Turing Machine

Lin Chen

Email: Lin.Chen@ttu.edu

Grader: zulfi.khan@ttu.edu

TEXAS TECH
UNIVERSITY

Turing Machine

☐ Handicapped machines

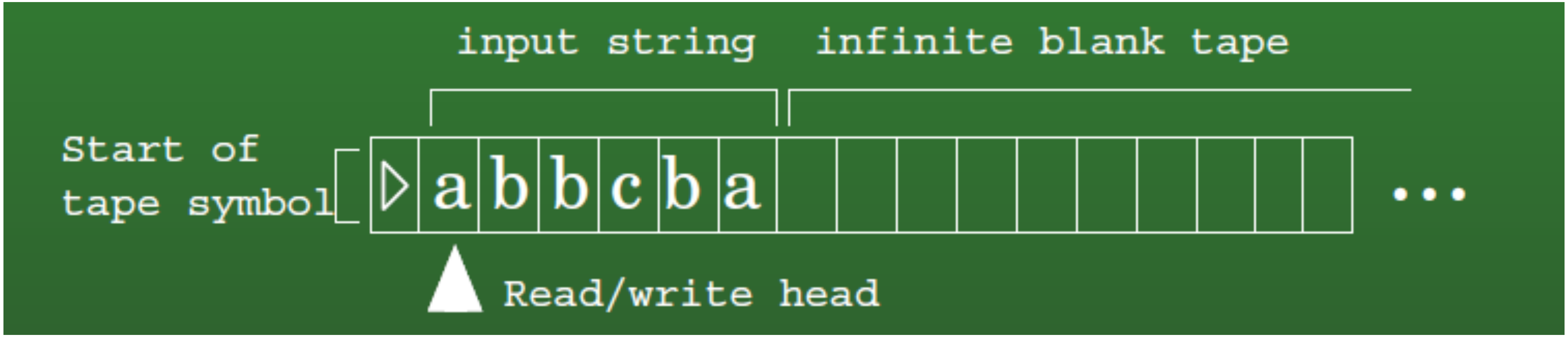
- **DFA limitations**
  - Tape head moves only one direction
  - Tape is read-only
  - Tape length is a constant

- **PDA limitations**
  - Tape head moves only one direction
  - Tape is read-only, but stack is writable
  - Stack has only LIFO(last-in, first-out) access
  - Tape length is constant, but stack is not bounded.

Turing Machine

- **What about**
  - Writable, 2-way tape?
  - Random-access ‘stack?

Turing Machine



- Head can both read and write, and move in both directions
- Tape has unbounded length.
- ⊔ is blank symbol. In practice, all but a finite number of tape squares are blank.

Turing Machine

A ***Turing machine*** is a 7-tuple, $(K, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, where K, Σ, Γ are all finite sets and

1. K is the set of states,
2. Σ is the input alphabet not containing the ***blank symbol*** $\sqcup$,
3. Γ is the tape alphabet, where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$,
4. δ: $K \times \Gamma \longrightarrow K \times \Gamma \times \{L, R\}$ is the transition function,
5. $q_0 \in K$ is the start state,
6. $q_{accept} \in K$ is the accept state, and
7. $q_{reject} \in K$ is the reject state, where $q_{reject} \neq q_{accept}$.

Turing Machine

Example:

Consider the Turing Machine $M = (K, \sum, \delta, s, H)$, where

$K = \{q_0, h\}$

$\sum = \{a, \sqcup, \triangleright\}$

$s = q_0$

$H = \{h\}$

and δ is given by the right table.

q	σ	$\delta(q,\sigma)$
q_0	a	$(q_0, \leftarrow)$
q_0	$\sqcup$	$(h, \sqcup)$
q_0	$\triangleright$	$(q_0, \rightarrow)$

$\cdots$	$\triangleright$	a	a	$\sqcup$	$\cdots$

$\uparrow$

Never stop!

