Automata

Unit - 4

Turning Maching :-

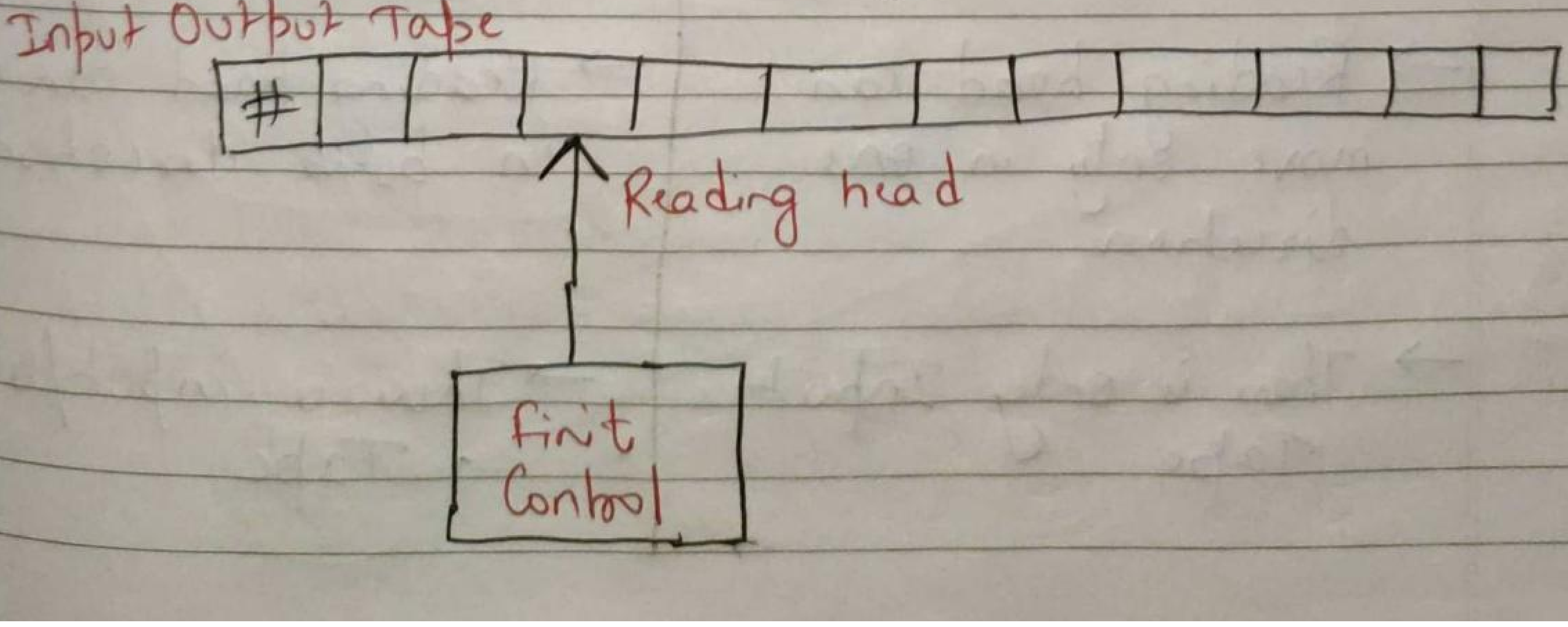
→ unrestricted
→ Compotational device

* For $a^n b^n$ $n \geq 1$, in PDA we were doing Push() and Pop(). This Push() & Pop() is not available in Turing Machine.

* Instead, Read / Writ operation is there which 'read' reads the data and 'writ' Operation perform modification (updation / change) in data.

This modification was not possible in PDA.

Block Diagram



Scanned by TapScanner

* Turing Machine has input/output tape as compared to just input tape in PDA.
* Tape is of infinite length with no ending.
* Reading head can move in both direction.

PDA vs TM

PDA	TM
→ Complexity is very high for complex problem	→ Complexity is low Compared to PDA
→ There is only Read operation in PDA	→ There is both Read & Write operation
→ Modification/updation is not allowed	→ Modification is allowed
→ Reading head can move only in one direction	→ Reading head can move in both direction
→ There is only Input Tape	→ There is input/output Tape

Formal Definition of TM :-

A Turing Machinge is expressed as a 7 tuples

$$(Q, \cancel{\Gamma}, \Sigma, \Gamma, q_0, h, S, B)$$

Q → Set of States

Σ → Set of input alphabets

Γ → tape symbols

q_0 → Start

h → Halt State $(\overset{\to \text{Blank}}{B}, \#)$

B →

δ → Transition function

$$Q \times P \to Q \times P \times (\underset{\text{Left Movement}}{L}, \underset{\text{Right movement}}{R}, \underset{\text{No change}}{N})$$

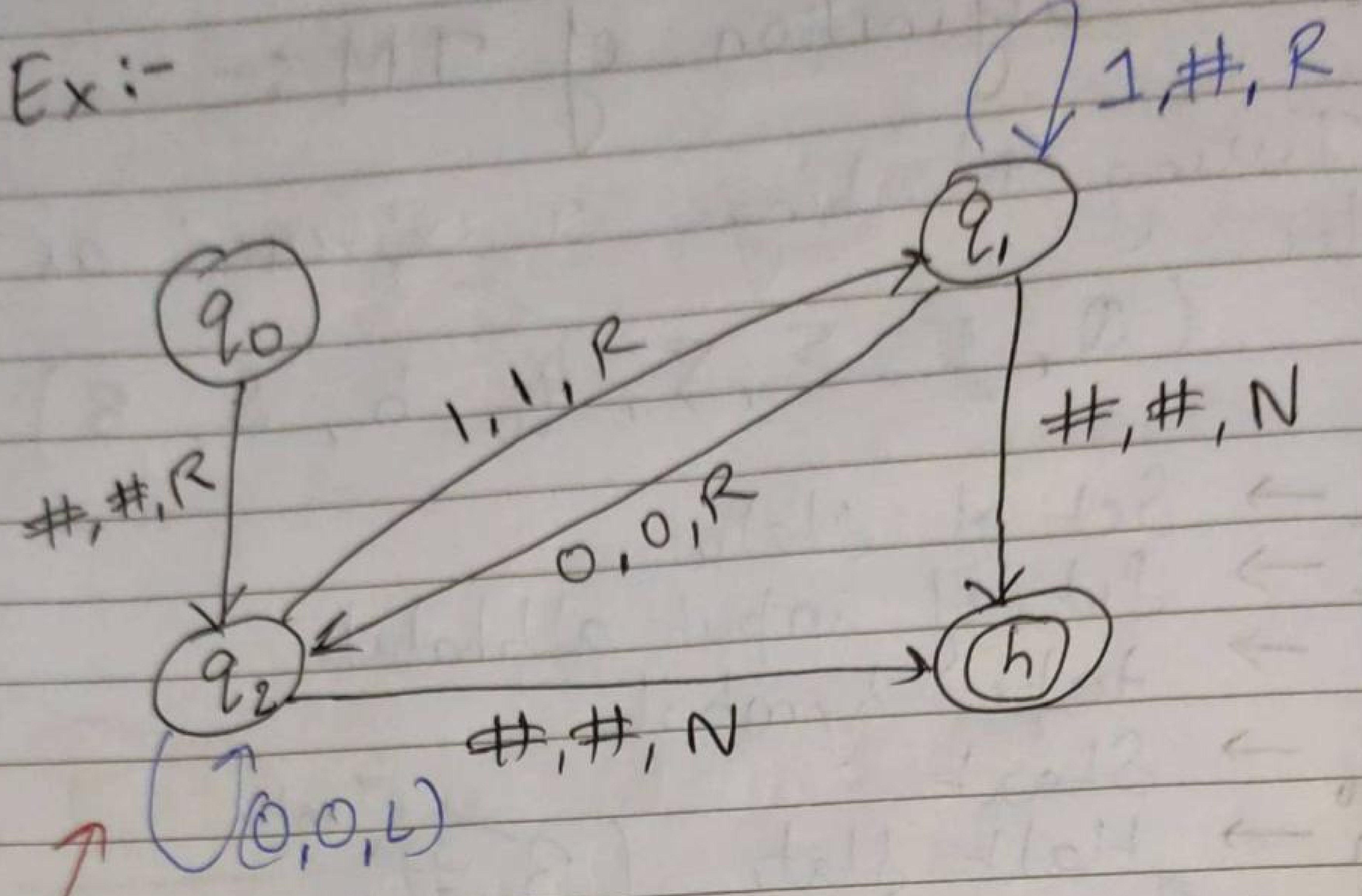
for input alphabet (a, b)

