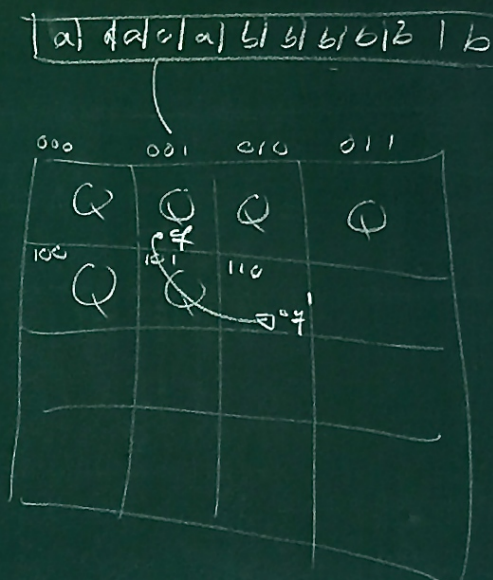
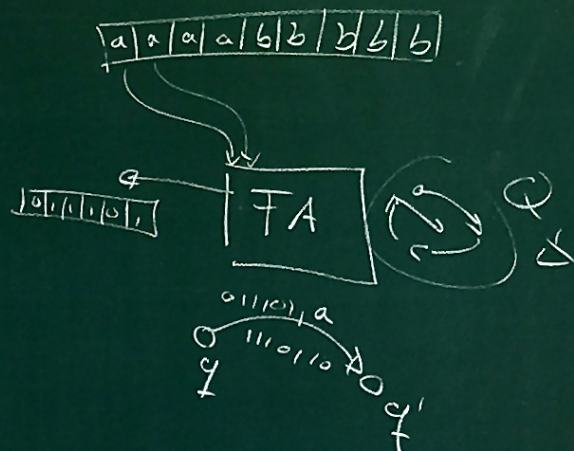


2. Memory, Tapes and TMs

Finite Automata

Memory 2.

- Registers, variable $\{0,1\}^c$



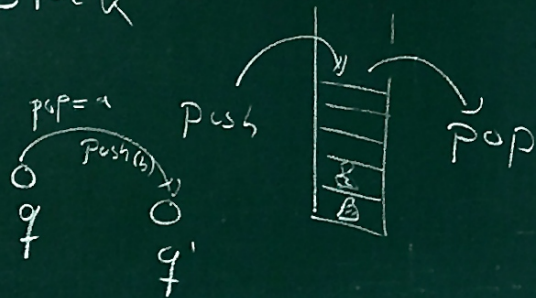
Storing of
finite memory
in the state space

2. Memory, Tapes and TMs

Finite Automata

Memory 2.

- Registers, variable $\{0,1\}^c$
- Stack



- $FA + \text{stack} \subseteq CFL$ (context-free languages)
- det. Pushdown-Automata

$$L = \{a^n b^n c^n \mid n \in \mathbb{N}\}$$

$$\{ww \mid w \in \Sigma^*\} = \text{COPY}$$

- $FA + 2\text{stacks}$
- infinite tapes

2. Memory, Tapes and TMs

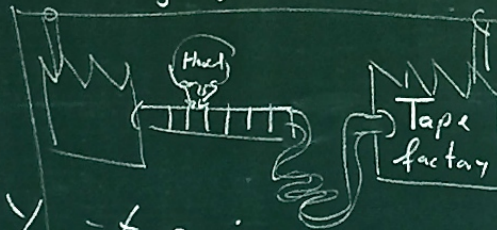
- $FA + \text{stack} \subseteq CFL$ (context-free languages)
det. Pushdown-Automata

