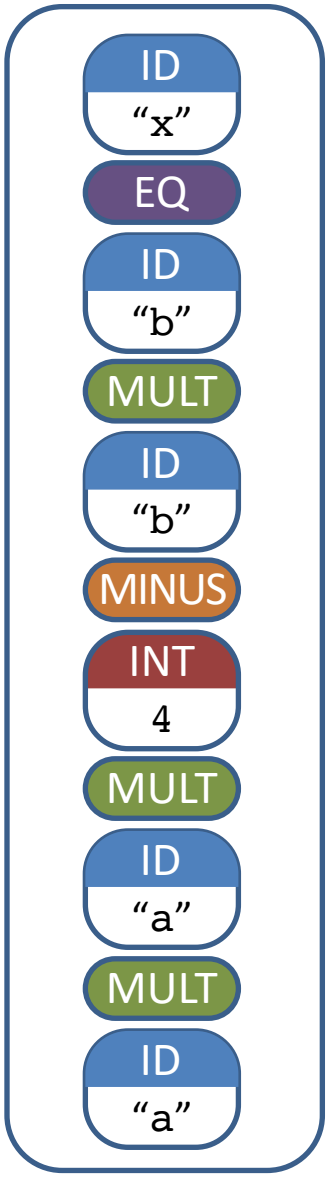


Journey inside a compiler



Lexical Analysis | Syntax Analysis | Semantic Analysis | IR Opt. | Code Gen.