Tape symbol → $(a, b, B/\#)$

→ Difference b/w two is that Blank Symbol can be a part of Tape Symbol whereas it can't be a part of input Symbol.

halt → end

Ex:-



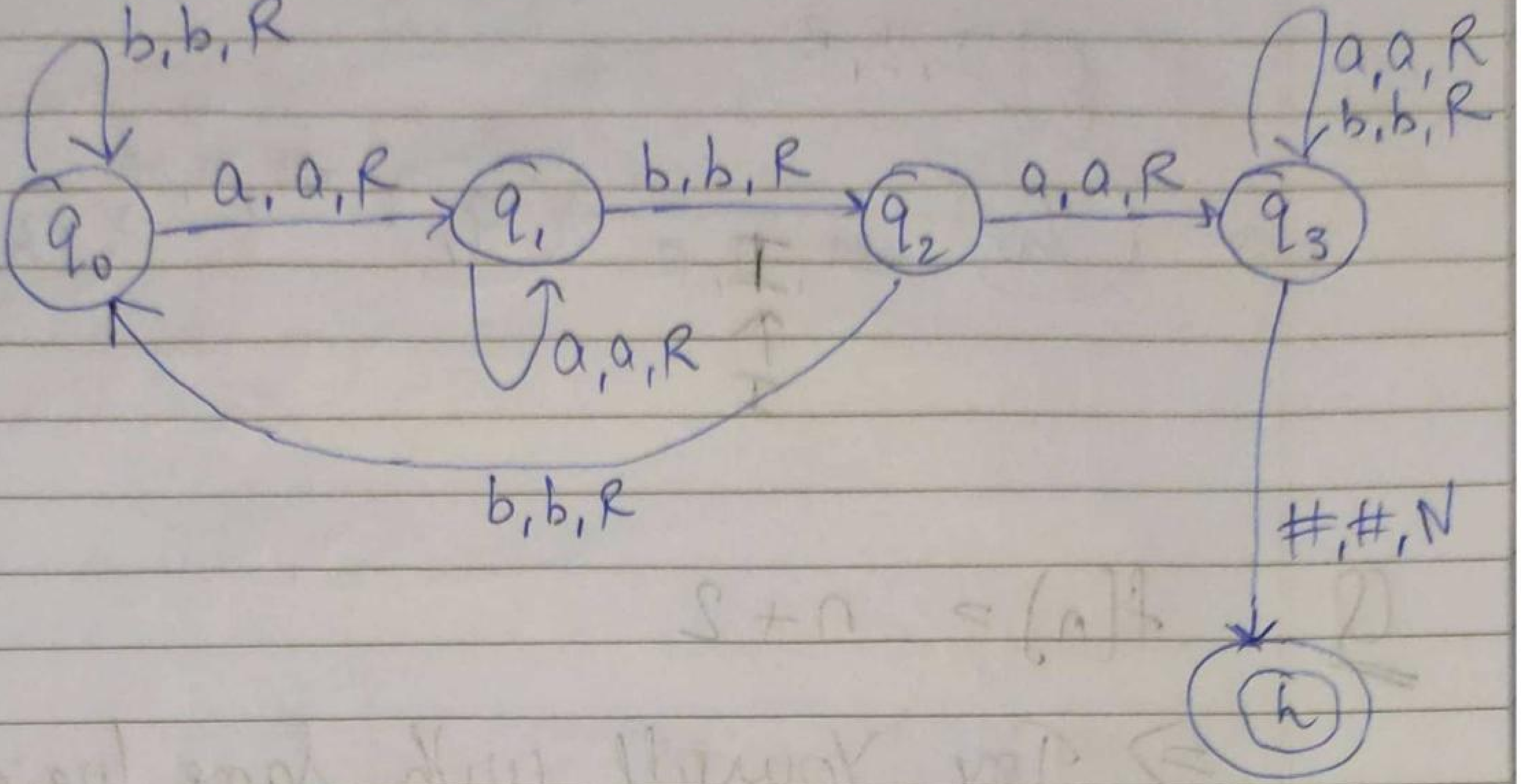
Transition Diagram

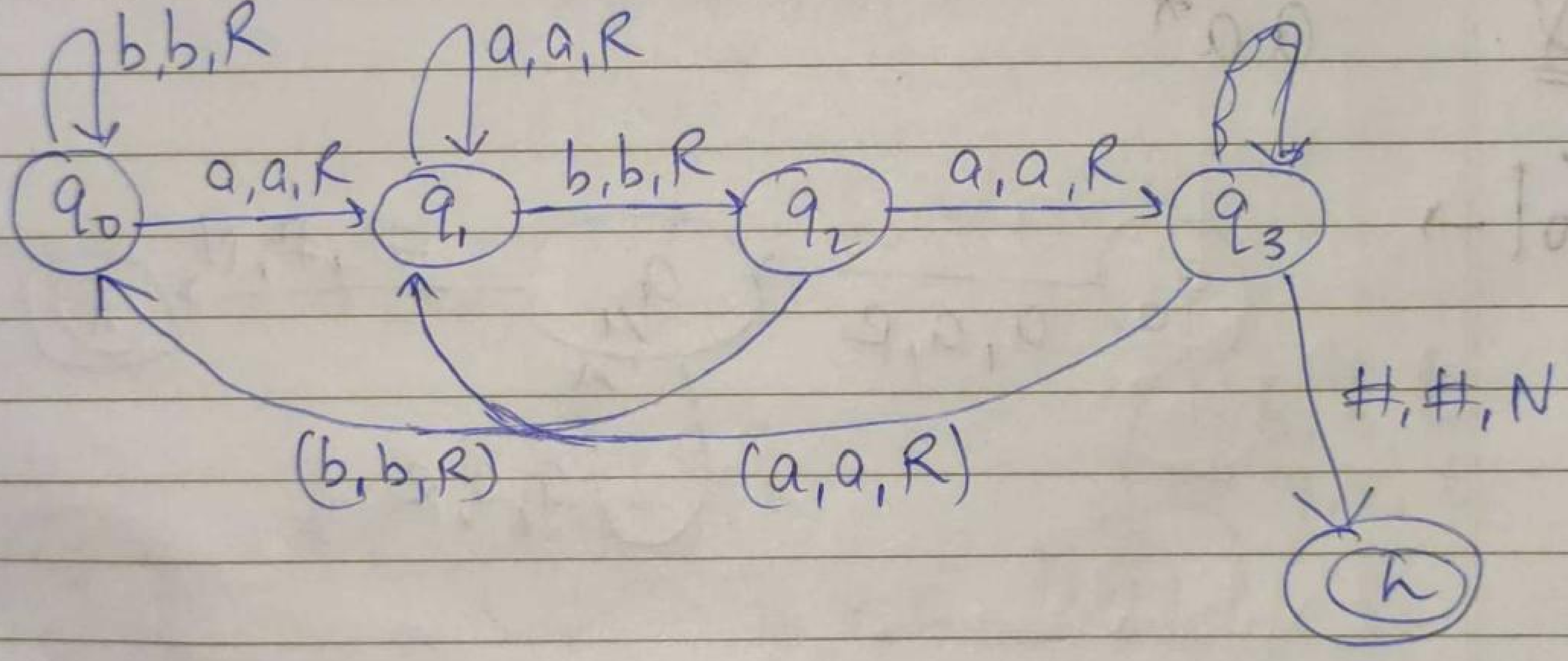
Transition Table :-

	0	1	#/B
q_0	—	—	$(q_2, \#, R)$
q_1	$(q_2, 0, R)$	$(q_1, \#, R)$	$(~~q~~h, \#, N)$
q_2	$(q_2, 0, L)$	$(q_2, \#, R)$	$(h, \#, N)$
h	—	—	—

