VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Turing Machine

1 laboratory work

Lecturers: dr. Pavel Stefanovič, Rokas Štrimaitis, dr. Tomas Petkus

Alan Mathison Turing

1912.06.23 – 1954.06.07

- Mathematician, logician, cryptographer, philosopher and marathon runner.
- Guided the team to decipher the Enigma code.
- The beginner of artificial intelligent (Turing test).
- Father of „Computer science“, started with the concept of algorithm; universal machine concept; software, etc.
- Films: „Codebreaker“ and „The Imitation Game“.

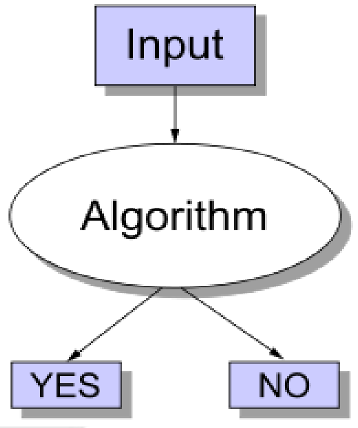
https://www.youtube.com/watch?v=gtRLmL70TH0&

Turing Machine (1)

- At that time, computing was done by human computers; machines were designed to perform a single function; huge computers.
- Universal machine concept.

Turing Machine (2)

- Is there an algorithm which would return the result in finite time „yes“ or „no“ according to the given initial axioms? (does program always stops, *halting problem*).
- Turing, by designing Turing machines, has proven that this is impossible (Church-Turing thesis).



- *Example: Does x : y ?*

$$x^n + y^n = z^n$$

Turing Machine (3)

- Computing machine mathematical model.
- Formal description:

$M = (Q, \Gamma, \Sigma, b, q_0, F, \delta)$;

Q – a finite set of states;

Γ – a finite set of alphabet tape;

$\Sigma \subseteq \Gamma$ – a finite alphabet input (all symbols which can be used);

$b \in \Gamma$ – „blank symbol“, which are used to fill blank cells on the tape;

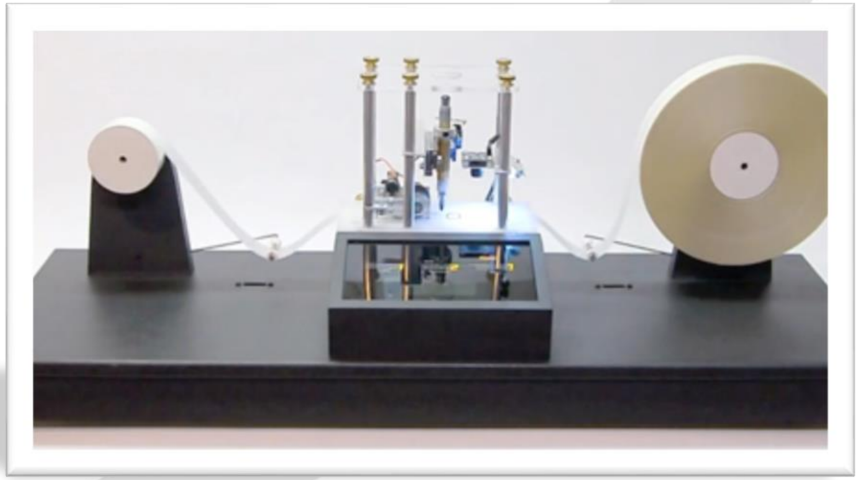
$q_0 \in Q$ – initial state;

$F \subseteq Q$ – a set of final or accepted states;

$\delta: (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ – moving function; L – left, R – right.

