12/11/2020

CS303 Tutorial 9

Name : M.Maheeth Reddy   Roll No. 1801CS31

Ans 1:

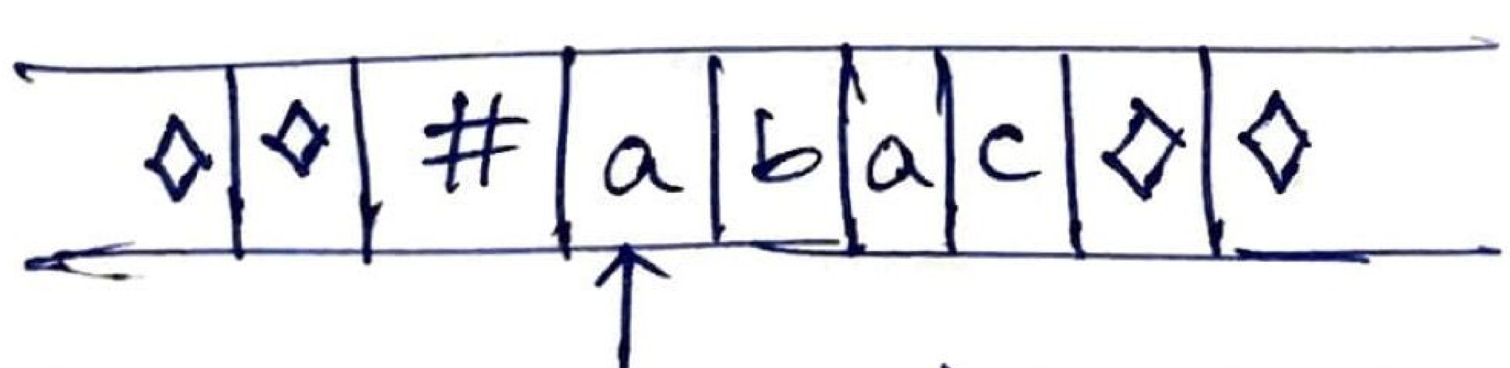
To prove : Semi-Infinite Machines have same power as Standard Turing Machines

Proof:

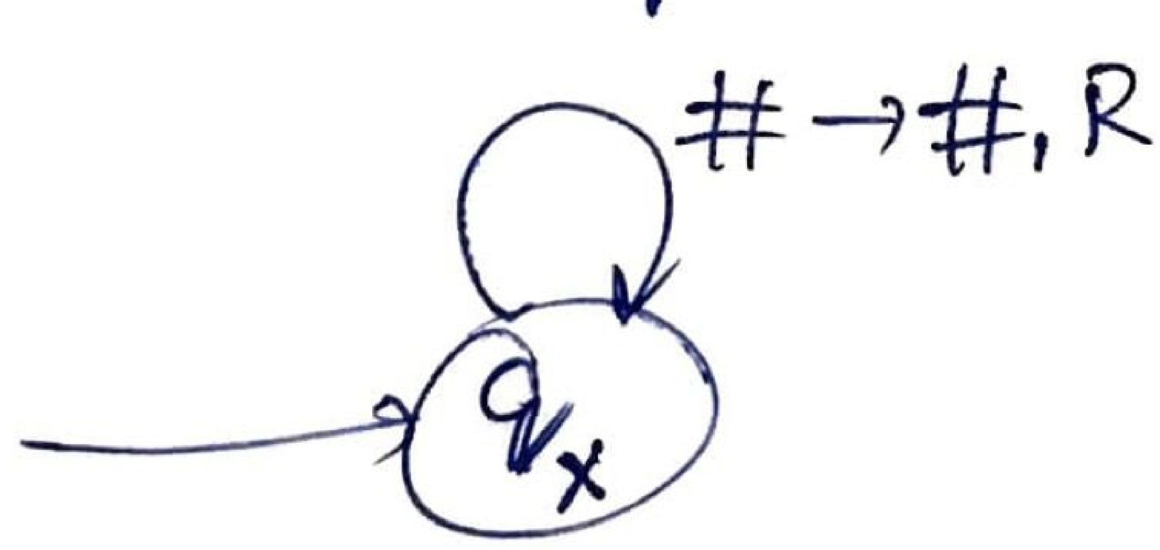
I) Standard Turing Machines simulate Semi-Infinite Machines.

Consider a Standard Turing Machine and do the following modifications to it.

① Insert a special symbol # at the left of input string.



