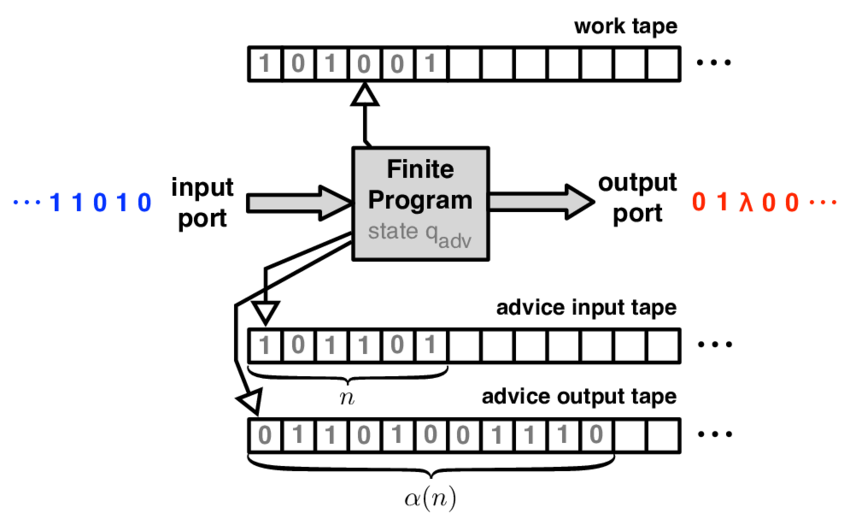
FINAL REPORT

TURING MACHINE

---o0o---

Computational Theory

2020-2021



Student Info

Name: Lê Việt Anh

ID: BI9-035

Email: anhlv.bi9035@st.usth.edu.vn

**Contents**

[HISTORY OF TURING MACHINE 3](#_Toc77878879)

[IMPACTS OF TURING MACHINE 3](#_Toc77878880)

[REFERENCES 3](#_Toc77878881)

# **HISTORY OF TURING MACHINE AND HOW IT AFFECTS COMPUTERS**

Turing machine was a computing device in theory invented by Alan Turing, an English mathematician, in 1936. This machine hypothetically reads a tape of 1, 0 and symbols in divided squares, one by one, followsing predefined instructions, i.e. states, to either change or leave the current symbol, move one step either left or right, shift to a new state or remain in the current one, and produce the output tape.

The Turing machine was rather a theoretical model for computation than an actual equipment, being one of the first (if not first) models for computers. Subsequent computers are designed based on Turing machine architecture.

Turing machine is believed to be the predecessor of modern computers. The concept for this machine is regarded as simple, yet so powerful, that it can do everything modern computers are capable of. All the essentials features such as input/output devices, memory and CPU present in computers could be seen in Turing machine as taper and reader, control mechanism’s storage and central processing unit, respectively.

Originally, Alan Turing was inspired by David Hilbert’s question: “Is there an effective procedure for deciding the truth or falsity of any math statement?”. The Turing machine was his answer to the “effective procedure” part.

Following certain procedures to solve a problem and give a result, emulating what a human would do when given the same problem and instructions, his invention thus lead to the crucial concept in computer science: “algorithm”.

# **Turing Machine Simulator – Calculate Sum Of 2 Binary Numbers**

## **Project Struture**

|---turing\_machine.py

|---set\_internal\_states\_table.py

|---add\_2\_binary\_numbers\_2.txt

## **Github Source Code**

<https://github.com/vietanh2000april/turing_machine_simulator>

## **Code**

import set\_internal\_states\_table as sist

## References

# https://sandipanweb.wordpress.com/2020/08/08/simulating-a-turing-machine-with-python-and-executing-programs/

# https://stackoverflow.com/questions/59045832/turing-machine-for-addition-and-comparison-of-binary-numbers

N = 1000 # tape length, initialize to a large value

class TuringMachine:

    def \_\_init\_\_(self, algo, input, state=0):

            # init dict

        self.trf = {}

            # state to string

        self.state = str(state)

            # init tape with underscores

        self.tape = ''.join(['\_']\*N)

            # set head position

        self.head = N // 2   # head is in the middle

            # insert input into the middle of the tape

        self.tape = self.tape[self.head:] + input + self.tape[:self.head]

        print('\nInternal states table')

        print('---------------------')

            # assign value to key

        for line in algo.splitlines(): # for each line in the internal states table

                # s, s1: current state and next state; d: shift direction;

            s, a, r, d, s1 = line.split(',') # a: value read; r: value write

            self.trf[s, a] = (r, d, s1) # current state and accept value

            print(line)

            print('---------')

    def shift\_one\_step(self, i):

            # H means halt

        if self.state != 'H':

                # assert self.head >= 0 and self.head < len(self.tape) here

            a = self.tape[self.head]

                # get current state based on state and a (value read)

            action = self.trf.get( (self.state, a) )

            if action: # if action = (r0, d0, s1\_0)

                    # assign r = r0, d = d0, s1\_0

                r, d, s1 = action

                    # insert input into tape

                self.tape = self.tape[:self.head] + r + self.tape[self.head+1:]

                    # move head and shift state

                if d != '\*':

                    self.head = self.head + (1 if d == 'r' else -1)

                    self.state = s1

                    print(str(i+1) + ' ' + self.tape.replace('\_', ''), self.state)

                    print('---------')

    def execute(self, max\_iter=10000):

        print('\ni in   state')

        print('------------')

        i = 0

            # execute through the tape

        while self.state != 'H' and i < max\_iter: # prevent infinite loop

            self.shift\_one\_step(i)

            i += 1

            # clear the tape

        res = self.tape.replace('\_', '')

        print('\nRESULT')

            # results output

        print("{} + {} = {} (binary)".format(first\_bin\_num, second\_bin\_num, res))

        print("{} + {} = {} (decimal)".format(int(first\_bin\_num, 2), int(second\_bin\_num, 2), int(res, 2)))

if \_\_name\_\_ == '\_\_main\_\_':

        # header

    print("Adding two binary numbers x + y")

        # take inputs

    first\_bin\_num = str(input("x = "))

    second\_bin\_num = str(input("y = "))

    input = first\_bin\_num + '\_' + second\_bin\_num

        # read internal states table

    algo = open(sist.sum\_of\_two\_binary\_numbers).read()

        # execute Turing machine

    turing = TuringMachine(algo, input)

    turing.execute()

# **REFERENCES**

Report

<https://plato.stanford.edu/entries/turing-machine/>

<https://en.wikipedia.org/wiki/Turing_machine>

<https://www.encyclopedia.com/science-and-technology/computers-and-electrical-engineering/computers-and-computing/turing-machine>

<https://www.youtube.com/watch?v=dNRDvLACg5Q>

<https://www.technologyreview.com/2012/02/08/187598/turings-enduring-importance/>

<https://www.youtube.com/watch?v=PLVCscCY4xI&t=823s>

Code

<https://www.python-course.eu/turing_machine.php>

<https://sandipanweb.wordpress.com/2020/08/08/simulating-a-turing-machine-with-python-and-executing-programs/>

<https://www.youtube.com/watch?v=PLVCscCY4xI>

<https://cs.stackexchange.com/questions/9341/practical-importance-of-turing-machines>

<https://stackoverflow.com/questions/59045832/turing-machine-for-addition-and-comparison-of-binary-numbers>