# Turing's Program

Cem Bozşahin

Cognitive Science Department, The Informatics Institute, METU

bozsahin@metu.edu.tr

2016

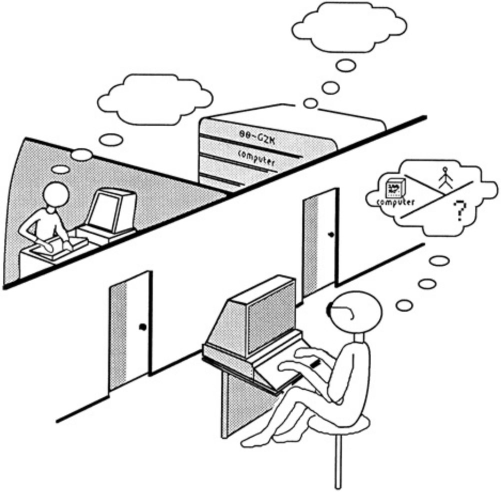
# Alan Mathison Turing: 1912–1954

| Mechanism | Behavior |
|---|---|
| simple | simple |
| complex | complex |

| Mechanism × | Behavior | |
|---|---|---|
| simple | simple | (not empirically relevant) |
| complex | simple | (not interesting) |
| complex | complex | (not scientifically interesting) |
| simple | complex | (real cogsci) (computer science) |

The last one was what he was after.

# Turing Test



(Ann Witbrock; Copeland, B.J., Artificial Intelligence Blackwell Publishers, Oxford, 1993.)

# What was Turing up to?

He tried just about everything

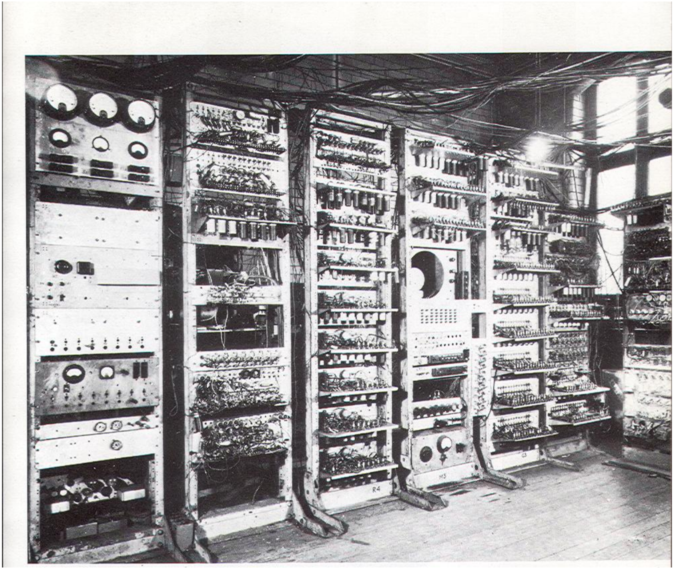
- Chemistry
- Cryptology (Enigma)
- Bio-genetics (morphogenesis)
- Abstract computing (algorithms)
- Applied computing (programming)
- Mind-body problem (round-the-house chess)
- Turing test
- Artificial life
- Foundations of mathematics
- Artificial Intelligence
- Marathon

# AMT the athlete

# Perhaps the right question is:

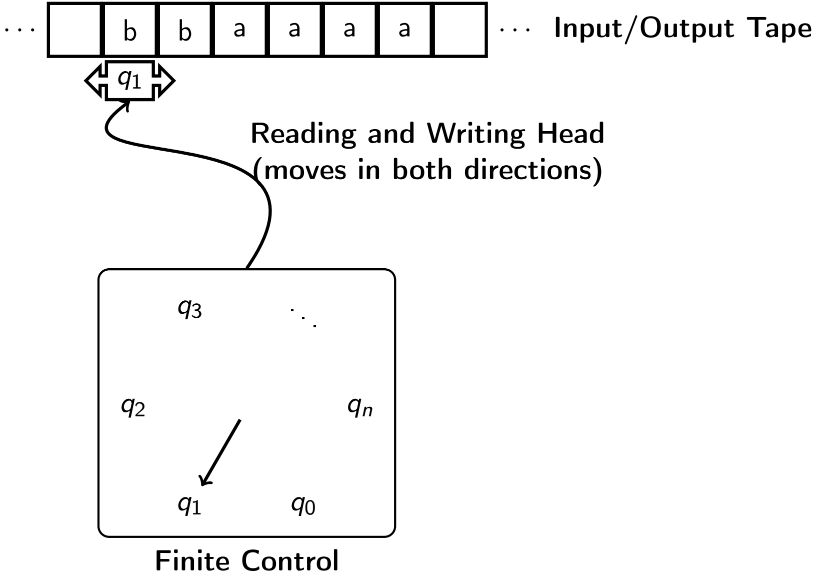
How was Turing trying to do everything he did?

