

# PROGRAMMING LANGUAGES FOUNDATIONS

## Computer Science III

Author: Eng. Carlos Andrés Sierra, M.Sc.  
[cavirguezs@udistrital.edu.co](mailto:cavirguezs@udistrital.edu.co)

Lecturer  
Computer Engineer  
School of Engineering  
Universidad Distrital Francisco José de Caldas

2024-III



# Outline

1 Programming Languages

2 Finite State Machines

3 Generative Grammars



Alan Turing

Noam Chomsky



# Outline

1 Programming Languages

2 Finite State Machines

3 Generative Grammars



# Babbage Machine

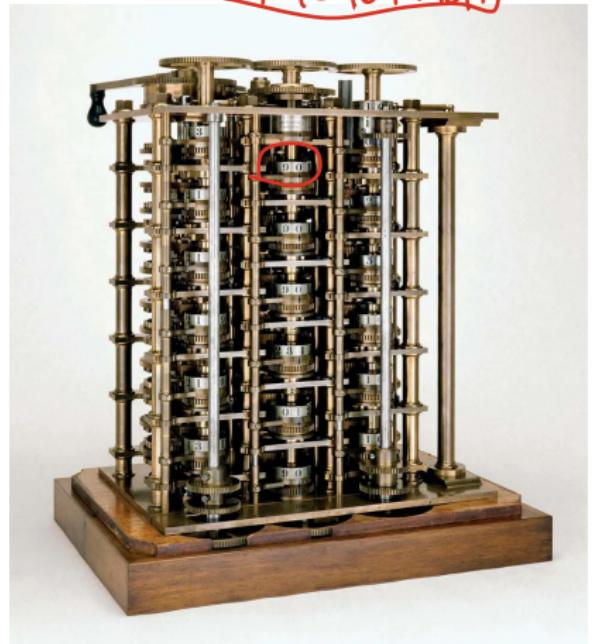


Figure: Analytical Machine



0 1 2 3 4 5 6 7 8 9 → métodos numéricos

- Charles Babbage (1791 — 1871) was an English mathematician, philosopher, inventor, and mechanical engineer.
- He originated the concept of a digital programmable computer.
- Considered the “father of the computer”. He creates the Analytical Engine.
- The Analytical Engine was a general-purpose mechanical computer.



# Ada Lovelace

- Ada Lovelace (1815 — 1852) was an English mathematician and writer. } Tops
- She is known for her work on Charles Babbage's early mechanical general-purpose computer, the Analytical Engine.
- She was the first to recognize that the machine had applications beyond pure calculation, and to have published the first algorithm intended to be carried out by such a machine.



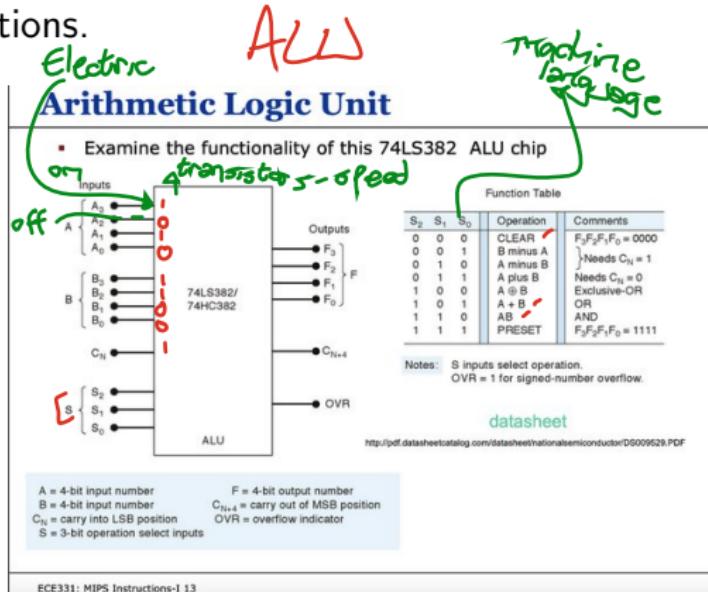
Figure: Ada Lovelace



## Physical Binary Language

Leibniz (1600s)

- **Binary** is a base-2 number system. It uses only **two symbols**: typically off-0 (zero) and on-1 (one).
  - The **bit** is the **basic unit** of information in computing and digital communications.