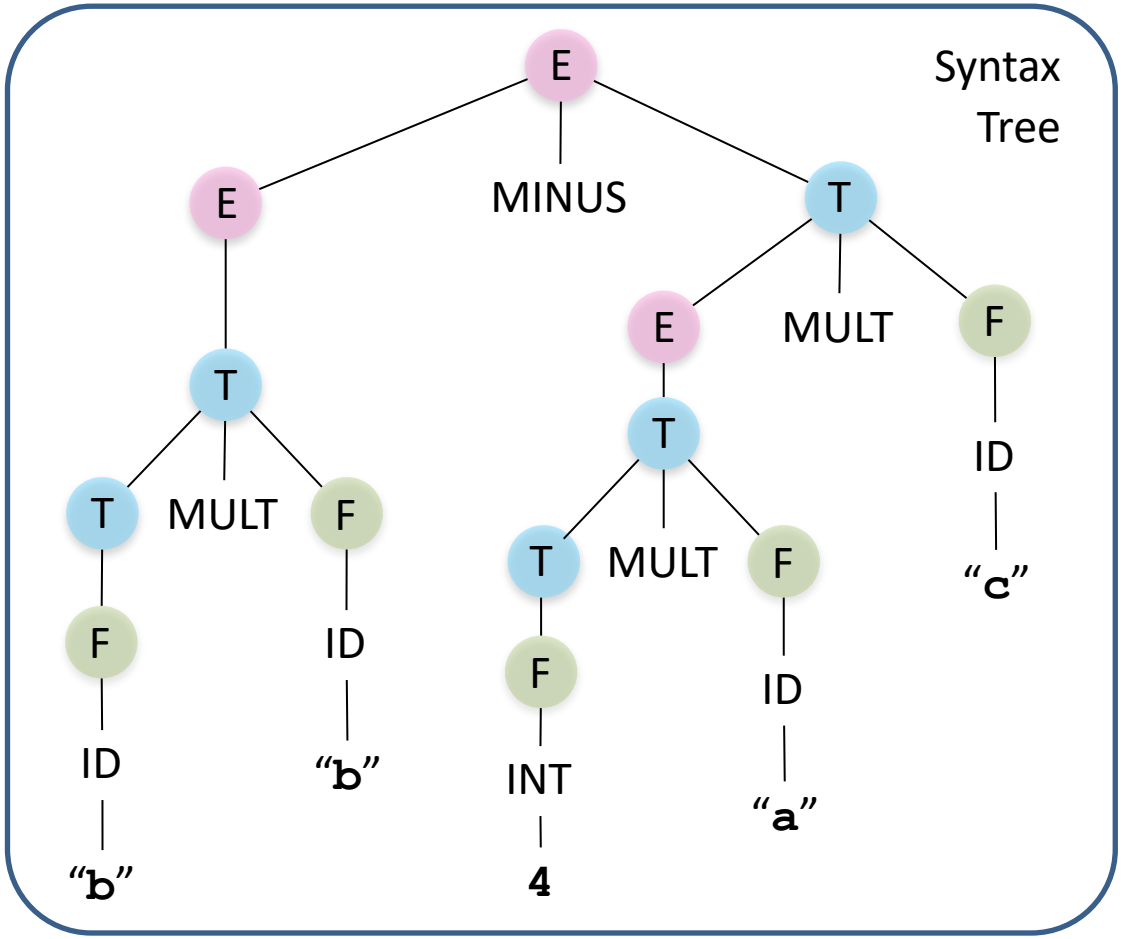
Journey inside a compiler



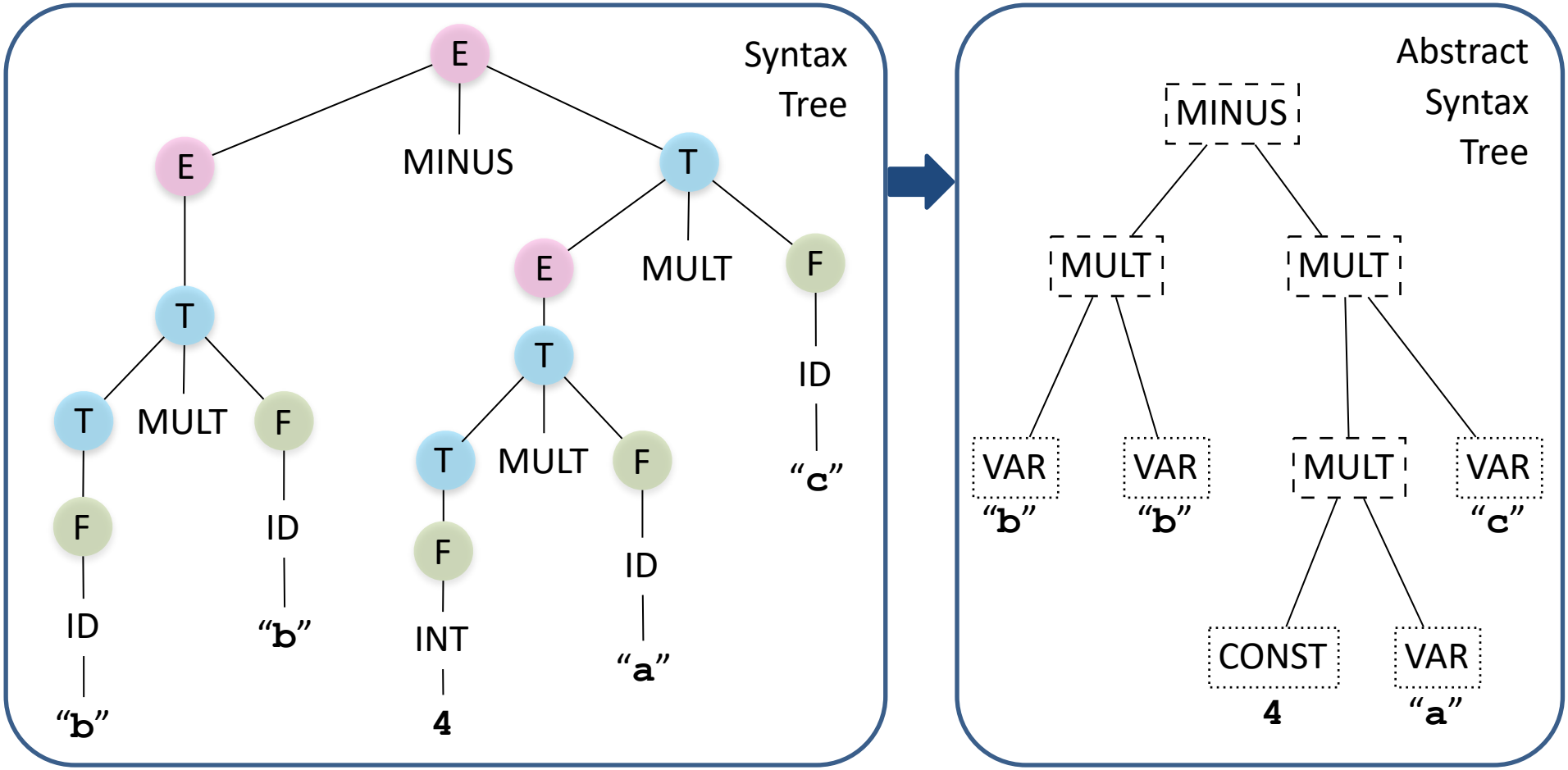
Token Stream

