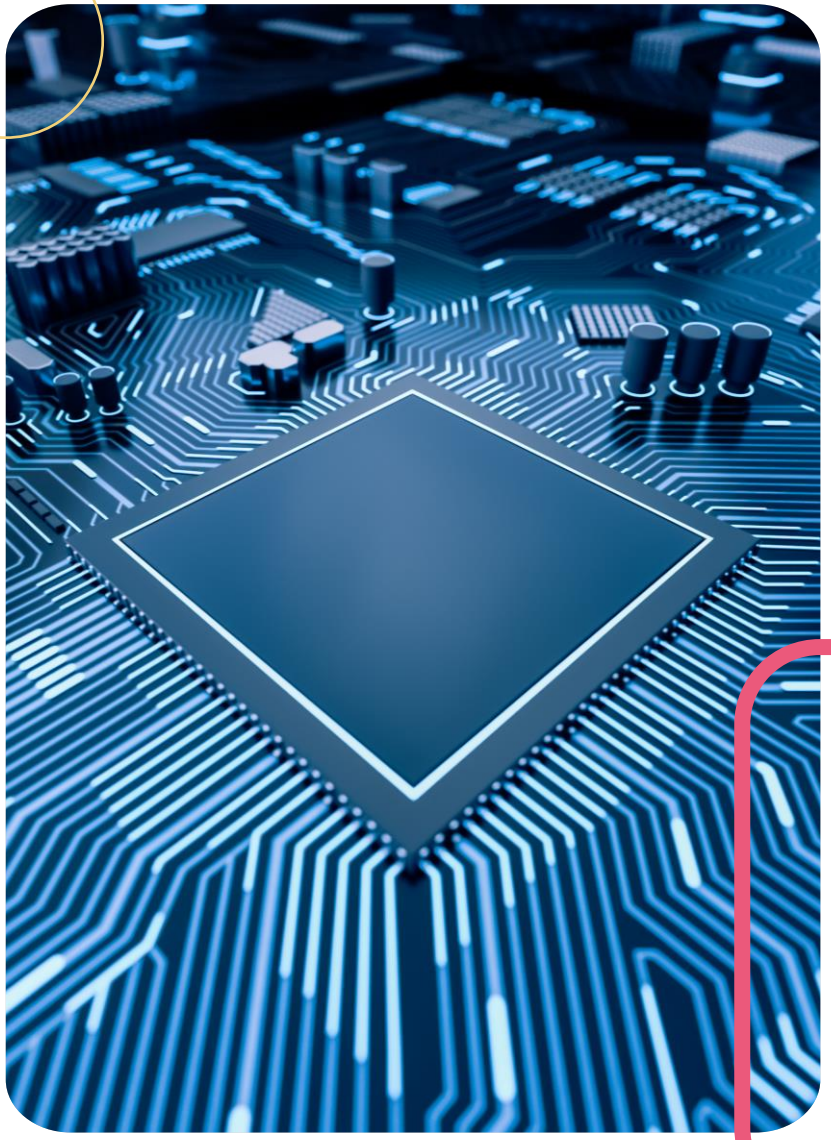
Diploma in

Computer Science

The language of computers

Question

Have you thought about your next step?



Objectives

- Explore language as a means of communication
- Discuss how computer components communicate
- Explore the theory of programming language and modelling using abstract machines

DID YOU KNOW?

“Typewriter” is the longest word that you can write using only the letters of the first row of a computer keyboard.

Human language

Natural language

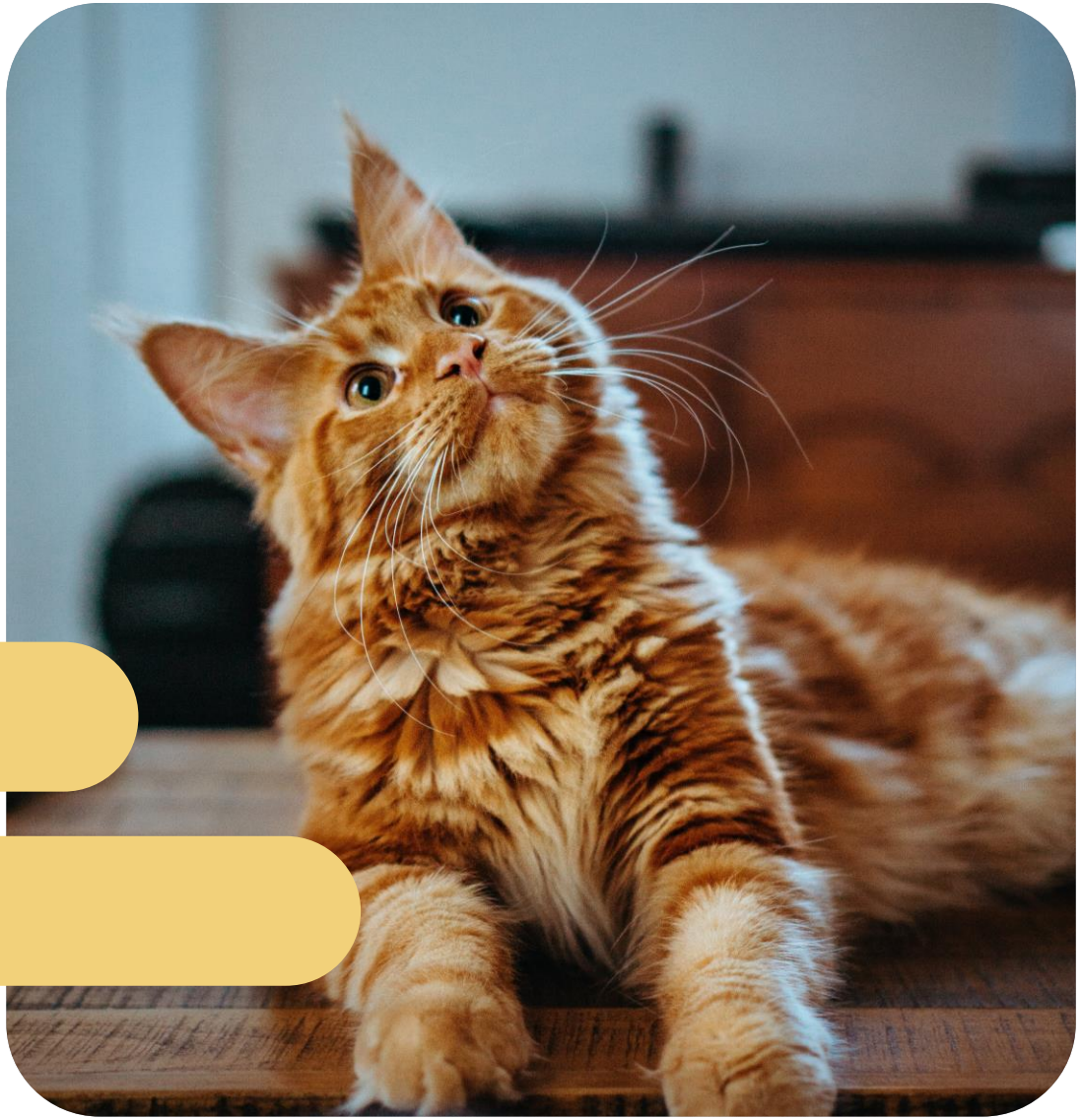
- In Computer Science, natural language means human language.
- Computer Science tries to make computers completely understand human language.
- Computers still struggle with human language.