# Bits and Bytes

Byte  $\Rightarrow$  Binary Array - Decimal

$0101\ 1110\ 0001 \cdot 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1000$

$\downarrow$

$\dots + 128 + 64 + 32 + 16 + 8 + 4 + 1 = 255$

$$\begin{array}{r} 1+2+4+8+16+32+64+128=255 \\ 0+0+0-\dots-\dots+0=0 \end{array} \quad \boxed{256} \quad \text{Poss. 0..1.000}$$

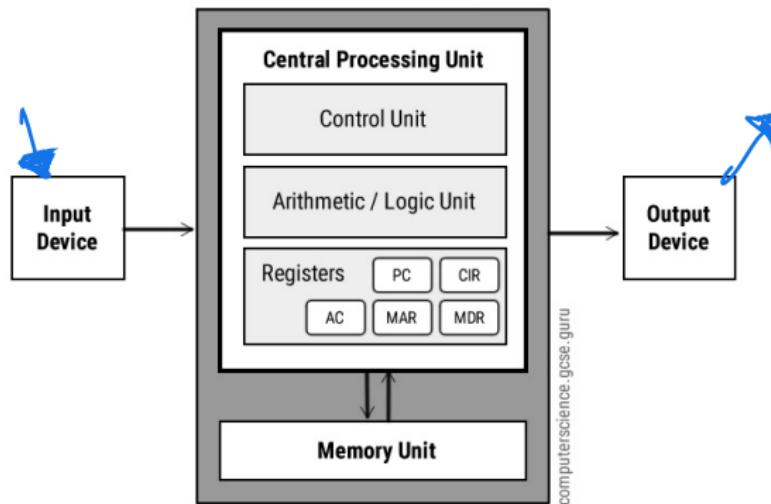
ASCII  $\Rightarrow$  Unicode  $\Rightarrow$  Hexadecimal