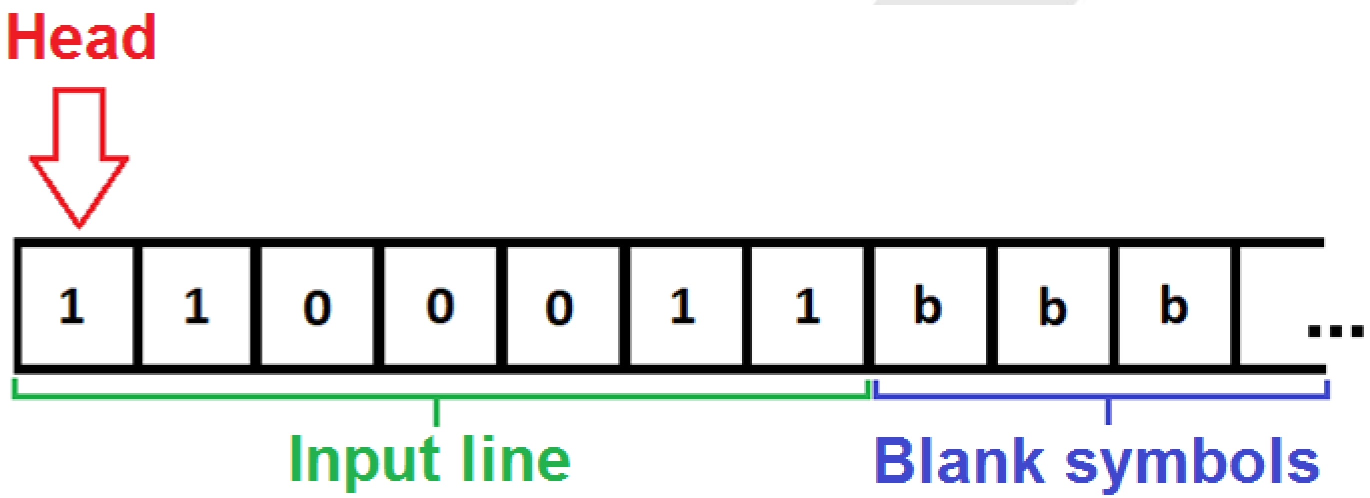
Components of Turing Machine

- Infinite **Tape,** which are devided into cells and are filled with defined alphabet 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 and Head

- Semi infinite tape which has a beginning but no end.
- It is divided into cells filled with defined possible symbols and a blank symbol.
- 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.



States

- Keep the state of the Turing machine.
- The number of states is finite.
- The initial state must always be defined.
- The states change as long as the program is running.

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 Turing machine is deterministic, so for each state, all possible variants must be described.

Conclusions

- The first machine which is capable to perform a several functions and program stored in memory – UTM (Software idea).
- Can solve all algorithms that nowadays computers solve, only at a much slower rate.
- Different from current computers, but computing itself is very similar (current computers use von Neumann architecture).
- Still used in learning/teaching and research as an easy way to show what's going on in the processor.

Example No. 1 (1)

- Lets say we have any binary number $\Gamma = \Sigma\{0,1\} \cup \{b\}$
- Turing machine will find the last digit of the binary number and 0 replace to 1, or 1 replace to 0.

1	0	1	1	b	b	b	…

q0

Current state	Current symbol	New symbol	Direction	New state
q0	0	0	R	q0
q0	1	1	R	q0
q0	b	b	L	q1
q1	0	1	R	qf = HALT
q1	1	0	R	qf = HALT
q1	b	b	R	qf = HALT

Example No. 1 (2)

- Lets say we have any binary number $\Gamma = \Sigma\{0,1\} \cup \{b\}$
- Turing machine will find the last digit of the binary number and 0 replace to 1, or 1 replace to 0.

1	0	1	1	b	b	b	…

q0

Current state	Current symbol	New symbol	Direction	New state
q0	0	0	R	q0
q0	1	1	R	q0
q0	b	b	L	q1
q1	0	1	R	qf = HALT
q1	1	0	R	qf = HALT
q1	b	b	R	qf = HALT

Example No. 1 (3)

- Lets say we have any binary number $\Gamma = \Sigma\{0,1\} \cup \{b\}$
- Turing machine will find the last digit of the binary number and 0 replace to 1, or 1 replace to 0.

1	0	1	1	b	b	b	...

q0

Current state	Current symbol	New symbol	Direction	New state
q0	0	0	R	q0
q0	1	1	R	q0
q0	b	b	L	q1
q1	0	1	R	qf = HALT
q1	1	0	R	qf = HALT
q1	b	b	R	qf = HALT

Example No. 1 (4)

- Lets say we have any binary number $\Gamma = \Sigma\{0,1\} \cup \{b\}$
- Turing machine will find the last digit of the binary number and 0 replace to 1, or 1 replace to 0.

1	0	1	1	b	b	b	…

q0

Current state	Current symbol	New symbol	Direction	New state
q0	0	0	R	q0
q0	1	1	R	q0
q0	b	b	L	q1
q1	0	1	R	qf = HALT
q1	1	0	R	qf = HALT
q1	b	b	R	qf = HALT

Example No. 1 (5)

- Lets say we have any binary number $\Gamma = \Sigma\{0,1\} \cup \{b\}$
- Turing machine will find the last digit of the binary number and 0 replace to 1, or 1 replace to 0.