Natural human language

- Language is a system of spoken, manual or written symbols that humans use to express themselves, their identity, imagination and emotions.
- The main function of language is communication.
- The same principle applies to computers.

Natural human language

- Structure is common to all languages on Earth.
- Language structure is primarily divided into syntax and semantics.
- Language structure is important; it can make or break your program!

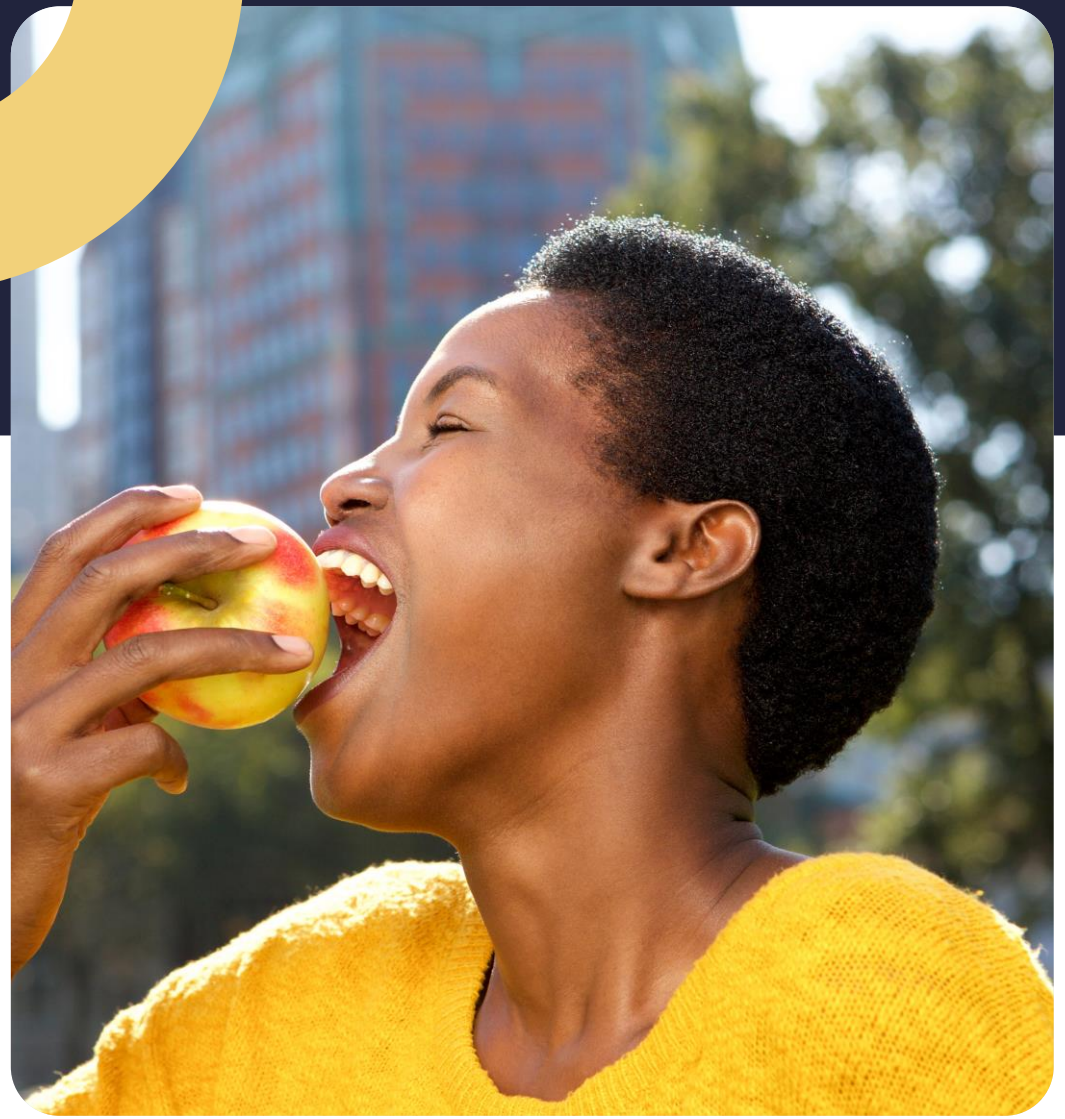


Language structure

- **Syntax** is the set of rules that tells us how to arrange words and combine phrases.
- **Semantics** is the meaning of a word or phrase and its connection to a certain concept.
- Syntax and semantics work together to create statements.

Language construction

- Morphology is the study of words and their information.
- Understanding how languages work is important in programming.



Human language example

Ambiguity

I didn’t say we should eat it.
I **didn’t** say we should eat it.
I didn’t **say** we should eat it.
I didn’t say **we** should eat it.
I didn’t say we **should** eat it.
I didn’t say we should **eat** it.
I didn’t say we should eat **it.**

- In computer language, every statement has to mean only one thing.
- Computer code is a group of instructions and instructions should be as clear as possible.

Clarity

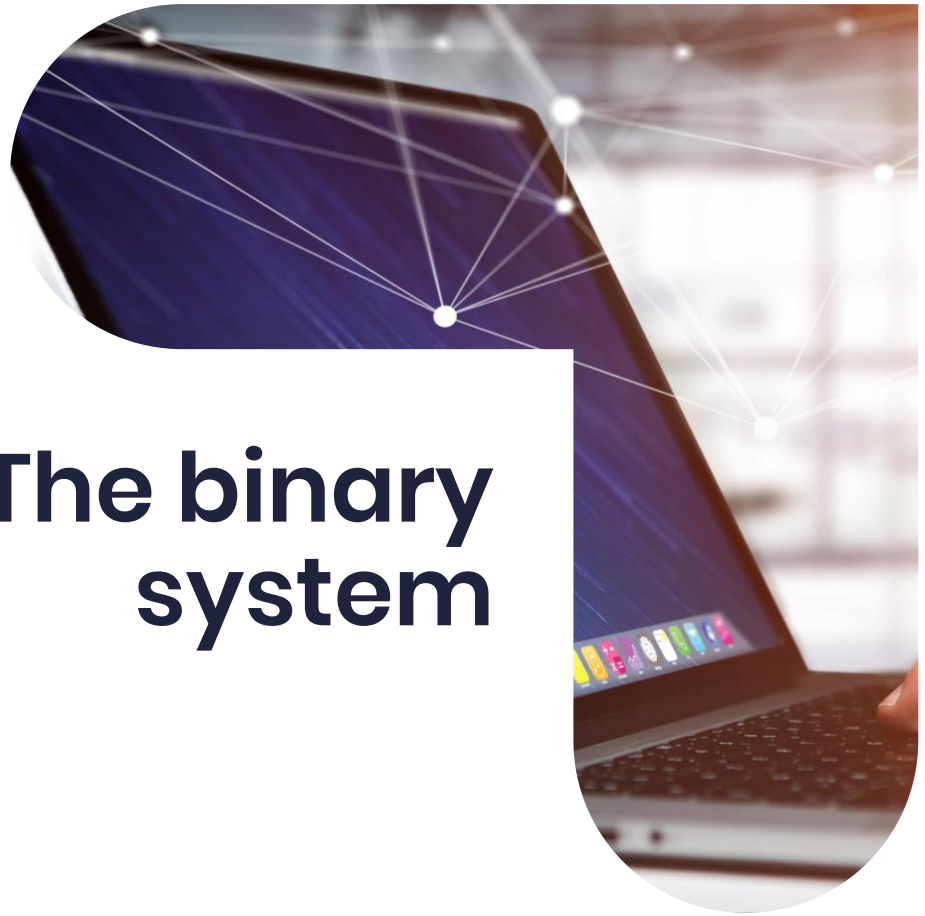
Background to computer language

- Programming language is a lot simpler than human language.
- Programming languages were originally nothing more than a whole lot of ones and zeros.