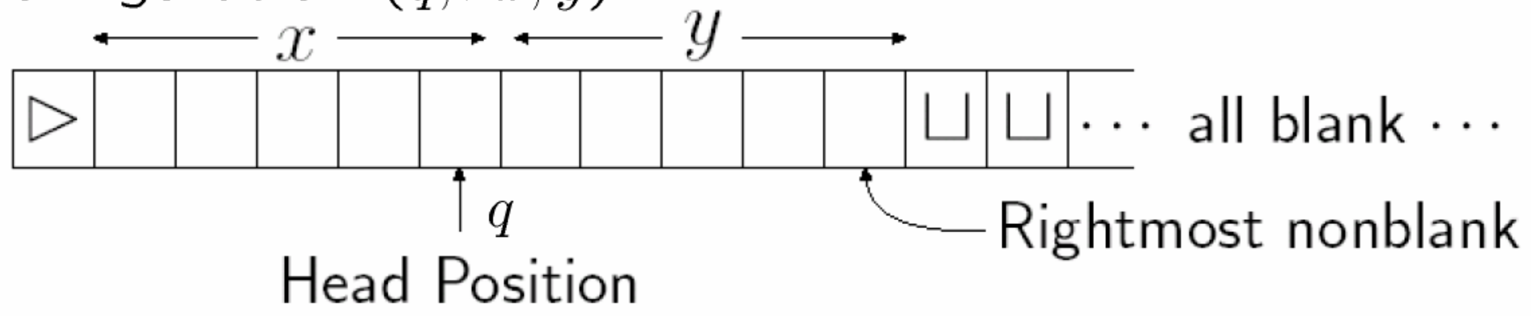
(The machine goes into a loop if no $\sqcup$ can

Turing Machine

Turing Machines Configuration

Definition: A configuration of a TM $M = (K, \sum, \delta, s, H)$ is a member of $K \times \triangleright \sum^* \times (\sum^*(\sum - \{\sqcup\}) \cup \{e\})$.

Configuration $(q, \triangleright x, y)$:



Remark:

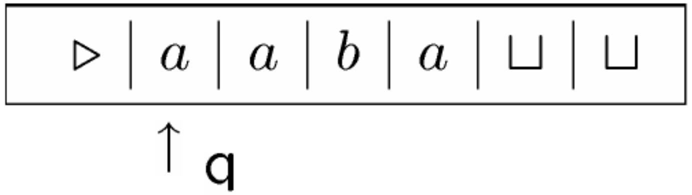
- A configuration whose state component is in H will be called halted configuration.

Turing Machine

- A simplified notation of configuration:

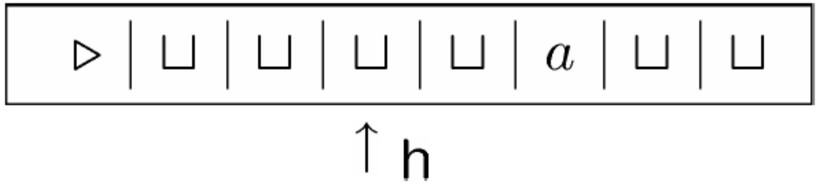
$$(q, wa, u) \Rightarrow (q, w\underline{a}u)$$

Example of configuration:



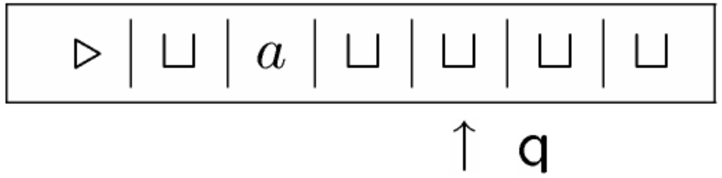
$(q, \triangleright a, aba)$
or $(q, \triangleright \underline{a}aba)$

$\triangleright\, qaaba$



$(h, \triangleright \sqcup \sqcup\sqcup, \sqcup a)$
or $(h, \triangleright \sqcup \sqcup\underline{\sqcup} \sqcup a)$

$\triangleright\sqcup\sqcup\, h \sqcup\sqcup\, a$



$(q, \triangleright \sqcup a \sqcup \sqcup, e)$
or $(q, \triangleright \sqcup a \sqcup \underline{\sqcup}e)$

$\triangleright\sqcup\, a \sqcup q \sqcup$

Turing Machine

Remark:

- For any Turing Machine M, let $\vdash_M^*$ be the Reflexive, transitive closure of $\vdash_M$.
Configuration C_1 yields configuration C_2 if $C_1 \vdash_M^* C_2$.

- A **computation** by M is a sequence of configuration C_0, C_1, $\cdots$, C_n, for some $n \geq 0$ such that

$$C_0 \vdash_M C_1 \vdash_M \cdots \vdash_M C_n$$

we say that the computation is of length n or that it has n steps, denoted by $C_0 \vdash_M^n C_n$.