byte | dec | symbol

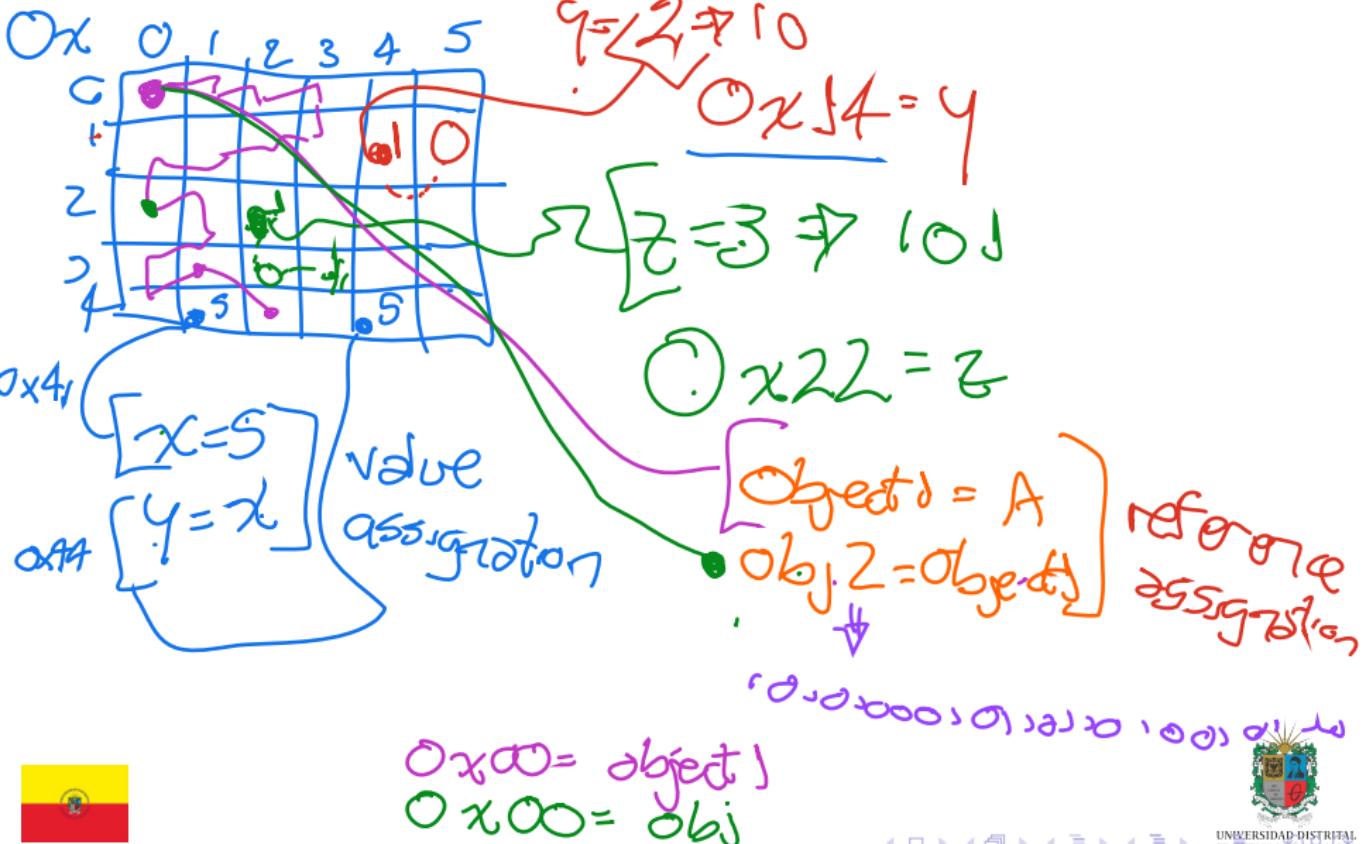


# Von Neumann Architecture

- The **Von Neumann Architecture** is a computer architecture based on the stored-program computer concept.
- The **design** is based on the concept of an instruction set.
- The **program** and **data** are both stored in the same memory unit.



# Memory and Bit Storage



# Machine Programming Language

$$\begin{array}{r} \overline{101} \\ \overline{5} \\ + \quad \overline{111} \\ \hline \end{array}$$

Physical Machine

$\oplus$

Digital Electronic

a - 10110

b - 10111

c - 10101

d - 110000

e - 11010

f - 111100

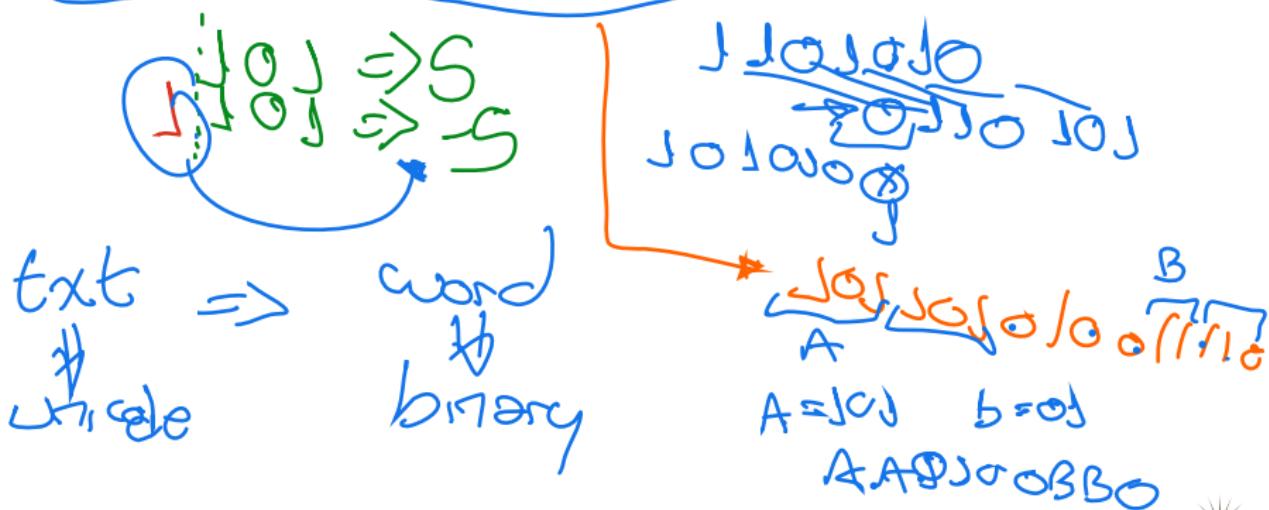
g - 110100

50 Mgbps  $\Rightarrow$  50.000.000 bytes/s  
400.000.000 bits/s



# Bit Operations

- **Bitwise operations** are operations that directly manipulate bits.
- They are used in low-level programming for performing calculations, file processing, and data compression.



