



# Turing Machine

*1 laboratory work*

Lecturer: doc. dr. Pavel Stefanovič

# Alan Mathison Turing



1912.06.23 – 1954.06.07

- Alan Mathison Turing was an English mathematician, logician, cryptanalyst, and computer scientist. Besides his works in crypto analysis (1939-1945 he helped to crack down German crypto-machine called “Enigma”, he also proposed an idea to perform crypting using prime numbers) he was worked on formalization of algorithms and especially, computation itself. What we call now “Computer science”, was in big part started by AMT.

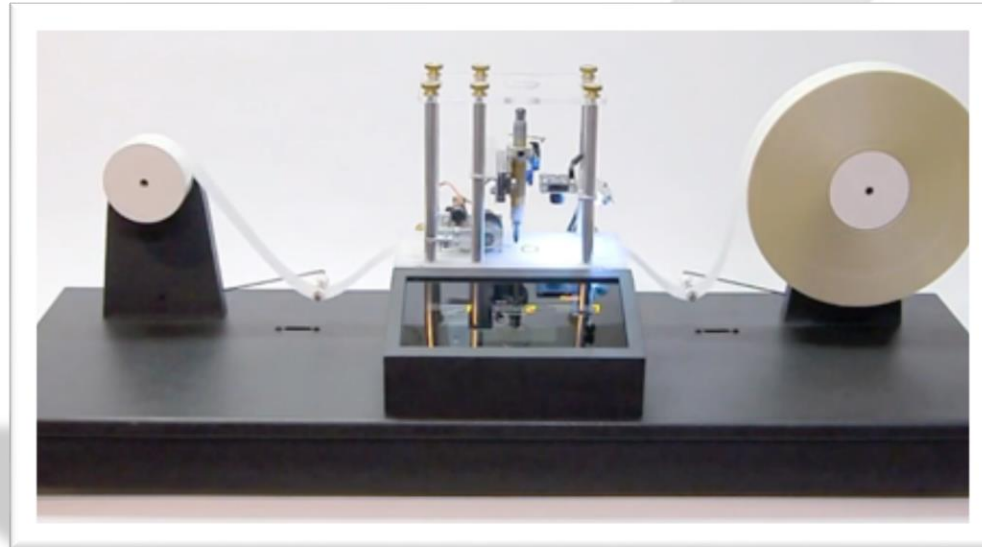
<https://www.youtube.com/watch?v=gtRLmL70TH0&t=371s>

# Turing machine?

- This is simplest virtual machine. Not only computers, but also VMWare, VirtualBox and friends are working using same principles.
- This should give you understanding that Turing based computation is a state + rules to change the state.
- This should broaden your mind.
- Everybody who study IT **must** to know about Turing machine.
- It have to be fun (**for some of you**).

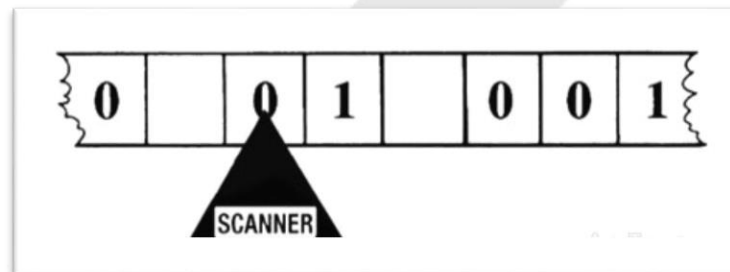
# Components of Turing machine

- **Tape** – collection of symbols.
- **Head** – read/write/move – capable device.
- **State** – value contained in some store.
- **Program** – 5 column table that denotes what machine has to do.



# Tape

- A tape is collection of cells, which is infinite, it means that we do not know how many cells there are.
- What is contained in the cell is a symbol. All distinct symbols from the tape is the grammar of partical machine. All symbols, characters can be used.
- Grammar is finite, means we know exactly what symbols are in the tape, but do not know how many.
- Generally, a grammar is defined by the Turing machine programmer, very like when one is naming variables in various programming language.