Q. Construct a Turing Machine for a string containing substring aba.

Sol →





Q. $f(n) = n + 1$

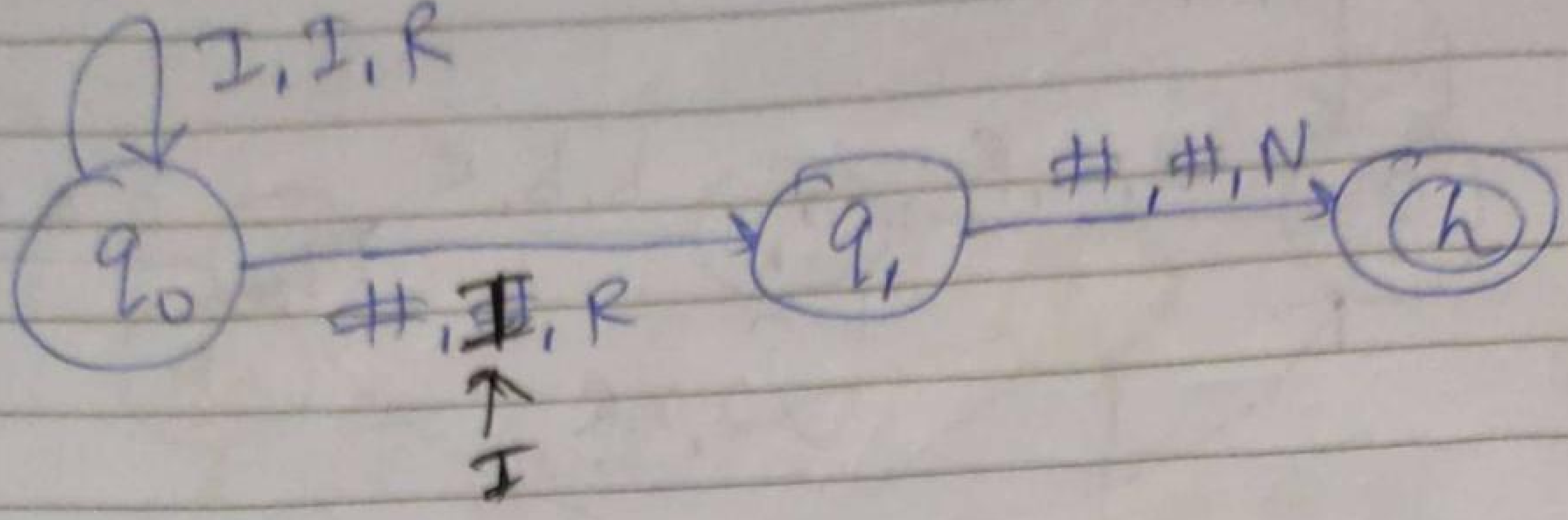
Sol → $n \rightarrow 2$ O/P → 3

$n \rightarrow 5$ O/P → 6

⇒ I/P :- # I I $\boxed{\#}$ #

O/P :- # I I $\boxed{I}$ #

└→ chaning 1st # to I.

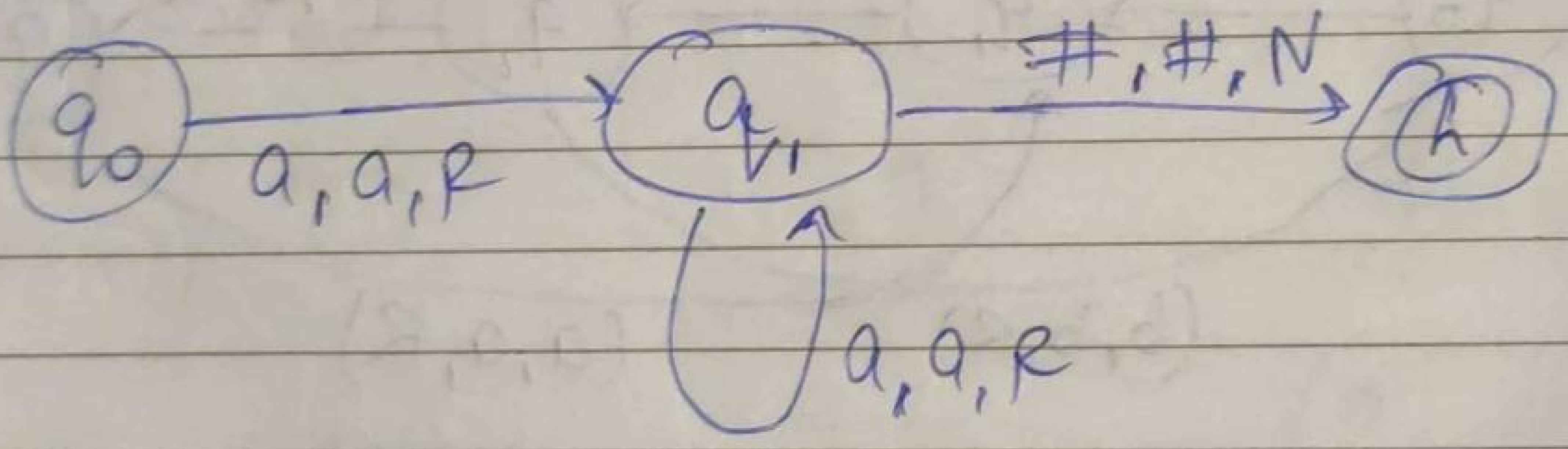


Q f(n) = n + 2

⇒ Try Yourself with same logic.