② Add a self loop to each state of the Standard Turing Machine

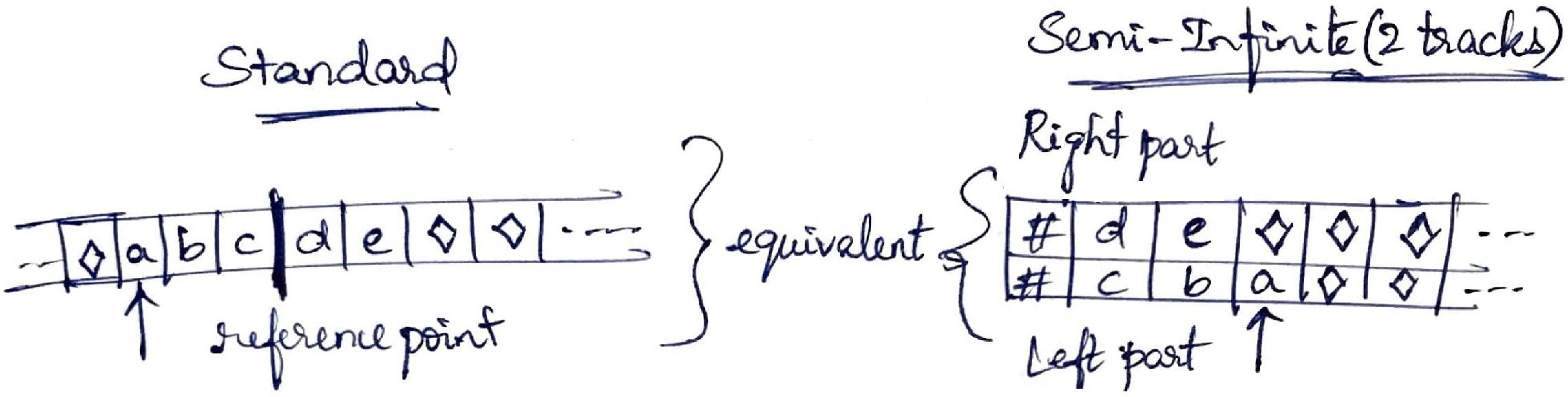


Exclude states with no outgoing transition.

This will ensure the Standard Turing Machine will not go beyond the #, so left infinity can never be reached. In this way Semi-Infinite tape machine is simulated.

II) Semi-Infinite Machine simulates Standard Turing Machines

Since Semi-Infinite Machine has one infinity only, whereas Standard Turing Machine has two, we use a Semi-Infinite machine with 2 tracks for simulation



The head of the Semi-infinite machine will read both tracks. each state q_1 in standard turing machine has 2 counterparts q_1^L, q_1^R in semi-infinite machine.

With the following transitions included in the semi-infinite machine, we ensure the functionality is same in both machines.

1

Right Part $\qquad q_1^R \xrightarrow{(a,x)\to(g,x),R} q_2^R$

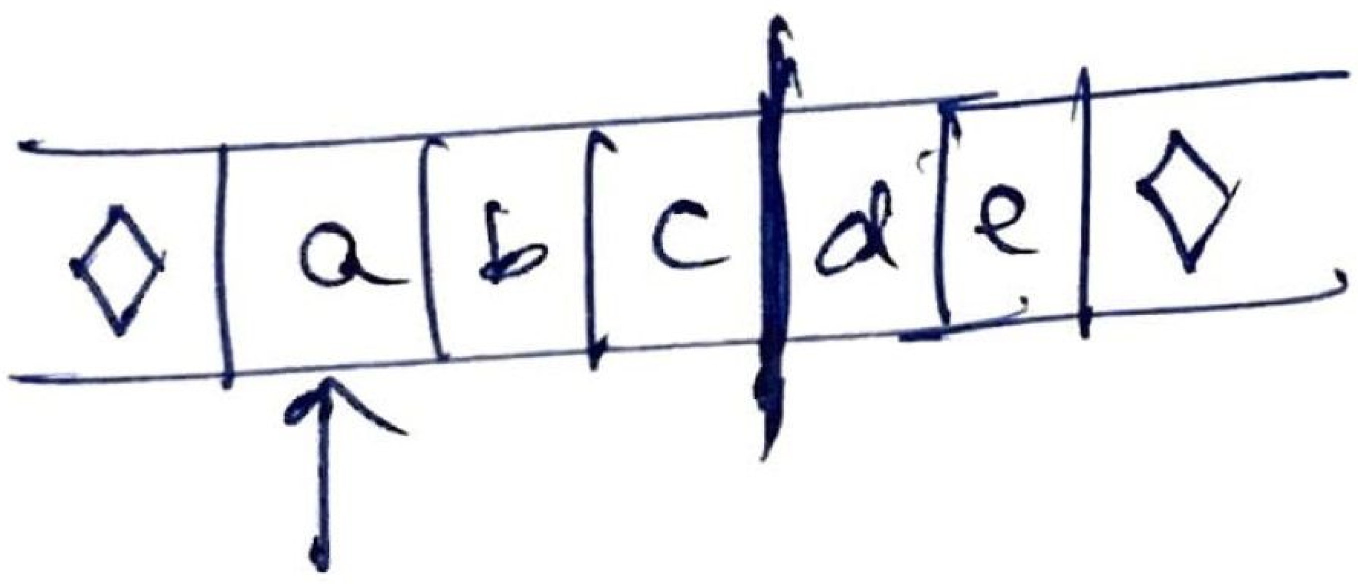
Left Part $\qquad q_1^L \xrightarrow{(x,a)\to(x,g),L} q_2^L$