Computer language

The binary system

- Computers use binary codes to represent characters, images, and sound.
- A well-known binary code is ASCII. Here, a 7-bit binary code is used to represent text and other characters.
- In computing, even zero is counted!

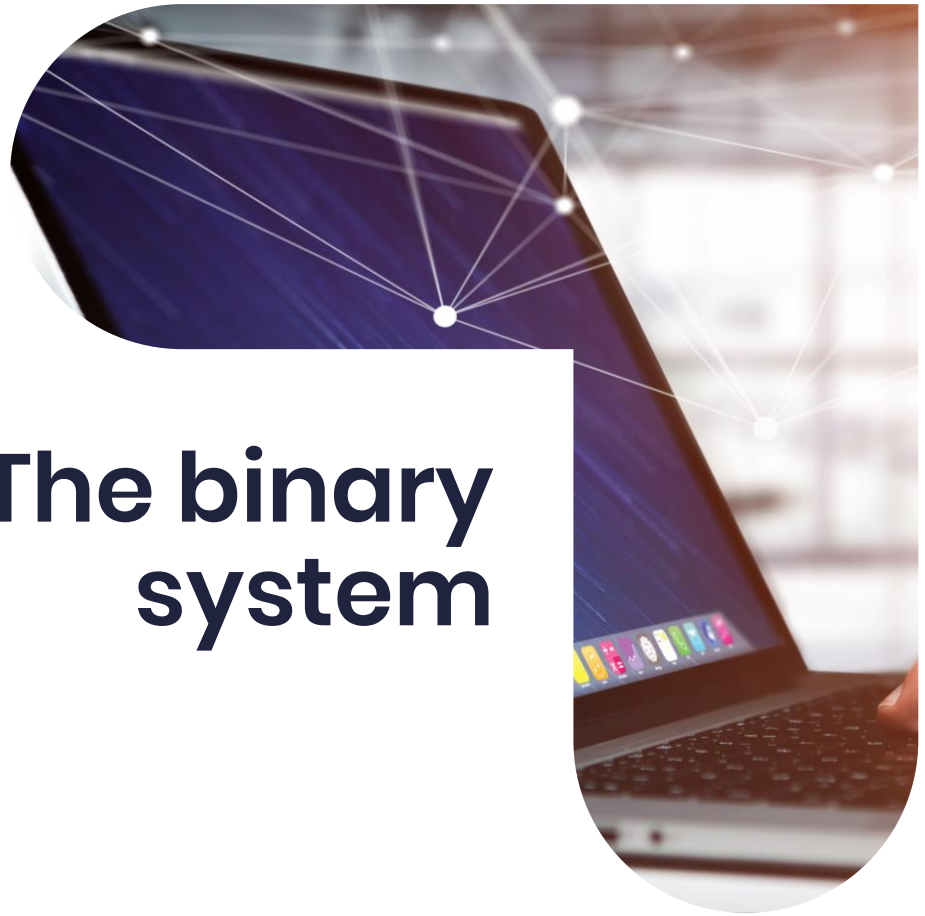


The binary system

- ASCII only represents 128 characters.
- Unicode comes in four forms: UTF-7, UTF-8, UTF-16, and UTF-32.
- UTF stands for Unicode Transformation Format.

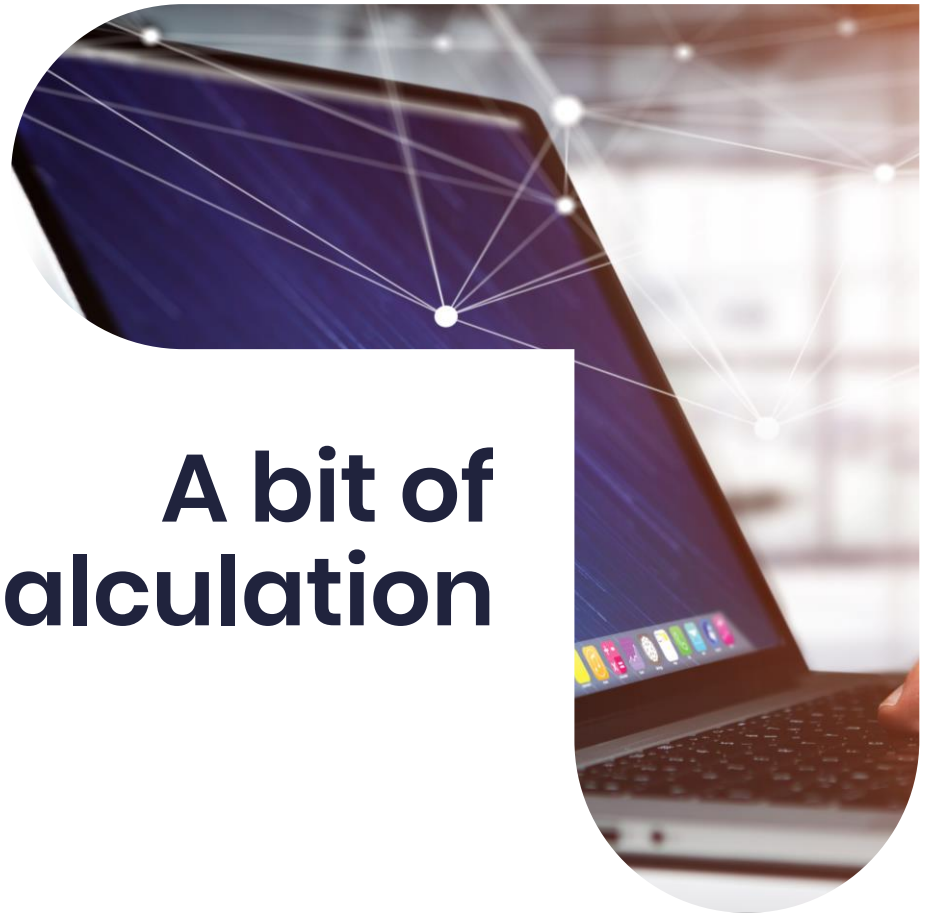
The binary system

- UTF-7 was designed to represent ASCII characters in email messages that required Unicode encoding.
- UTF-8 uses 8 bits to maximise compatibility with ASCII, but allows for variable-width encoding expanding to 16, 24, 32, 40 or 48. It is the most popular type of Unicode encoding.



The binary system

- UTF-16 is less flexible, starting off at 16 bits with the ability to extend to 32 bits. It can represent up to one million characters.
- ITF-32 is as rigid as ASCII; it can represent a whole lot of characters.



A bit of calculation

To find out how many characters a particular binary code can hold, you raise 2 to the number of bits that the binary code uses.

For example:

- ASCII uses 7 bits, so 2^7 gives us 128.
- UTF-32, 2^32 gives us a whopping 4.2 billion characters!

DID YOU KNOW?