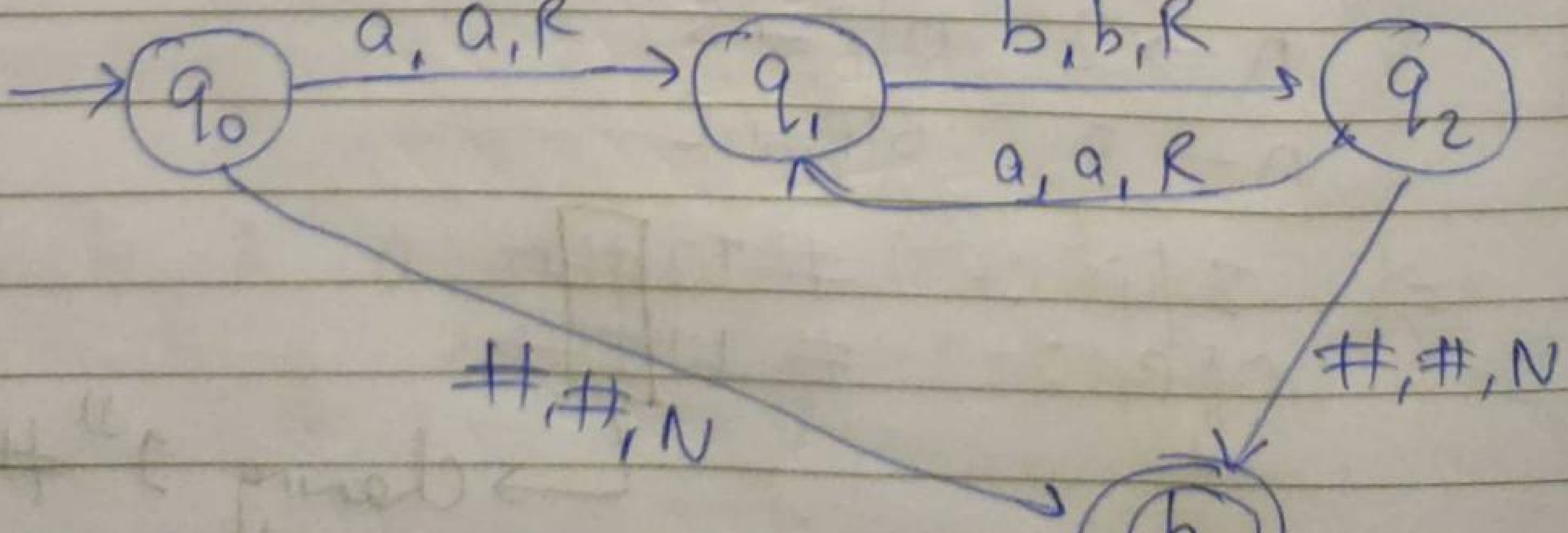
Q. aa*

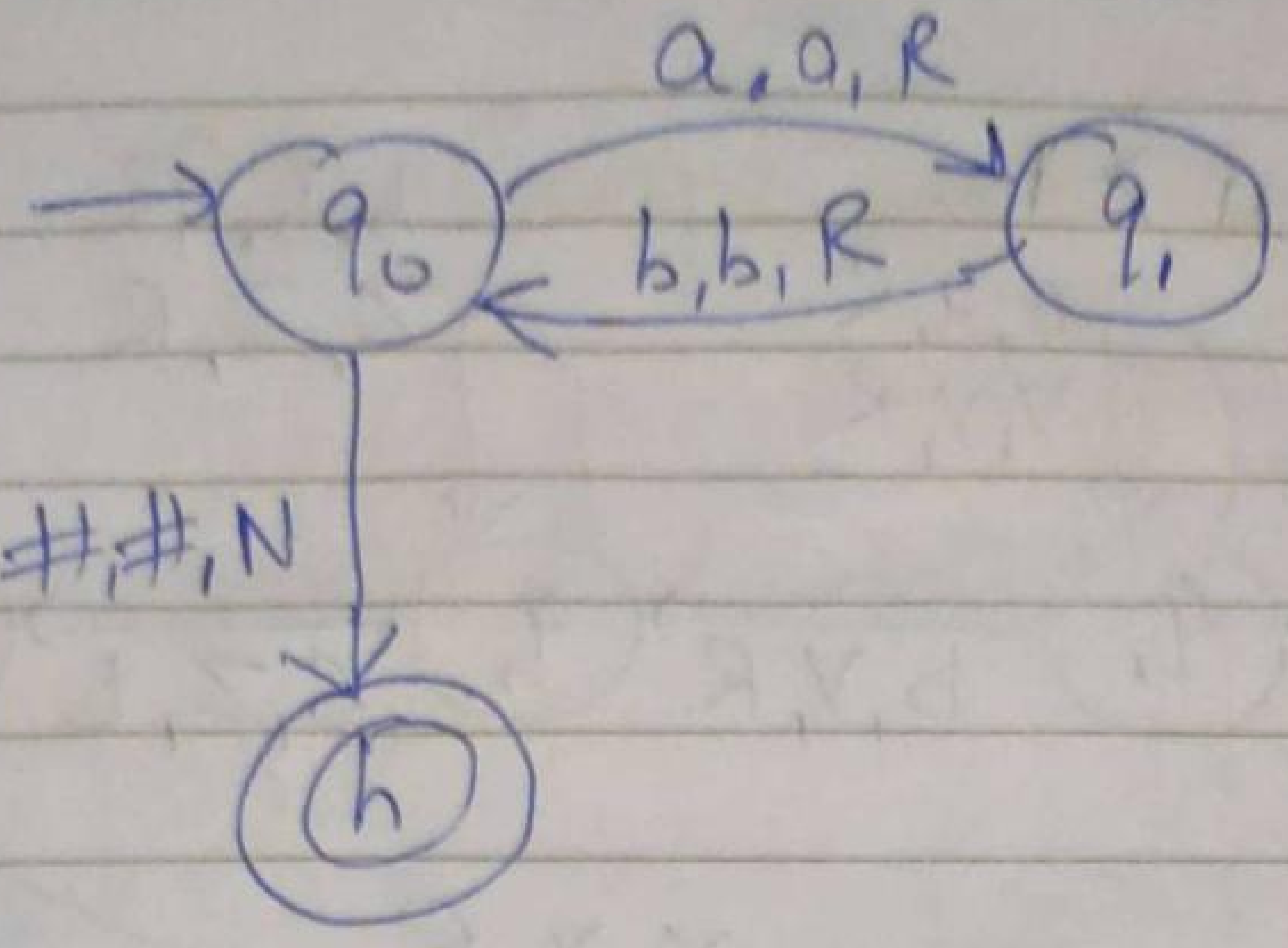
Sol →



Q $(ab)^n$, $n \geq 0$

Sol →



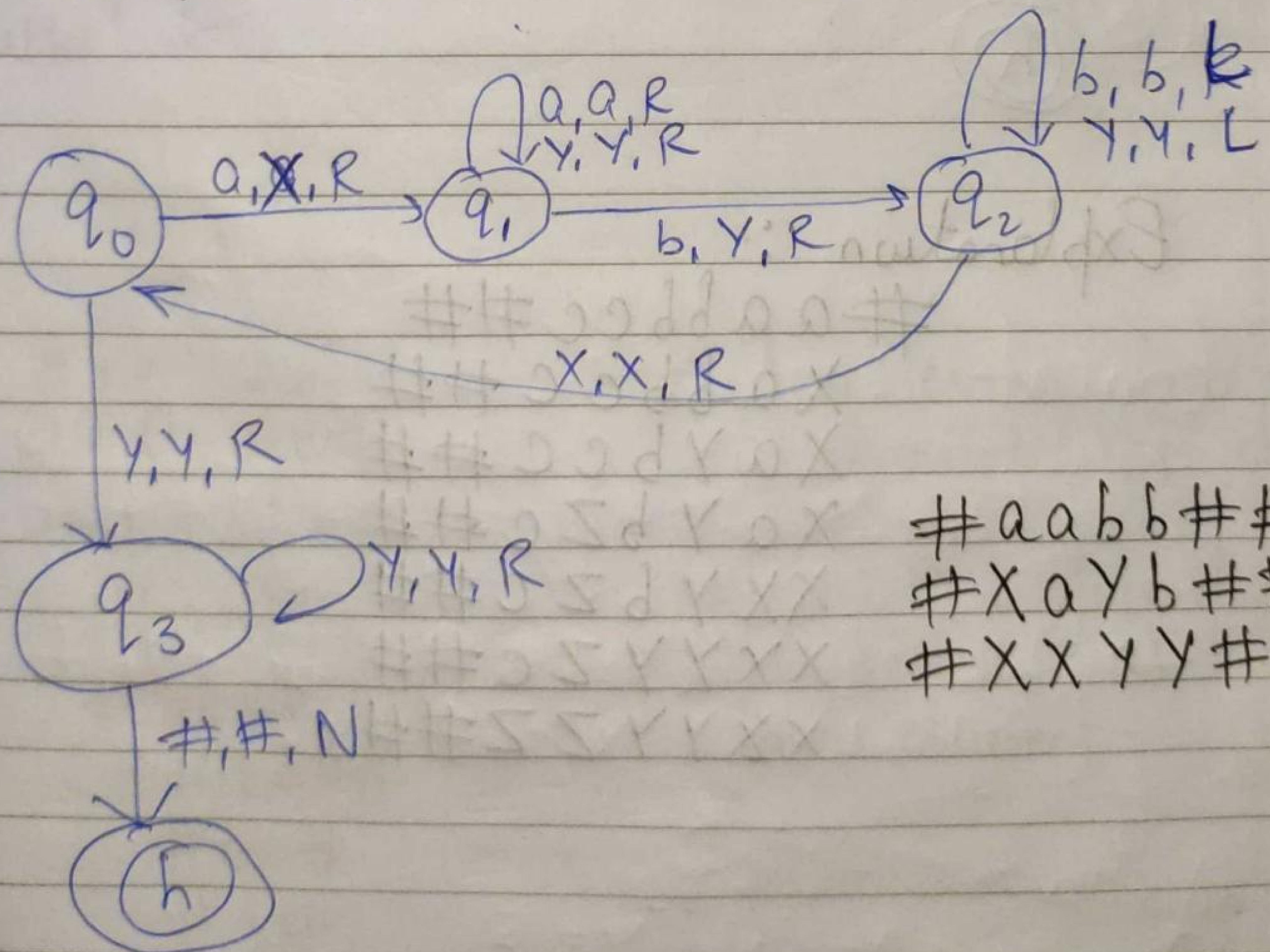


Not → for input → # a a b b # #

Blank Symbol

Scan Start from here

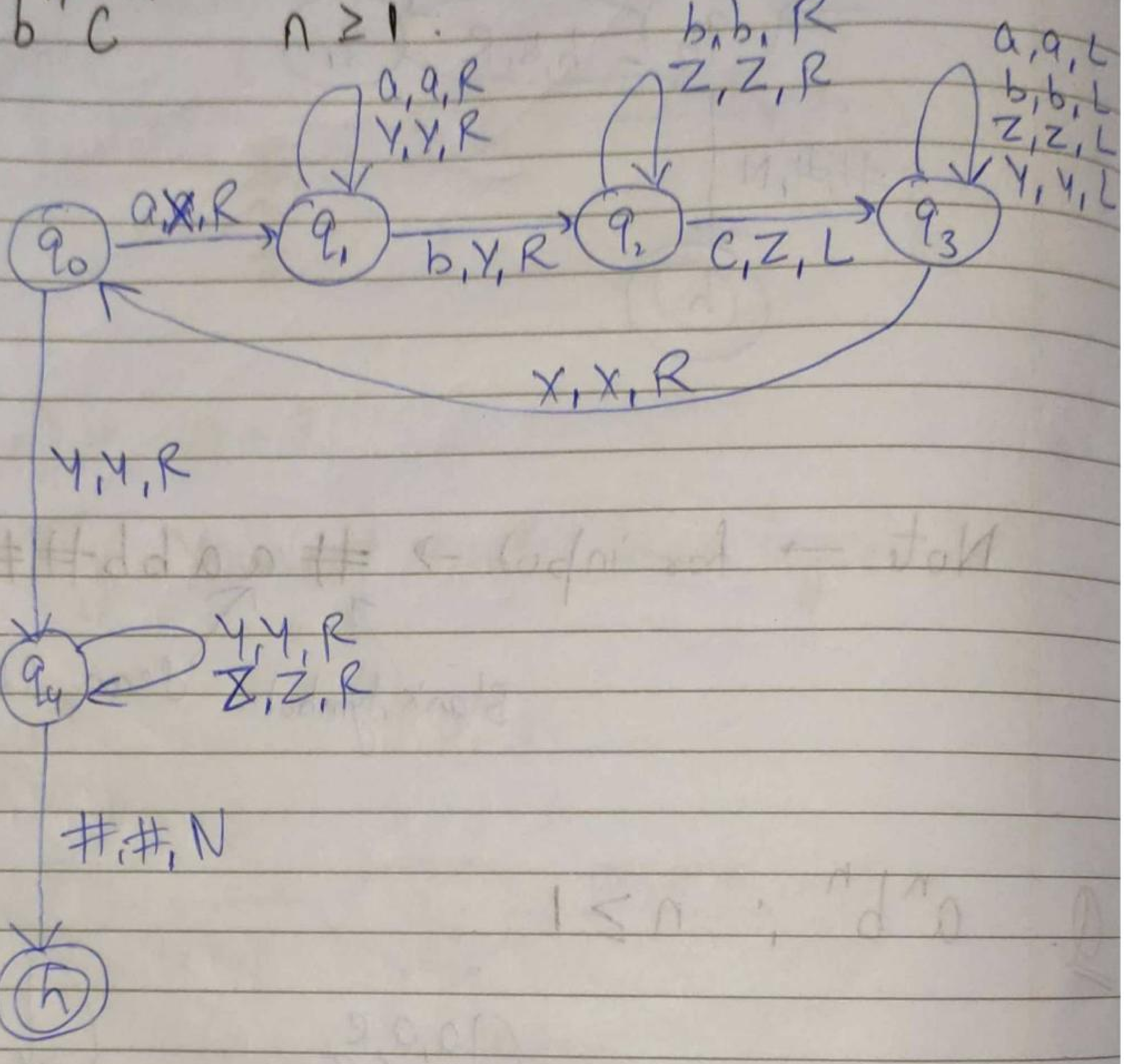
Q. $a^n b^n$; $n \geq 1$



a a b b #
X a Y b #
X X Y Y #

Q $a^n b^n c^n$ $n \geq 1$.

Sol →