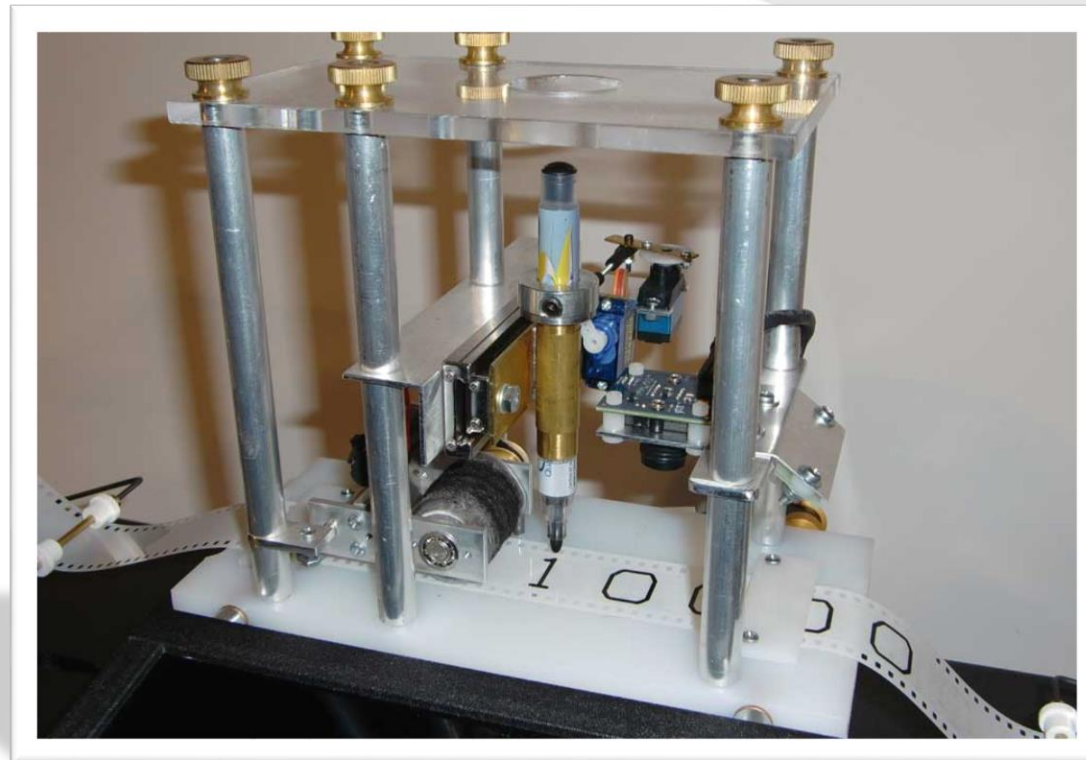


# States

- A state is some value which is stored somewhere (a variable, for example). Turing machine is abstract model, it is possible to construct mechanical prototype of TM, in this case a state would probably be encoded in an angle of some gear. A state can only take values from it's own set of symbols (also a grammar, but distinct from tape grammar).
- We need to know what is a starting state of machine.
- The state will change while machine will work (execute a program).

# Head

- Head (a scanner in Turing's original) is a device (even mechanical one) that is moving along the tape, left or right, and does reading/writing.





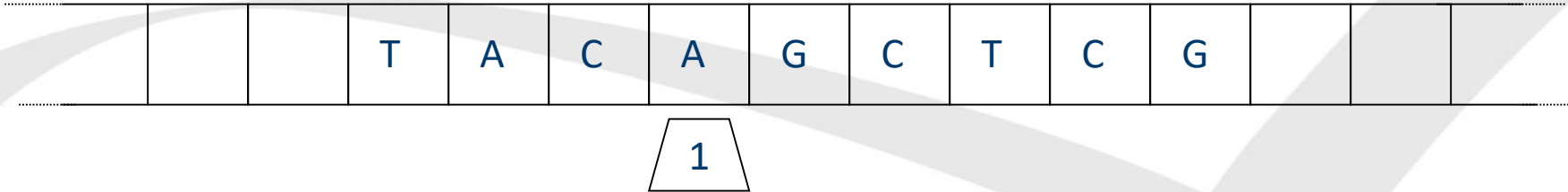
# Program notation

| Current state | Current symbol | New symbol | Direction | New state |
|---------------|----------------|------------|-----------|-----------|
|---------------|----------------|------------|-----------|-----------|

- Current state – the name of state (example: 0, 1, 2).
- Current symbol – the symbol from the tape.
- New symbol – the new symbol which will be written to the tape.
- Direction – left (L), right (R).
- New state – the name of new state.