1	0	1	1	b	b	b	…

q0

Current state	Current symbol	New symbol	Direction	New state
q0	0	0	R	q0
q0	1	1	R	q0
q0	b	b	L	q1
q1	0	1	R	qf = HALT
q1	1	0	R	qf = HALT
q1	b	b	R	qf = HALT

Example No. 1 (6)

- Lets say we have any binary number $\Gamma = \Sigma\{0,1\} \cup \{b\}$
- Turing machine will find the last digit of the binary number and 0 replace to 1, or 1 replace to 0.

1	0	1	1	b	b	b	…

q0

Current state	Current symbol	New symbol	Direction	New state
q0	0	0	R	q0
q0	1	1	R	q0
q0	b	b	L	q1
q1	0	1	R	qf = HALT
q1	1	0	R	qf = HALT
q1	b	b	R	qf = HALT

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Example No. 1 (7)

- Lets say we have any binary number $\Gamma = \Sigma\{0{,}1\} \cup \{b\}$
- Turing machine will find the last digit of the binary number and 0 replace to 1, or 1 replace to 0.

1	0	1	1	b	b	b	...

q1

Current state	Current symbol	New symbol	Direction	New state
q0	0	0	R	q0
q0	1	1	R	q0
q0	b	b	L	q1
q1	0	1	R	qf = HALT
q1	1	0	R	qf = HALT
q1	b	b	R	qf = HALT

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Example No. 1 (8)

- Lets say we have any binary number $\Gamma = \Sigma\{0,1\} \cup \{b\}$
- Turing machine will find the last digit of the binary number and 0 replace to 1, or 1 replace to 0.

1	0	1	0	b	b	b	…

qf

Current state	Current symbol	New symbol	Direction	New state
q0	0	0	R	q0
q0	1	1	R	q0
q0	b	b	L	q1
q1	0	1	R	qf = HALT
q1	1	0	R	qf = HALT
q1	b	b	R	qf = HALT

Example No. 2

- Lets say we have any binary number $\Gamma = \Sigma\{0,1\} \cup \{b\}$
- Turing machine will add 1 to the any digit of given binary number.

…	b	b	1	0	1	1	b	b	…

q0

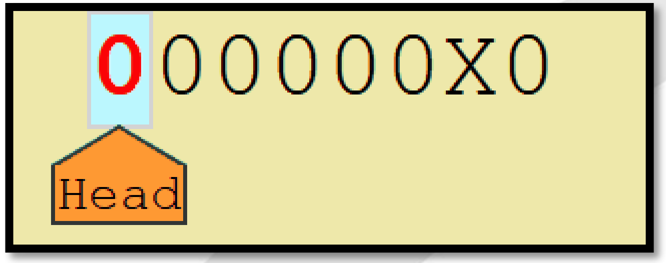
Current state	Current symbol	New symbol	Direction	New state
q0	0	0	R	q0
q0	1	1	R	q0
q0	b	b	L	q1
q1	0	1	L	qf = HALT
q1	1	0	L	q1
q1	b	1	L	qf = HALT

HOMEWORK:

What is wrong with this program?

http://morphett.info/turing/turing.html

Task



- Binary counter.

				0	0	0	0	0	0	X	0		

0

Current State	Current symbol	New Symbol	Direction	New State

The main aim of laboratory work

- Improve the skills to solve halting problem – one of the biggest programming problems. It has been proven that it is not guaranteed that the program will always stop, so it is up to the programmer to think through the possible options.
- Improving programming skills.
- Find out how the Turing machine works.

1 laboratory work (part I)

- You will have to program the universal Turing machine, which can run the rules from a file, where:
    1. Tape.
    2. Head position in the tape.
    3. 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.

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

1 laboratory work (part II)

- You need to create a simulation of parallel Turing.
- Universal Turing machine have to show the working of four input files at once.
- When one of the programs stops, the others need to keep running.
- In order to get the **MAXIMUM EVALUATION** of this part, the *thread* must be used.

The main rules of laboratory work

- The program must have all stopping conditions.
- Practical program – easily and clearly presented how to use the program; the menu; the information is clearly displayed on the screen (no unnecessary information, no ambiguity, everything can be clearly seen, etc.); clear program code.
- Universal program – program must be able to run all files with a given structure.
- Optimization.

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Evaluation

MAX: 10 points

MIN: 5 points

REALIZATION

COLLECTED SCORE

PRESENTATION

PRESENTED ON TIME

GENERAL REQUIREMENTS (6 points)

1. Works with all four files (2).
2. Program is practical (2).
3. Program code is optimized (1).
4. Program is universal (1).

HALT/LIMITATIONS (2 points)

1. Tape limitation (1).
2. Halt limitation (0.5).
3. Infinite working (0.5).

PARALLEL TURING (2 points)

1. *Threads* has to be used (1.5).
2. After one program stops, other programs are still running (0.5).