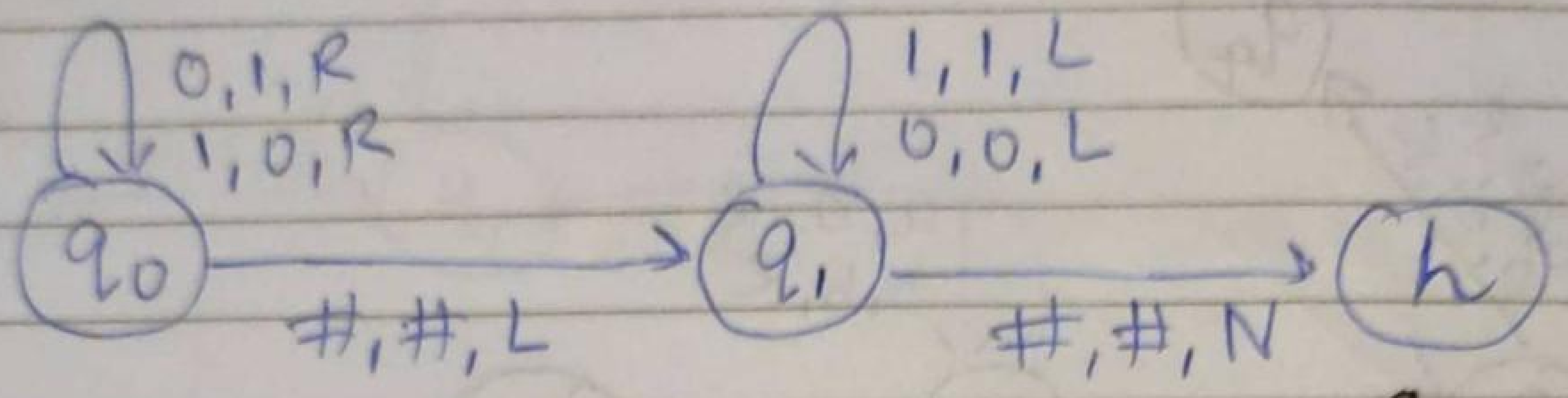
Explanation :-

```
#aabbcc##
  Xabbcc##
  XaYbcc##
  XaYbZc##
  XXYbZc##
  XXYYZc##
  XXYYZZ##
```

transducer

Q. 1's Complement :-

I/P :- #0101#
O/P :- #1010#



See no final stat Symbol

Q. ww^r :-

Sol → $w\,w^r$

~~aba~~ ~~aba~~

① ③ ⑤ ⑥ ④ ②

→ for every first element eliminat last element

as for ww^r, first element is same as last element.