Grammar

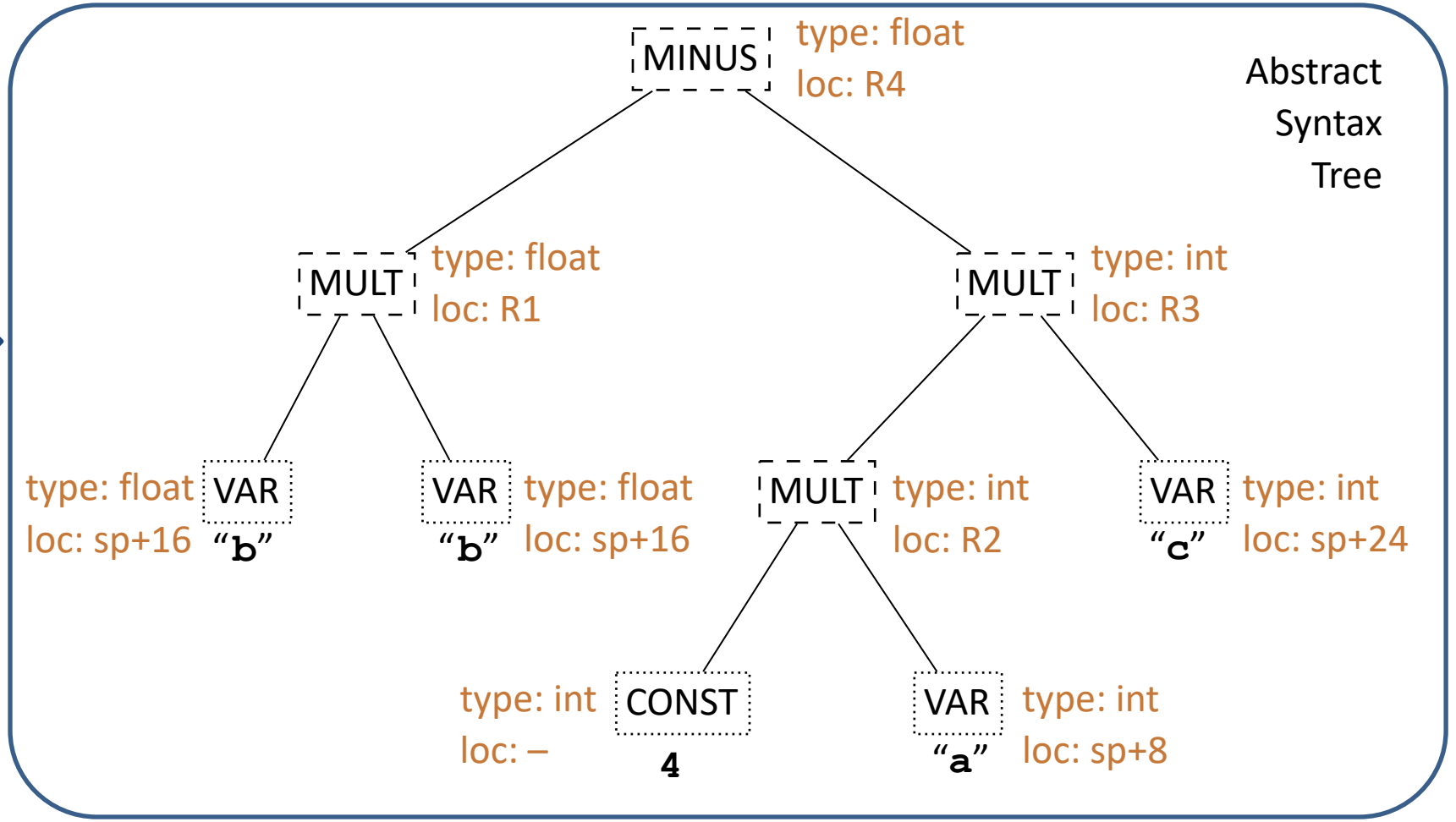
E ⟶ E (PLUS) T
E ⟶ E (MINUS) T
T ⟶ T (MULT) F
T ⟶ T (DIV) F
F ⟶ (ID)
F ⟶ (INT)