# Assembly Programming Language

*C5X OxJ24 voice*

```

1 ; Example of a basic conditional structure in x86
2 assembly
3 f eax=10
4 cmp eax, 10      ; Compare the value in eax with 10
5 je equal_label   ; Jump to equal_label if eax is equal to
6 10
7
8 ; Code for not equal case
9 jmp end_label    ; Jump to end_label to avoid executing the
10 equal case code
11
12 equal_label:
13 ; Code for equal case
14 jmp $6
15
16 end_label:
17 ; Continue execution

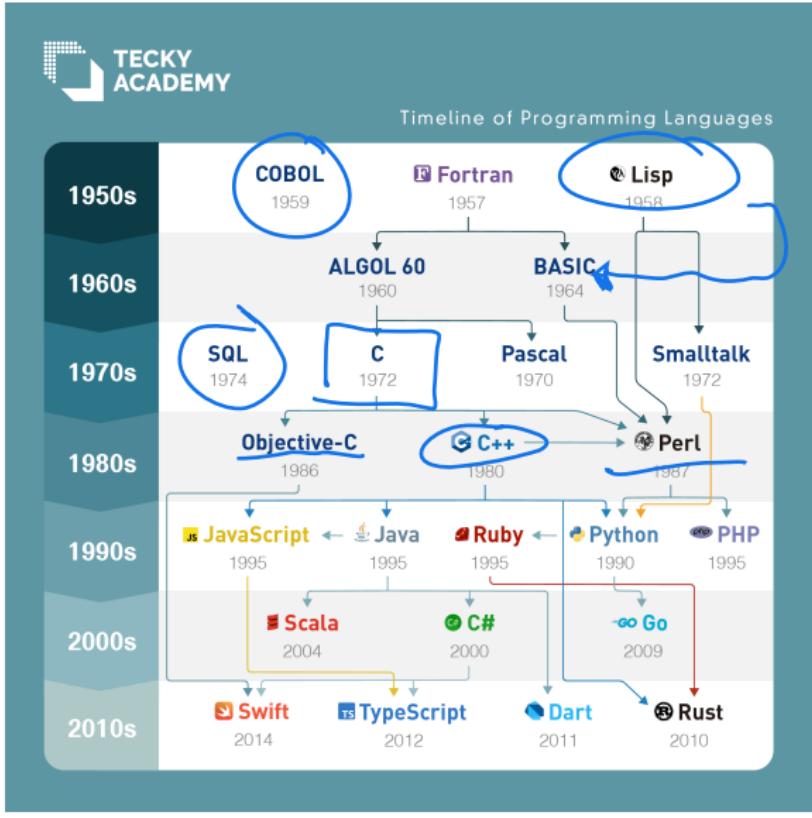
```

