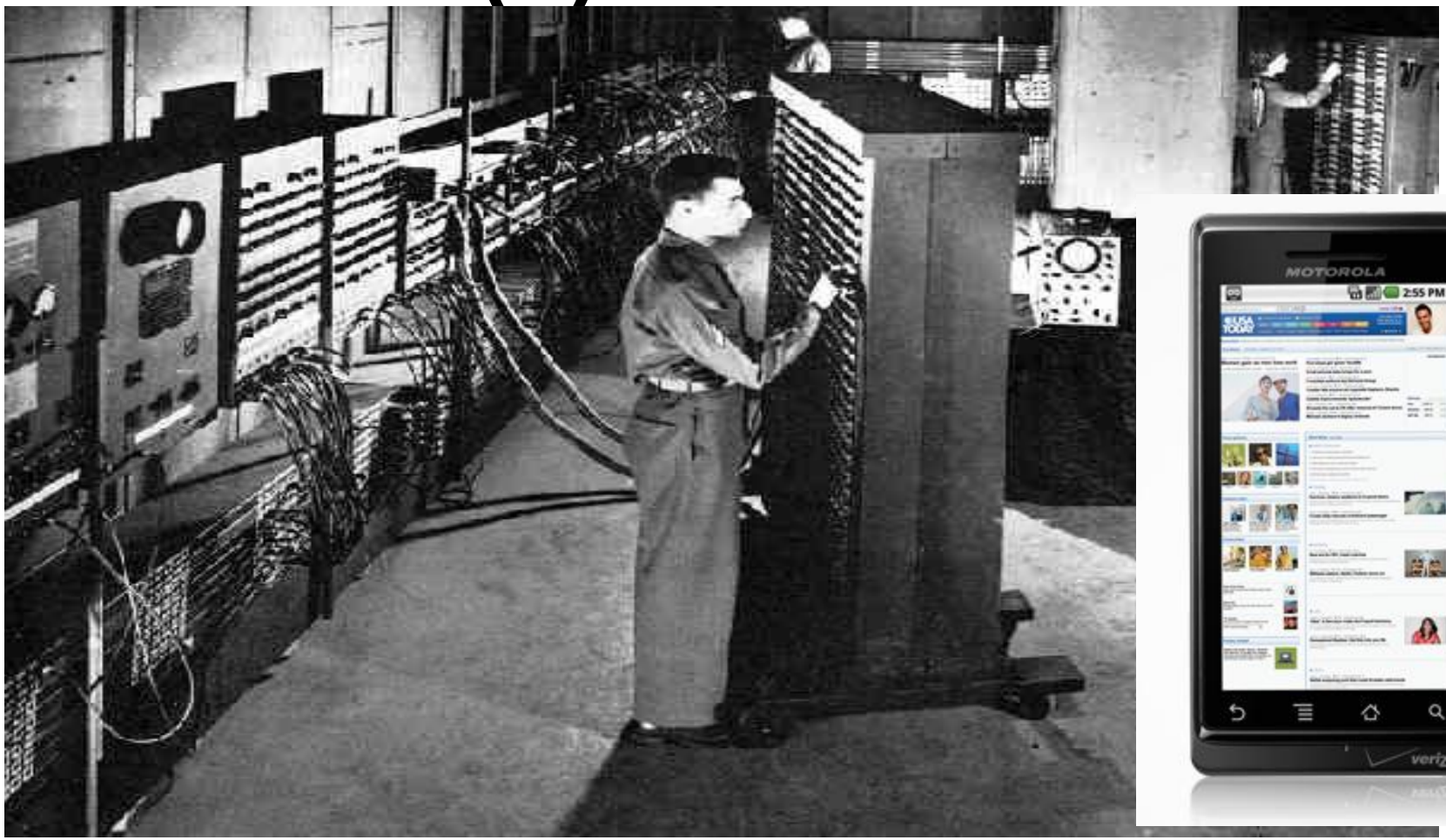
Keerpunten in de Informatica

HOVO, 12 oktober 2011

Frits Vaandrager

Institute for Computing and Information Sciences
Radboud Universiteit Nijmegen

Computing Technology (R)Evolution



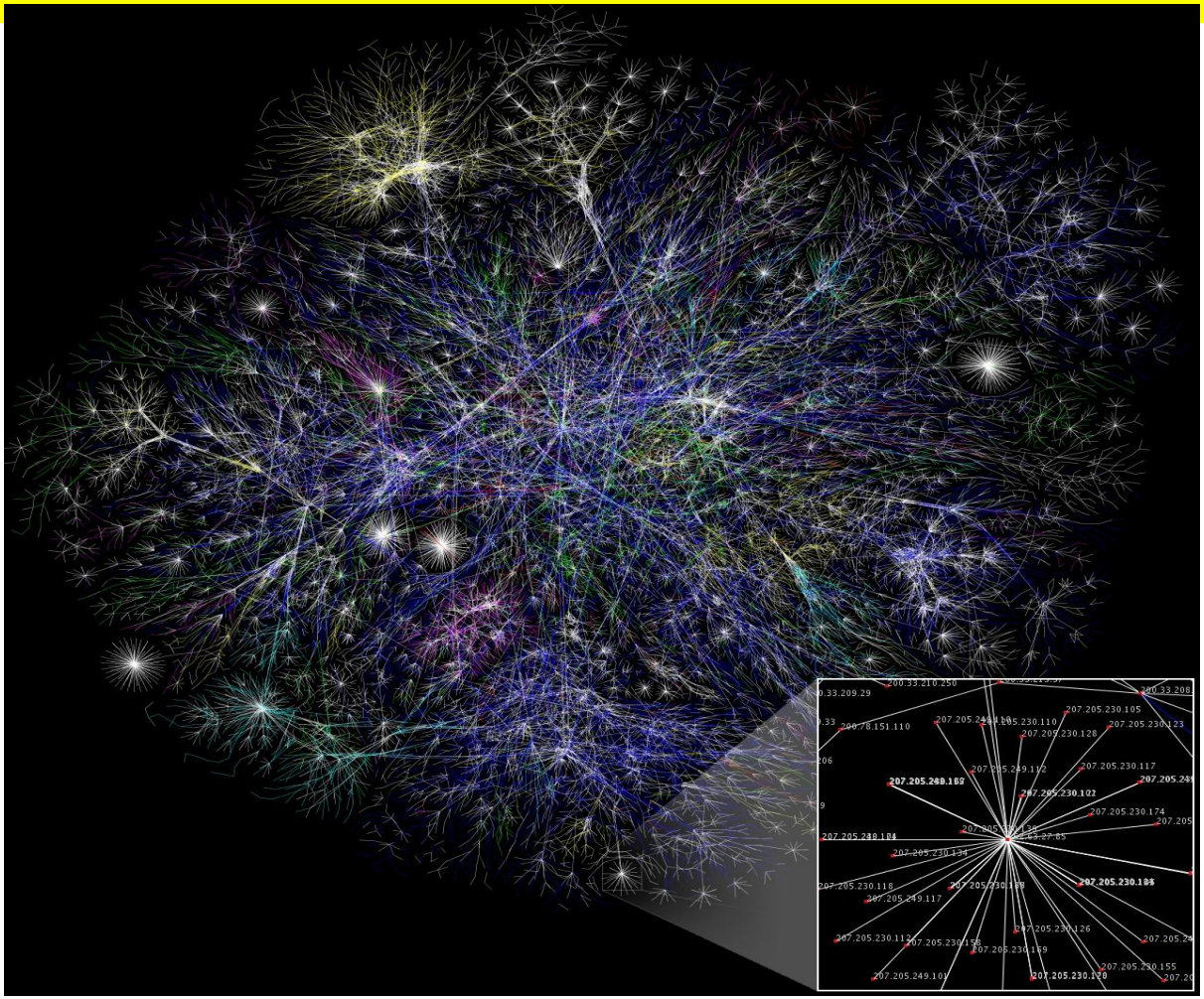
1945 1956 2010

2

Economic Impact

Social Impact

Layers of Abstraction



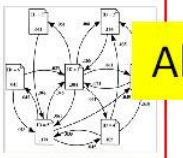
5

Search

Natural Language Processing, Text and Information Retrieval, User Interfaces

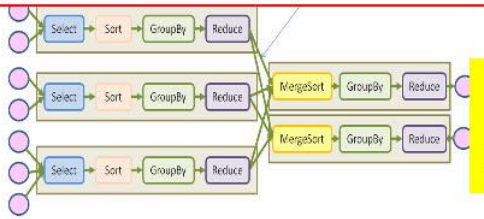
PageRank

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)}$$



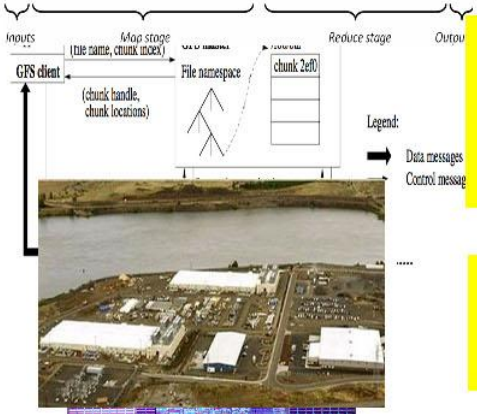
Algorithms, Data Structures

MapReduce



Programming Languages, Software Engineering

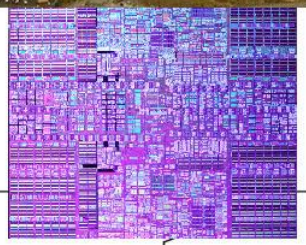
GFS, BigTable, Chubby