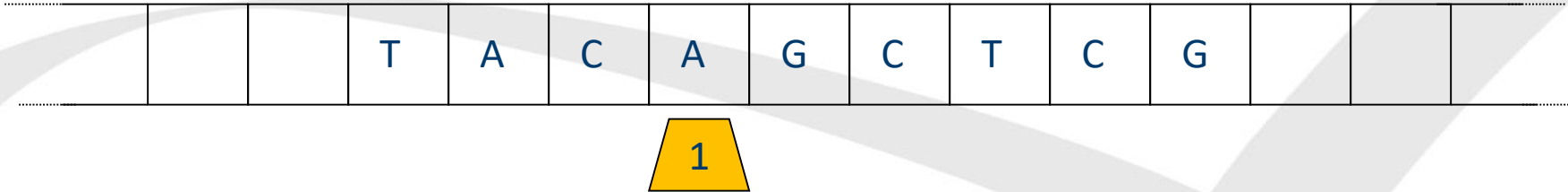
# The example of Turing machine (1)



| Current state | Current symbol | New symbol | Direction | New state   |
|---------------|----------------|------------|-----------|-------------|
| 1             | A              | A          | R         | 2           |
| 2             | G              | T          | R         | 3           |
| 3             | C              | A          | R         | <b>Halt</b> |

**Task:** change letters GC to TA

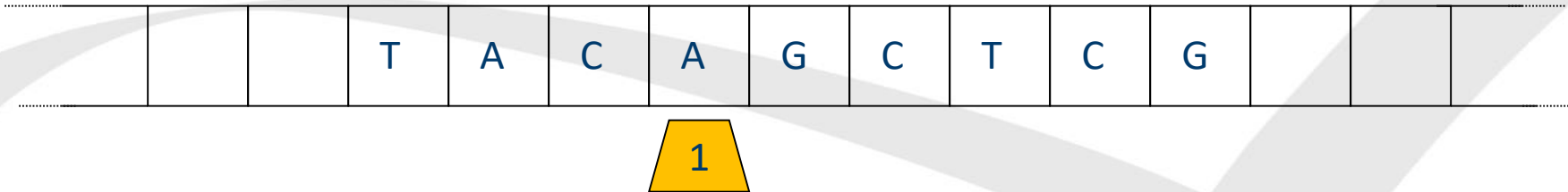
## The example of Turing machine (2)



| Current state | Current symbol | New symbol | Direction | New state   |
|---------------|----------------|------------|-----------|-------------|
| 1             | A              | A          | R         | 2           |
| 2             | G              | T          | R         | 3           |
| 3             | C              | A          | R         | <b>Halt</b> |

**Task:** change letters GC to TA

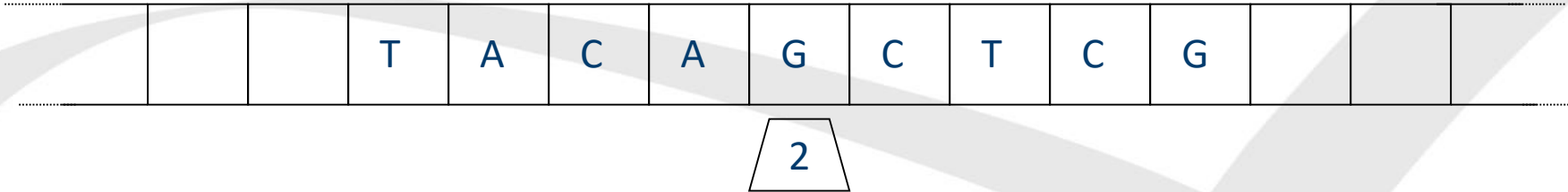
## The example of Turing machine (3)



| Current state | Current symbol | New symbol | Direction | New state |
|---------------|----------------|------------|-----------|-----------|
| 1             | A              | A          | R         | 2         |
| 2             | G              | T          | R         | 3         |
| 3             | C              | A          | R         | Halt      |