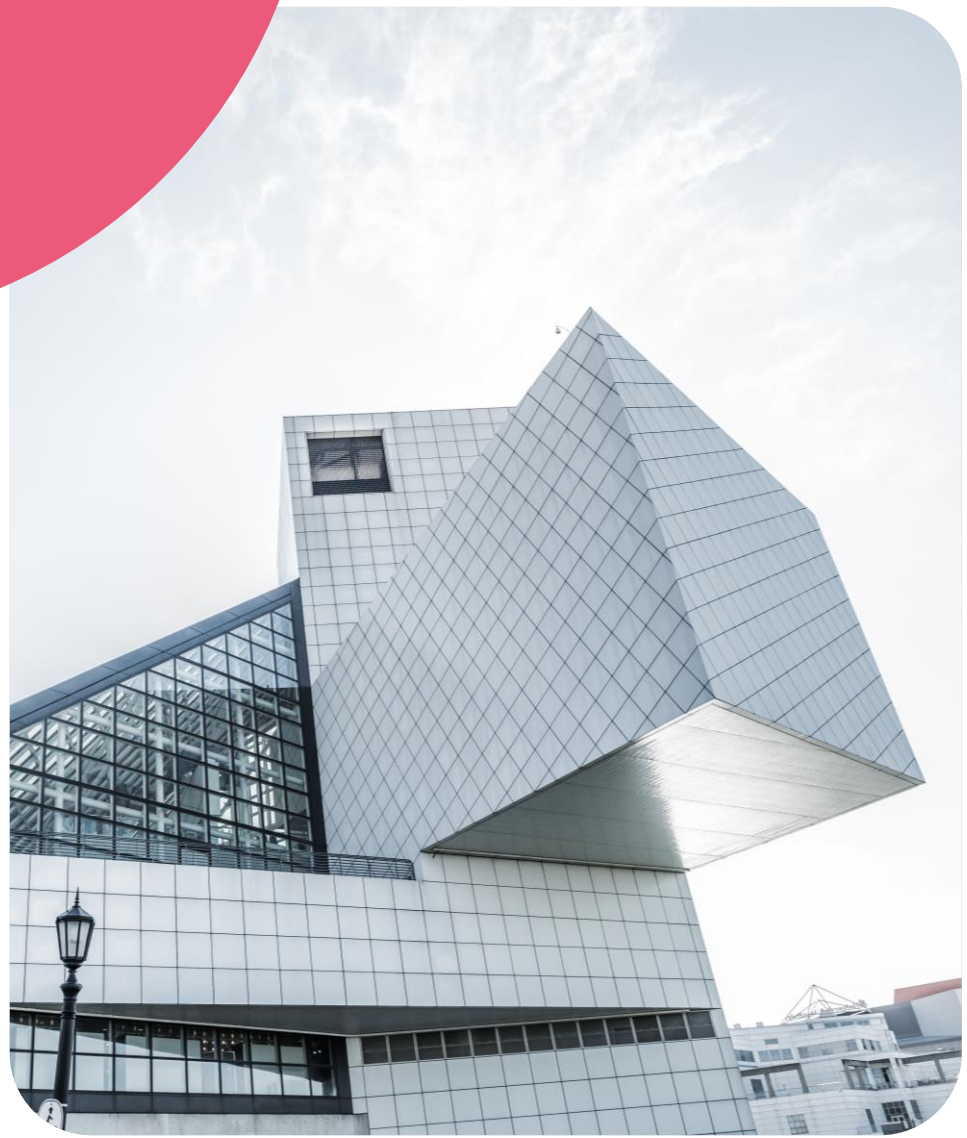
The word “bit” is short for **bi**nary digi**t.**

Parity

- Computers were originally not very reliable, and computer scientists needed to check data integrity.
- We need parity because we constantly transmit data over mediums that can distort it.

Parity

- Parity is used to check data for integrity.
- It can be implemented as odd parity or even parity.
- In odd parity, if the number of 1s in the 7 bits of information is odd, then the parity bit is set as 0.
- In even parity, if the number of 1s is even, then the parity bit is set to 0.



- Computers use machine language for all operations.
- All programming is converted to machine language.
- The instruction set is the set of basic instructions that a processor uses.
- The instruction set architecture is the set of basic operations the computer supports. It offers a level of abstraction from the combinational and sequence circuits.

Computer instructions

Instruction set

- CPU instruction set typically refers to the word size of the processor.
- Computer scientists agreed to standardise this unit to be 32 bits or 64 bits.
- This loosely defines the amount of data the CPU can manipulate in one go.

Computer interpretation

- A language statement generally corresponds to a single processor instruction.
- Each instruction is a designated unique sequence of 1s and 0s.
- These 1s and 0s tell the CPU which transistors to switch on or off.

Computer instructions

A computer instruction is made up of three parts:

- The operation (op) code field specifies the operation to be performed.
- The address field designates the memory address or register.
- The mode field specifies the way the operand of effective address is determined.

Transducers

- Data is not always available in an ideal form for the computer to use.
- Transducers have a sensing component that converts environmental phenomena to electrical signals.
- Your smartphone is loaded with electrical transducers!

Programming language theory

Programming language theory

- Programming languages are a way of describing tasks to a computer.
- Programming languages were invented to make machines easier to use.

Abstract machines

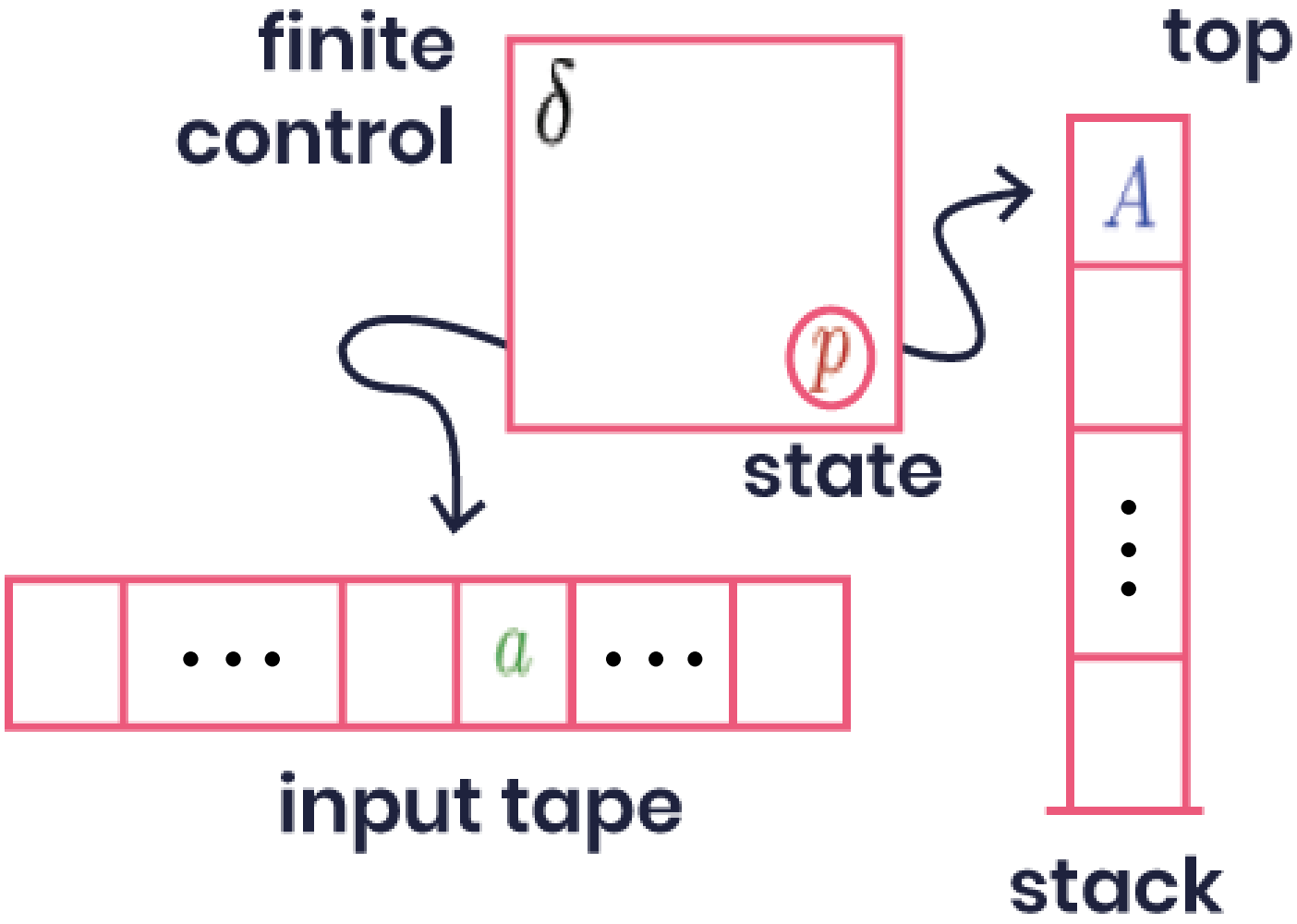
- The computer’s language describes in the finest detail what the computer will be doing.
- Before anything is built, an abstract machine is drawn up.
- An abstract machine presents a detailed and precise analysis of how the computer system will work.