# Mark I, 1948. Manchester's Ferranti Bros.



# Mark I, 2011

# Turing Machine

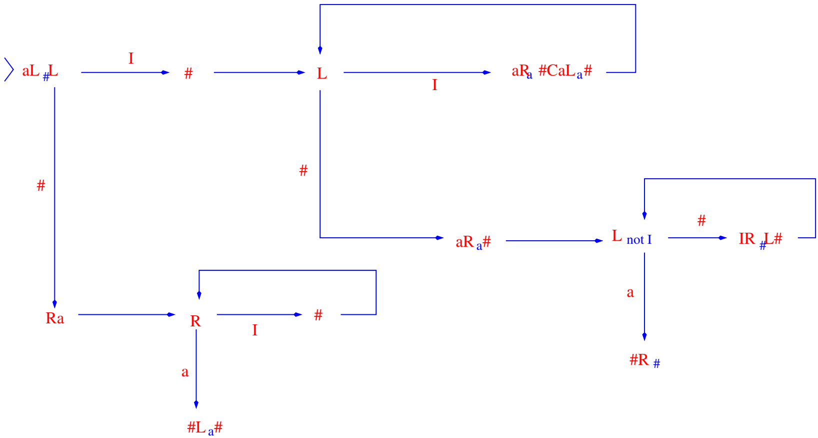


Input/Output Tape

Reading and Writing Head (moves in both directions)

Finite Control

# A TM



A Turing machine to compute f(n,m)=n x m

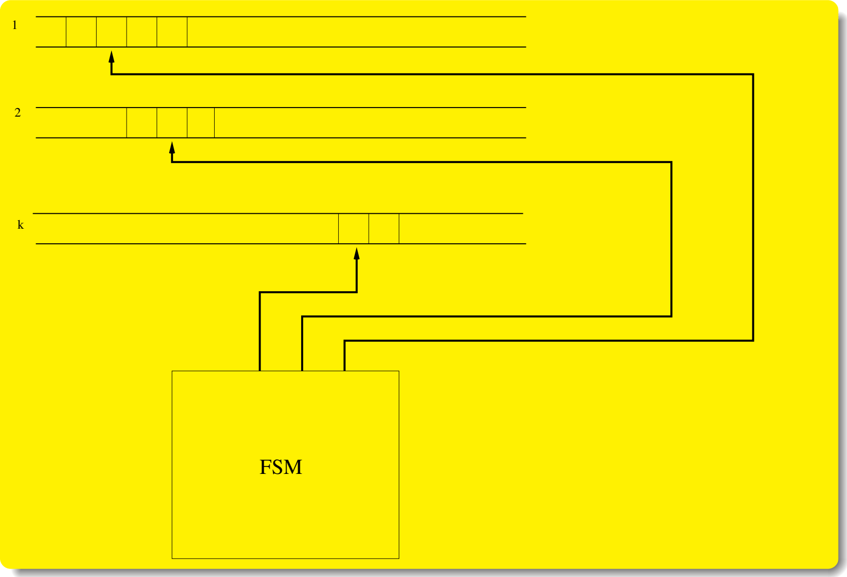
It copies the second string a number of times given by the first string.

From Lewis and Papadimitriou (1981).

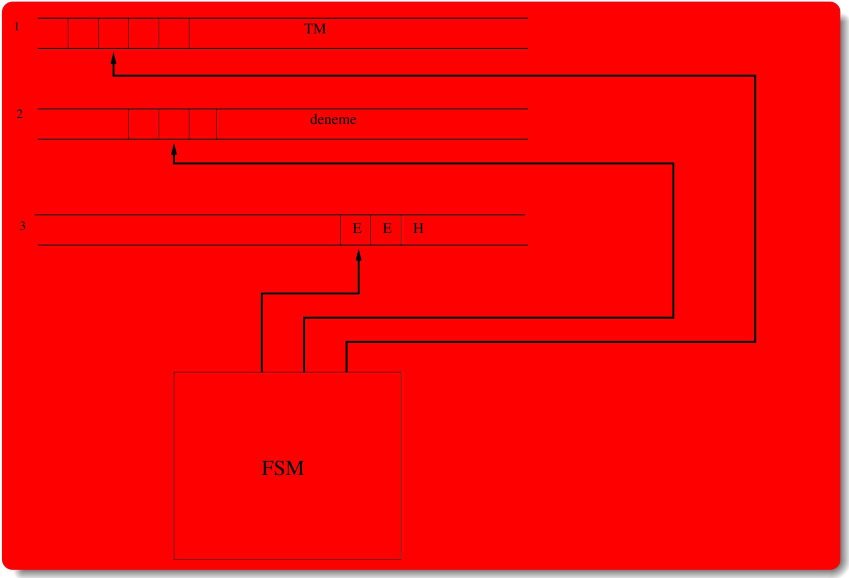
Representation: maps $\#I^m\#I^n\underline{\#}$ to $\underline{\ }aI^m\,aI^n$ first

| | |
|---|---|
| ‘#’ | blank |
| x | write x |
| L | move left |
| R | move right |
| $L_x$ | find first x on left |
| $R_x$ | find first x on right |