Turing Machine

Run a Turing machine on an input, the Turing machine may:

- Accept (enter q_{accept})
- Reject (enter q_{reject})
- Loop (running forever)

Turing Machine

Run a Turing machine on an input, the Turing machine may:

- Accept (enter q_{accept})
- Reject (enter q_{reject})
- Loop (running forever)

• M **accepts** $w \in (\Sigma - \{\sqcup, \triangleright\})^*$ if $(s, \triangleright\underline{\sqcup}w)$ yields an accepting configuration; M **rejects** w if $(s, \triangleright\underline{\sqcup}w)$ yields an rejecting configuration.

• Let $\Sigma_0 \subseteq (\Sigma - \{\sqcup, \triangleright\}$ be a alphabet — input alphabet of M.

M **decides** $L \subseteq \Sigma_0^*$ if $\forall w \in \Sigma_0^*$ the following is true:

- □ $w \in L$ iff M accepts w;
- □ $w \notin L$ iff M rejects w.

Turing Machine

M **decides** $L \subseteq \sum_0^*$ if $\forall w \in \sum_0^*$ the following is true:

- ☐ $w \in L$ iff M accepts w;
- ☐ $w \notin L$ iff M rejects w.

A language is Turing-decidable or simply decidable if some Turing machine decides it.

Turing Machine

M **decides** $L \subseteq \sum_0^*$ if $\forall w \in \sum_0^*$ the following is true:

- ☐ $w \in L$ iff M accepts w;
- ☐ $w \notin L$ iff M rejects w.

A language is Turing-decidable or simply decidable if some Turing machine decides it.

The collection of the strings that a Turing machine accepts is the language recognized by the machine.

A language is Turing-recognizable (or semi-decidable) if some Turing machine recognizes it.

Can we use Turing machine as checker for the language it recognizes/decides?

Turing Machine-example

Turing machine (TM) M_1 that decides $B = \{w\#w \mid w \in \{0,1\}^*\}$

M_1 = "On input string w:

1. Zig-zag across the tape to corresponding positions on either side of the # symbol to check whether these positions contain the same symbol. If they do not, or if no # is found, *reject*. Cross off symbols as they are checked to keep track of which symbols correspond.
2. When all symbols to the left of the # have been crossed off, check for any remaining symbols to the right of the #. If any symbols remain, *reject*; otherwise, *accept*."

