

Models of Computation

(5)

history: beginning of 20th century: people asking for "mechanical procedures" for solving math problems - e.g., solving integer polynomial equations

1930s: Gödel, Church, Kleene, Turing: formal definitions of computation (all of them equivalent, yet different)

What problems we want to solve?

↳ fix "language"

- Σ - finite alphabet of symbols (characters) - examples: $\{0,1\}$, $\{a \dots z\}$, math symbols
- Σ^* - set of all words (finite sequences) over Σ

• ϵ ... empty word

• $|x|$... length

• $\alpha\beta$... concatenation

• symbol $\approx$ 1-symbol word

• $\alpha[i]$... i-th symbol (starting with 0)

• $\alpha[i:j] = \alpha[i] \dots \alpha[j-1]$... subword

• $\alpha[:j] = \alpha[0:j]$... prefix

• $\alpha[i:] = \alpha[i:|\alpha|]$... suffix

every alphabet can be encoded in every other (assuming $|\Sigma| \geq 2$)

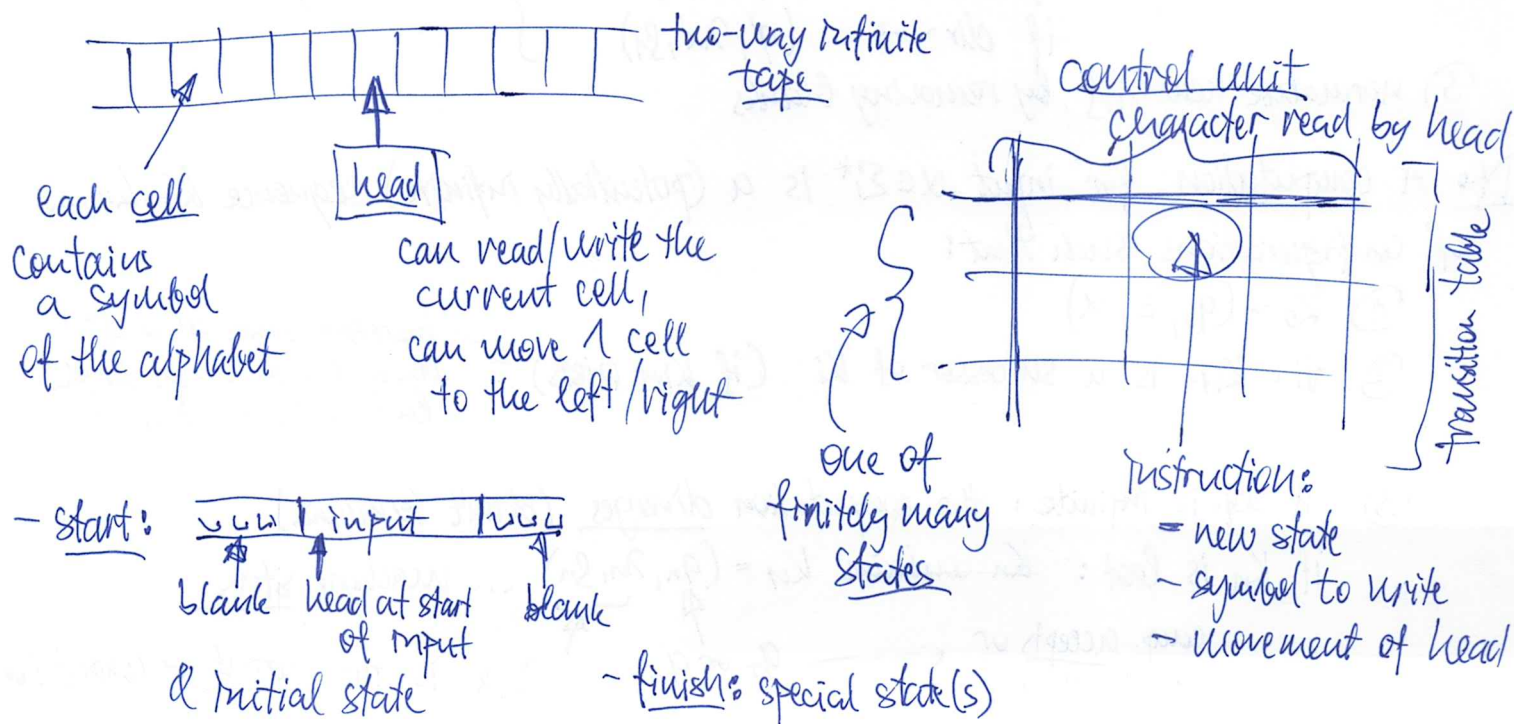
$$\alpha^k = \underbrace{\alpha \dots \alpha}_k, \alpha^0 = \epsilon$$