Finite state machine

- It produces regular language to simulate sequential logic and some computer logic.
- Regular languages are the subset of all strings languages. They are used in computing models to describe and group problems.

Pushdown automata

- Pushdown automata are non-deterministic finite state machines that are hooked up with additional memory to form a stack.
- In a stack, the last item placed is the first item to be removed.

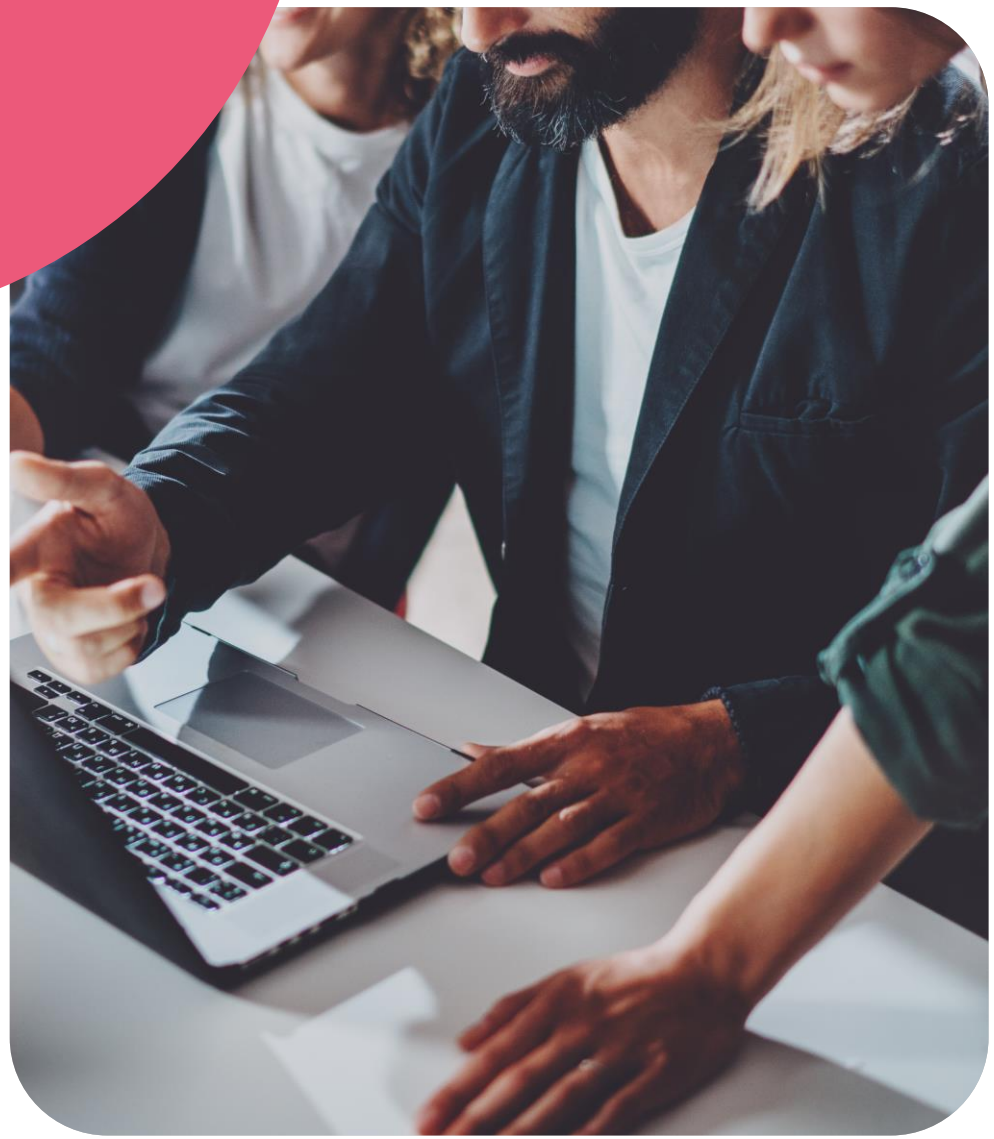

finite
control
δ
top
A
p
state
a
input tape
stack

Linear bound automata

- This model is a slight improvement on the pushdown automata.
- It includes a tape of finite length which holds the symbols.
- It recognises context-sensitive grammar.

Turing machine

An abstract machine that is the most powerful model used to solve problems and test the limits of computation.



- Alan Turing invented the Turing machine in 1936.
- A Turing machine can solve a common problem that a computer can solve, assuming it has enough memory, and vice versa.
- It uses infinite tape as the memory, a tape head that reads the memory and a transition table.
- Each cell of the tape can have one of a predetermined finite set of symbols.

Turing machine

Turing machine

- At the start, the head points to the first cell of the string.
- A Turing machine can either halt or run forever.
- A Turing machine generates unrestricted languages.

Try this!