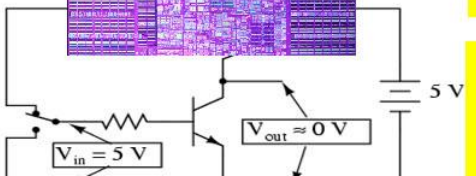
Reliability, File Systems, Operating Systems, Consensus

Server Farm

Distributed Systems, Networking, Storage Systems



Computer Architecture, Parallel Computing



Electronics, Digital Circuits, Signal Processing

Driver puting



Society

Science

- What is computable?
- P = NP?
- What is intelligence?
- What is information?
- (How) can we build complex systems simply?

Technology

7

Scientific Turning Points?

- Numerous great scientific results in Computer Science
- Computer Science has been extremely effective in combining results to create a powerful technology
- Scientific turning points???
- Computer Science good example of great science without major turning points
- OK, one maybe: Turing

ALAN MATHISON TURING

1912-1954

Father of Computer Science

- PIONEER OF COMPUTER SCIENCE
- HIS IDEAS WERE VERY ADVANCED FOR THE TECHNOLOGY OF HIS TIME
- ONE OF THE FIRST TO BELIEVE IN ARTIFICIAL INTELLIGENCE

Turing's Childhood

Born: June 23, 1912
in London England

- Second son to Julius Mathison Turing and Ethel Sara Stoney
- Spent childhood in foster households while parents lived in India
- Was not seen as a good student by teachers

SCHOOLING

- At age 13 enrolled in Sherbourne School
- After two failed attempts, earned a fellowship at King's College in Cambridge

"He refused to adapt and ignored subjects that did not interest him." BBC News, 1999.

"His genius drove him in his own direction rather than those required by his teachers." The History of Computing Projects, 2006.