- problem: function from Σ^* to Σ^*
- decision problem: $f: \Sigma^* \rightarrow \{0,1\}$

↳ also viewed as language $L \subseteq \Sigma^* : x \in L \Leftrightarrow f(x) = 1$ (characteristic function of a set)

- usually we find encoding of inputs (e.g. polynomials) to strings
 - concrete encoding doesn't matter (they can be converted algorithmically)
 - what happens if the input string is not a valid encoding?
 - ↳ suppose we always answer ϵ or 0 in such cases.

Turing Machines motivation: a mathematician with finite mind working on an infinite blackboard



Now formally...

(6)

Df: A Turing machine consists of:

- Q ... a finite set of states
 - $q_0 \in Q$... initial state
 - $q_+, q_- \in Q$... final states (accepting & rejecting)
 - Σ ... non-empty finite input alphabet
 - $\Gamma \supseteq \Sigma$... finite work alphabet
 - $\sqcup \in \Gamma \setminus \Sigma$... blank symbol
 - $\delta: (Q \setminus \{q_+, q_-\}) \times \Gamma \rightarrow Q \times \Gamma \times \{\leftarrow, \cdot, \rightarrow\}$... transition function
- } q_0, q_+, q_- all distinct

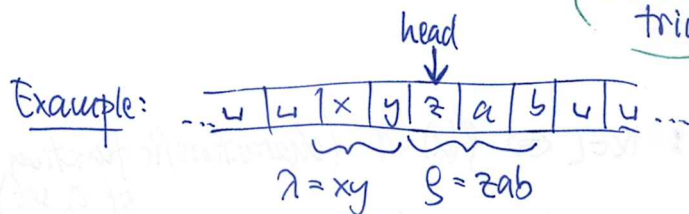
Df: Configuration of the TM: $(q, \lambda, \rho) \in Q \times \Gamma^* \times \Gamma^*$

replaced by
Page 600

current
state

left part of tape
before the head
with leading blanks
trimmed

right part of tape
from the head onwards
with trailing blanks trimmed
normalization



$\lambda \rho$ = non-blank part of tape
on empty tape, all positions of the head
are the same configuration

Df: A configuration (q, λ, ρ) , $q \neq q_+, q_-$ has a successor defined in this way:

- ① extend λ by a leading $\sqcup$, ρ by a trailing $\sqcup$
 - ② now, $\lambda = \lambda'x$, $\rho = x\rho'$ for some x, λ', ρ'
 - ③ evaluate $\delta(q, x)$... get (q', x', dir)
 - ④ execute instruction:
 - if $\text{dir} = \cdot$... $(q', \lambda, x'\rho')$
 - if $\text{dir} = \leftarrow$... $(q', \lambda', x\rho')$
 - if $\text{dir} = \rightarrow$... $(q', \lambda x', \rho')$
 - ⑤ normalize new λ, ρ by removing blanks
- } new config.

Df: A computation for input $\alpha \in \Sigma^*$ is a (potentially infinite) sequence $K_0, K_1, \dots$ of configurations such that:

① $K_0 = (q_0, \epsilon, \alpha)$

② $\forall i: K_{i+1}$ is a successor of K_i (if K_{i+1} exists)

therefore state q_+ or q_-
never occurs except for the
last config of a finite seq.