$\Rightarrow$  if (eax == 10) {  
    ...  
} else {  
    ...  
}

Listing 1: Basic Conditional Structure in Assembly

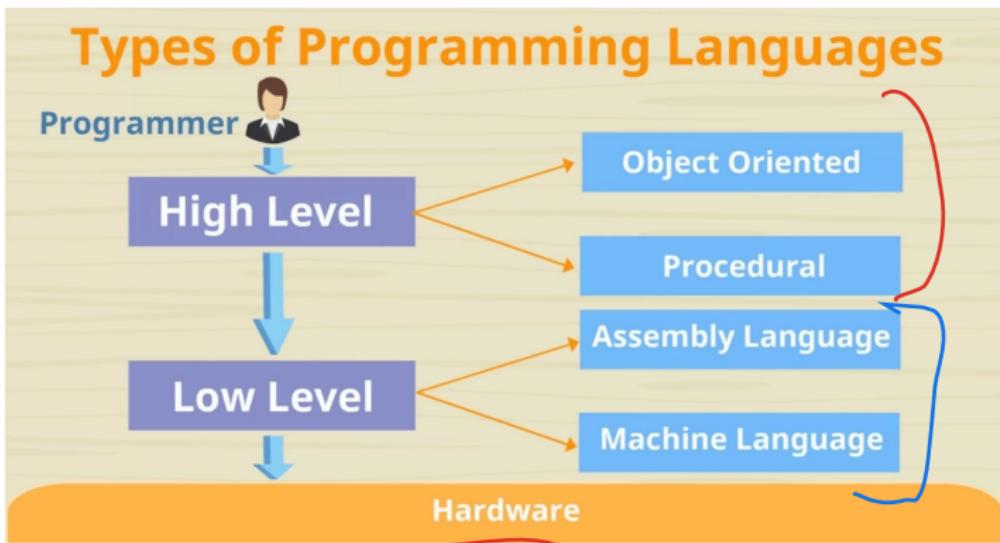


# History of Programming Languages



# High-Level Programming Languages I

## High-Level vs Low-Level



**inprogrammer**



# High-Level Programming Languages II

## Purpose of High-Level Languages

### Ease of Use

- Simplifying programming
- Minimizing learning curve
- Enhancing productivity
- Automated memory management
- Clear syntax
- Readability and maintainability



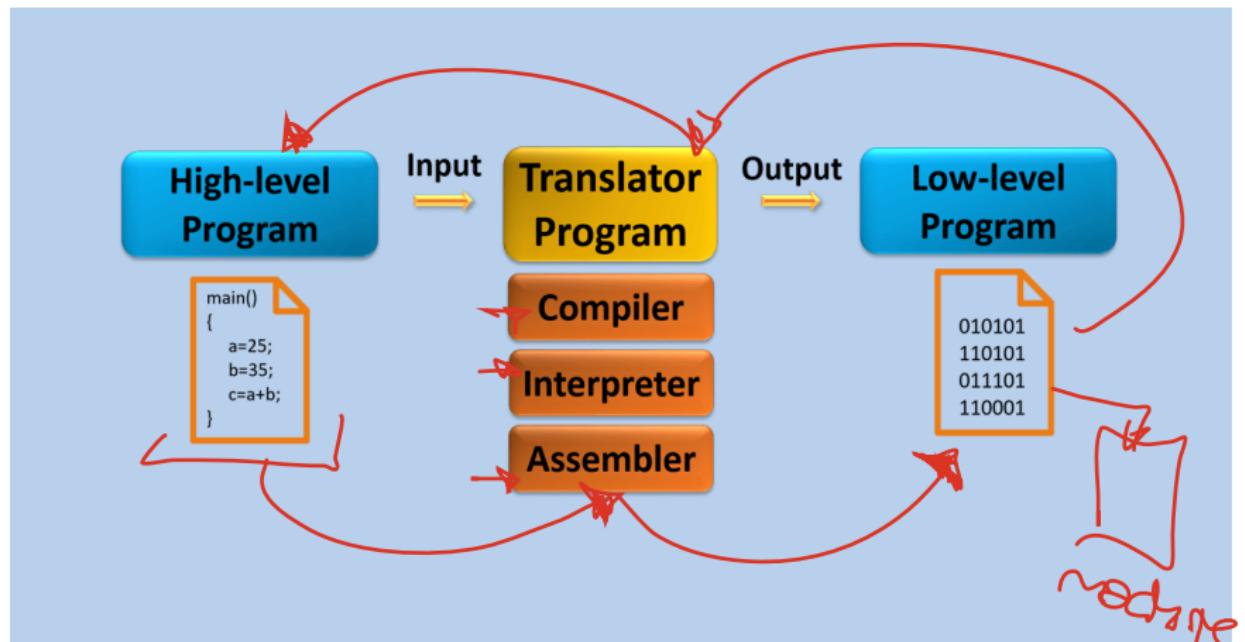
### Portability Across Systems

- Cross-platform compatibility
- Utilization of compilers and interpreters
- Seamless execution on various platforms
- Reduction of platform-specific modifications
- Enhanced flexibility across environments