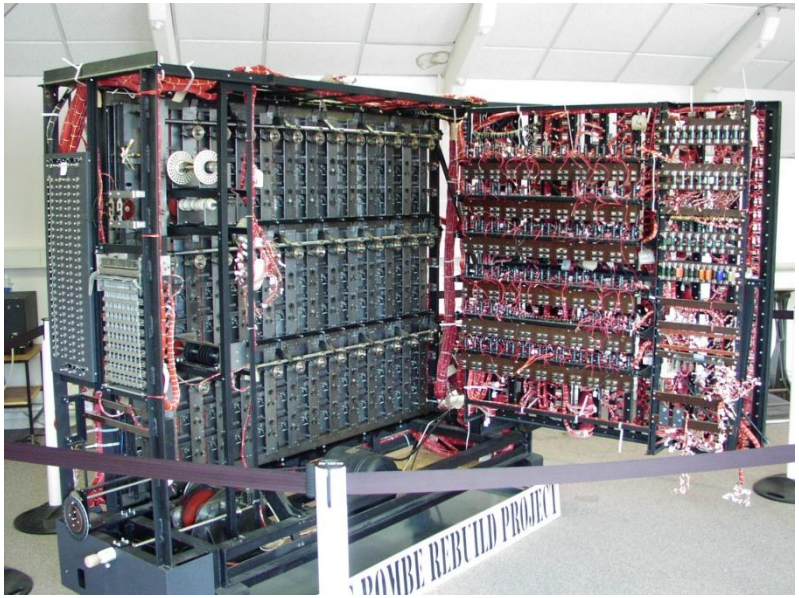
The Turing Machine

"The concept of the Turing Machine has become the foundation of the modern theory of computation and computability"

- **Turing's idea for the Turing Machine was years ahead of the technology needed to build the machine**
- **He wrote about the Turing Machine in 1936 in his paper, *On Computable Numbers***
- **This was the first digital computer programme and brought together the logical and physical world**

BREAKING THE CODE

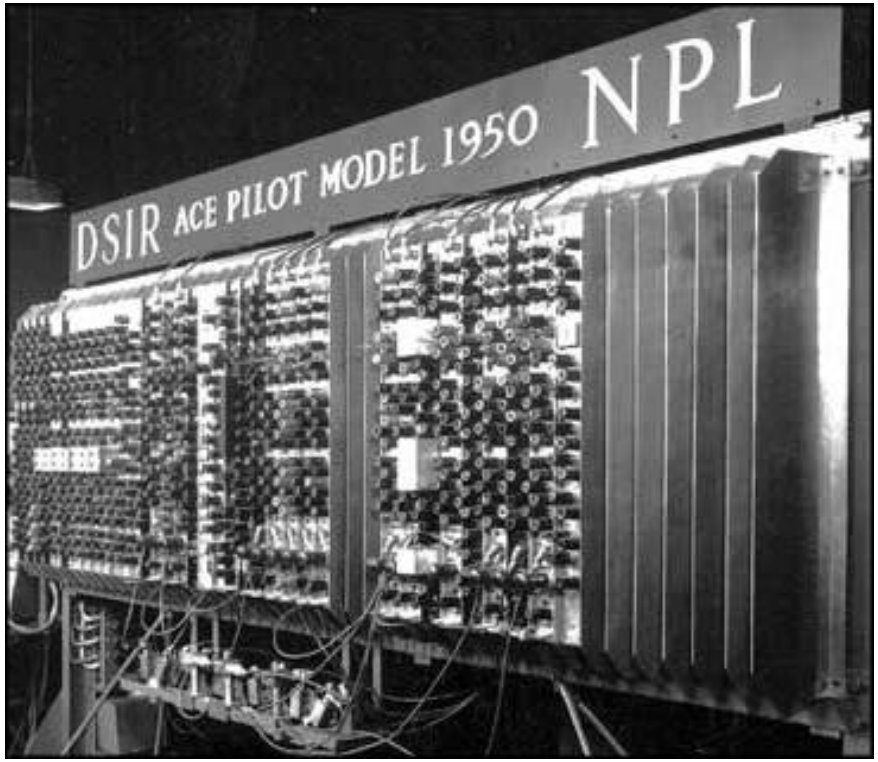
- Worked for the British Code and Cypher School during WWII
- Helped construct a machine, the Bombe, with W.G. Welchman to break the encrypted code Germans were communicating with
- Credited for saving many lives and helping Britain win the war