③ if seq. is infinite: the computation diverges (doesn't terminate)

if K_n is last: K_n contains $K_n = (q_n, \lambda_n, \rho_n)$... machine stops

comp. accepts or
rejects the input

q_+ or q_-

$\lambda_n \rho_n$ is the output of computation

Df: Configuration of the TM: (q, π, i)

$q \in Q$ current state

$i \in \mathbb{Z}$ head position

tape contents $\pi: \{1, \dots, r\} \rightarrow \Gamma$

positions visited by the head so far

outside $1, \dots, r$ the tape contains only blanks

(6a)

Df: Successor of a configuration (q, π, i) is a config. (q', π', i') $\left[\begin{array}{l} \text{a single step} \\ \text{of computation} \end{array} \right]$

$q \neq q_+, q_-$

- defined in this way:
- ① evaluate $\delta(q, \pi[i]) \rightarrow \text{instruction } (q', x', \text{movement})$
 - ② write x' on the tape: let $\pi' = \pi$ everywhere except $\pi'[i] = x'$
 - ③ move the head:
 - if movement = $\bullet \Rightarrow i' = i$
 - if movement = $\leftarrow \Rightarrow i' = i - 1$
 - if movement = $\rightarrow \Rightarrow i' = i + 1$
 - ④ if $\pi'[i']$ is not defined yet, put $\pi'[i'] = \sqcup$

Df: Computation of the TM for input $\alpha \in \Sigma^*$ is a (potentially infinite) sequence $K_0, K_1, \dots$ of configurations s.t.

- $K_0 = (q_0, \alpha, 0)$ extended by $\alpha[0] = \sqcup$ if we had $\alpha = \epsilon$
- $\forall i$ K_{i+1} is a successor of K_i
- if the computation is finite, the last K_i contains q_+ or q_-
- ...! no other K_i contains q_+ nor q_-

Comp. is finite $\rightarrow$ the machine stops and accepts (q_+) or rejects (q_-) the input α .

The final contents of the tape is the output (after trimming blanks)

$\hookrightarrow$ we assume that the output ^{leading/trailing} doesn't contain characters in $\Gamma \setminus \Sigma$

Comp. is infinite $\rightarrow$ the TM diverges (doesn't terminate)

Df: Computability:

Function $f: \Sigma_1^* \rightarrow \Sigma_1^*$ is computable
 $\equiv \exists$ M Turing machine s.t.
 $\forall x \in \Sigma_1^* \quad M(x)$ halts and outputs $f(x)$
 ↑
 M on input x
 (& its computation)

Language $L \subseteq \Sigma_1^*$ is computable
 $\equiv \exists$ M T.m. s.t.
 $\forall x \in \Sigma_1^* \quad M(x)$ always halts
 & (accepts $x \iff x \in L$)
 ends in q^+

→ equivalent to char. fn of L computable

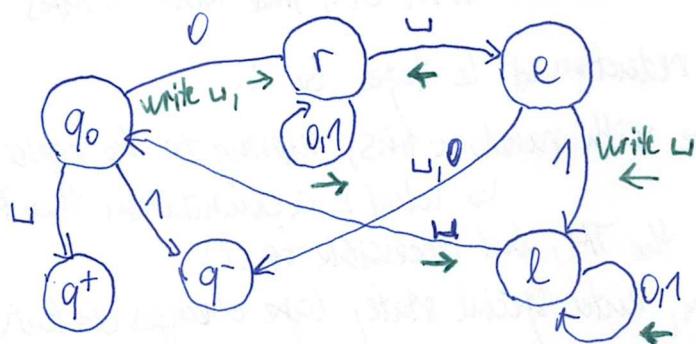
Function $f: \Sigma_1^* \rightarrow \Sigma_1^* \cup \{\uparrow\}$ ($\uparrow \notin \Sigma_1$)
 is partially computable $\equiv \exists$ M T.m.
 s.t. $\forall x \in \Sigma_1^* \quad$ if $f(x) = \uparrow$: $M(x)$ diverges
 else $M(x)$ halts & outputs $f(x)$

Language $L \subseteq \Sigma_1^*$ is partially computable
 $\equiv \exists$ M T.m. s.t.
 $\forall x \in \Sigma_1^* \quad M(x)$ halts $\iff x \in L$.
 in state q^+
 ↓
 also recursively enumerable
 → equivalent to $\chi_L(x) = \begin{cases} 1 & \text{if } x \in L \\ \uparrow & \text{if } x \notin L \end{cases}$ partially computable

Idea: Time & Space Spent by computation
 ↑
 # configurations visited
 $\equiv$ # instructions executed
 # cells visited by the head

} will serve as basis
 for complexity theory
 (later)

Example: Recognizing $\{0^n 1^n\} \subseteq \{0,1\}^*$ by accepting/rejecting.
 Idea: erase first 0 & final 1, repeat.