**Task:** change letters GC to TA

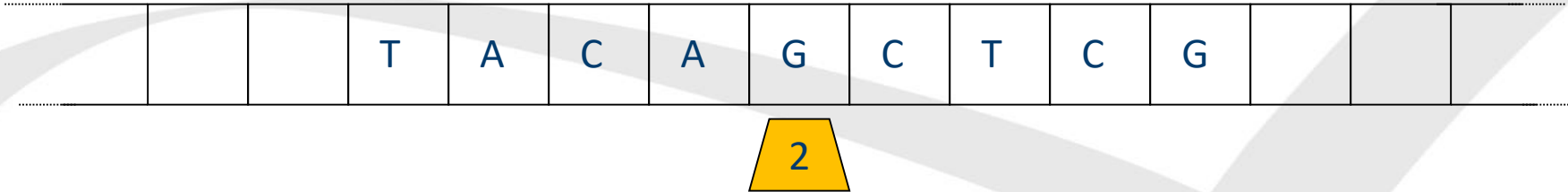
## The example of Turing machine (4)



| Current state | Current symbol | New symbol | Direction | New state   |
|---------------|----------------|------------|-----------|-------------|
| 1             | A              | A          | R         | 2           |
| 2             | G              | T          | R         | 3           |
| 3             | C              | A          | R         | <b>Halt</b> |

**Task:** change letters GC to TA

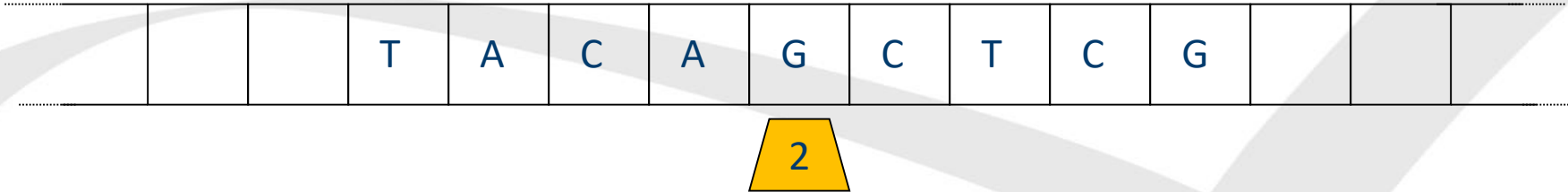
# The example of Turing machine (5)



| Current state | Current symbol | New symbol | Direction | New state   |
|---------------|----------------|------------|-----------|-------------|
| 1             | A              | A          | R         | 2           |
| 2             | G              | T          | R         | 3           |
| 3             | C              | A          | R         | <b>Halt</b> |

**Task:** change letters GC to TA

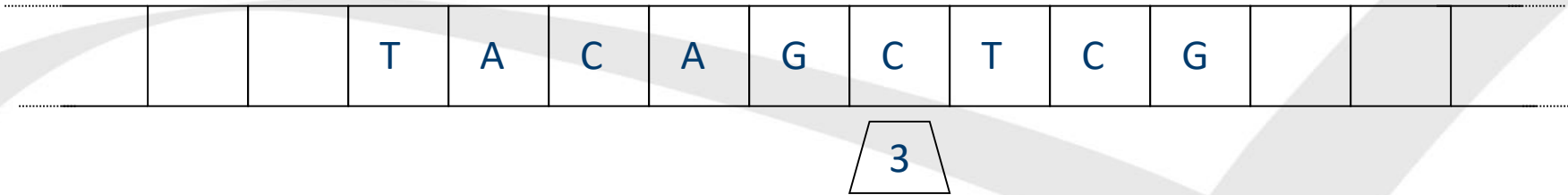
## The example of Turing machine (6)



| Current state | Current symbol | New symbol | Direction | New state |
|---------------|----------------|------------|-----------|-----------|
| 1             | A              | A          | R         | 2         |
| 2             | G              | T          | R         | 3         |
| 3             | C              | A          | R         | Halt      |

**Task:** change letters GC to TA

# The example of Turing machine (7)

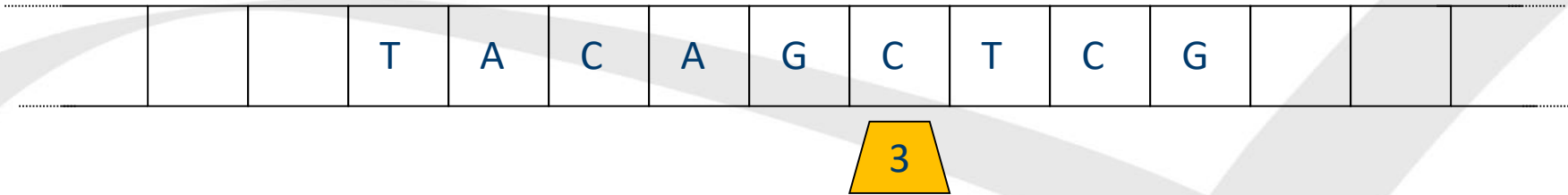


| Current state | Current symbol | New symbol | Direction | New state   |
|---------------|----------------|------------|-----------|-------------|
| 1             | A              | A          | R         | 2           |
| 2             | G              | T          | R         | 3           |
| 3             | C              | A          | R         | <b>Halt</b> |

