# Equivalent Models of Turing Machines

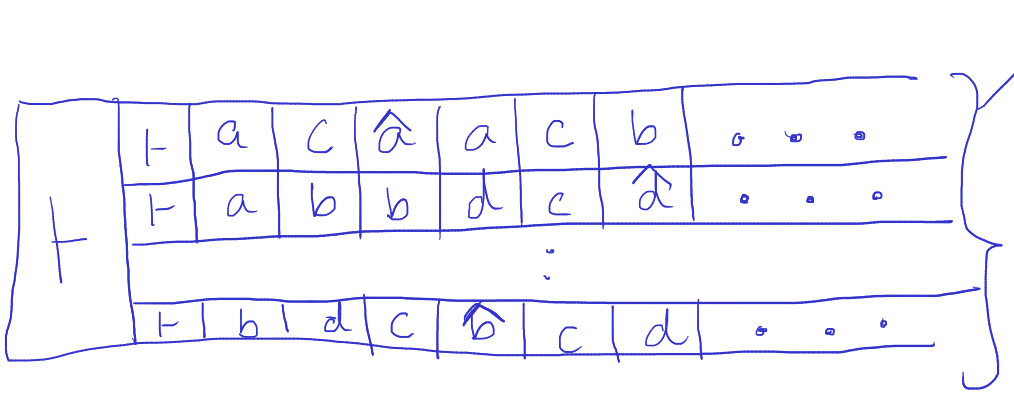
# Multiple Tape Turing Machine

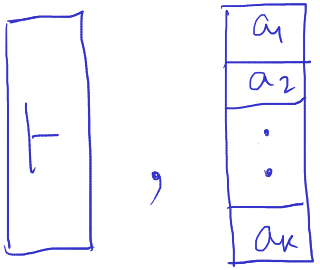
① Simulating a multi-tape TM is equivalent to simulating a single-tape TM.

k-tape TM M: (a) k read/write tapes
(b) k independent tape heads
(c) Same finite control

Transition function for M: $\delta_k : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R\}^k$

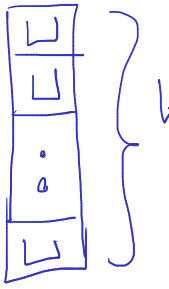
Single-tape TM N: Needs to simultaneously simulate all k tapes of M.

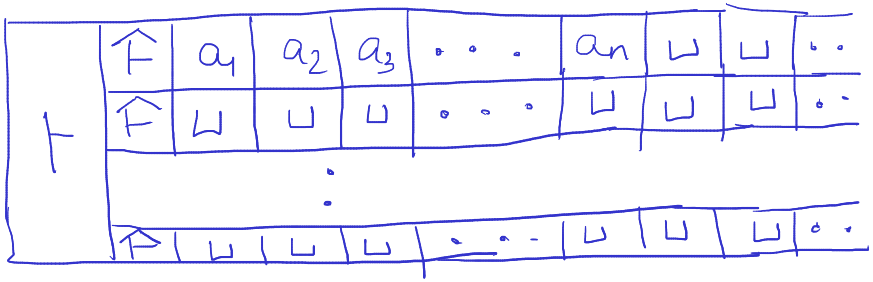



A tape symbol:  , where $a_1, a_2 \ldots a_k \in \Gamma \cup \hat{\Gamma}$

$\hat{\Gamma} = \{\hat{a} \mid a \in \Gamma\}$.

Left end-marker: 

Blank symbol: 


On input $x = a_1 a_2 \ldots a_n$, N starts: 

1 step of M is simulated by several steps of N:

① N starts from the left-most cell. Remembers state p of M in finite control.

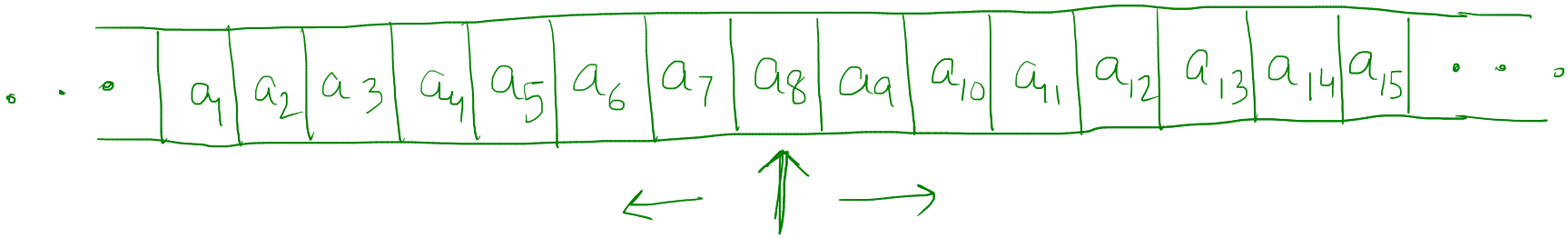
① Moves right until it sees alphabet in first track marked with ^ → remembers alphabet in finite control.