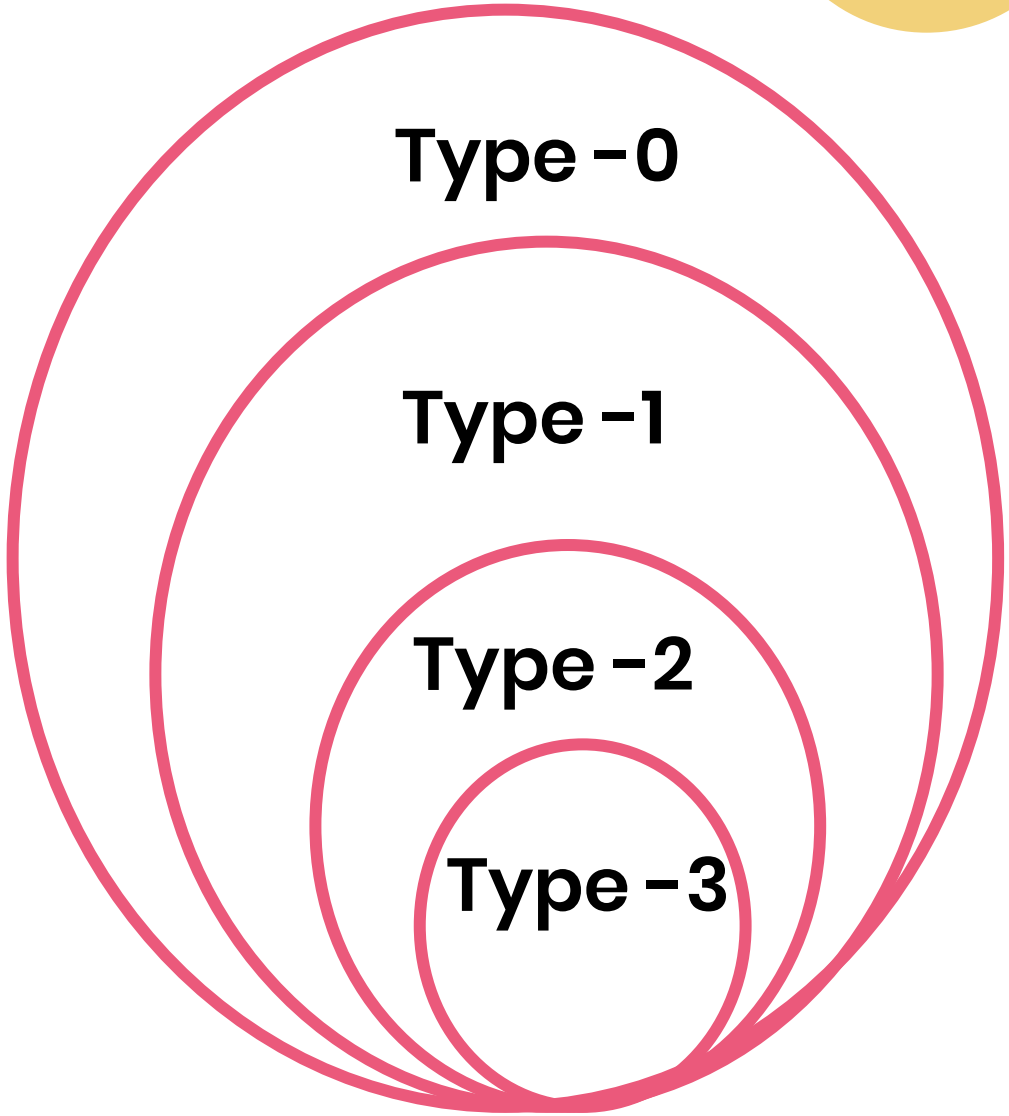
turingmachine.io/

Turing machine

- It can simulate machines acting on different data in memory.
- It can recognise many different kinds of languages and problems.
- Decidable languages are recognisable but recognisable languages are not decidable.