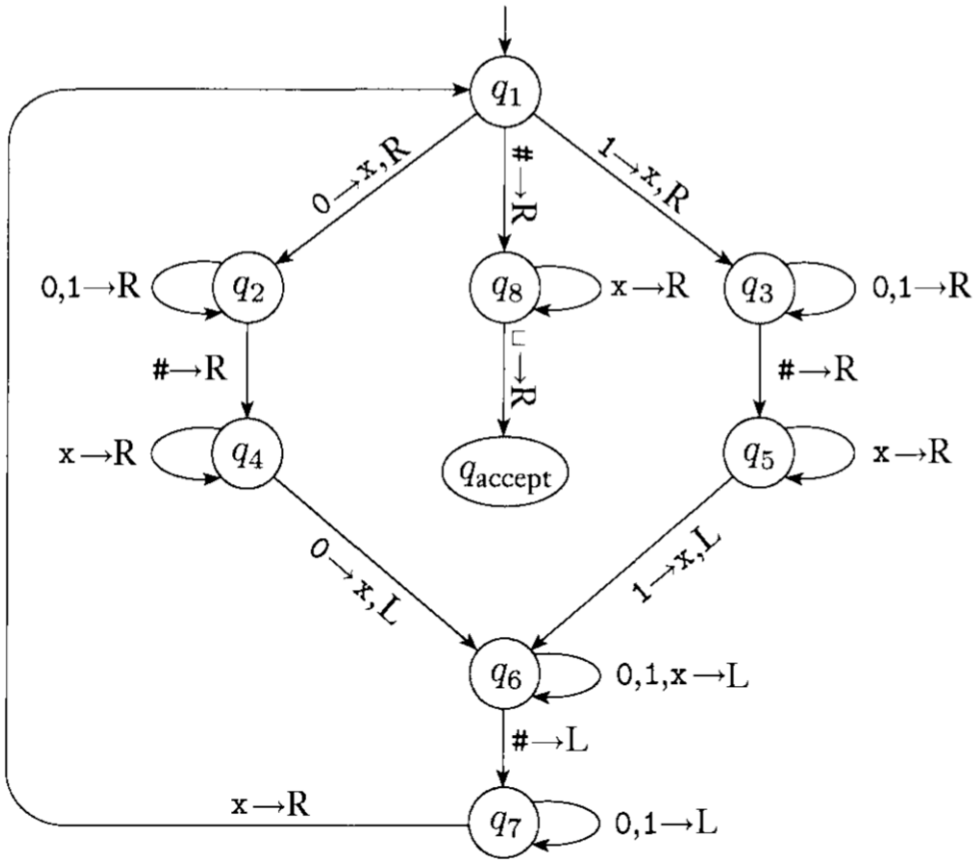
Turing Machine-example

Turing machine (TM) M_1 that decides $B = \{w\#w \mid w \in \{0,1\}^*\}$

```
0 1 1 0 0 0 # 0 1 1 0 0 0 ⊔ ...
x 1 1 0 0 0 # 0 1 1 0 0 0 ⊔ ...
x 1 1 0 0 0 # x 1 1 0 0 0 ⊔ ...
x 1 1 0 0 0 # x 1 1 0 0 0 ⊔ ...
x x 1 0 0 0 # x 1 1 0 0 0 ⊔ ...
x x x x x x # x x x x x x ⊔ ...
                          accept
```

Turing Machine-example

Turing machine (TM) M_1 that decides $B = \{w\#w \mid w \in \{0,1\}^*\}$



Incomplete, move to a reject state once lacking out-edge

Turing Machine-example

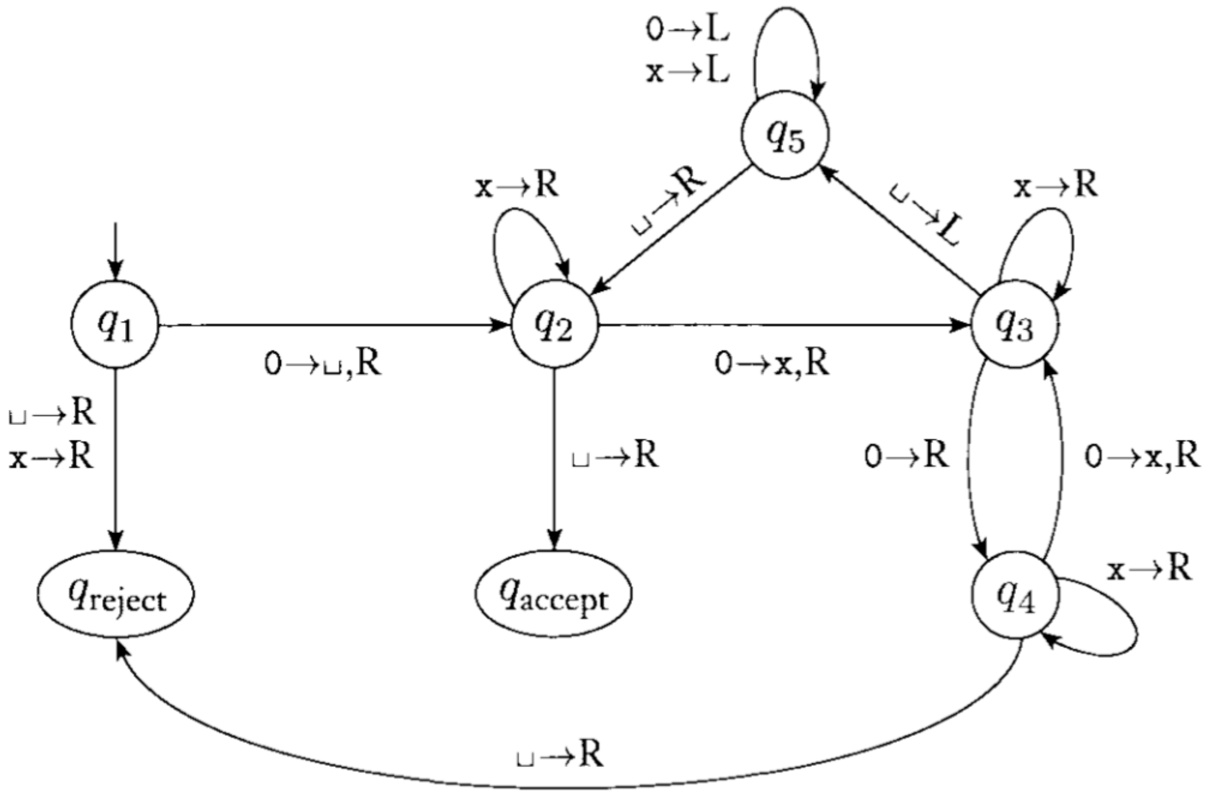
Turing machine (TM) M_2 that decides $A = \{0^{2^n} \mid n \geq 0\}$