**Task:** change letters GC to TA



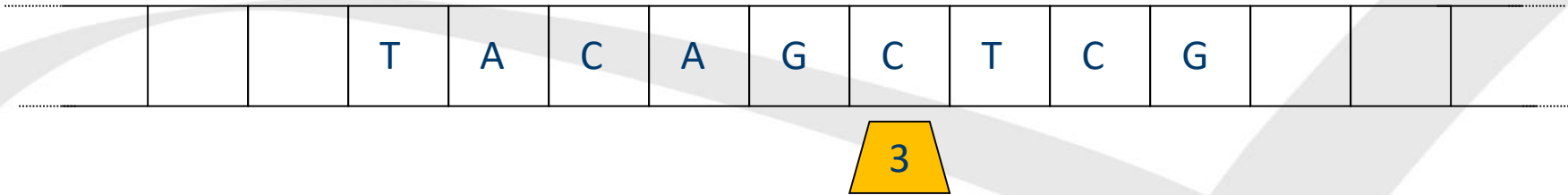
# The example of Turing machine (8)



| Current state | Current symbol | New symbol | Direction | New state   |
|---------------|----------------|------------|-----------|-------------|
| 1             | A              | A          | R         | 2           |
| 2             | G              | T          | R         | 3           |
| 3             | C              | A          | R         | <b>Halt</b> |

**Task:** change letters GC to TA

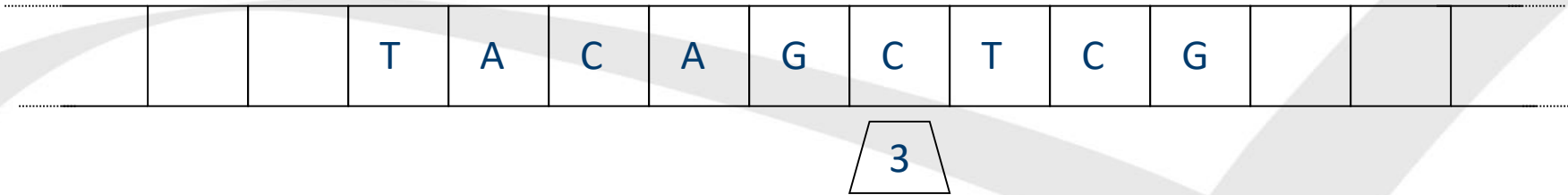
# The example of Turing machine (9)



| Current state | Current symbol | New symbol | Direction | New state |
|---------------|----------------|------------|-----------|-----------|
| 1             | A              | A          | R         | 2         |
| 2             | G              | T          | R         | 3         |
| 3             | C              | A          | R         | Halt      |