# k-tape TM



# Putnam-Gold TM



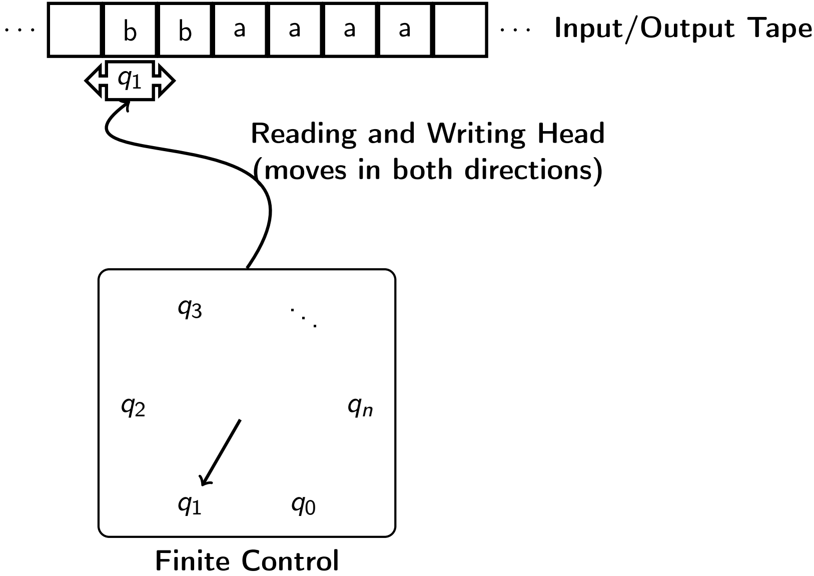
# Non-Turing extensions to Computing

- Analog devices (where everything represents itself)
- Quantum computing
    - Bankers beware
    - New measures of complexity

# What is essential to Turing?

- P-G machine is a TM
- The nature of relation between representation and computing.
- Complexity is defined over representation rather than physical time and space.
- One more move of a TM makes the solution one step more involved.

# TM

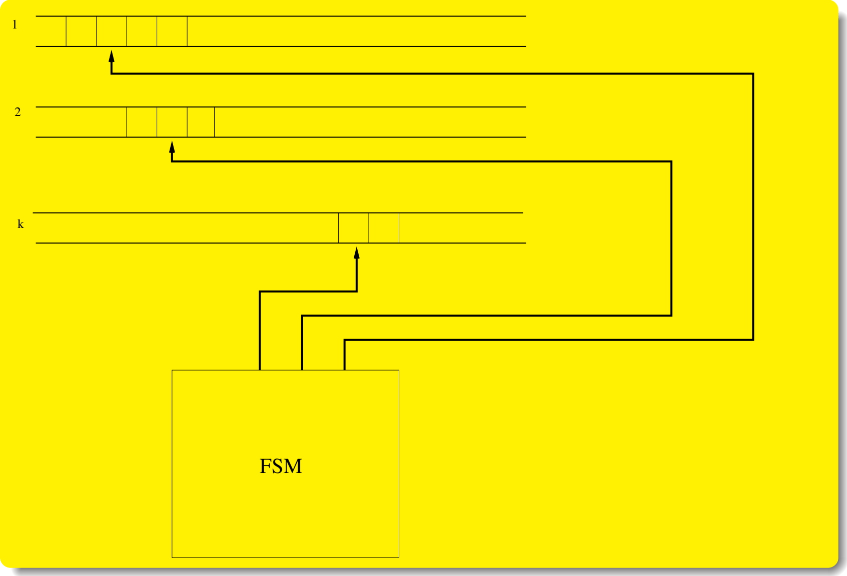


Input/Output Tape

Reading and Writing Head
(moves in both directions)

Finite Control

# k-TTM



# Problems

- Formulable but unsolvable problems

*diagonal(X)*:
*a*: if *halt(X,X)* goto *a* otherwise halt.

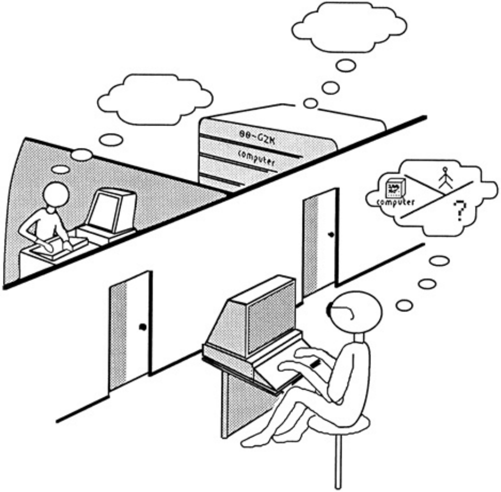
*diagonal(diagonal)* ?