A replica of the Bombe

Constructing the Machine

After the war Turing was invited to the British National Physical Lab where he thought he would have the opportunity to build the Automatic Computing Engine, based on his Turing machine idea. Construction was halted though and Turing went to the University of Manchester



Pilot Model of the ACE

TURING TEST

- **ONE OF THE FIRST TO BELIEVE IN ARTIFICIAL INTELLIGENCE**
  - a machine can mimic the human brain
  - Many people did not want to believe a machine could do the same thing as a human
- **TURING TEST- A PERSON ASKED QUESTIONS ON A KEYBOARD TO A PERSON AND A MACHINE, IF THEY COULD NOT TELL THE DIFFERENCE AFTER SOME TIME THE MACHINE WAS CONSIDERED INTELLIGENT**

Last Days

- Work on mathematical biology
- Arrested in 1952 for being homosexual
- Instead of jail time, Turing had female hormones injected into him
- On June 4, 1954 Turing committed suicide by lacing an apple with potassium cyanide

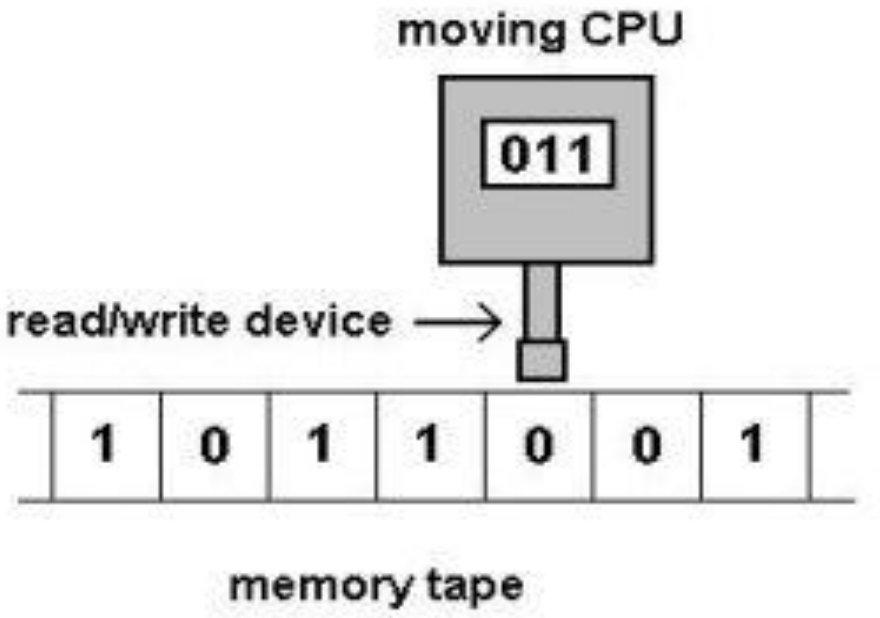
Legacy of Turing

- Today it would be hard to imagine life without computers. It is thanks to Turing's work that we have been able to develop this technology!

- "Everyone who taps at a keyboard, opens a spreadsheet or word processing program, is working on an incarnation of a Turing machine" (Gray, 1999).

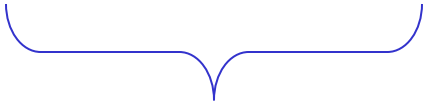
- On 10 September 2009 Prime Minister Gordon Brown apologized for the way in which Turing was treated after the war.