Techopedia



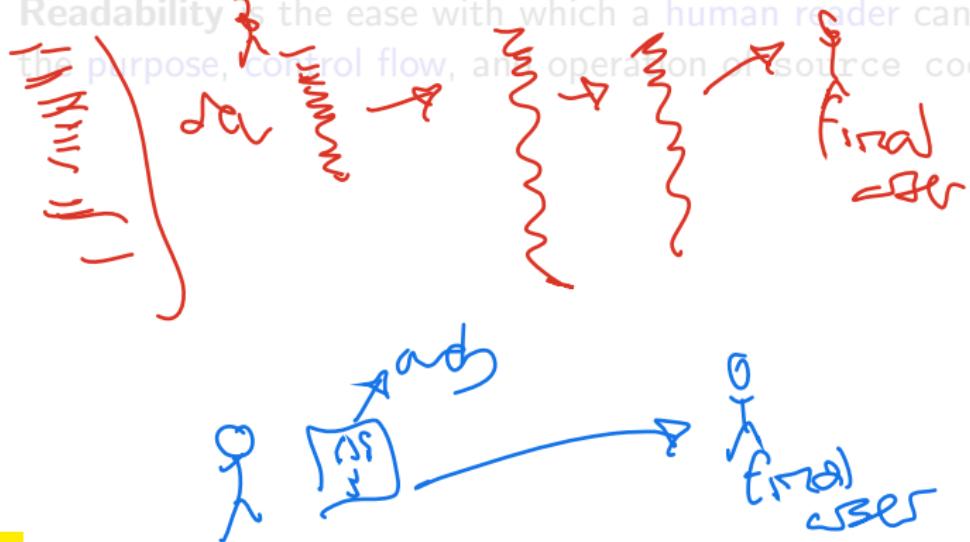
# Translation Process



# Efficiency and Readability

- **Efficiency** is the ability to avoid wasting materials, energy, efforts, money, and time in doing something or in producing a desired result.

- Readability is the ease with which a human reader can understand the purpose, control flow, and operation of a program code.



# Efficiency and Readability

- **Efficiency** is the ability to avoid **wasting materials, energy, efforts, money, and time** in doing something or in producing a desired result.
- **Readability** is the ease with which a **human reader** can understand the **purpose, control flow, and operation** of source code.

$f(x=2)\{$   
     $y=3$   
}

op ex, 2  
je ~  
y ASG @x 123  
y ADD 3  
j9



# Outline

1 Programming Languages

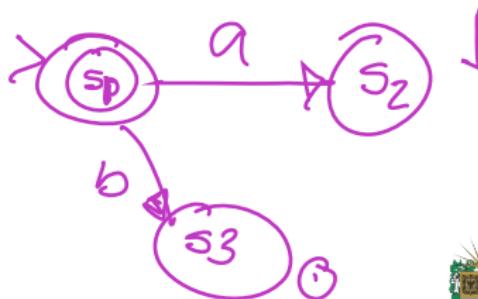
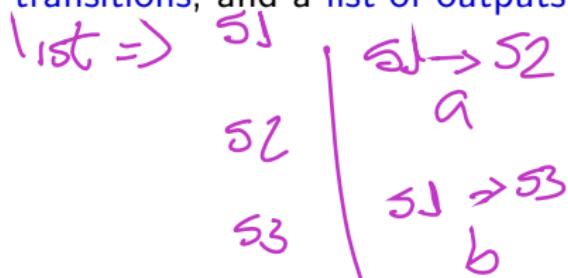
2 Finite State Machines