⇒ #abaaba#

#baaba

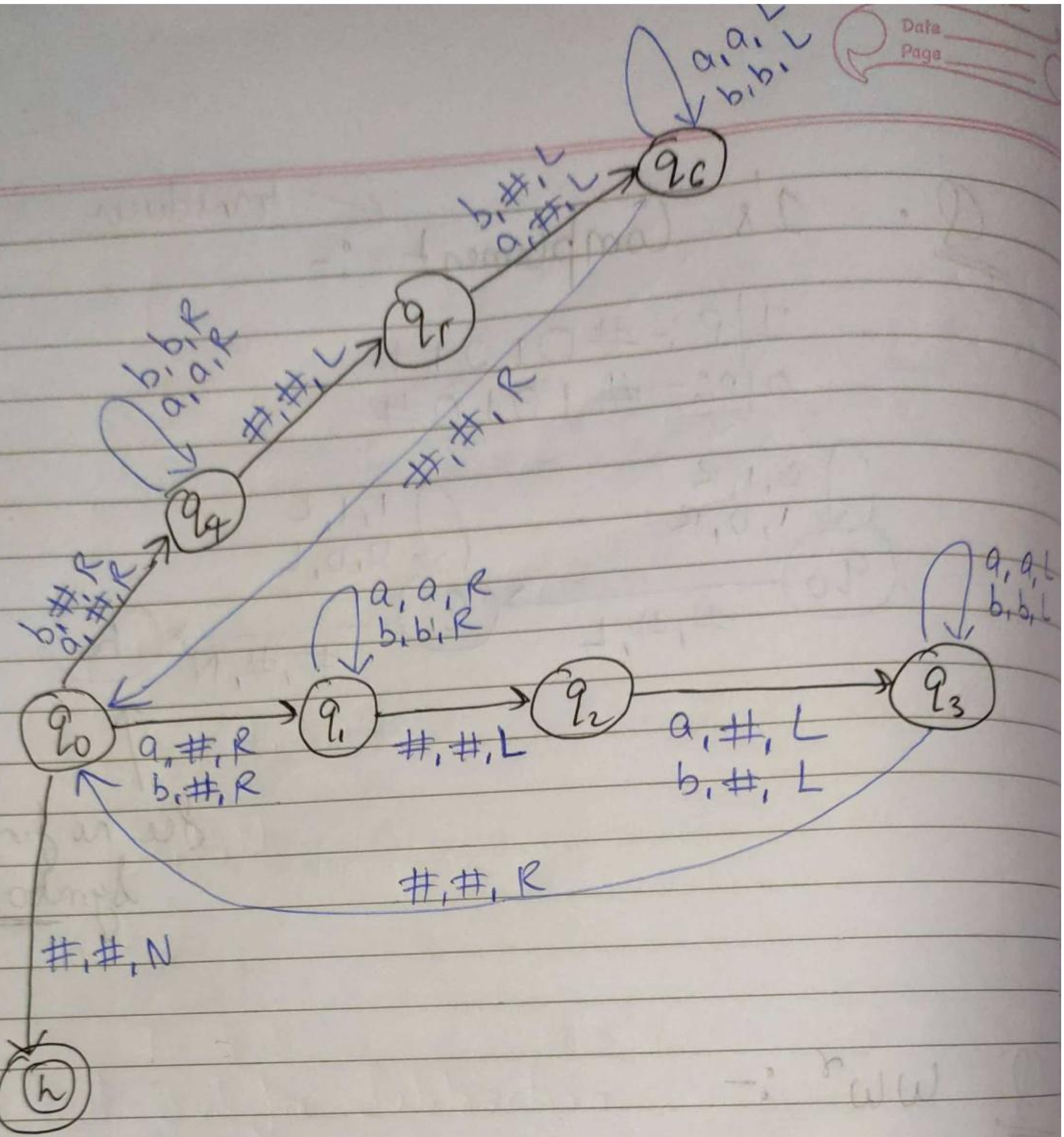
#baab#

##~~b~~ab#

#aa#

#a#

##



* Concat :- ww

* Addition :- f(x, y) = x + y.

↳ We use delimiter to be able to bifurgat (differentiate) X and Y.

f(x, y).