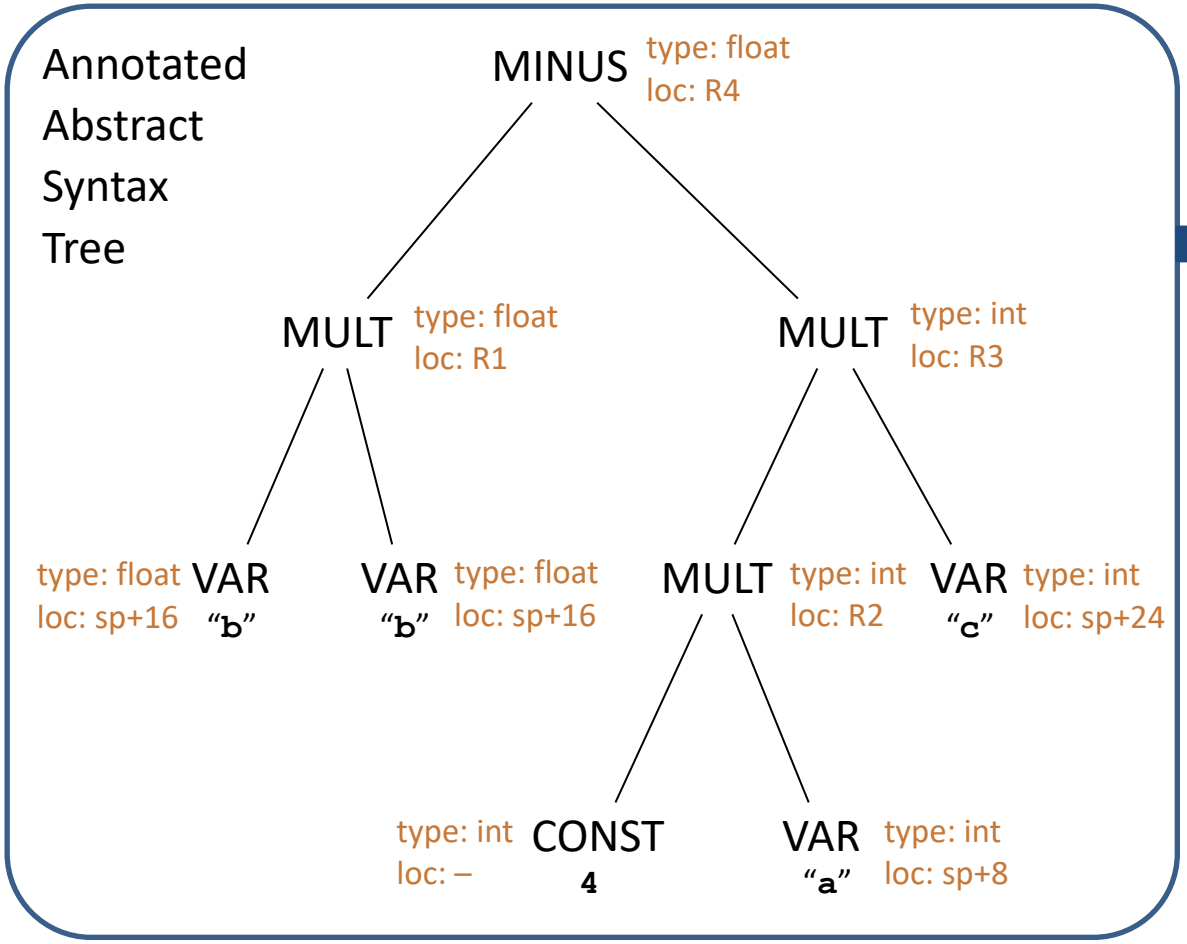
Journey inside a compiler

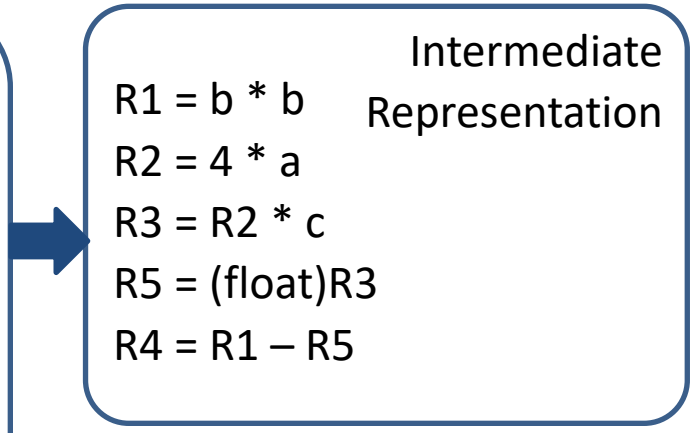


Journey inside a compiler

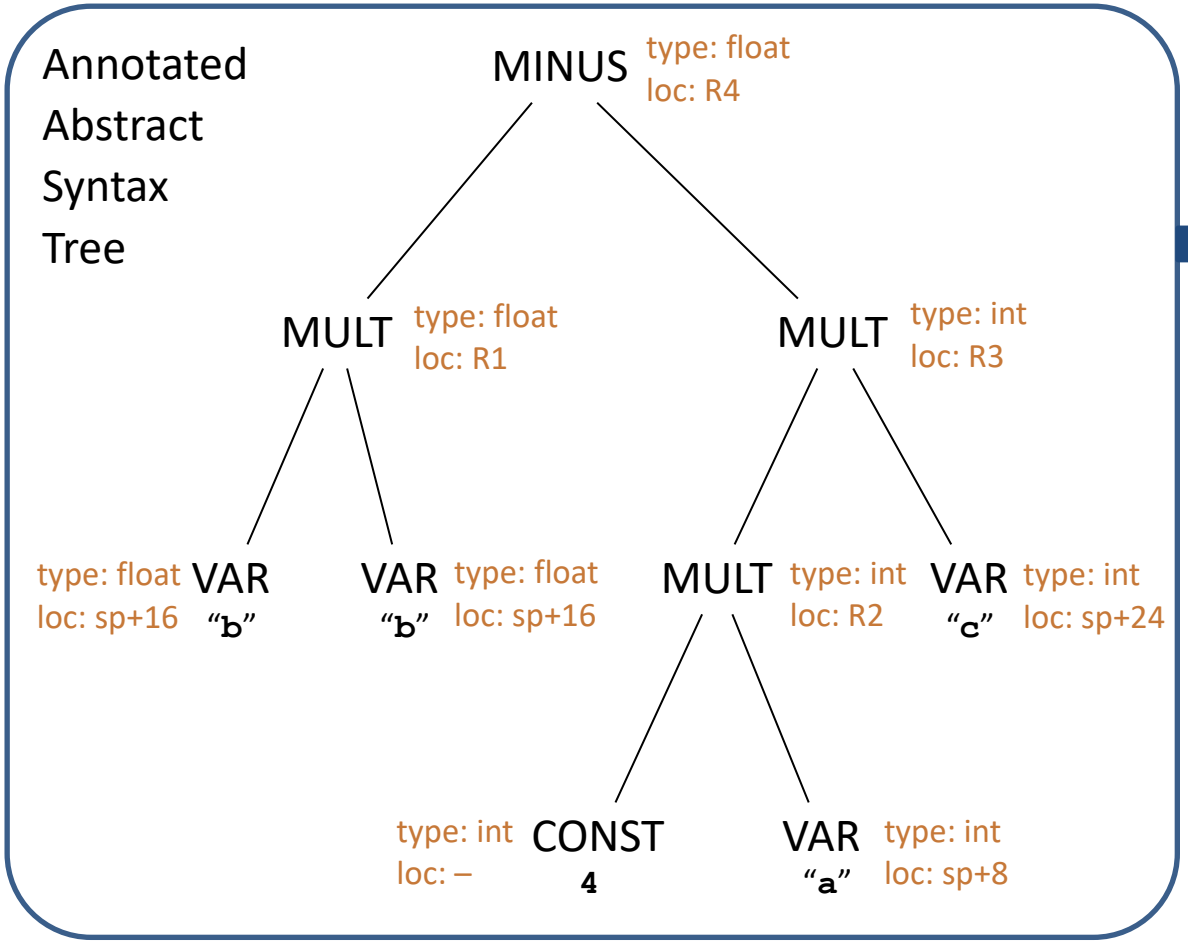


Journey inside a compiler





Journey inside a compiler



Intermediate Representation

```
R1 = b * b
R2 = 4 * a
R3 = R2 * c
R5 = (float)R3
R4 = R1 – R5
```

Assembly Code

```
fild    [esp+16]
fmul    st0, st0
mov     ebx, [esp+8]
sal     ebx, 2
mul     ebx, [sp+24]
fild    ebx
fsub    st0, st1
```

Lexical Analysis | Syntax Analysis | Semantic Analysis | IR Opt. | Code Gen.

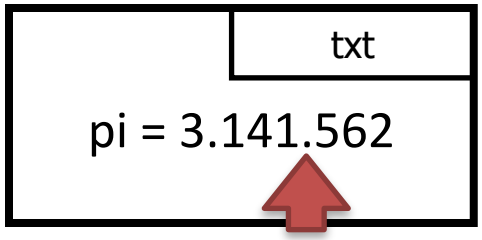
Error Checking

Check on every stage.
Report as soon as possible.