$\Sigma = \{0,1\}$

$\Gamma = \{0,1, \sqcup\}$

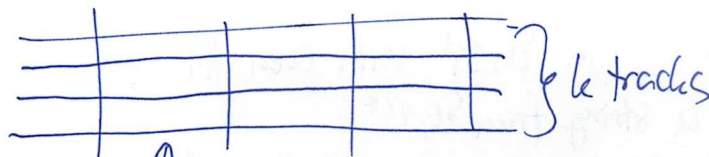
$Q = \{q^0, q^+, q^-, r, e, l\}$

doesn't matter

$\bar{q}$	0	1	$\sqcup$
q_0	$(r, u_1 \rightarrow)$	$(q^-, ?, ?)$	$(q^+, ?, ?)$
r	$(r, 0, \rightarrow)$	$(r, 1, \rightarrow)$	$(e, \sqcup, \leftarrow)$
e	$(q^-, ?, ?)$	$(l, \sqcup, \leftarrow)$	$(q^-, ?, ?)$
l	$(l, 0, \leftarrow)$	$(l, 1, \leftarrow)$	$(q_0, \sqcup, \rightarrow)$

Ideas: • Encode multiple variables with finite domains in the state — state is a tuple

• Multi-track tape



head reads/writes
 k -tuples

↑ each cell contains k -tuple
 Remember to en/decode tape at start/end of computation.

But all tracks share
 head position

Variants of the TM (robustness of definition)

(8)

- ① One-way infinite tape ... 1-way $\rightarrow$ 2-way trivial
 2-way $\rightarrow$ 1-way "fold tape in half" simulation
 ↳ equally powerful
 (set of computable functions remains unchanged...)
- $\dots -2 \ -1 \ 0 \ 1 \ 2 \ \dots \rightarrow \begin{array}{cccc} 0 & +1 & +2 & +3 & \dots \\ -1 & -2 & -3 & -4 & \dots \\ * & & & & \end{array} \left. \vphantom{\begin{array}{cccc} 0 & +1 & +2 & +3 & \dots \\ -1 & -2 & -3 & -4 & \dots \\ * & & & & \end{array}} \right\} 3 \text{ tracks}$
- mark end of tape in track #3

state contains "positive half-tape" switch.

- ② k tapes with independent heads (but sharing a common work alphabet (Σ))

• transition function: $(Q \setminus \{q^+, q^-\}) \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{\leftarrow, \rightarrow, \bullet\}^k$

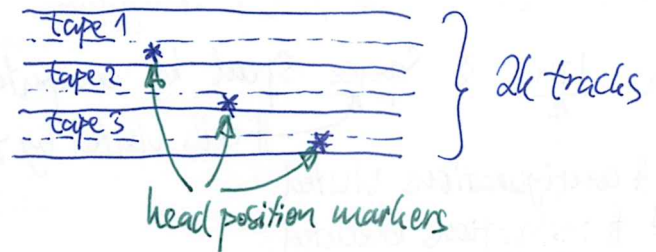
• configuration, successor, computation easy to extend

- start: tape #1 contains input, other tapes empty
- end: tape #1 contains output

• Equally powerful ... k-tape $\rightarrow$ 1-tape:

1 step of orig. machine can be simulated by scanning the whole tape 2 times:

- find all heads, record symbols they see in state
- write symbols back & move heads



actually, you can do it in $O(n \log n)$ time (exercise)

• But complexity changes: $\{0^n 1^n\}$ requires super-linear time with 1 tape but can be solved in $O(n)$ time with 2 tapes

• Later: there is a more efficient reduction of k tapes to 2.

- ③ Randomized TM ... read-only tape with random bits, moving to the right only
 ↳ what is a computation then? ...

- ④ Oracles (functions defined outside the TM, but accessible to it)

• oracle tape: write query there, enter special state, tape changes contents to the answer

- ⑤ Interactive TM ... outside world can be modelled as an oracle $\Rightarrow$

- ⑥ Exercise: 2-Dimensional tape ... head moves $\leftarrow, \rightarrow, \uparrow, \downarrow, \bullet$

... in some sense, the most powerful physically feasible computer is a TM with 3-D tape (or maybe 2-D only to allow head spreading...)

Exercises: - accept strings in $\{0,1\}^*$ with even #1

- reverse a string from $\{0,1\}^*$

- add / multiply numbers written in binary ... what's the complexity?
 ↑ non-negative integers