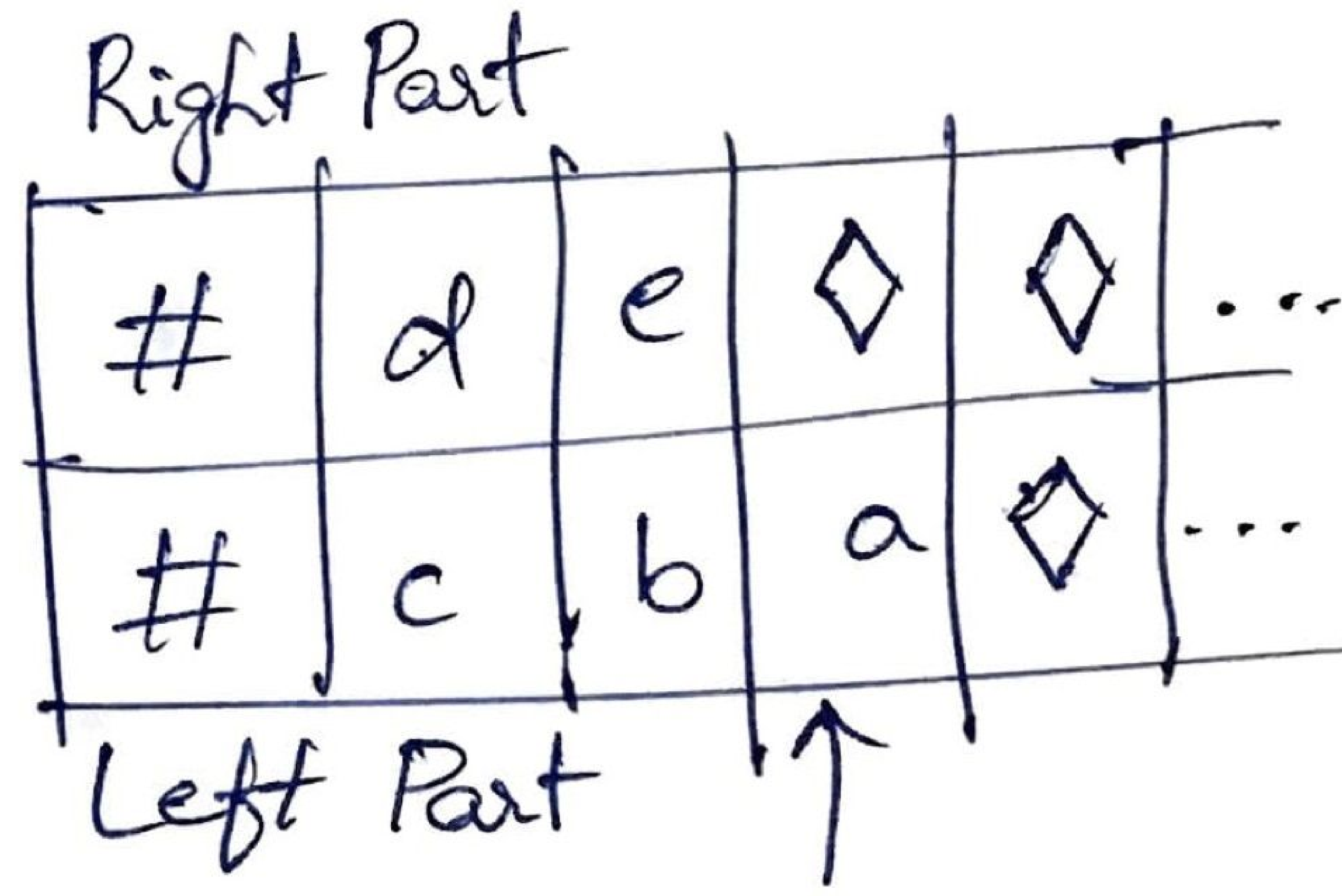
For any tape symbol x.

