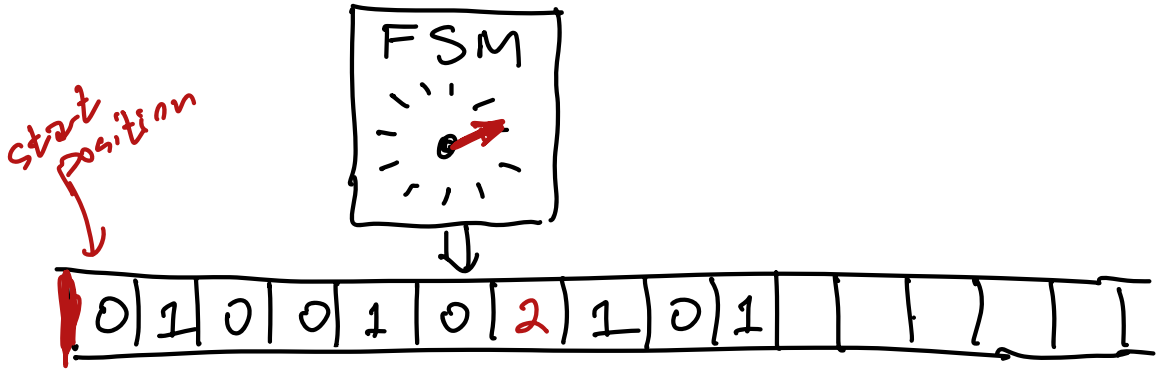
	Languages	Machine
Sequencing Branching Repetition	Regular	DFA / NFA
Recursion	Context-free	Pushdown automaton Recursive NFA
Memory	Recursive decidable	Turing machine Python RISC "Arbitrary computation"



At each step:

Read symbol at curr pos on tape

State

Write symbol at curr pos on tape

Change state

Move 1 step →or←

Entsheidungsproblem "Decision problem"

Gödel	Nope	Incompleteness
Church	Nope	λ-calculus
Turing	Nope	TM halting problem

There is no TM that can decide if an arbitrary TM halts.

There is no Python program that can decide, given source code for any Python program, whether that program halts.

CODE is DATA

Spose this is false


```
IHateMyself(M):
    if HaltOMatic(M) = True
        hang
    else
        return False
```

IHateMyself(IHateMyself)

Turing Machine $(\Sigma, \Gamma, \square, Q, \text{start}, \text{accept}, \text{reject}, \delta)$

Σ – input alphabet ← finite

$\Gamma \supseteq \Sigma$ – tape alphabet ← finite

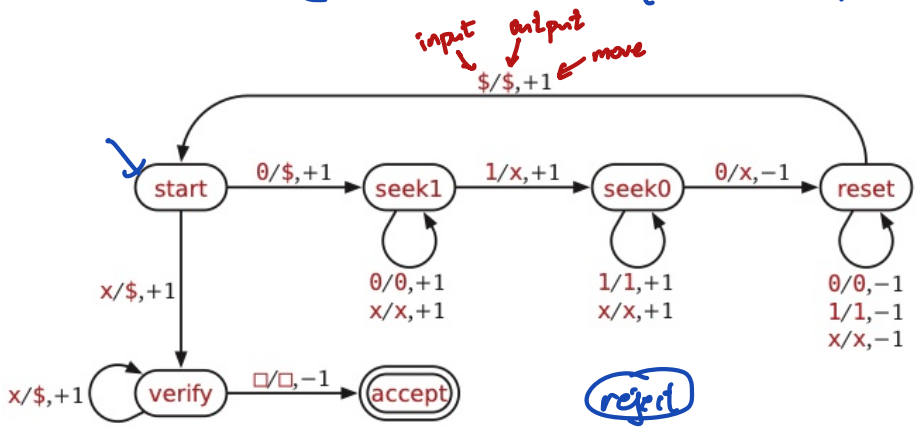
blank $\square \in \Gamma \setminus \Sigma$

Q – states ← finite

start, accept, reject $\in Q$

transition $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{-1, +1\}$

$\{0^n 1^n 0^n \mid n \geq 0\}$



$\delta(p, a) = (q, b, \Delta)$	explanation
$\delta(\text{start}, 0) = (\text{seek1}, \$, +1)$	mark first 0 and scan right
$\delta(\text{start}, x) = (\text{verify}, \$, +1)$	looks like we're done, but let's make sure
$\delta(\text{seek1}, 0) = (\text{seek1}, 0, +1)$	scan rightward for 1
$\delta(\text{seek1}, x) = (\text{seek1}, x, +1)$	
$\delta(\text{seek1}, 1) = (\text{seek0}, x, +1)$	mark 1 and continue right
$\delta(\text{seek0}, 1) = (\text{seek0}, 1, +1)$	scan rightward for 0
$\delta(\text{seek0}, x) = (\text{seek0}, x, +1)$	
$\delta(\text{seek0}, 0) = (\text{reset}, x, +1)$	mark 0 and scan left
$\delta(\text{reset}, 0) = (\text{reset}, 0, -1)$	scan leftward for $
$\delta(\text{reset}, 1) = (\text{reset}, 1, -1)$	
$\delta(\text{reset}, x) = (\text{reset}, x, -1)$	
$\delta(\text{reset}, \$) = (\text{start}, \$, +1)$	step right and start over
$\delta(\text{verify}, x) = (\text{verify}, \$, +1)$	scan right for any unmarked symbol
$\delta(\text{verify}, \square) = (\text{accept}, \square, -1)$	success!