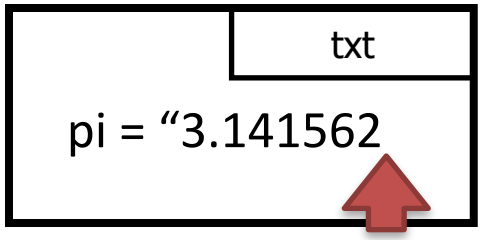
- Lexical analysis: illegal tokens
- Syntax analysis: illegal syntax
- Semantic analysis: incompatible types, undefined variables, …
- Even at runtime: division by zero, array bounds,...

- Every phase tries to recover and proceed with compilation

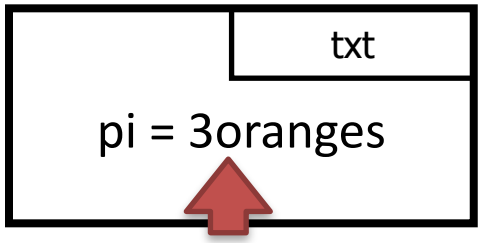
Lexical Errors



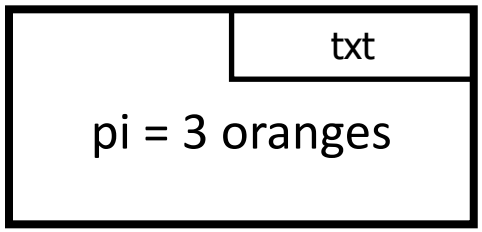
Illegal token



Illegal token (*unterminated string*)

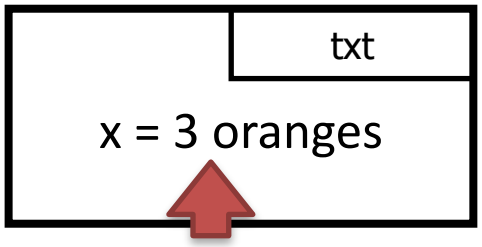


Illegal token (*malformed identifier*)

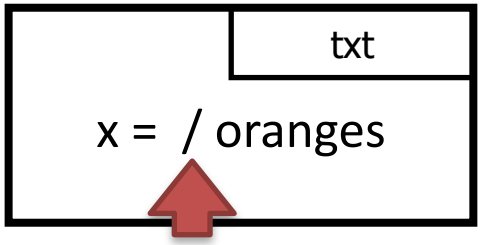


⟨ID,“`pi`”⟩, ⟨EQ⟩, ⟨INT, 3⟩, ⟨ID,“`oranges`”⟩

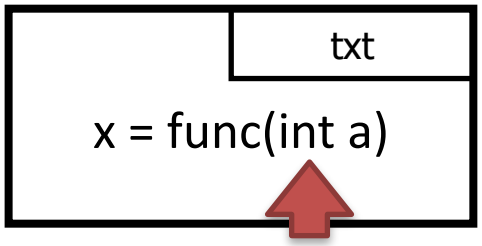
Syntax Errors



Missing operator

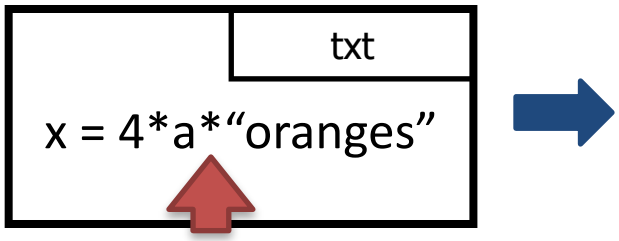


Wrong number of arguments to operator “/”

