HW 4 today

Midterm 1 is Mon 29th

Fill out conflict exam form by Fri the 26th.

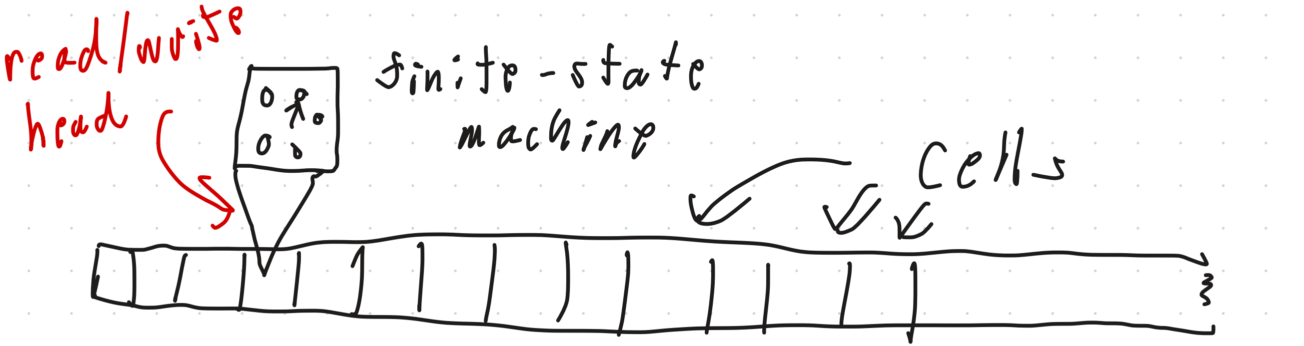
Capabilities	Languages	Machine
Sequencing Branching Repetition	Regular	DFA/NFA
Recursion	Context-free	Recursive NFA Pushdown automaton
Memory	Decidable/ Recursive	"Arbitrary computation" Python RISC Turing machine

Motivation: Entscheidungsproblem "decision problem"

Gödel, Church, Turing: NO!

Church-Turing thesis: λ-calculus and Turing-machines can execute "any" "calculation by finite means"

Turing machine

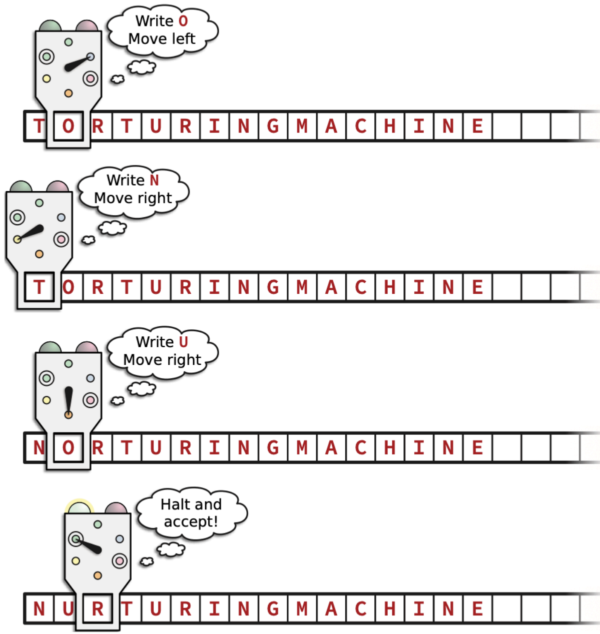


semi-infinite tape

transition based on current state and what is written on head's cell

write new symbol at head's cell, change state, and move one cell left or right

continue until you get to accept or reject state



Formally ...

Arbitrary finite Γ with $|\Gamma| \geq 2$: tape alphabet

A blank symbol $\square \in \Gamma$

$\Sigma \subseteq (\Gamma \setminus \{\square\})$: input alphabet

Finite Q : the states

Three distinct special states
start, accept, reject $\in Q$

Transition function

$\delta : (Q \setminus \{\text{accept}, \text{reject}\}) \times \Gamma \to$

$$Q \times \Gamma \times \{-1, +1\}$$

At all times, <u>configuration</u> $(q, x, i) \in Q \times \Gamma^* \times \mathbb{N}$

position of head

internal state q

tape has x followed by infinite blanks

Given input $w \in \Sigma^*$, start at $(\text{start}, w, 0)$.

If in configuration $(p, x\,a\,y, i)$ with $|x| = i$

(so ith symbol is a) and $\delta(p, a) = (q, b, \pm 1)$,

move to configuration $(q, x\,b\,y, i \pm 1)$.

Machine <u>accepts</u> w if (after a finite sequence of transitions) it reaches (accept, ·, ·).

It <u>rejects</u> w if it reaches (reject, ·, ·)

Machine could <u>crash</u> (position goes negative)

- or -

it might never get to accept or reject state, looping forever.

M recognizes or accepts language L if it accepts w iff $w \in L$.

If such a machine exists, L is recognizable, semi-computable, or recursively enumerable

M decides L if it accepts all $w \in L$ and rejects all $w \notin L$.