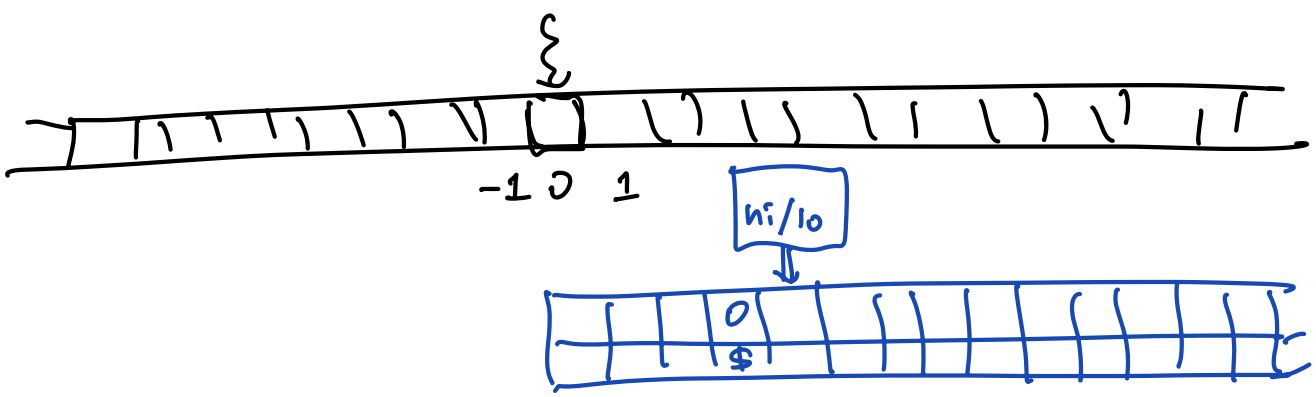
The transition function for a Turing machine that decides the language $\{0^n 1^n 0^n \mid n \geq 0\}$.

$\delta(\ p\ , a) = (\ q\ , b, \Delta)$
$\delta($ start $, 0) = ($ seek1 $, \$, +1)$
$\delta($ start $, x) = ($ verify $, \$, +1)$
$\delta($ seek1 $, 0) = ($ seek1 $, 0, +1)$
$\delta($ seek1 $, x) = ($ seek1 $, x, +1)$
$\delta($ seek1 $, 1) = ($ seek0 $, x, +1)$
$\delta($ seek0 $, 1) = ($ seek0 $, 1, +1)$
$\delta($ seek0 $, x) = ($ seek0 $, x, +1)$
$\delta($ seek0 $, 0) = ($ reset $, x, +1)$
$\delta($ reset $, 0) = ($ reset $, 0, -1)$
$\delta($ reset $, 1) = ($ reset $, 1, -1)$
$\delta($ reset $, x) = ($ reset $, x, -1)$
$\delta($ reset $, \$) = ($ start $, \$, +1)$
$\delta($ verify $, x) = ($ verify $, \$, +1)$
$\delta($ verify $, \square) = ($ accept $, \square, -1)$

(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!**

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

Simulate bi-infinite tape using 2 tracks

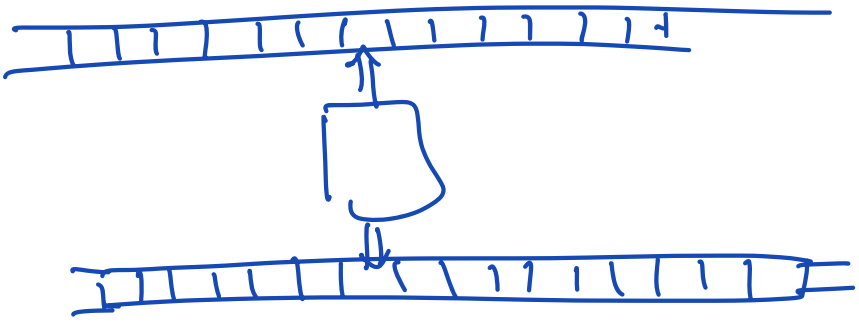


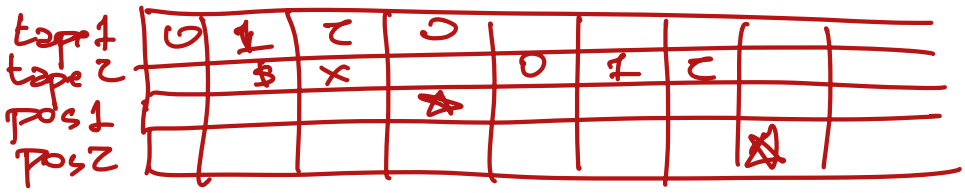
$\Gamma' = \Gamma \times \Gamma$

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

+
-

Simulate multiple tapes





2d tape