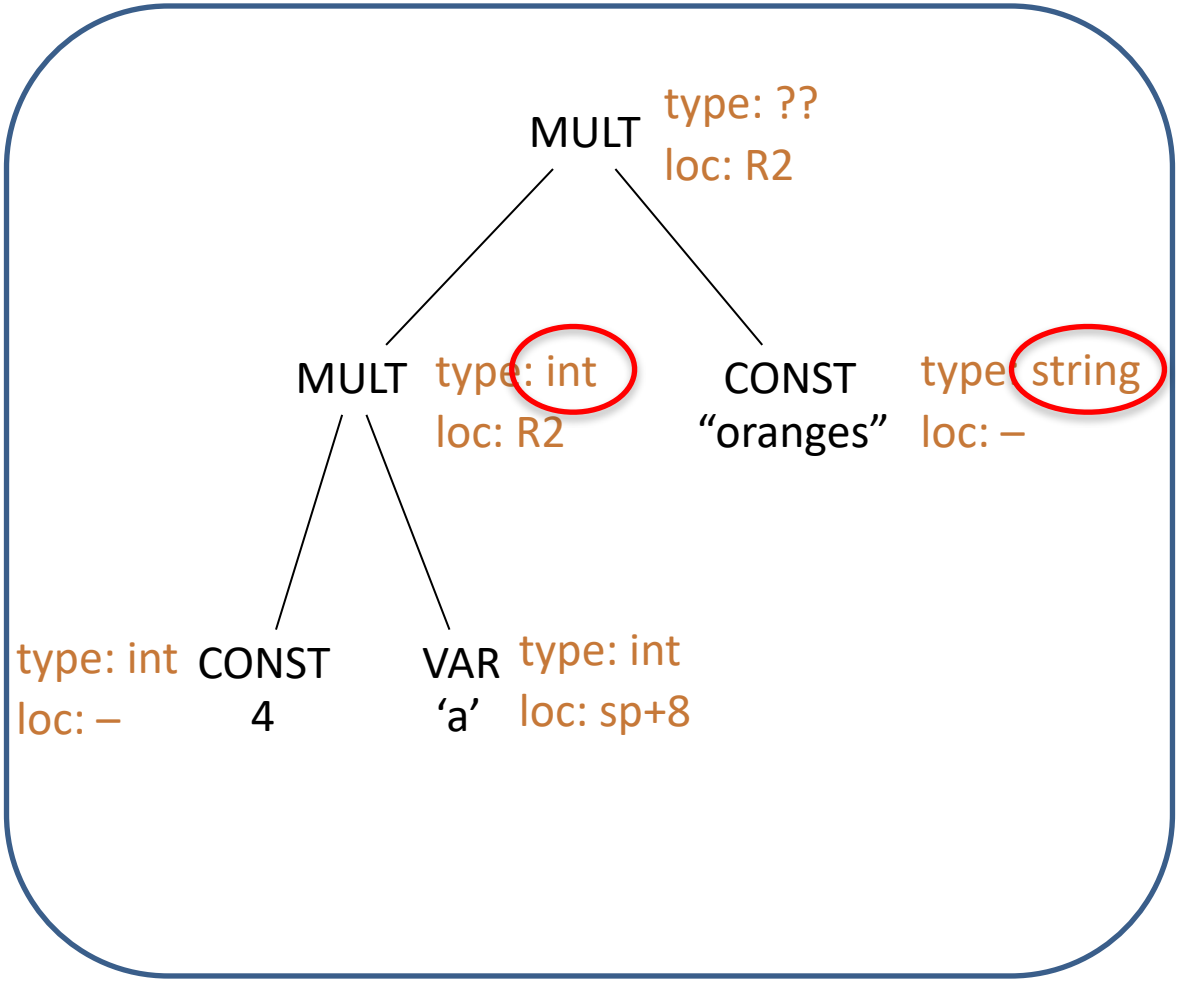


A declaration is not expected inside a call

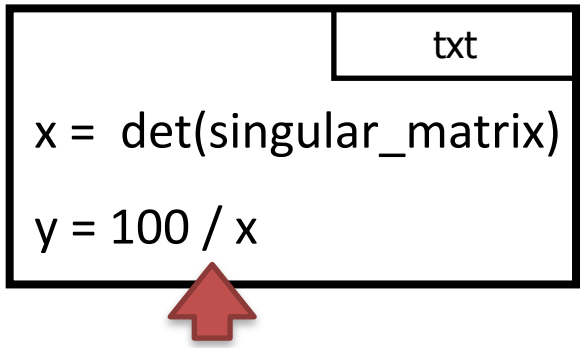
Semantic Errors: Type Checking



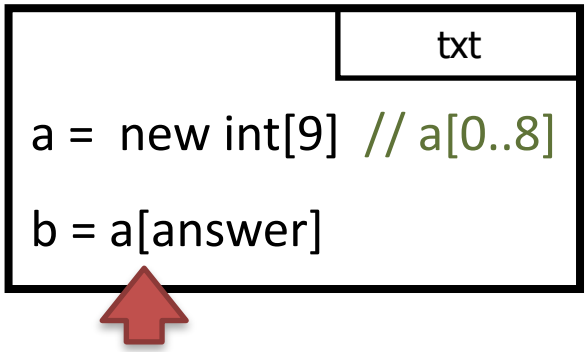
Type mismatch



Runtime Errors

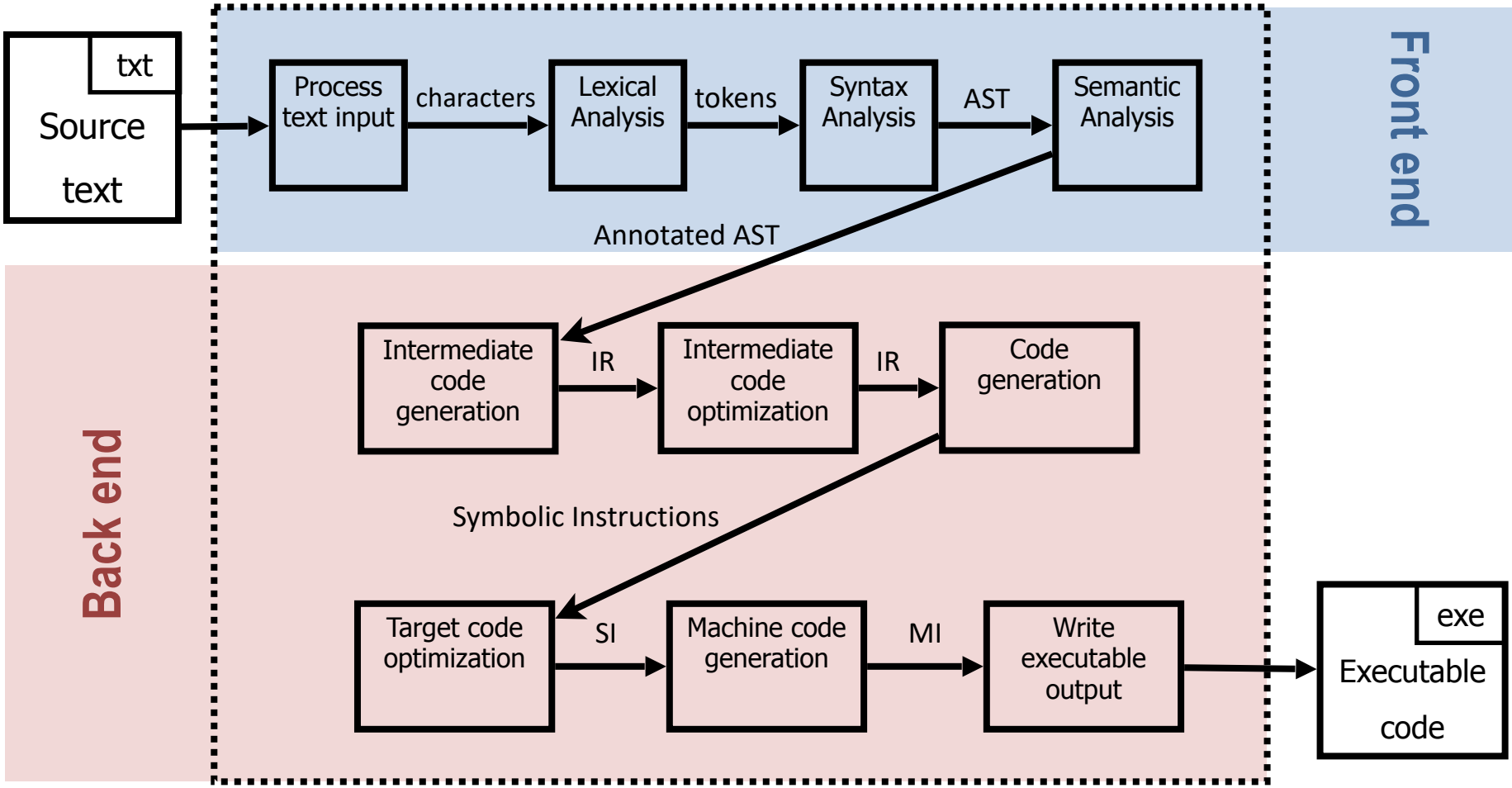


Division by zero



Array index out of bounds: 42 > 8

Detailed Anatomy of a Compiler



Compiler Construction Toolset

Summary

- ✓ Compiler = a program that translates code from source language (high level) to target language (low level)
- ✓ Compiler constructed from modular phases
  - Reusable components
  - Different front/back end combinations
- ✓ Compilers play a critical role
  - Bridge programming languages to the machine
  - Many useful techniques and algorithms
  - Many useful tools (e.g., lexer/parser generators)

Next