① Comes back to left-most cell and in the same way finds alphabets marked ^ in each of the k tracks → these are remembered in finite control.

① Transition from M that has to be applied is $\delta_k(p, a_1, a_2 \dots a_k)$ where p was the state in M, $a_1, \dots a_k$ are marked with ^.

① N goes back to do the rewrites and shift the ^ markings in each track. Also makes sure finite control remembers the new state of M.

# 2-way Infinite Tape

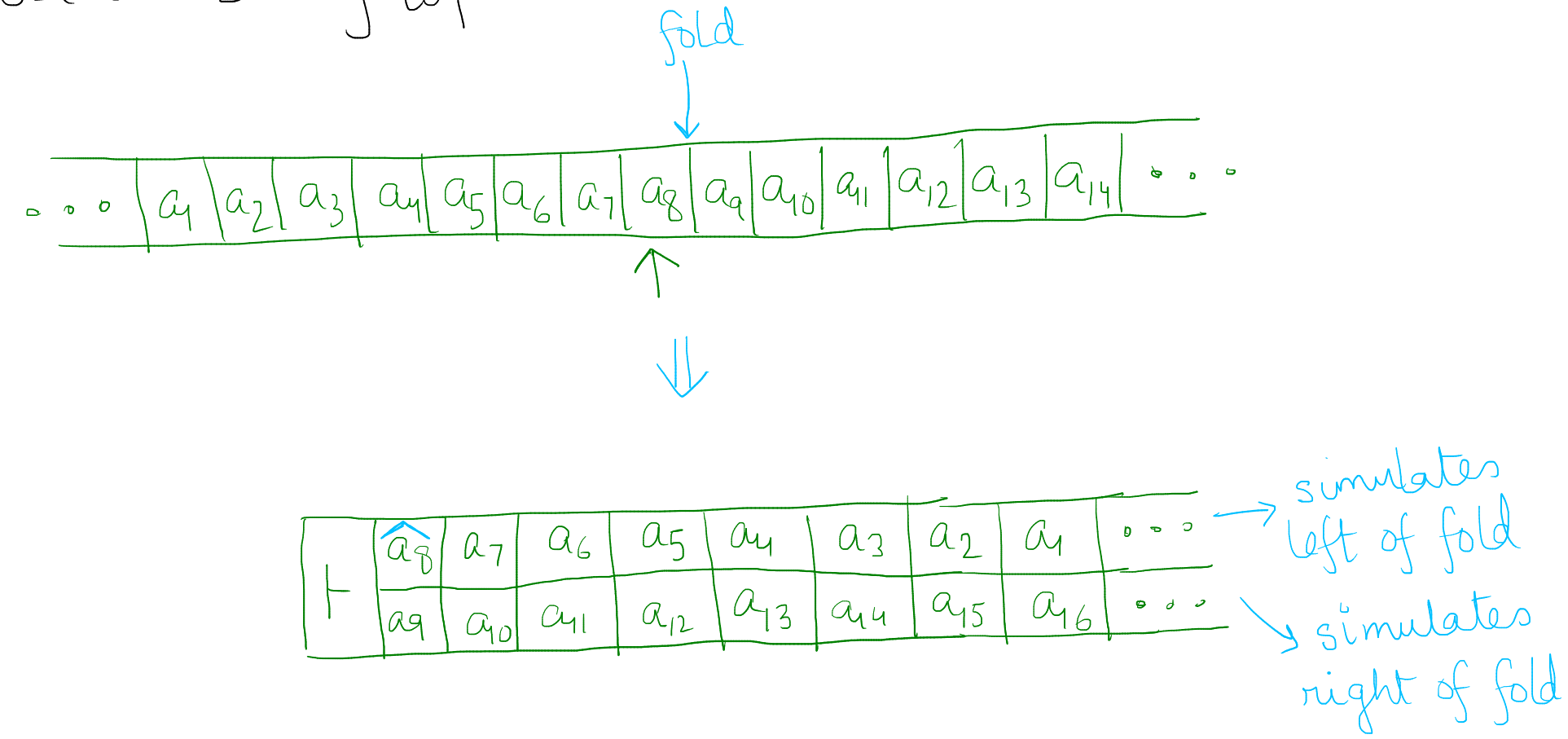



2-way tape : ① Tape head can move in either direction

② Tape is unbounded to the left and right.

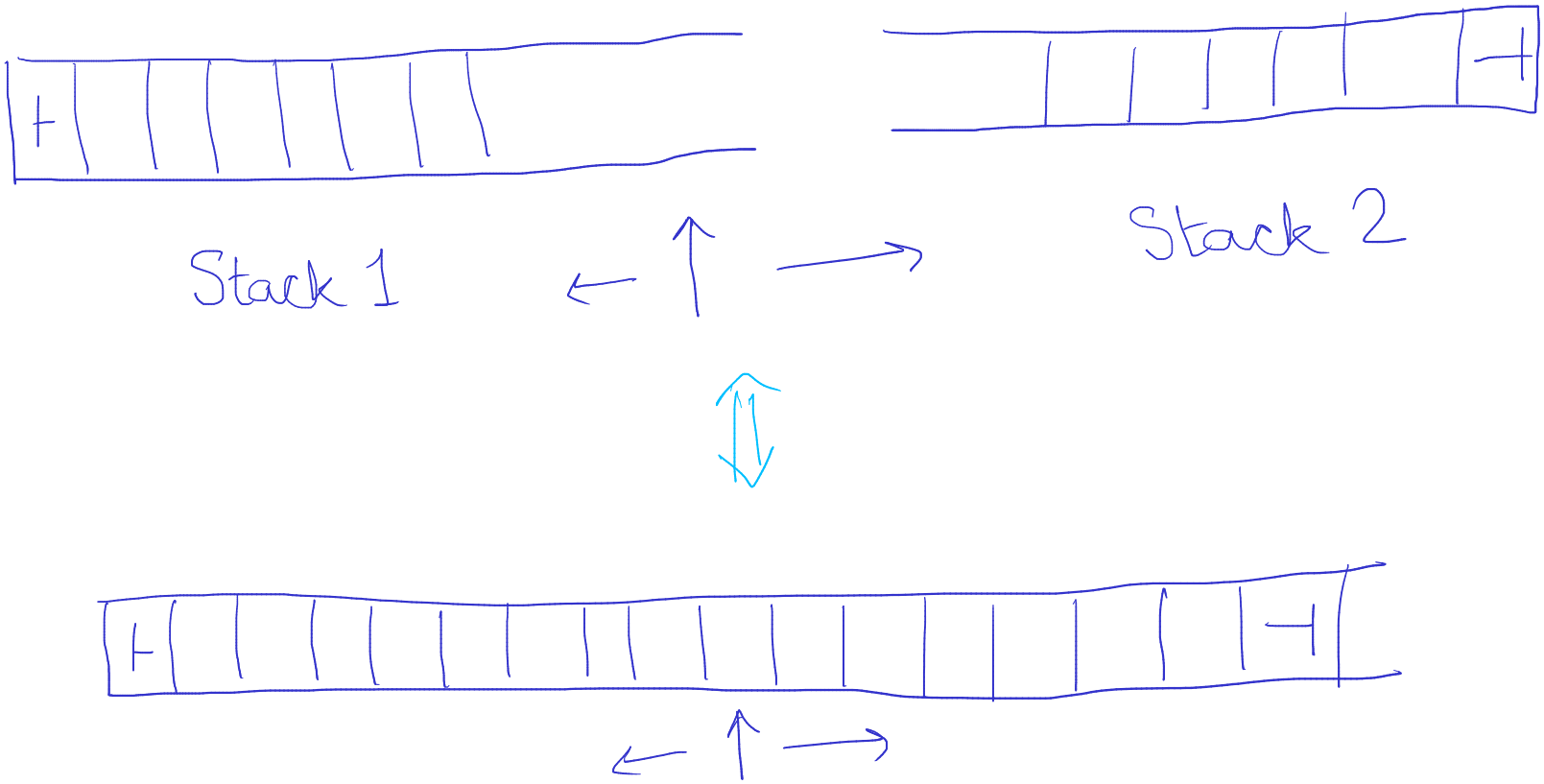
- Simulating a 2-way infinite tape M is equivalent to simulating a 1-tape (2-track) TM.