$$L = \{a^n b^n c^n \mid n \in \mathbb{N}\}$$

$$\{ww \mid w \in \Sigma^*\} = \text{COPY}$$

- $FA + 2 \text{ stacks}$

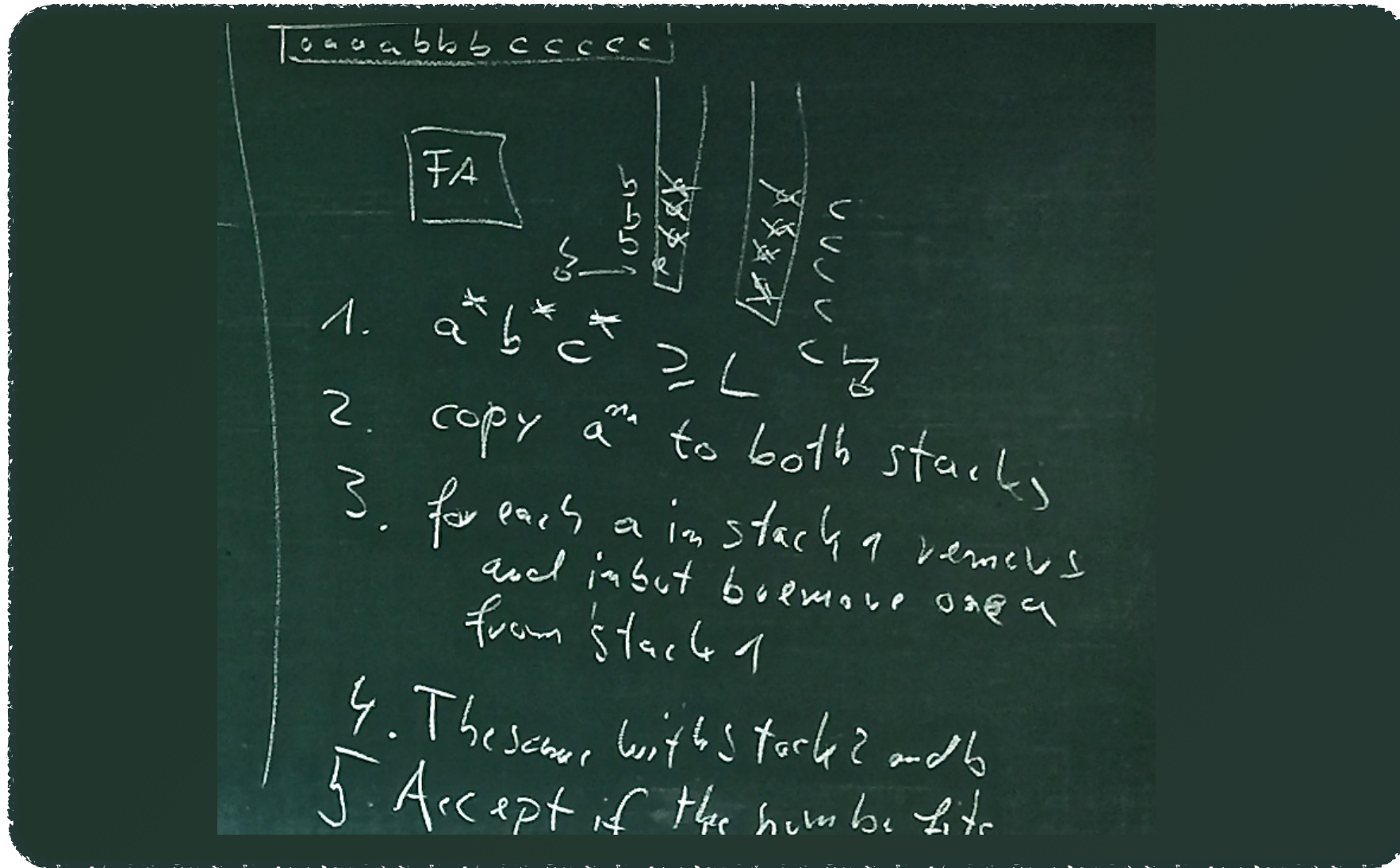
- infinite tape



- tape inscription = string $\in \Sigma^*$
- head position
- move left/right
- tape alphabet