3 Generative Grammars



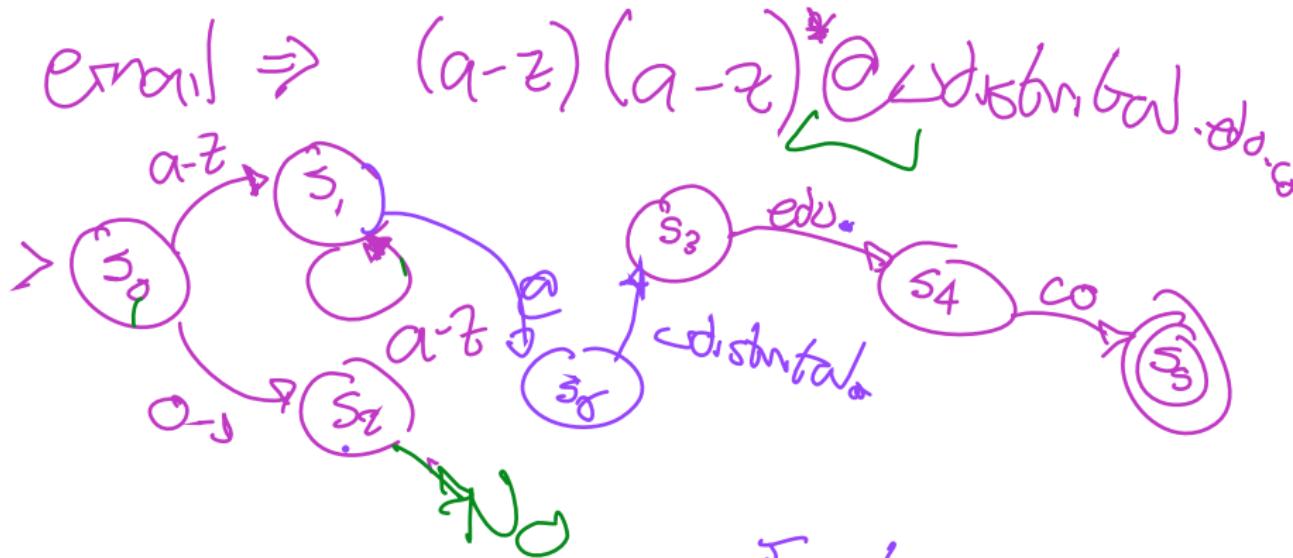
# Finite State Machines

- A **finite-state machine** (FSM) is a **mathematical model of computation**. *1920's*
- It is an **abstract machine** that can be in **exactly one of a finite number of states** at any given time.
- The FSM can change from one **state** to another in response to some **external inputs**.
- The FSM is defined by a **list of states**, a **list of inputs**, a **list of transitions**, and a **list of outputs**.





## Turing Master



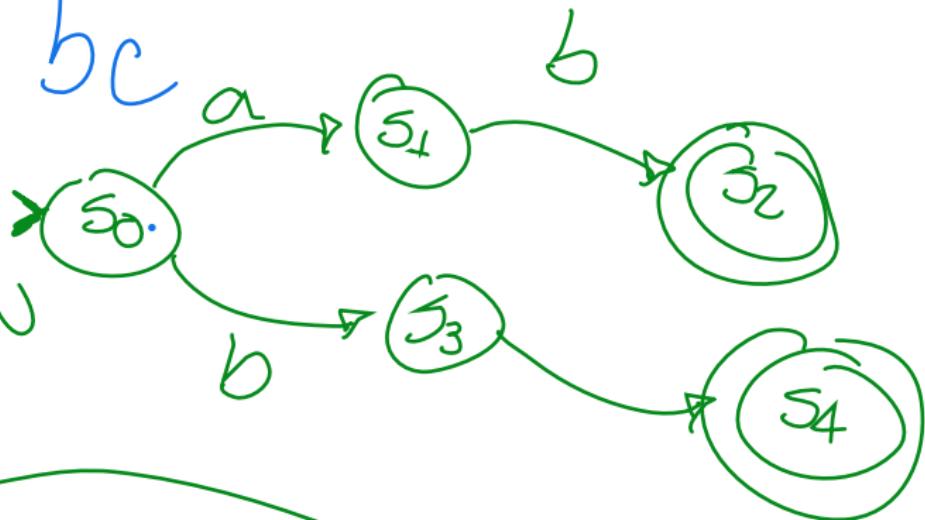
Finite  
Automata



## Finite Automata: Union, Kleene Star

$ab \cup bc$   
(or)