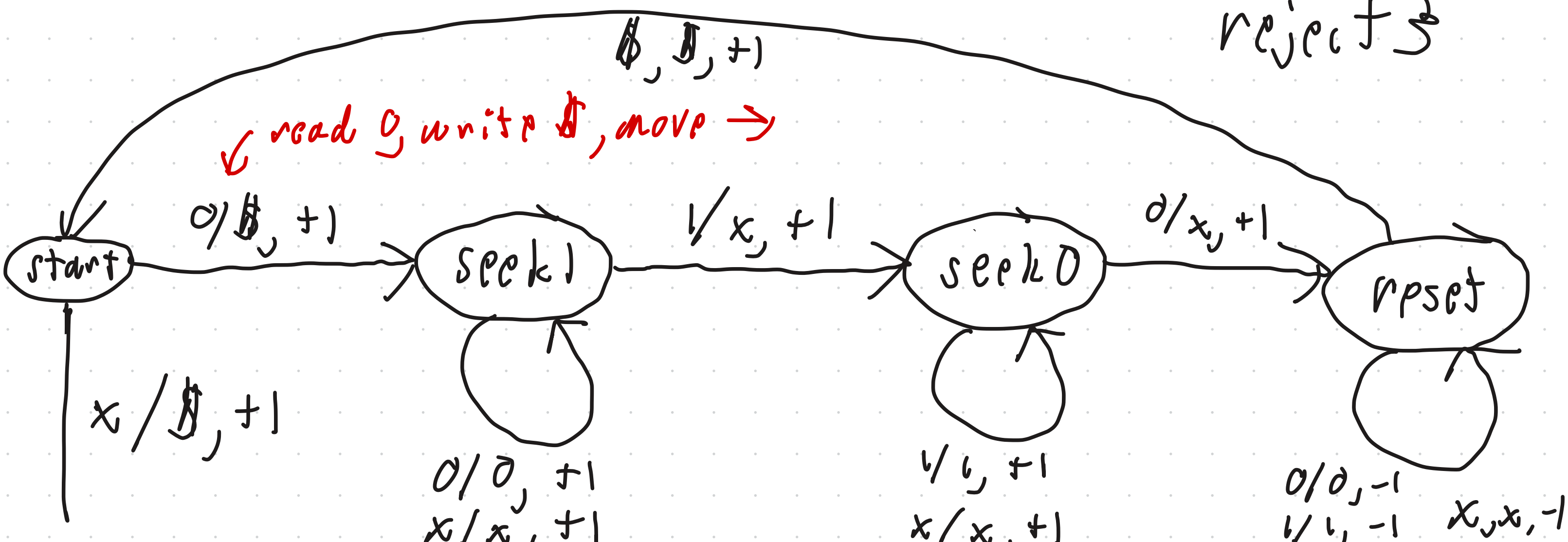
L is decidable, computable or recursive if some machine decides L.

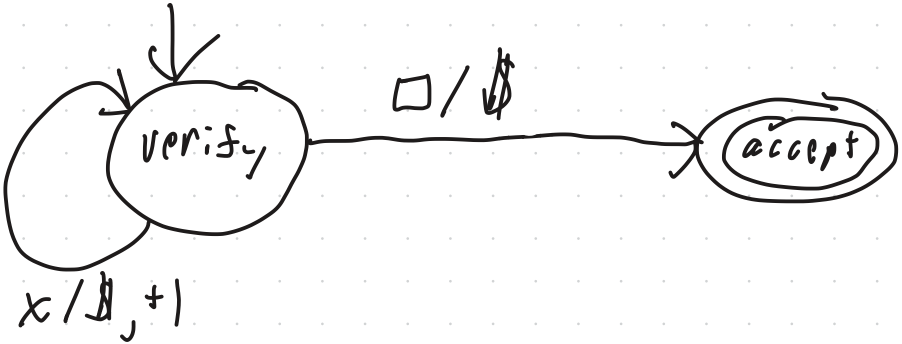
Ex: $\{0^n 1^n 0^n \mid n > 0\}$.

$\Gamma = \{0, 1, \$, x, \square\}$

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

$Q = \{$start, seek1, seek0, reset, verify, accept, reject$\}$





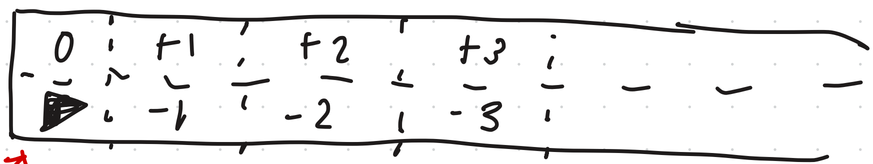
(start, 00100)
⇒ (seek1, $0100)
⇒ (seek1, $0100)
⇒ (seek0, $0x00)
⇒ (reset, $0xx0)
⇒ (reset, $0xx0)
⇒ (reset, $0xx0)
⇒ (start, $0xx0)
⇒ (seek1, $$xx0)
⇒ (seek1, $$xx0)
⇒ (seek1, $$xx0) ⇒ (~~reject!~~ seek1, $$xx0□)

(start, 001100)
⇒ (seek1, $01100)
⇒ (seek1, $01100)
⇒ (seek0, $0x100)
⇒ (seek0, $0x100)
⇒ (reset, $0x1x0)
⇒ (reset, $0x1x0)
⇒ (reset, $0x1x0)
⇒ (reset, $0x1x0)
⇒ (start, $0x1x0)
⇒ (seek1, $$x1x0)
⇒ (seek1, $$x1x0)
⇒ (seek0, $$xxx0)
⇒ (seek0, $$xxx0)
⇒ (reset, $$xxxx)
⇒ (reset, $$xxxx)
⇒ (reset, $$xxxx)
⇒ (reset, $$xxxx)
⇒ (start, $$xxxx)
⇒ (verify, $$$xxx)
⇒ (verify, $$$$xx)
⇒ (verify, $$$$$x)
⇒ (verify, $$$$$$□)
⇒ (accept, $$$$$$) ⇒ **accept!**

reject!

Doubly infinite tape...

$\Gamma' = \Gamma \times (\Gamma \cup \{\$\})$



$Q' = Q \times \{hi, lo\}$

so machine knows to swap between hi & lo states

Multiple tapes can be simulated.

Useful to define how much space is needed

Non-determinism...

Can model: loops,
function calls,
arrays,
Python code given as input...

Can simulate other Turing machines
given with a suitable encoding.

Universal Turing machine