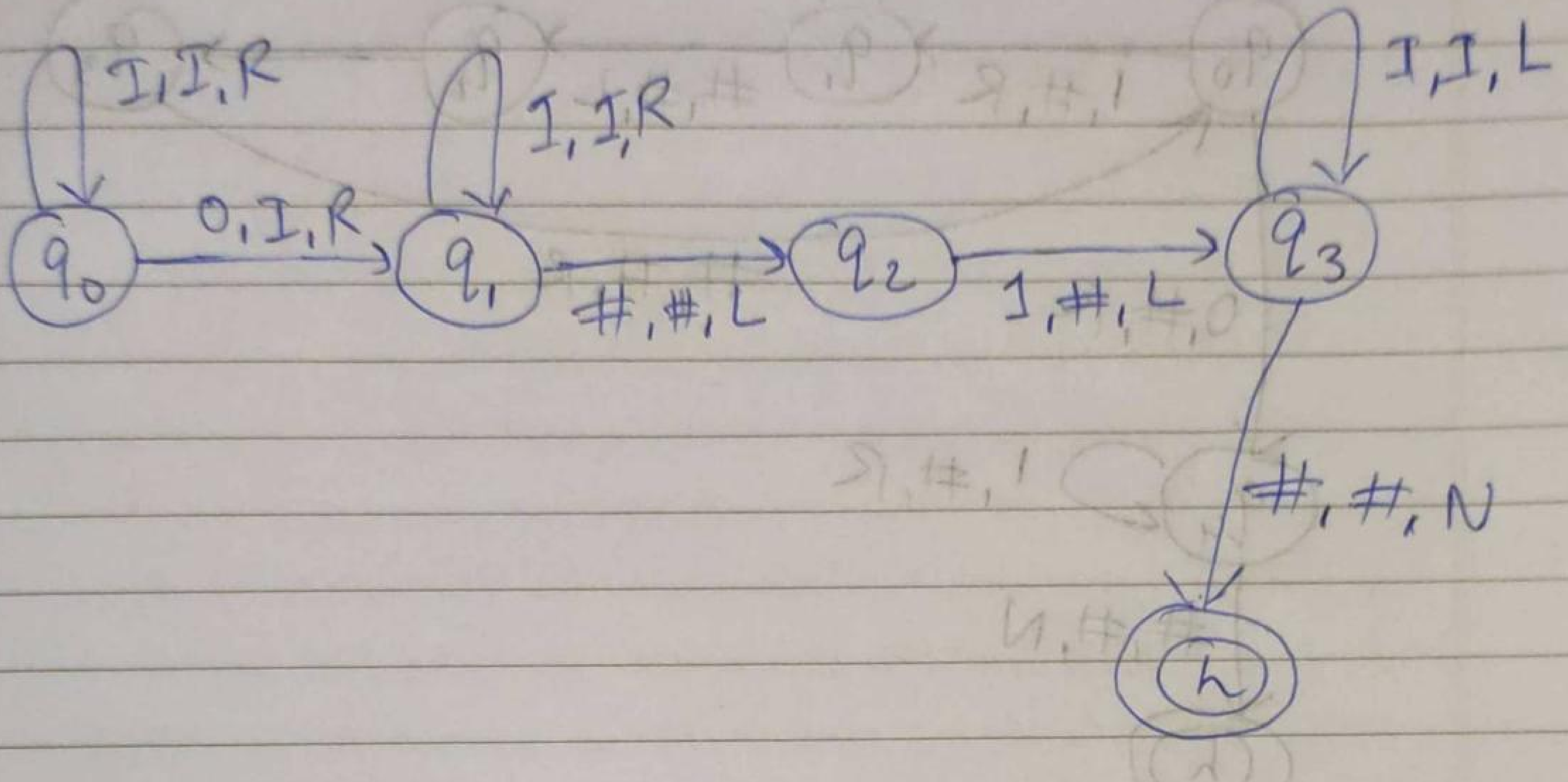
x = 2 ↓ y = 3

11 111

delimiter

⇒ 11 $\boxed{0}$ 111

Q. $f(x, y) = x+y$



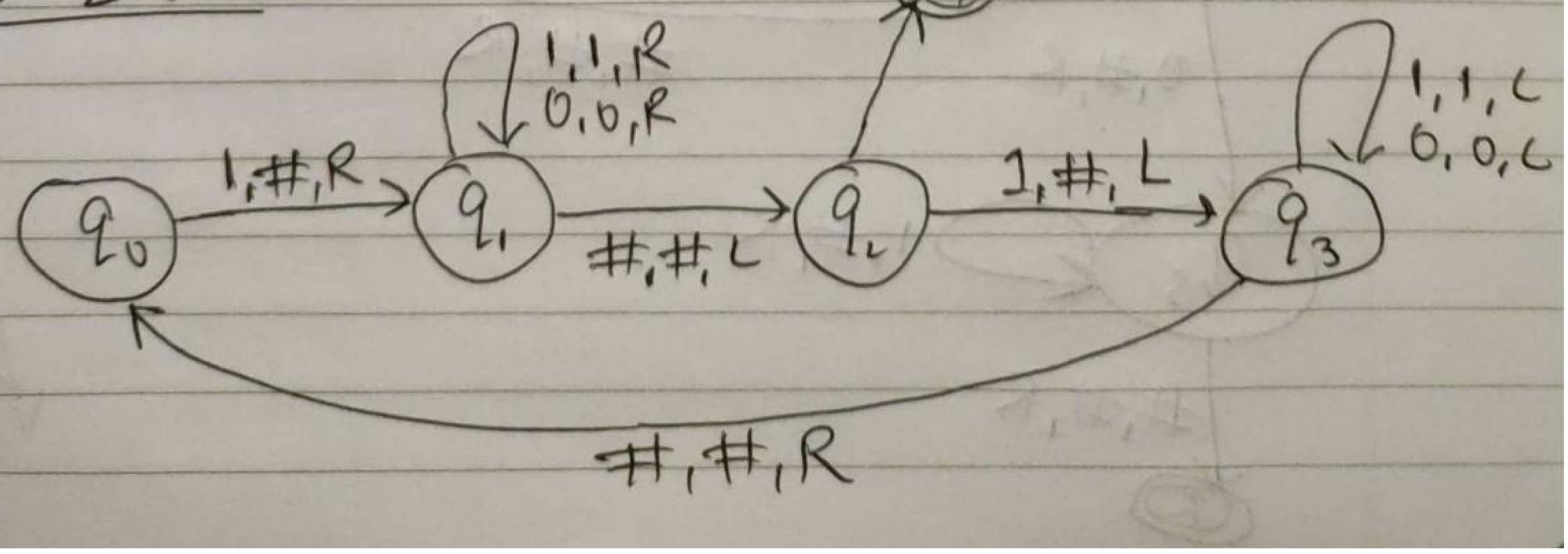
* Subtraction :- #XX1 0XX##

$\boxed{\# 1}$

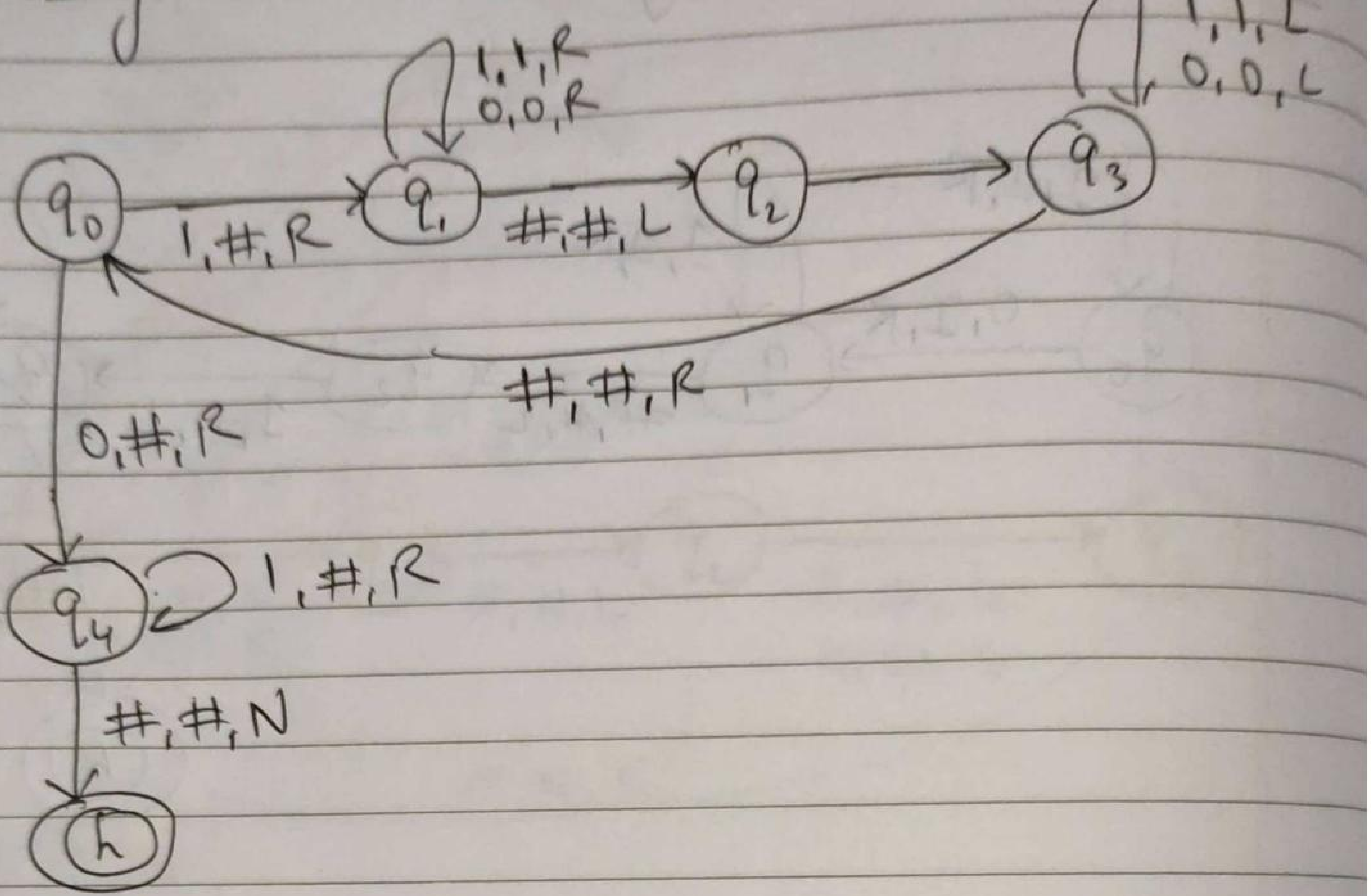
$f(x, y) = x-y$

$$\left\{ \begin{array}{ll} x > y & \longrightarrow x-y \\ x \leq y & \longrightarrow 0 \end{array} \right\}$$