M_2 = "On input string w:

1. Sweep left to right across the tape, crossing off every other 0.
2. If in stage 1 the tape contained a single 0, *accept*.
3. If in stage 1 the tape contained more than a single 0 and the number of 0s was odd, *reject*.
4. Return the head to the left-hand end of the tape.
5. Go to stage 1."

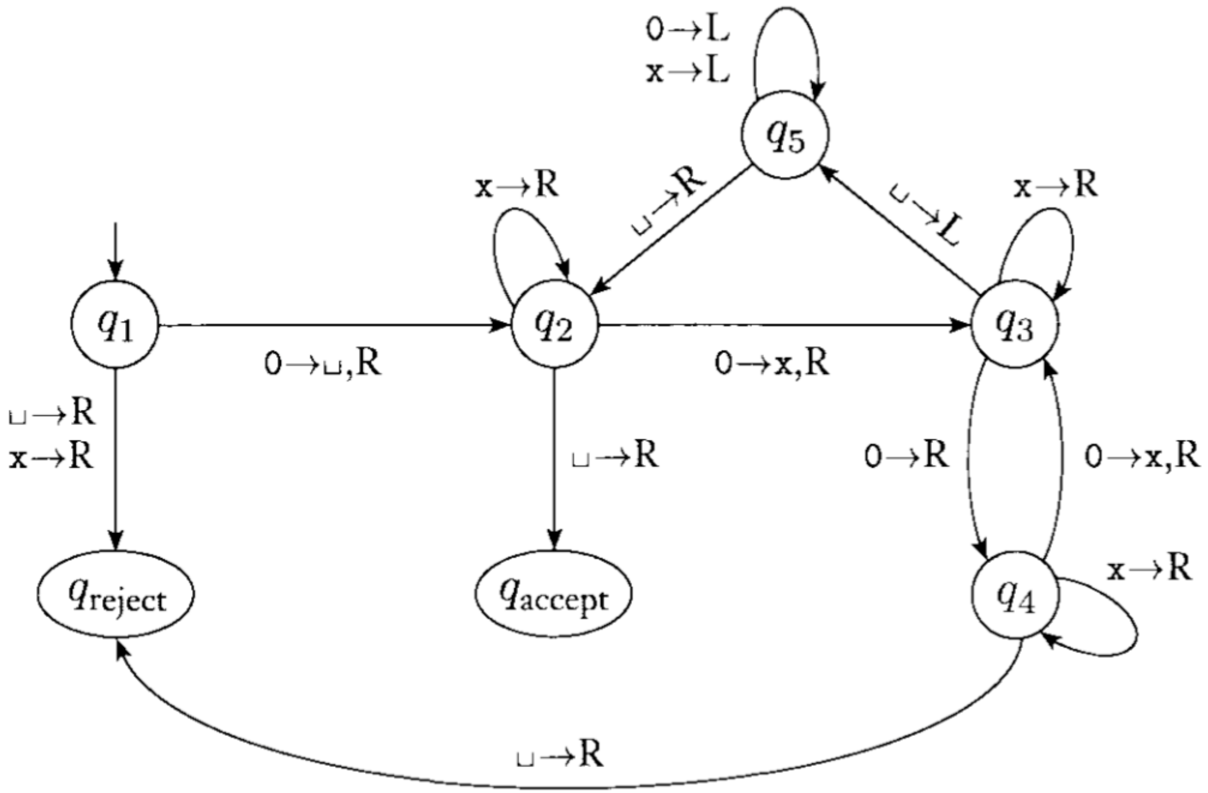
Turing Machine-example

Turing machine (TM) M_2 that decides $A = \{0^{2^n} \mid n \geq 0\}$



Turing Machine-example

Turing machine (TM) M_2 that decides $A = \{0^{2^n} \mid n \geq 0\}$



$q_1 0000$	$\sqcup q_5 x0x\sqcup$	$\sqcup x q_5 xx\sqcup$
$\sqcup q_2 000$	$q_5 \sqcup x0x\sqcup$	$\sqcup q_5 xxx\sqcup$
$\sqcup x q_3 00$	$\sqcup q_2 x0x\sqcup$	$q_5 \sqcup xxx\sqcup$
$\sqcup x0 q_4 0$	$\sqcup x q_2 0x\sqcup$	$\sqcup q_2 xxx\sqcup$
$\sqcup x0x q_3 \sqcup$	$\sqcup xx q_3 x\sqcup$	$\sqcup x q_2 xx\sqcup$
$\sqcup x0 q_5 x\sqcup$	$\sqcup xxx q_3 \sqcup$	$\sqcup xx q_2 x\sqcup$
$\sqcup x q_5 0x\sqcup$	$\sqcup xx q_5 x\sqcup$	$\sqcup xxx q_2 \sqcup$
		$\sqcup xxx\sqcup q_{accept}$

Variants of Turing Machine