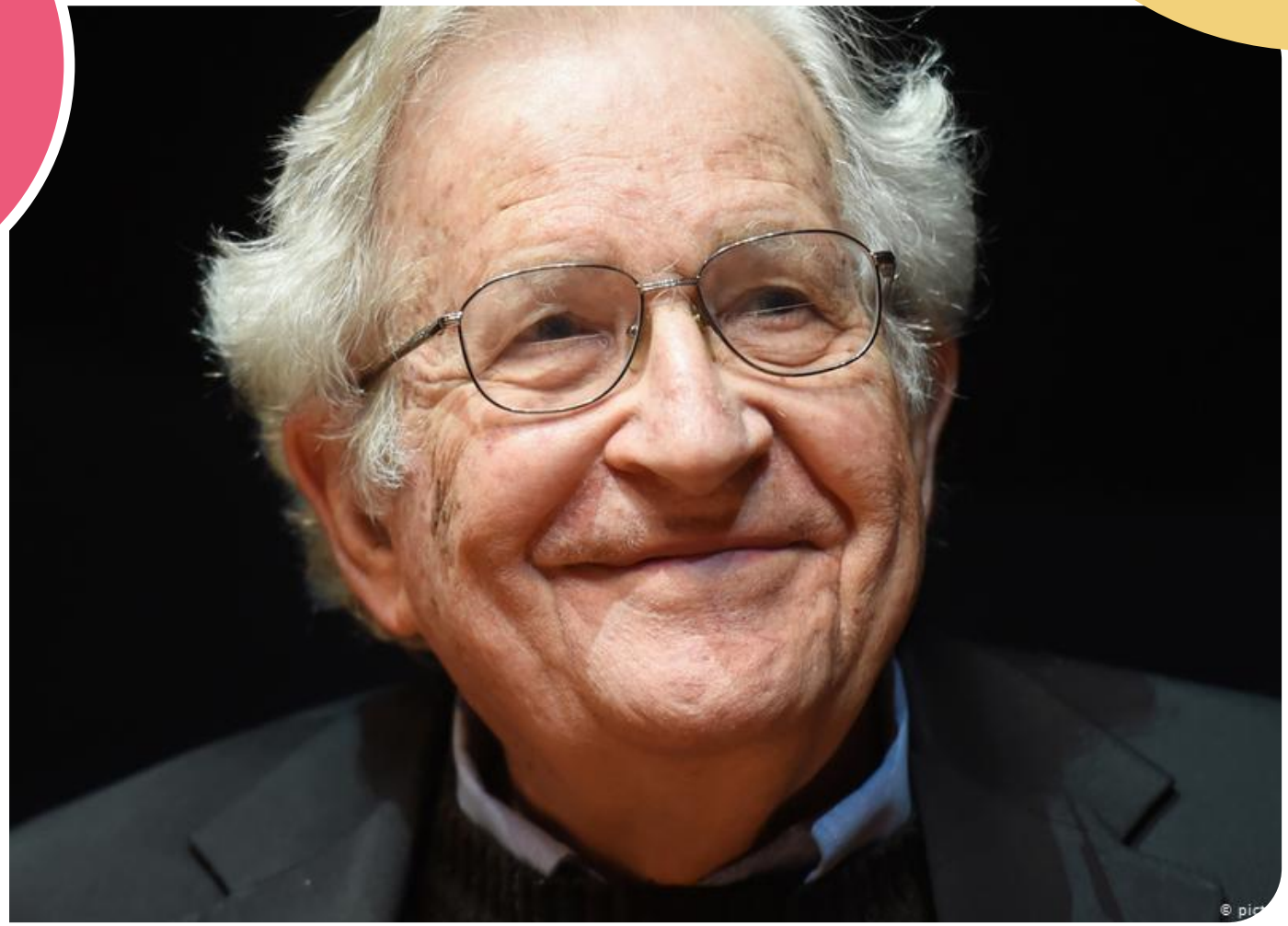
The Chomsky hierarchy

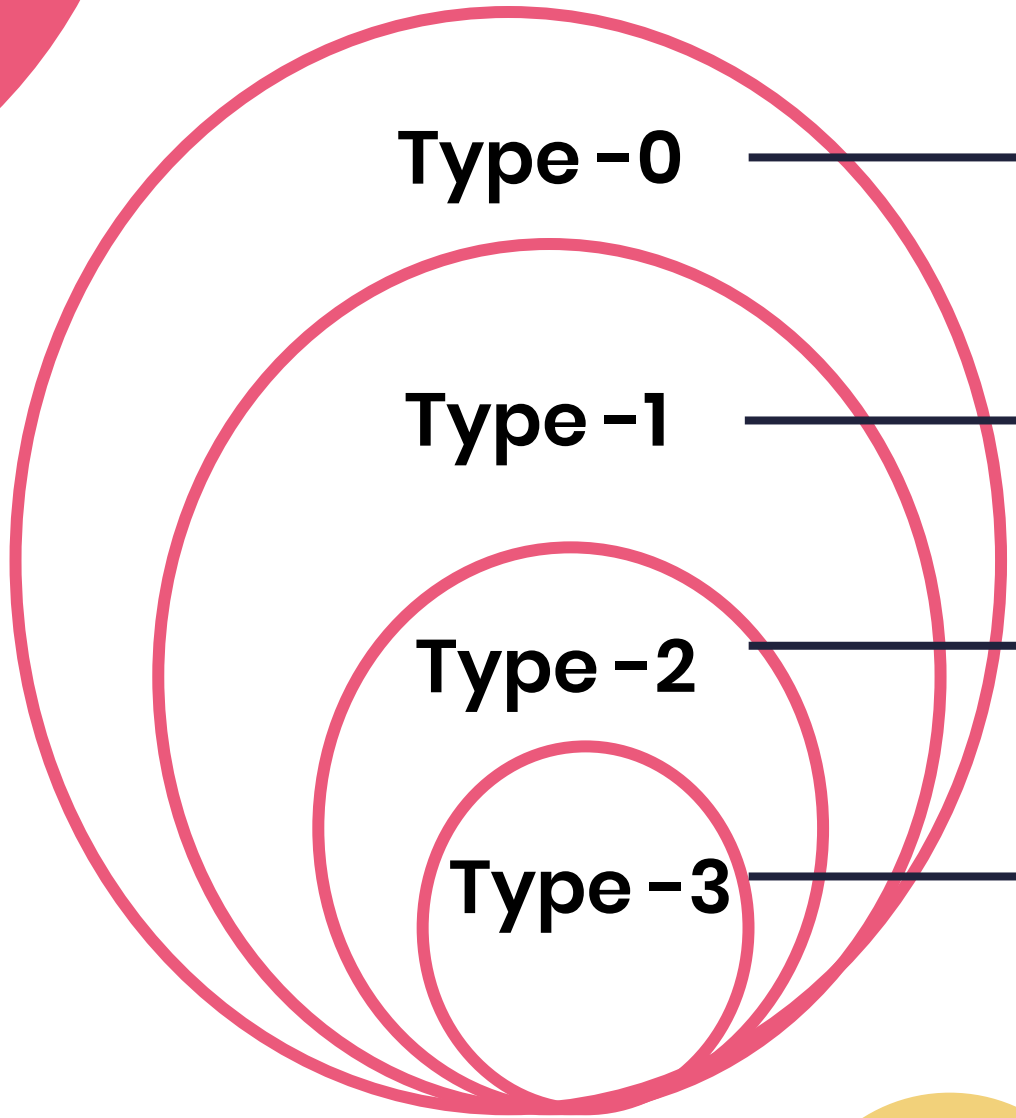
- A series of increasingly complex classes of formal languages and grammars.
- There are four basic levels.



DID YOU KNOW?

Noam Chomsky is an American linguist, philosopher, cognitive scientist, historian, social critic and political activist.





Unrestricted Grammar
(Accepted by Turning machine)

Context Sensitive Grammar
(Accepted by Linear bound automata)

Context Free Grammar
(Accepted by Pushdown automata)

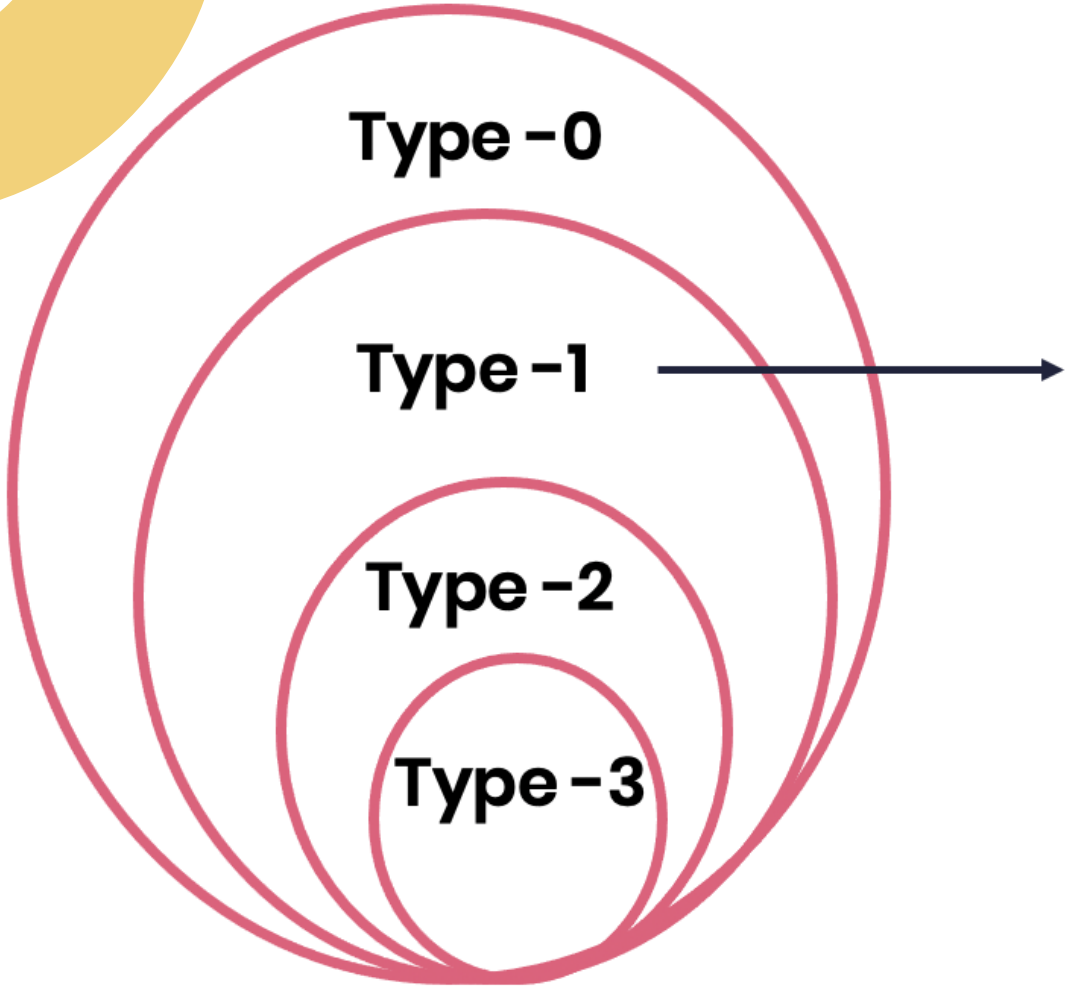
Regular Grammar
(Accepted by Finite automata)