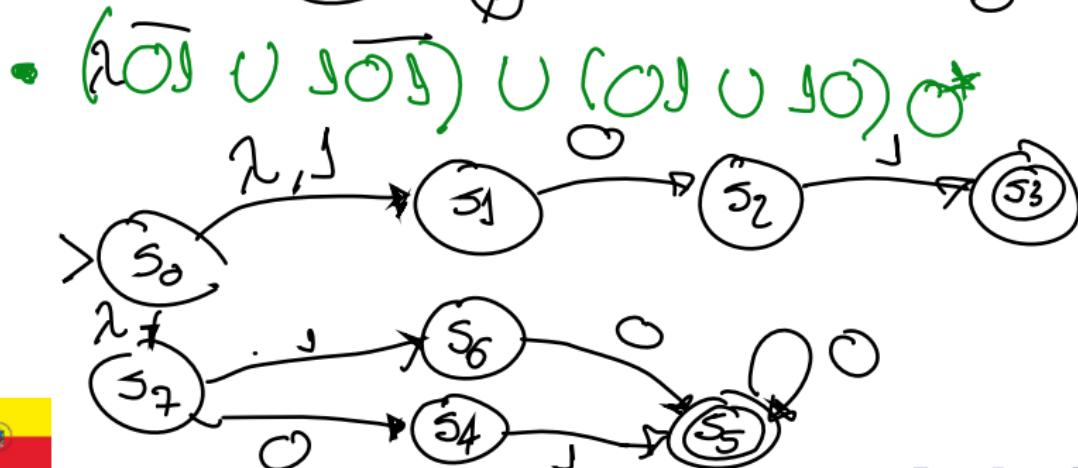
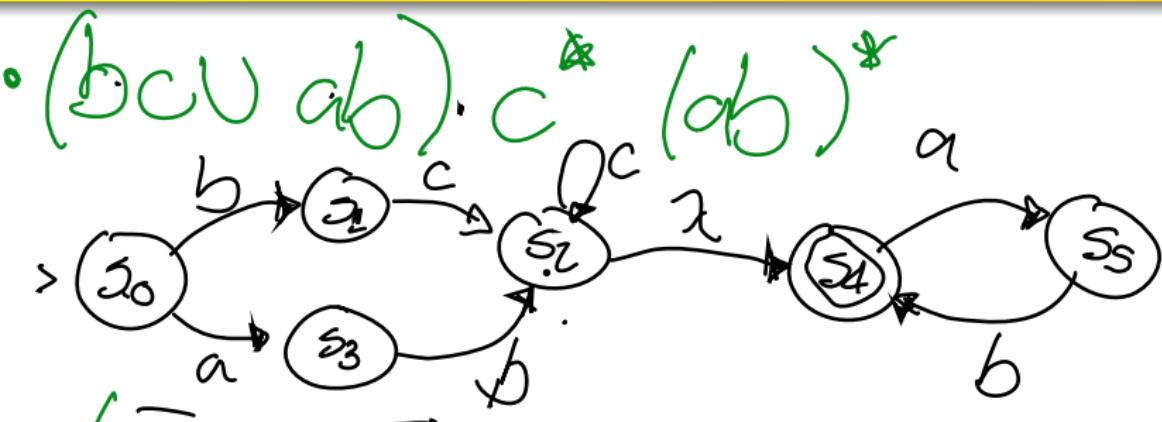
UNION



$ab^*a$



# Regular Expressions



# Strings Processing



# Alan Turing

- **Alan Turing** (1912 — 1954) was an English mathematician, computer scientist, logician, cryptanalyst, philosopher, and theoretical biologist.
- He is widely considered to be the **father of theoretical computer science** and **artificial intelligence**.
- He was highly influential in the development of **theoretical computer science**, providing a formalization of the concepts of **algorithm** and **computation** with the **Turing machine**.



Figure: Alan Turing



# Turing Machine



# Universal Turing Machine



# Outline

1 Programming Languages

2 Finite State Machines

3 Generative Grammars



# Noam Chomsky

- Noam Chomsky (1928 — ) is an American linguist, philosopher, cognitive scientist, historian, social critic, and political activist.
- He is considered the **father of modern linguistics**.
- He introduced the **Chomsky hierarchy**, a classification of formal languages.



Figure: Noam Chomsky



# Natural Processing Language



# Formal Languages



# Grammars Foundations



# Chomsky Formal Norm



# Regular Expressions



# Equivalence between Gramma and Finite Automatas



# Grammars Example



# Outline

1 Programming Languages

2 Finite State Machines

3 Generative Grammars



# Thanks!

## Questions?



Repo: <https://github.com/EngAndres/ud-public/tree/main/courses/computer-science-iii>