Simulation:

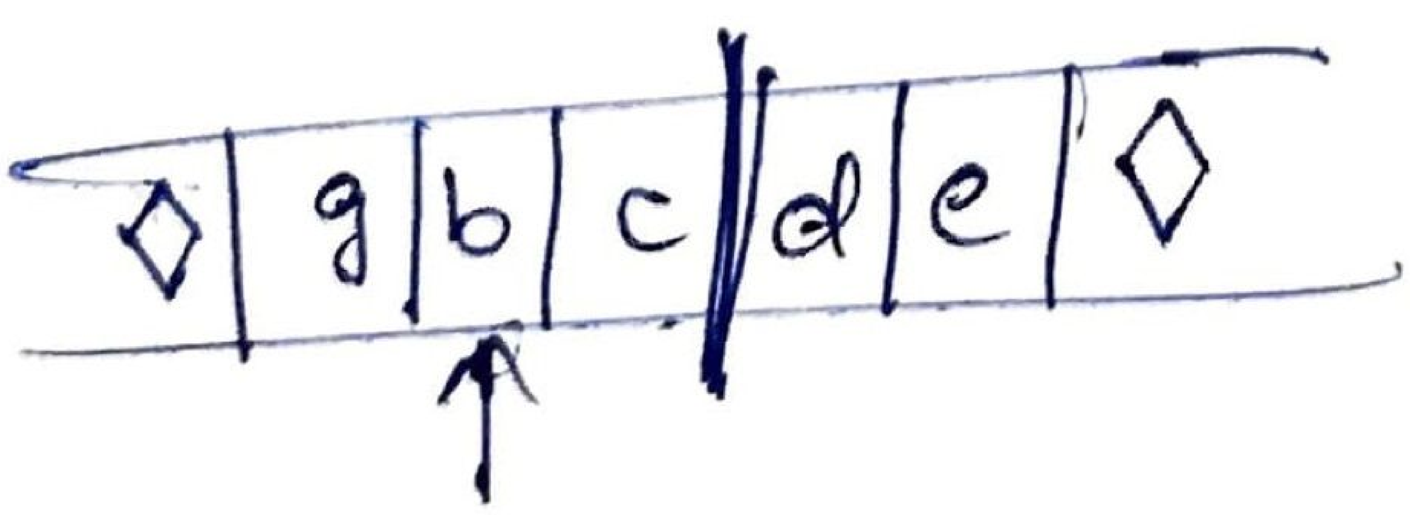
Standard | Semi-Infinite

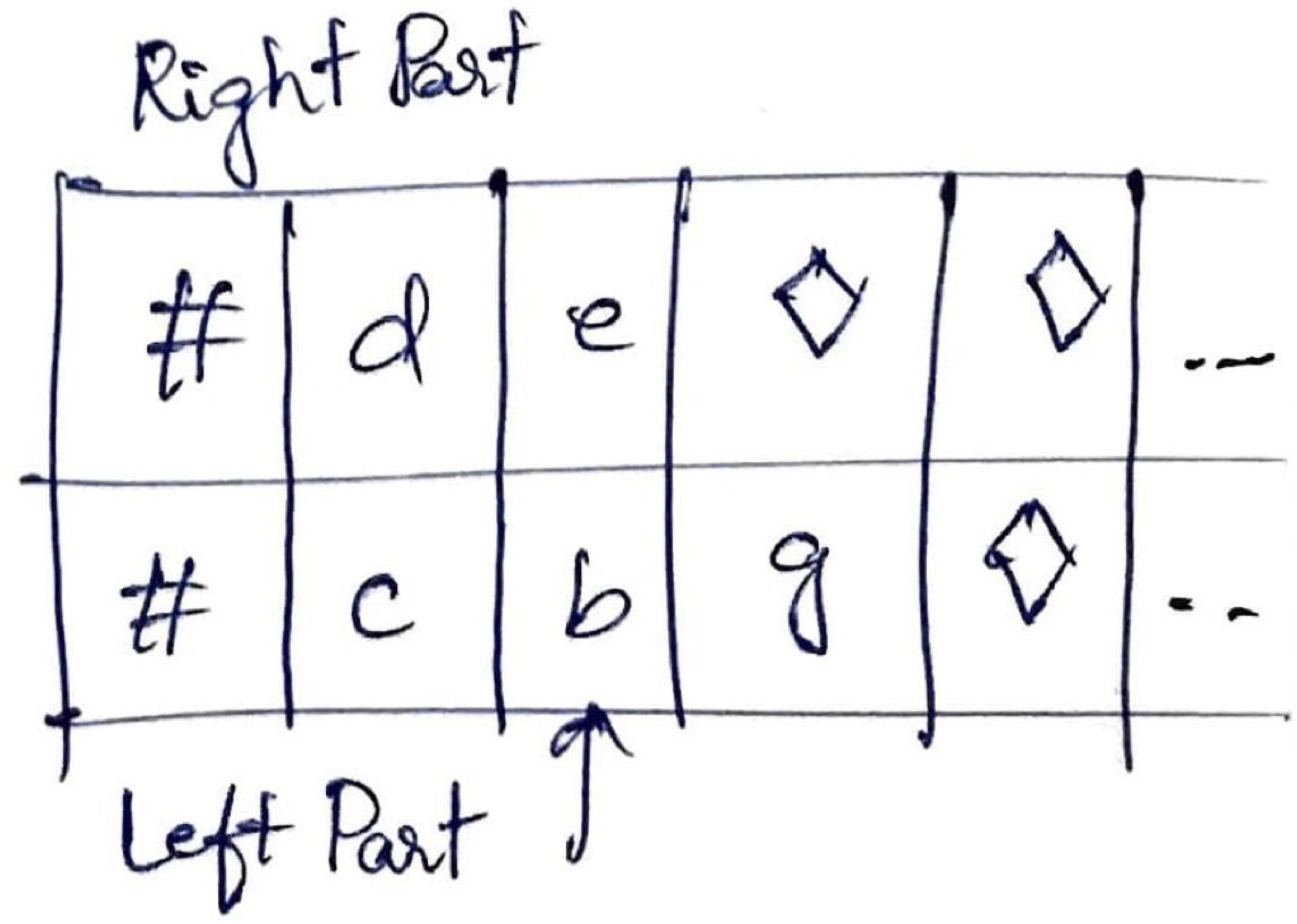
Time 1





Time 2





(2) At the border,

Right Part

$$q_1^R \xrightarrow[R]{(\#,\#)\to(\#,\#)} q_1^L$$

Left Part

$$q_1^L \xrightarrow[R]{(\#,\#)\to(\#,\#)} q_1^R$$

Simulation: **Semi-Infinite**

Time 1.

Right	#	d	e	◇	◇	...
Left	#	c	b	g	◇	...

↑ q_1^L

Time 2

Right	#	d	e	◇	◇	...
Left	#	c	b	g	◇	...

↑ q_1^R (now 'd' will be read so we moved to the right)

In this way, we can prove that Standard Turing Machine & Semi-Infinite Turing Machine have same power. Hence Proved!!!

Ans 2: $L = \{a^p \mid p \text{ is prime number}\}$

Algorithm:

① if ($p=0$ or $p=1$) Reject the string

② if ($p=2$) Accept the string.

③ if ($p>2$)

if ($\exists i \mid 3 \leq i \leq p-1$ and $i \mid p$ i.e., i divides p)
Reject the string

else
Accept the string

Description of Turing Machine, M

It has 2 tapes, one is for input string a^p.

If $a^p \in \{\epsilon, a\}$ it is rejected.