Can we strengthen a Turing machine by equipping it with more "resources"?

Variants of Turing Machine

Can we strengthen a Turing machine by equipping it with more "resources"?

Multi-tape Turing machine: Turing machine has only one read/write tape, what if we allow multiple tapes?

$$\delta: Q \times \Gamma^k \longrightarrow Q \times \Gamma^k \times \{\mathrm{L}, \mathrm{R}, \mathrm{S}\}^k$$

$$\delta(q_i, a_1, \ldots, a_k) = (q_j, b_1, \ldots, b_k, \mathrm{L}, \mathrm{R}, \ldots, \mathrm{L})$$

Variants of Turing Machine

Can we strengthen a Turing machine by equipping it with more "resources"?

Multi-tape Turing machine: Turing machine has only one read/write tape, what if we allow multiple tapes?

$$\delta: Q \times \Gamma^k \longrightarrow Q \times \Gamma^k \times \{\mathrm{L}, \mathrm{R}, \mathrm{S}\}^k$$

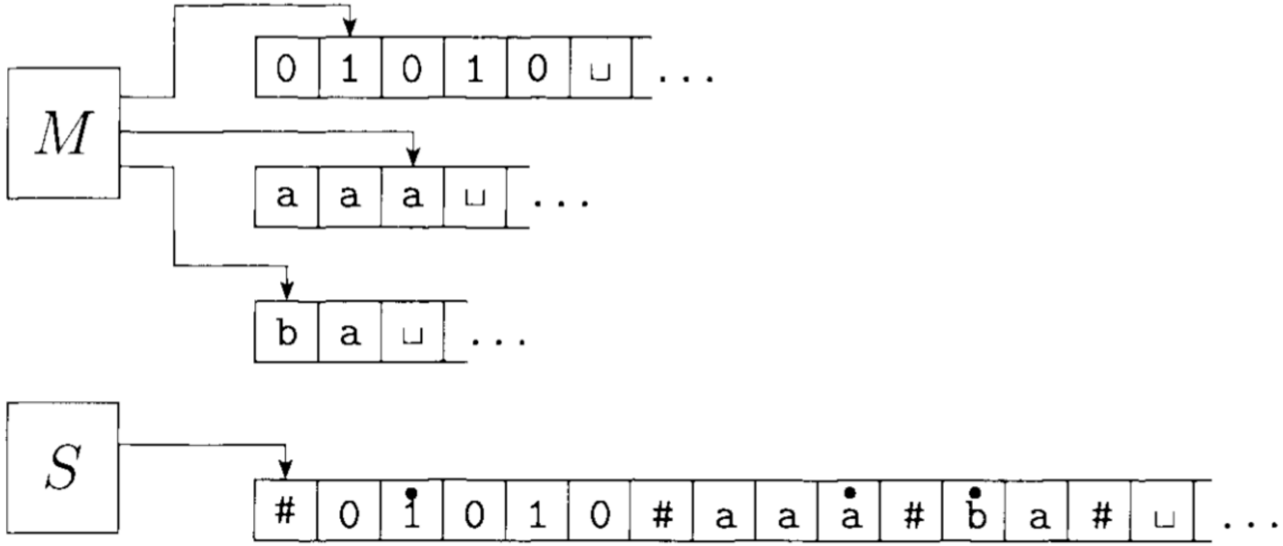
$$\delta(q_i, a_1, \ldots, a_k) = (q_j, b_1, \ldots, b_k, \mathrm{L}, \mathrm{R}, \ldots, \mathrm{L})$$

Can multi-tape Turing machine decides/recognizes more languages than Turing machine?