The Turing Machine



A limitation of Turing Machines:

Turing Machines are “hardwired”



they execute
only one program

Real Computers are re-programmable

Solution: Universal Turing Machine

Attributes:

- Reprogrammable machine
- Simulates any other Turing Machine

Universal Turing Machine

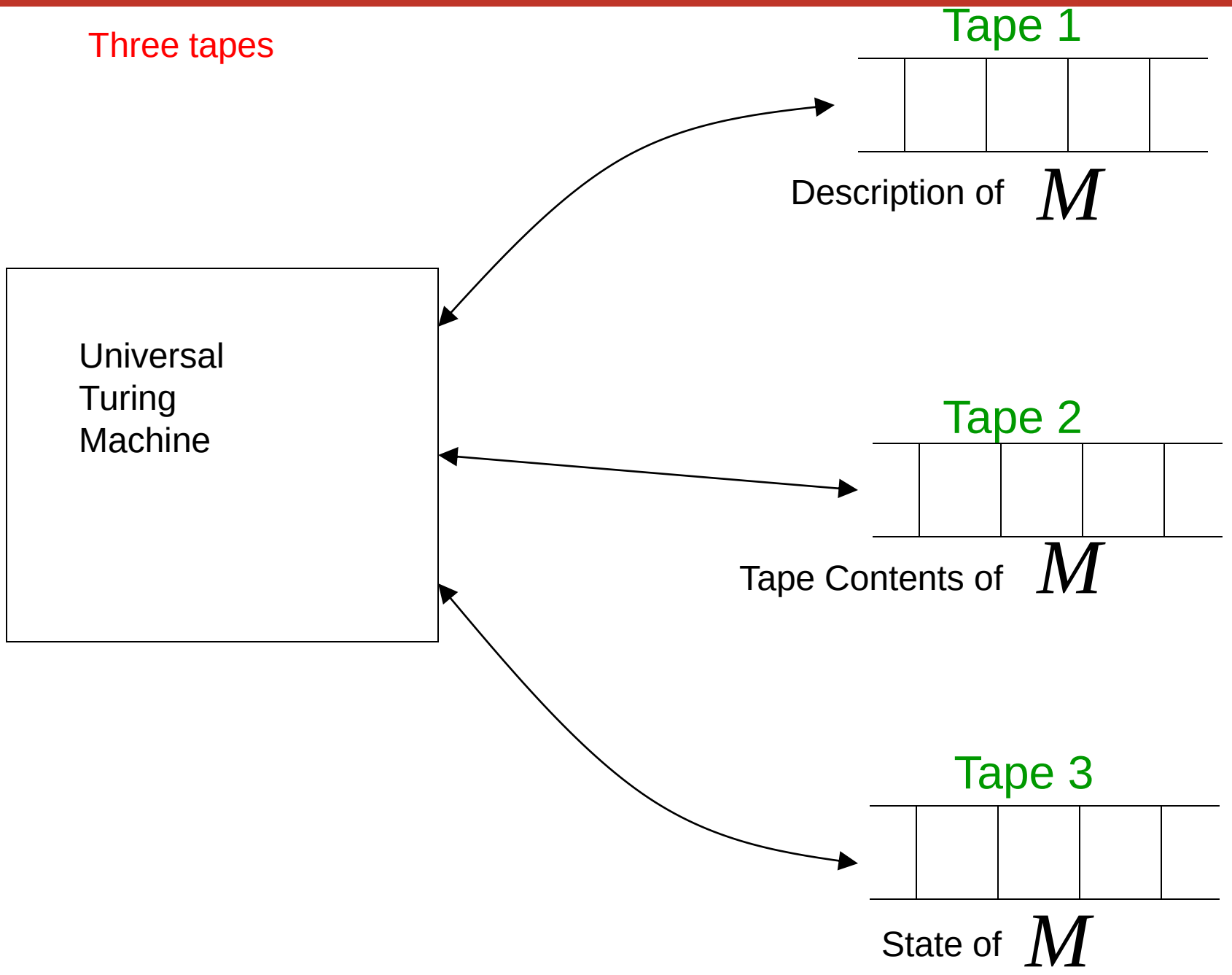
simulates any other Turing Machine M

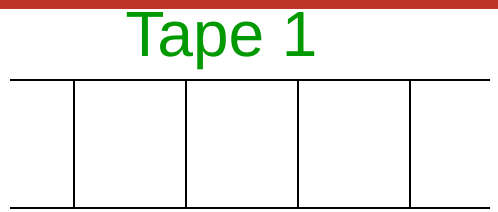
Input of Universal Turing Machine:

Description of transitions of M

Initial tape contents of M

Three tapes


Tape 1
Description of M
Universal
Turing
Machine
Tape 2
Tape Contents of M
Tape 3
State of M



Description of M

We describe Turing machine M as a string of symbols:

We encode M as a string of symbols

Alphabet Encoding

Symbols:	a	b	c	d	$\cdots$
Encoding:	1	11	111	1111	

States: q_1 q_2 q_3 q_4 $\cdots$

Encoding: 1 11 111 1111

Head Move Encoding

Move: L R

Encoding: 1 11

Transition Encoding

Transition: $\delta(q_1, a) = (q_2, b, L)$

Encoding: 1 0 1 0 1 1 0 1 1 0 1

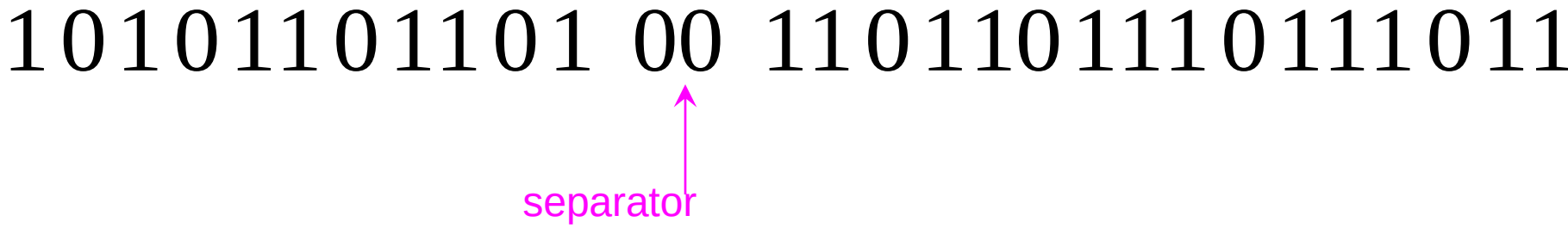
separator

Machine Encoding

Transitions:

$$\delta(q_1, a) = (q_2, b, L) \qquad \delta(q_2, b) = (q_3, c, R)$$

Encoding:



The halting problem

Does program number *x* halt on input of *x*?

$$h(x) \equiv \begin{cases} 0 \text{ if program } x \text{ halts on input } x \\ 1 \text{ otherwise} \end{cases}$$

Is there an algorithm to solve the halting problem, that is, to compute *h(x)*?

~~Suppose such an algorithm exists.~~

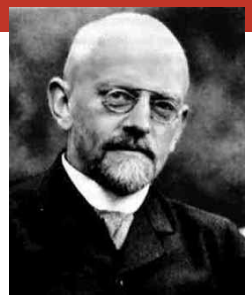
Let T be the program number for TURING.

PROGRAM: TURING(x)

```
IF h(x) = 1 THEN
    HALT
ELSE
    loop forever
```

Contradiction!

h(T) = 0 ⟺ TURING(T) halts ⟺ *h*(T) = 1



Q: Is there a general algorithm to determine whether a mathematical conjecture is true or false?

Church-Turing:
NO!

≡

Turing Machine

Computer science

***Ad hoc* empirical justification!**

Church-Turing thesis: Any algorithmic process can be simulated on a Turing machine

Literatuur

http://www.turing.org.uk/book/