If $a^p \in \{a^2\}$ it is accepted

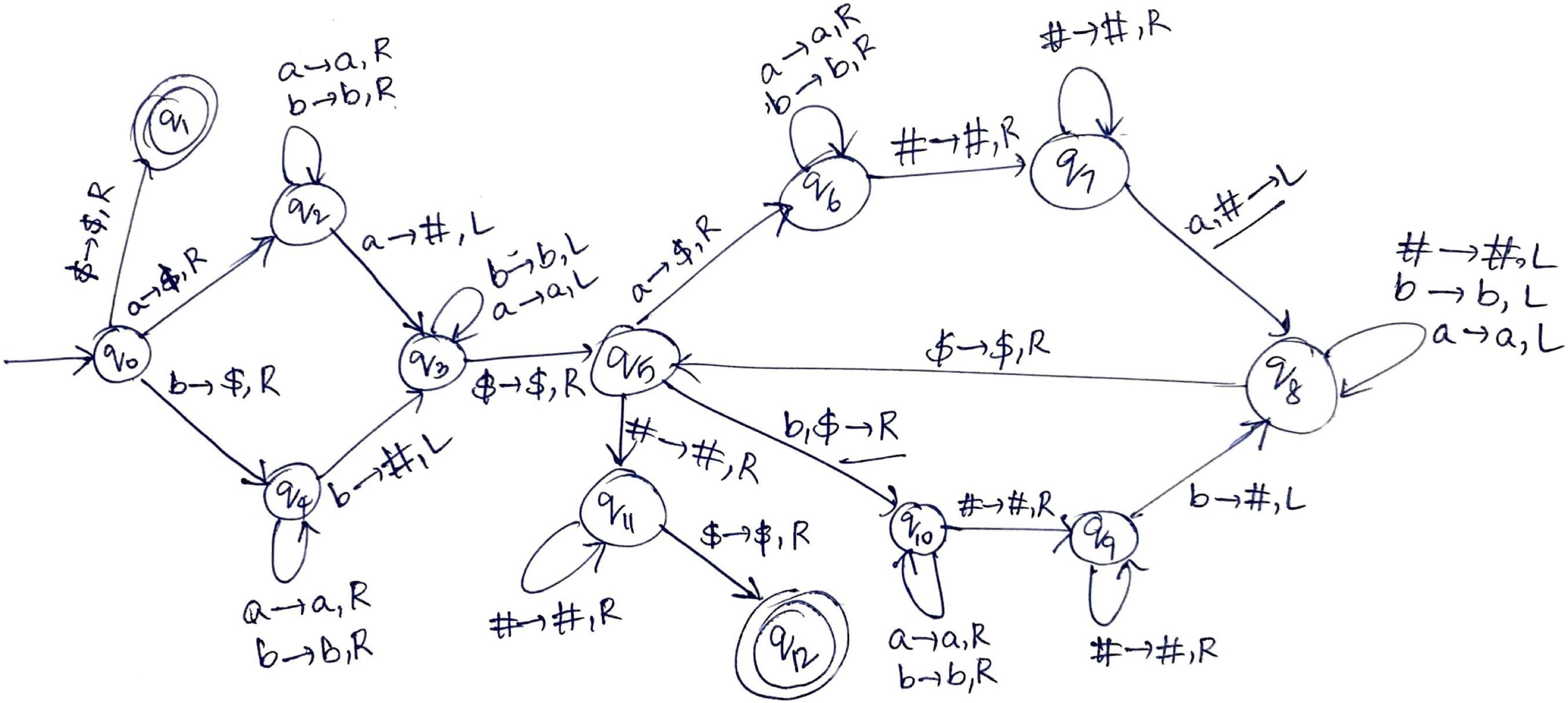
Otherwise, the second tape is placed with 2 a's. The head of both tapes point to their corresponding first a's.

It marks off copies of tape-2 string on tape-1 string until tape-2 head is at the end of its string. then moving tape-2 head back to the left end and repeating.

If it reaches the end of tape-1 string at the same time it reaches the end of the tape-2 string, it rejects (in this case p must be even).

Otherwise it adds a third 'a' to tape-2 string and again marks off copies of tape-2 string on tape-1 string. If it finds they are equal length, it accepts — otherwise it marks off copies and rejects if it finds the right ends of the two strings at the same time. Otherwise it continues increasing no. of 'a's in tape 2 (by 1) each time and marking off. It accepts if it finds the tape-1 and tape-2 strings to be the same length, and rejects if it first finds the tape-1 length to be a proper multiple of the tape-2 length.

3 Ans:



Algorithm: ① Find the middle of string. If there's a lone character, reject the string.

② Check characters at similar positions in both halves of the input. If they match, mark them (a as $, b as #) otherwise reject.

③ Repeat step ② until end of the strings. accept if everything has matched.