Variants of Turing Machine

Every multitape Turing machine has an equivalent single-tape Turing machine.

We can simulate a multi-tape Turing machine using a single-tape Turing machine.



Variants of Turing Machine

Every multitape Turing machine has an equivalent single-tape Turing machine.

A language is Turing-recognizable if and only if some multitape Turing machine recognizes it.

Variants of Turing Machine

Can we strengthen a Turing machine by equipping it with non-determinism?

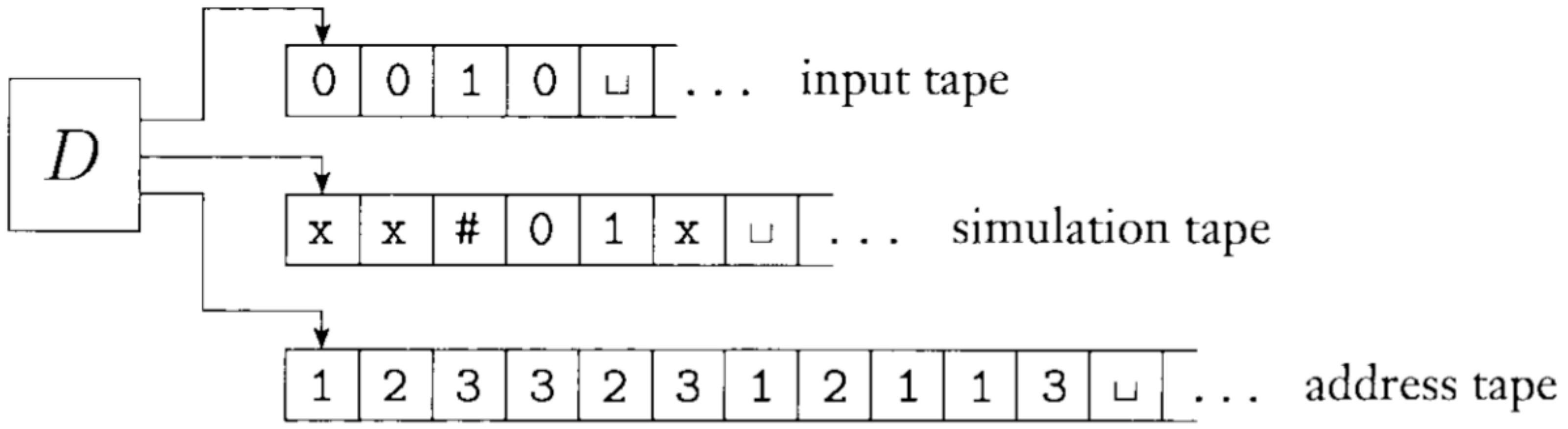
Nondeterministic Turing machine: from transition function to transition relation.

$$\delta \colon Q \times \Gamma \longrightarrow \mathcal{P}(Q \times \Gamma \times \{\mathrm{L}, \mathrm{R}\})$$

Variants of Turing Machine

Every nondeterministic Turing machine has an equivalent deterministic Turing machine.

We can simulate a non-deterministic Turing machine using a deterministic Turing machine.



Variants of Turing Machine

Every nondeterministic Turing machine has an equivalent deterministic Turing machine.

We can simulate a non-deterministic Turing machine using a deterministic Turing machine.

1. Initially tape 1 contains the input w, and tapes 2 and 3 are empty.

2. Copy tape 1 to tape 2.

3. Use tape 2 to simulate N with input w on one branch of its nondeterministic computation. Before each step of N consult the next symbol on tape 3 to determine which choice to make among those allowed by N's transition function. If no more symbols remain on tape 3 or if this nondeterministic choice is invalid, abort this branch by going to stage 4. Also go to stage 4 if a rejecting configuration is encountered. If an accepting configuration is encountered, *accept* the input.

4. Replace the string on tape 3 with the lexicographically next string. Simulate the next branch of N's computation by going to stage 2.

Variants of Turing Machine

Every nondeterministic Turing machine has an equivalent deterministic Turing machine.

A language is Turing-recognizable if and only if some nondeterministic Turing machine recognizes it.