Fold the 2-way tape:



# 2 Stacks

⊙ A machine M with a 2-way read-only input head and 2-stacks is equivalent to a TM.



# Counter Automata

k-counter automaton :

- 2-way read-only input head
- Read-only tape (with input)
- k integer counters - each counter can store an arbitrary non-negative integer.
- In each step, the automaton can increment/decrement its counters and test if a counter is zero; can move input head one cell in either direction.

- Equivalent to a TM.

Given a k-counter automaton : Can be simulated by a k-tape TM.

Given a TM : We show a stack can be simulated by 2 counters

$\Rightarrow$ 2 stacks can be simulated by 4 counters

Assume stack alphabet is $\{0,1\}$ $\rightarrow$ o/w encode alphabets with binary strings of fixed length m

So pushing/popping an element = pushing/popping m binary digits.

Under assumption - Stack string is a binary number.

Using 2 counters, this number is maintained:

- Pushing 0 : Value in counter needs to be doubled.
  - ① Enter loop of subtracting 1 from counter and adding 2 to 2nd counter, till first counter is 0.
  - ② Value in 2nd counter is 2× (initial value)
  - ③ Transfer to 1st counter.

- Pushing 1 : Same as before except in the end, 2nd counter is incremented by 1 and then the value is transferred to 1st counter

Popping = dividing counter value by 2.

(b) In a loop, decrement 1st counter by 1
increment 2nd counter every 2nd step.

At an odd no. step, if 1st counter has 0 then
1 was popped.

At an even no. step, if 1st counter has 0 then
0 was popped.

⊖ A TM ≡ 2-stacks ≡ 4 counter automaton

≡ 2 counter automaton

$i, j, k, l$ in 4 counters $\Longleftrightarrow$ $2^i 3^j 5^k 7^l$ in 1st counter.

⊖ If 3rd counter is incremented : Value in 1st counter should be multiplied by 5.

⊖ If 2nd counter is decremented : Value divided by 3.

⊖ Testing is 1st counter is 0 : Is value divisible by 2?
If yes → 1st counter not 0.
no → 1st counter is 0.

# Enumeration Machines

R.E sets : accepted by TMs.

Enumeration machines: ⊙ Finite control

⊙ Two tapes : Read/Write work-tape
Write-only o/p tape

⊙ Work-tape head moves 2-way
O/p tape head moves only right
Only writes symbols $\in \Sigma$.

⊙ No i/p string, no accept/reject state.

Machine plugged in : ⊙ From tapes, it uses transition fn. and reads / writes.

⊙ At some pt. "enumeration state" entered

- Then the string currently on o/p is said to be "enumerated".
- O/p tape then gets automatically erased, o/p head moves back to leftmost cell.
  [Work tape remains intact]
- Machine continues. Runs forever.

$L(E)$ = strings in $\Sigma^*$ enumerated by E.

- $L(E) = \phi \Rightarrow$ enumeration state was never visited
- Same string could be enumerated twice

Lemma: A set is $L(E)$ for some enumeration machine $E \iff L(M) = L(E)$ for a TM $M$.

$\Rightarrow$ Given $E$, construct $M$:

On input $x$: $M$ simulates $E$;

Everytime "enumeration state" is entered $M$ checks if $x$ has been enumerated.

Then continues.

$x \in L(M) \iff x \in L(E)$

$\Leftarrow$ Given M, construct E:

① Would want E to write down all possible strings in $\Sigma^*$ on its work tape and see which ones are accepted by M by simulating M on each string.

② What if M loops on a string? E will be stuck and won't be able to move to the other strings
[Note: Not a problem if M is total]

③ Time-sharing strategy:
Step 1: Simulate 1 step of string 1, Step 2: 1 step of strings 1,2
Step 3: 1 step of strings 1,2,3 ...

In some strings M may enter a loop.
In accepted strings, M accepts after finite steps.

$$L(E) = L(M).$$