FA_1 | FA_2

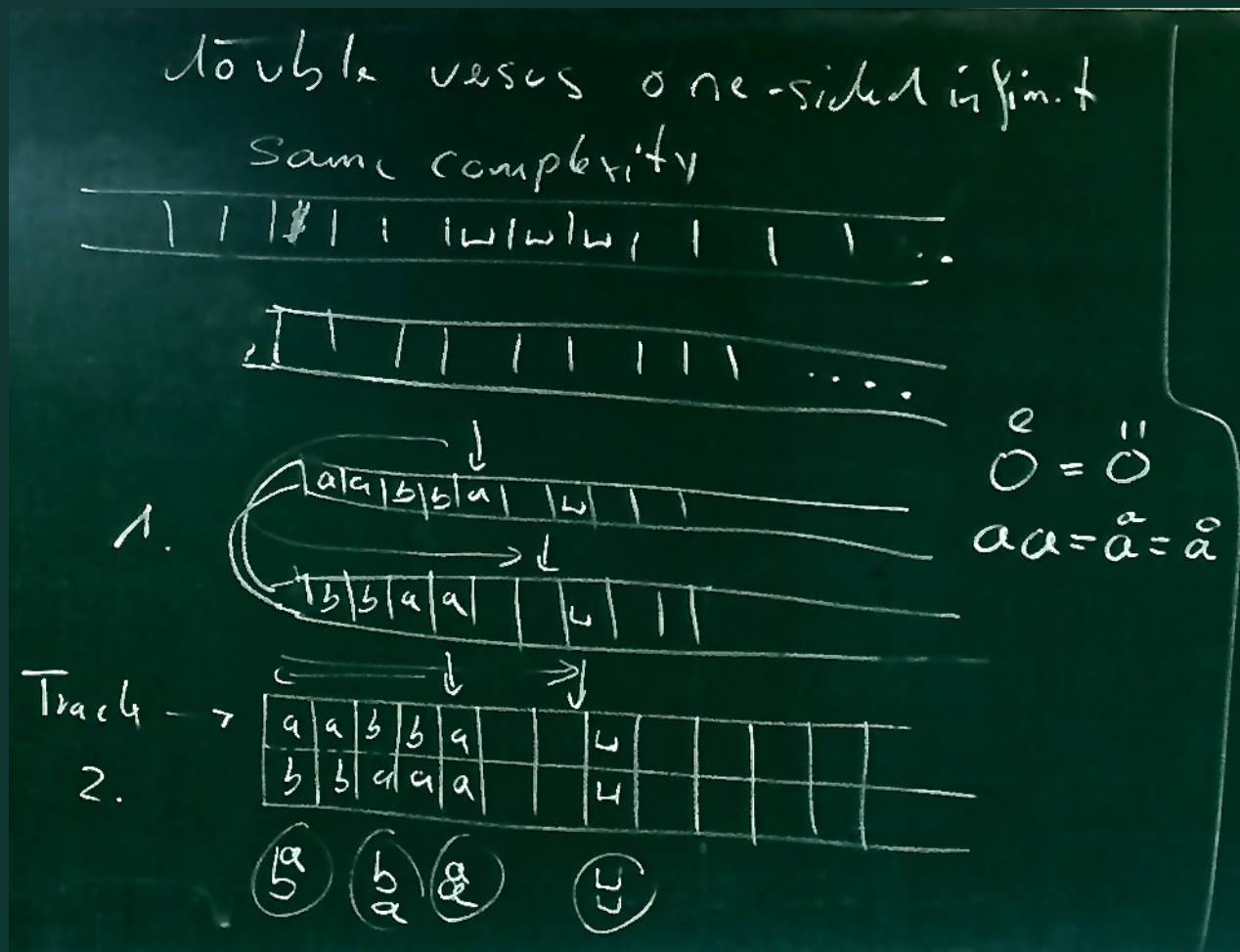
2. Memory, Tapes and TMs



The diagram shows a Turing Machine (FA) with a tape containing the string $T a a b b b c c c c c$. The tape head is positioned over the first 'a'. To the right of the tape, there are two vertical stacks of symbols. The first stack contains 'a's and 'b's, and the second stack contains 'c's. Arrows indicate the flow of symbols from the tape to the stacks.

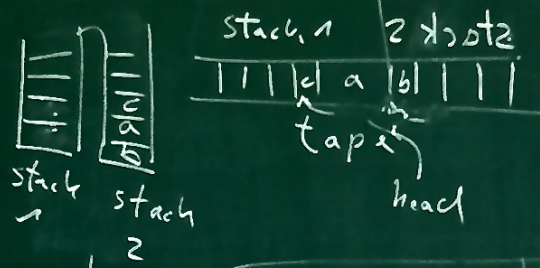
1. $a^* b^* c^* \geq L^c$
2. copy a^m to both stacks
3. for each a in stack 1 remove a and input b or remove a or c from stack 1
4. The same with stack 2 and b
5. Accept if the number is 1