- Expressible but unformulable problems

What is the next number after $\pi$?

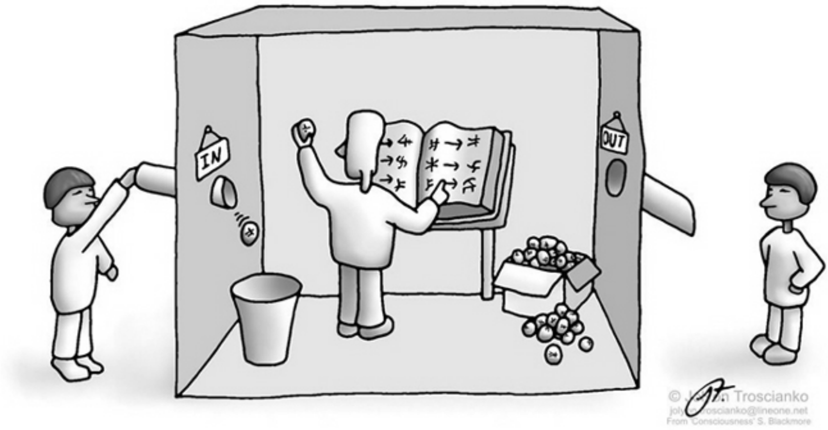
- The difference is TM representability.

# Turing Test



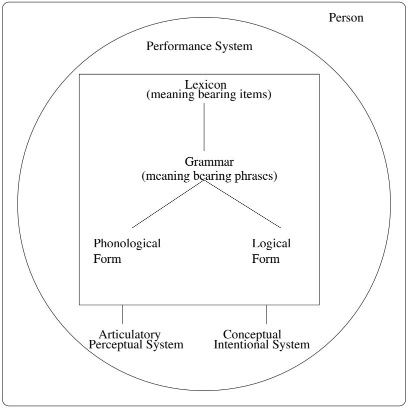
(Ann Witbrock; Copeland, B.J., Artificial Intelligence Blackwell Publishers, Oxford, 1993.)

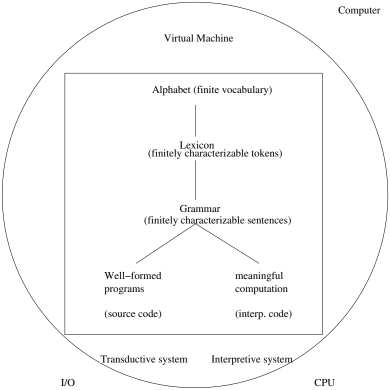
# Searle and Turing



© Jolyon Troscianko
jolyon.troscianko@lineone.net
From 'Consciousness' S. Blackmore

# Programming and Turing



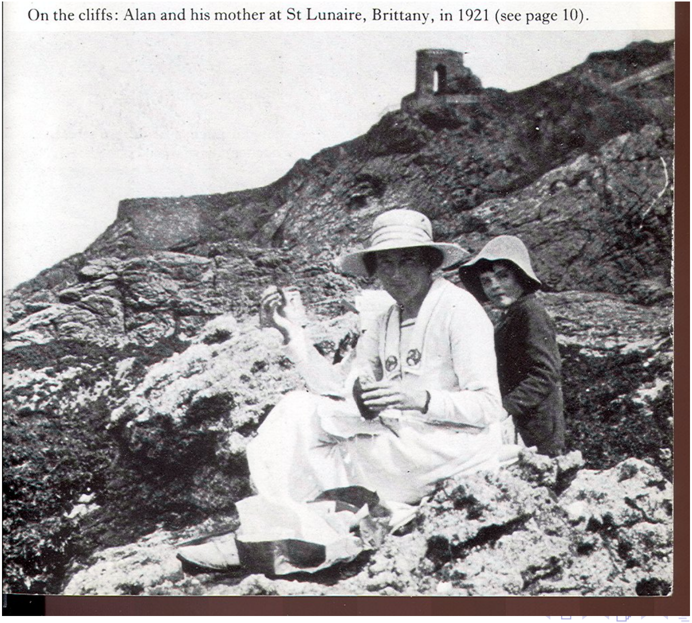


- does it look like a TM?
- AMT: intelligent behavior can arise from manipulation of representation.
- With representations, we can build models. Without them we talk about them.
- Turing representability is the key to Turing's computing.

# AMT @ 13

"I always seem to want to make things from the thing that is commonest in nature."

# AMT @ 9



On the cliffs: Alan and his mother at St Lunaire, Brittany, in 1921 (see page 10).