**Task:** change letters GC to TA

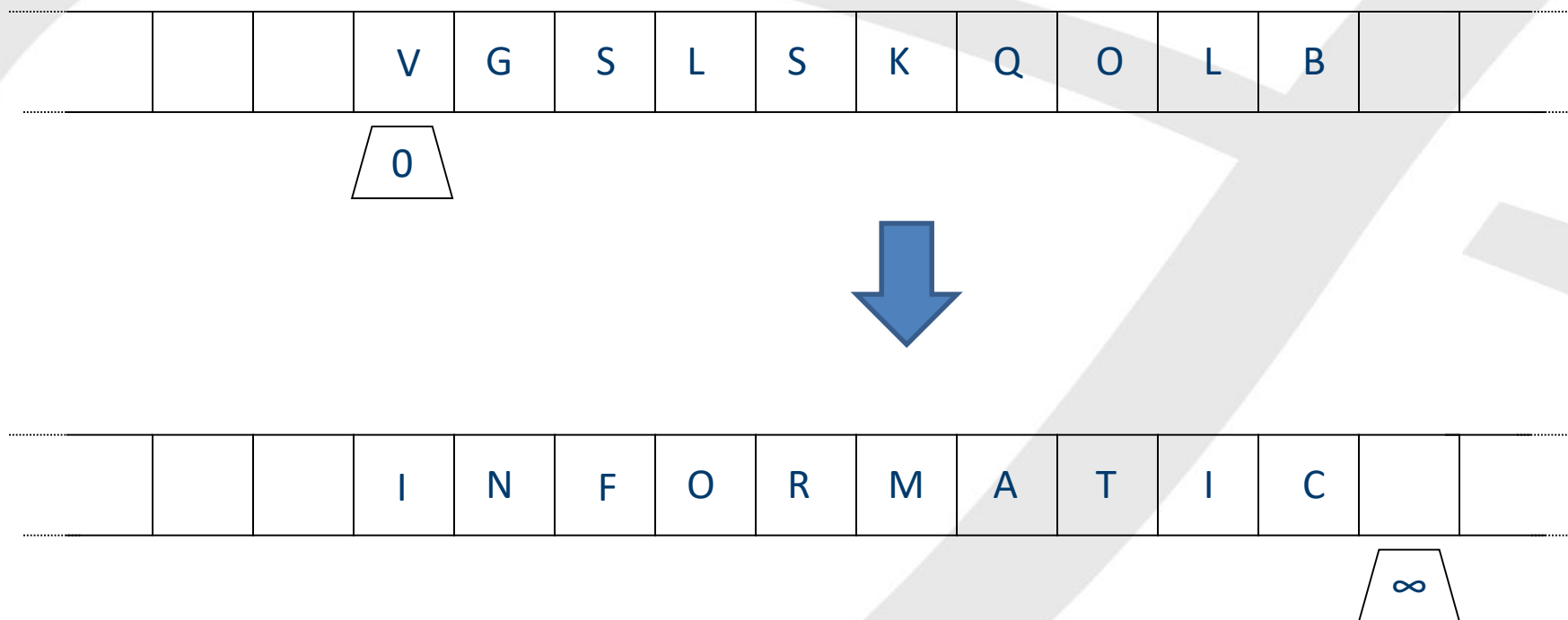
# The example of Turing machine (10)



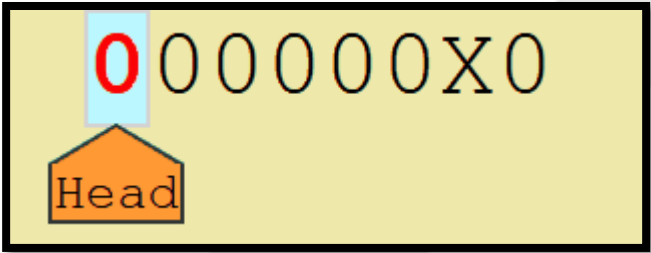
| Current state | Current symbol | New symbol | Direction | New state |
|---------------|----------------|------------|-----------|-----------|
| 1             | A              | A          | R         | 2         |
| 2             | G              | T          | R         | 3         |
| 3             | C              | A          | R         | Halt      |

**Task:** change letters GC to TA

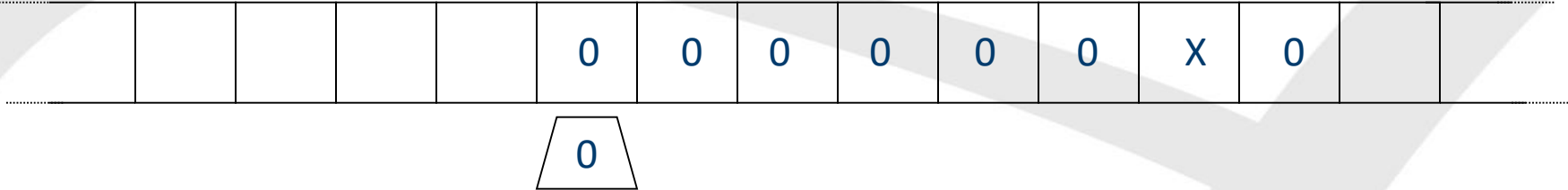
# I task



# II task



- Binary counter.



| Current state | Current symbol | New symbol | Direction | New state |
|---------------|----------------|------------|-----------|-----------|
|               |                |            |           |           |

# 1 laboratory work

- You will have to program the universal Turing machine, which can run the rules from a file, where:
  - **1 line.** Head position in the tape.
  - **2 line.** Tape.
  - **3 line.** Program.
- The program always starts from the “0” state.
- The machine is unchanged – the program have to start when one of the file name is given as a parameter. Program can’t be recompiled