2. Memory, Tapes and TMs



2. Memory, Tapes and TMs

2 stack vs tape Some complexity



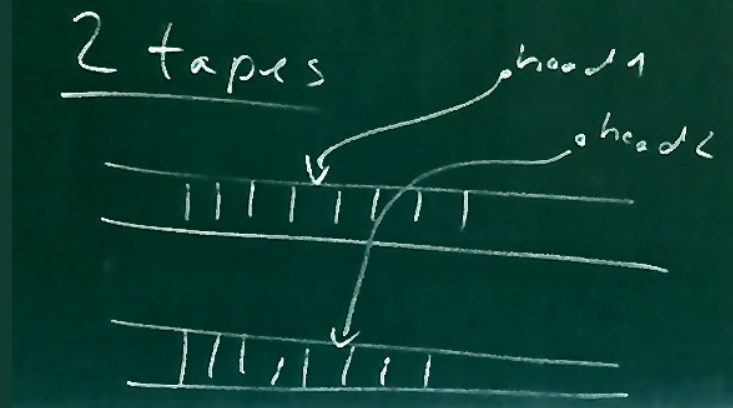
Tape: a a b b b c c c c c

FA

1. $a^* b^* c^* \geq L^c b$
2. copy a^m to both stacks
3. for each a in stack 1 remove a and input b remove one a from stack 1
4. The same with stack 2 and b
5. Accept if the number fits

2. Memory, Tapes and TMs

Lemma Every TM with k tapes with time t and space S can be simulated by a 1 tape TM in time $2 \cdot k \cdot s \cdot t$ and spaces.