The Chomsky hierarchy

Type 0

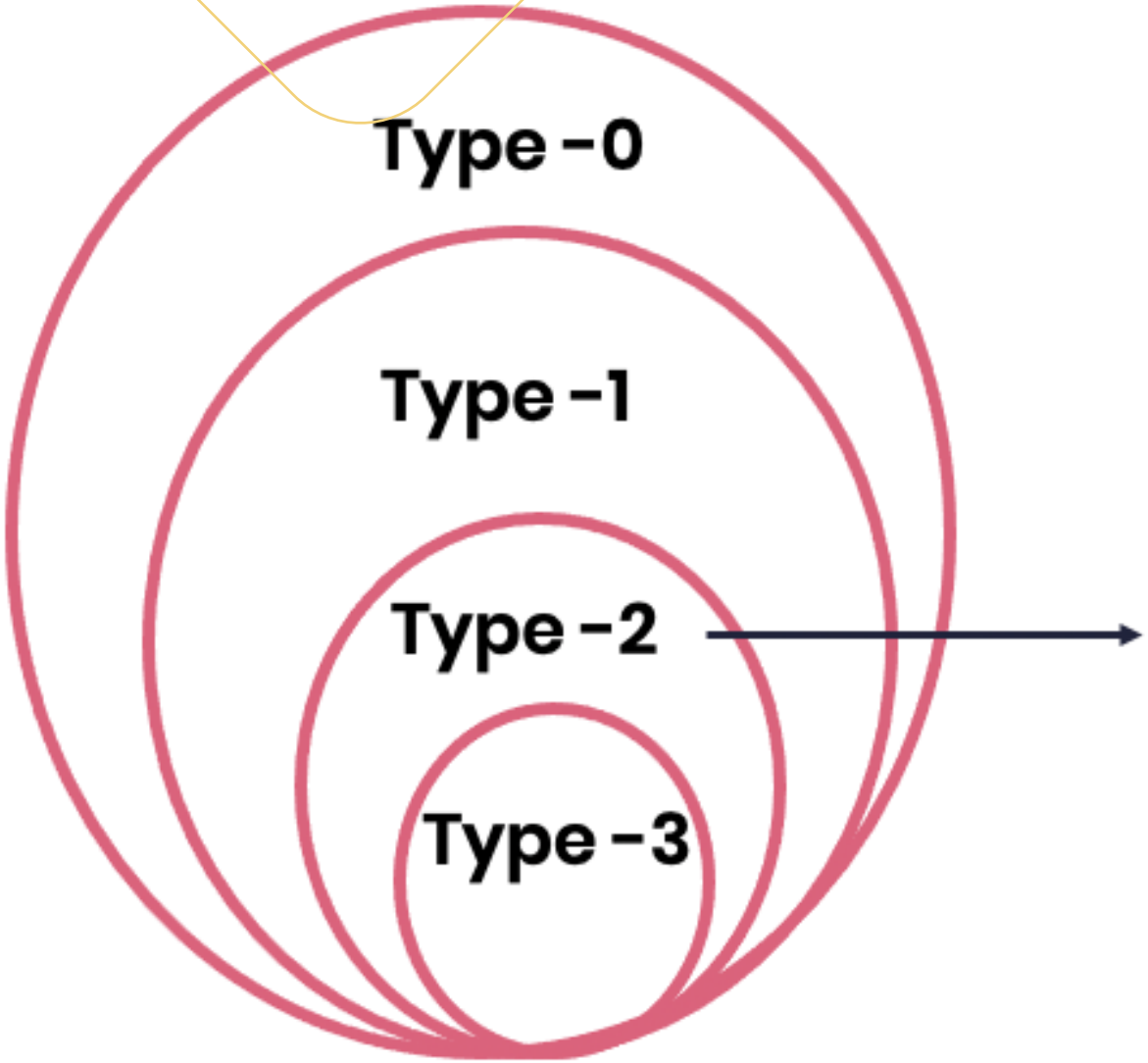
- Unrestricted languages generated by Turing machines.
- Turing machines are able to calculate everything you can imagine (everything is decidable).

The Chomsky hierarchy



Type 1

- Context-sensitive languages generated by linear-bound non-deterministic Turing machines.
- They can deal with different contexts because they are non-deterministic and have complete reference to the past.



The Chomsky hierarchy

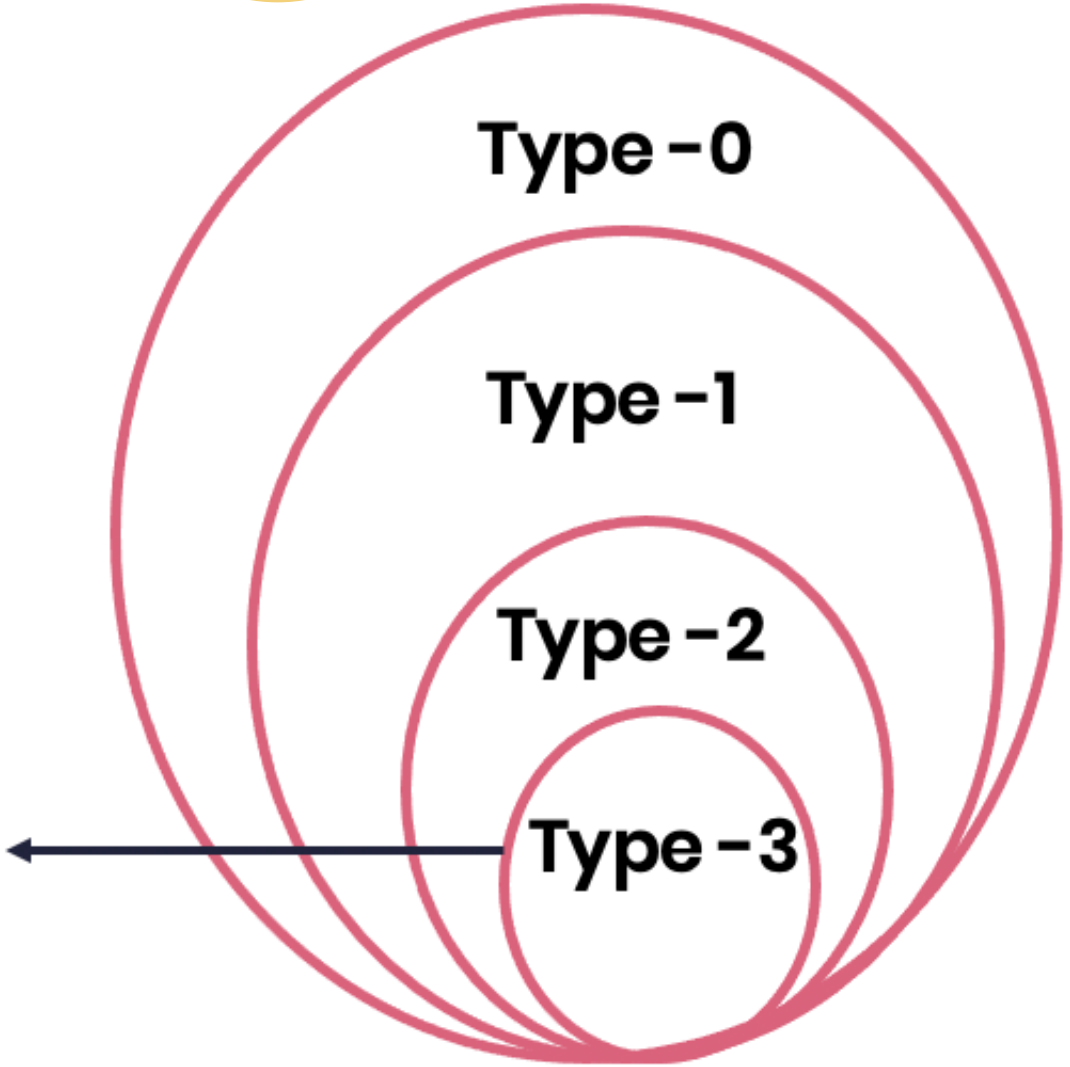
Type 2

- Context-free languages generated by non-deterministic pushdown automata.
- They have memory.

The Chomsky hierarchy

Type 3

- Regular languages generated by regular automata.
- They have limited compute memory.



Recap

- Human language
- Computer language
- Programming language theory