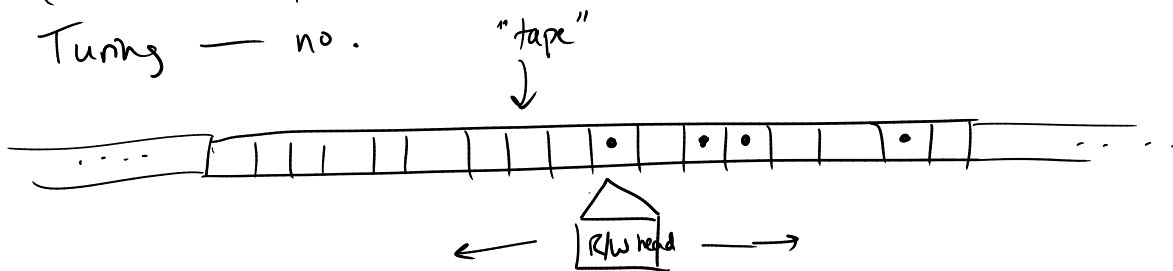# Turing machines

Alan Turing — 1936 PhD thesis.

  (Entscheidungsproblem — "decision problem" — Hilbert 1900)

Turing — no.   "tape"

Also has an internal "state"

Based on current state + contents of tape @ R/w head

  → decide what to write + which way to move + next state.

We will build a Turing machine — inspired computer.

— Memory (stores bits)

— Current memory location (R/w head)

— Instruction memory (RAM)

— PC

— A register

— 6 instructions:

		MR	move right
ML	move left
⌈ ER	erase
⌊ WR	write
⌈ JC	jump if clear
⌊ JS	jump if set.

2	1	0
1	0	0
1	0	1
0	0	0
0	0	1
0	1	0
0	1	1

$\neg(a \lor b)$
$\equiv \neg a \land \neg b$

• A instruction: 0 ⎵⎵⎵⎵⎵
                         5 bits.

(PC = 5 bits,
     programs ≤ 32 instructions)

∘ C instruction: ①11 ⎵⎵⎵
                        3 bits.

∘ Registers: — PC (5 bits)   — RAM (2^n bits)
                 — A (5 bits)
                 — Head (n bits)

Key ideas

- Always compute everything.
- Carefully control when each memory is loaded.
- With each instruction, what changes?

ROM (instruction)

addr /5 → ROM → instr /6

instr[0..4] /5 → in | A | load | out → a

load = instr[5]

Head:
in → Head → out → head

+1
-1

instr[0]

load = instr[5] || instr[2]

RAM ("tape"):
instr[0] /1 → in
head /n → addr | RAM ("tape") | out /1 → mem

load

instr[5]
instr[1] instr[2]

$$\text{instr}[5] \wedge \neg (\text{instr}[1] \vee \text{instr}[2])$$

a —/— in　PC　out —/— addr
　　5　　load　inc　　5