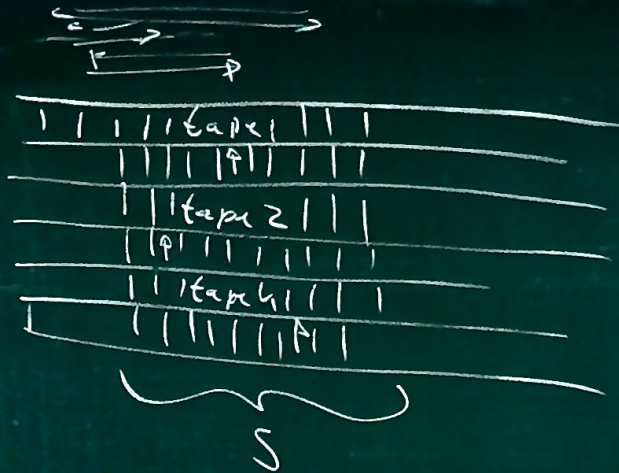


Proof

1. One tape with k tracks (copies of the k tapes) + special marks for the head (extra k tapes)

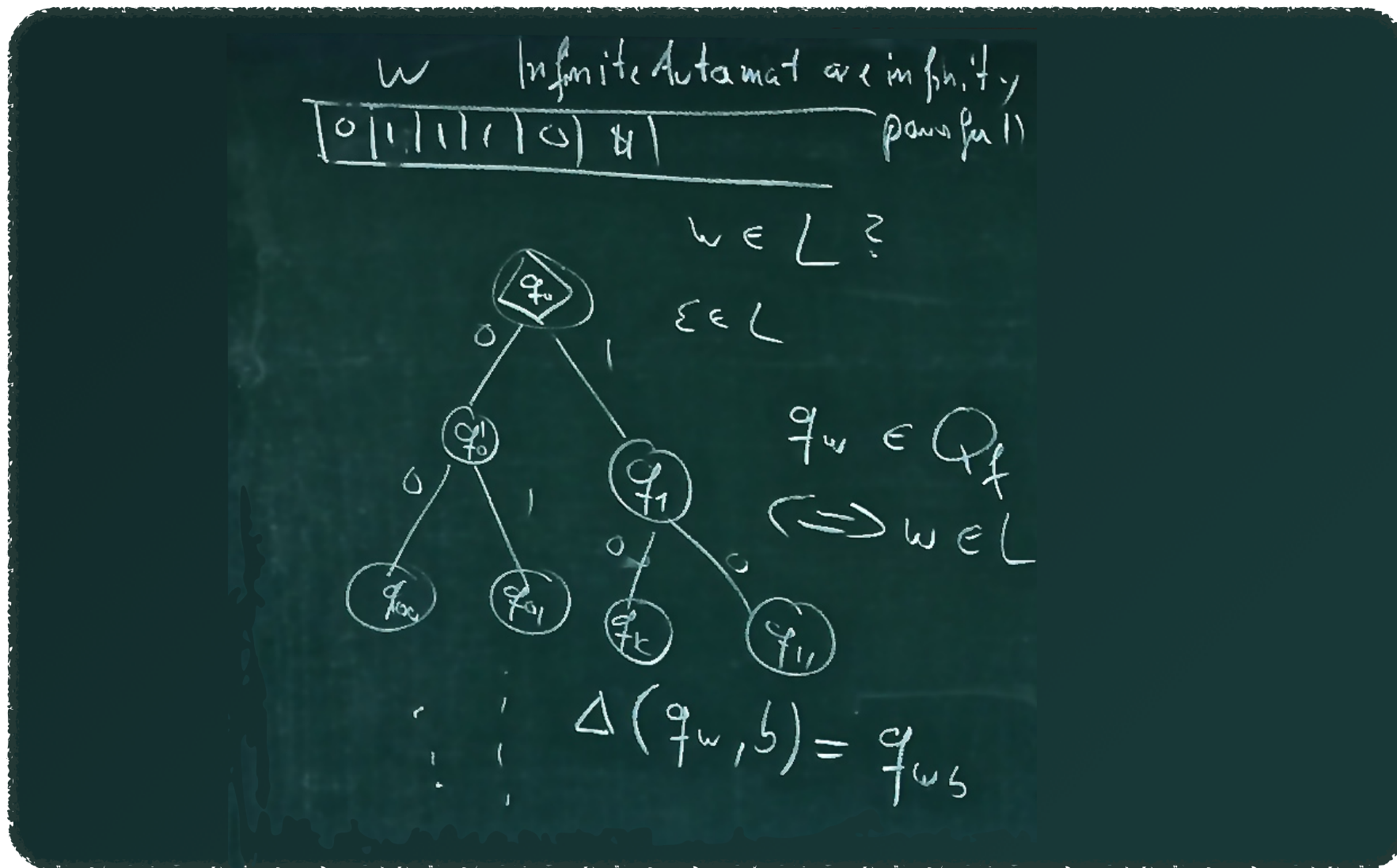


2. Memory, Tapes and TMs



Simulation
 time $2 \cdot s \cdot k$ { Read for each tape
 - go to head
 - store letter in fin. + mem.
compute - for fin.
 time $2 \cdot s \cdot k$ { Write & Move for each tape (to act)
 - go to head
 Repeat until finish - write & move head letter
 overall time: $2 \cdot k \cdot s \cdot t$

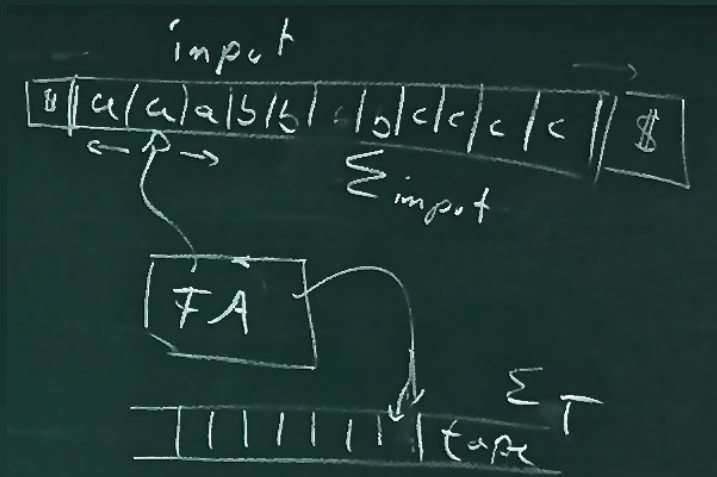
2. Memory, Tapes and TMs



2. Memory, Tapes and TMs

Church-Turing-Thesis

Everything computable can be
computed by a single tape TM



Def 1-Tape TM with input tape

$$M = (\Sigma_{input} \cup \{\emptyset\}, \Sigma_T, Q, q_0, q_{acc}, q_{rej}, \Delta)$$

$$\Delta: Q \times \Sigma_{input} \cup \{\emptyset\} \times \Sigma_T \rightarrow Q \times \{\ell, r\} \times \Sigma_T \times \{\ell, r\}$$

2. Memory, Tapes and TMs

configuration : global state of a TM M
at a time

- input string $x \in \Sigma_{\text{input}}^*$
- head pos. on input $h \in \mathbb{N}$
- all written letter word on tape $w \in \Sigma_T^*$
- head position on the tape
- state of F_A

2. Memory, Tapes and TMs

What happens when a configuration repeats?

$$C_{init} \xrightarrow{*} C_{acc} \iff x \in L(M)$$



$C_1 \vdash C_2 \iff$ one calculation step

C_{init} is first configuration

C_{acc} accepting configuration $q \in Q_{acc}$

C_{rej} rejecting conf $q \in Q_{rej}$