3-2 :-

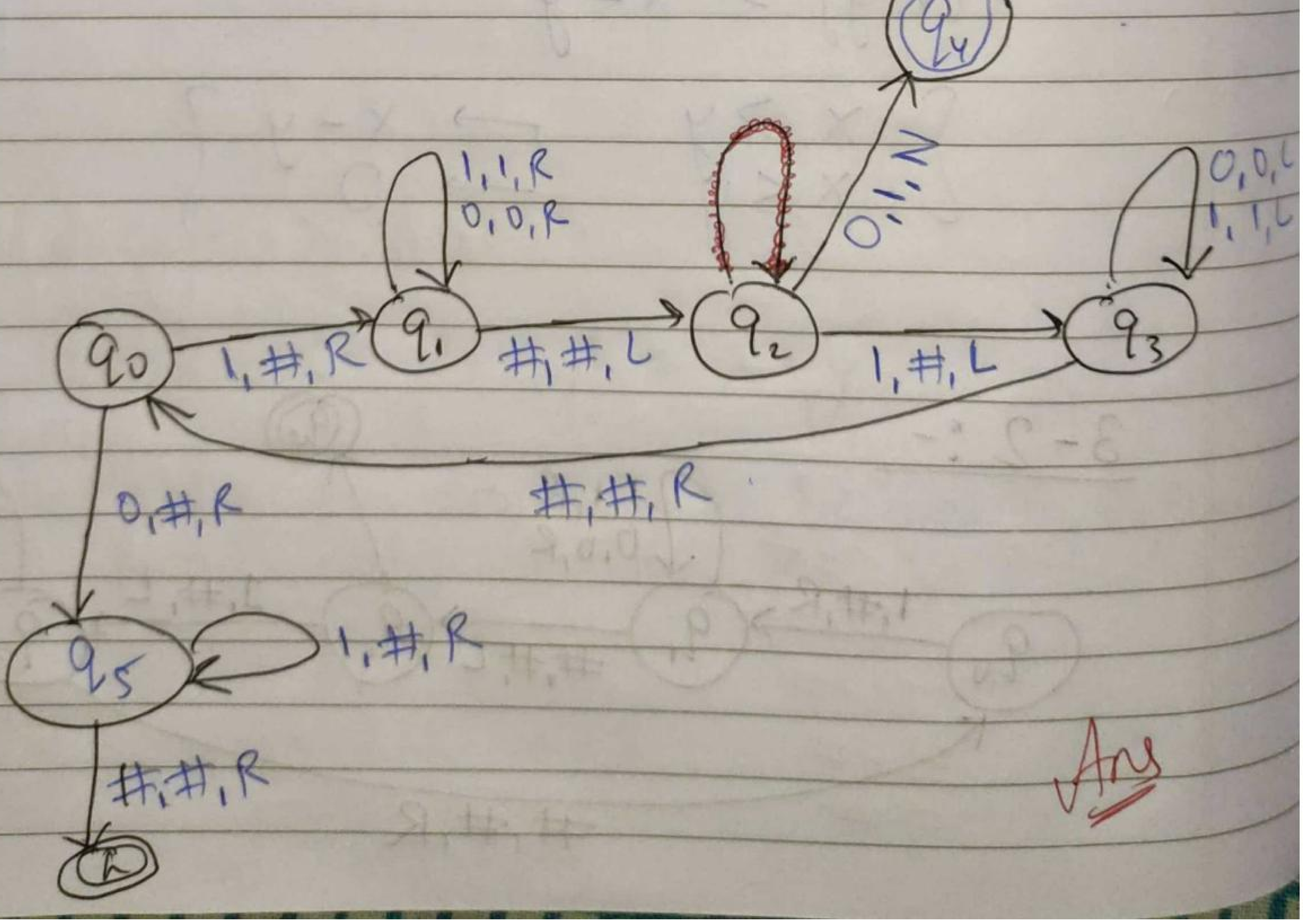


$x \leq y$:-



7-3 :-

→ final sol for Subtraction :-



Ans

Multiplication :-

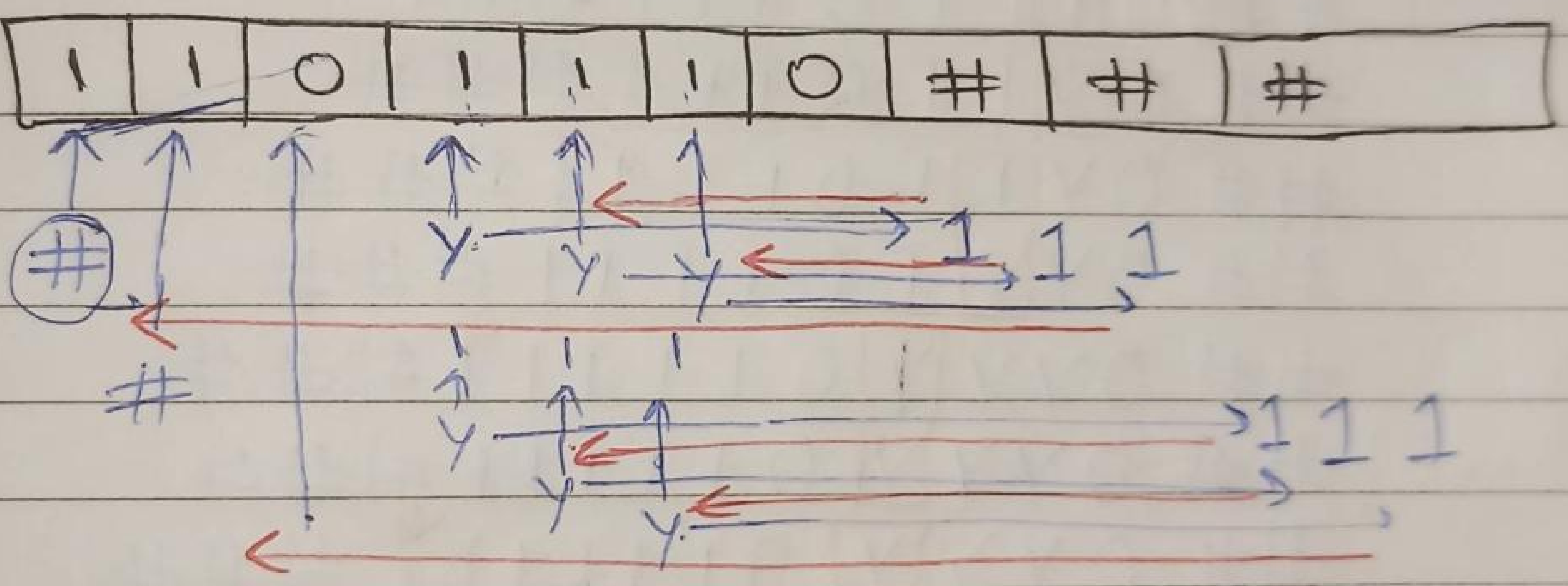
$f(n) = x * y$

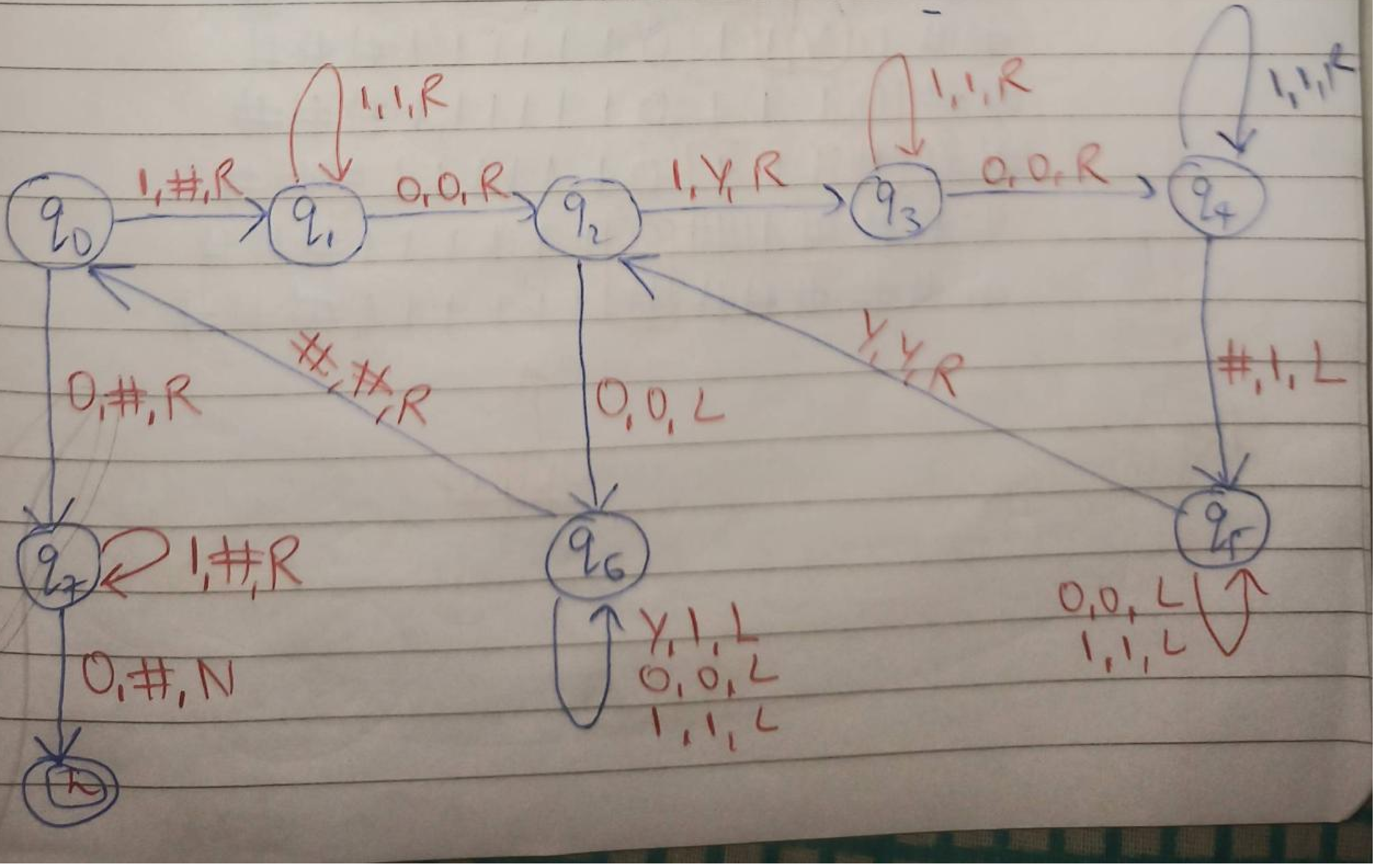
11 0 111 0 111111 ###

↑ delimiter to separat x & y.

↑ delimiter to separat input & output

* Multiplication follows Repeated Addition Rule -





Explanation (Process step by step) :-

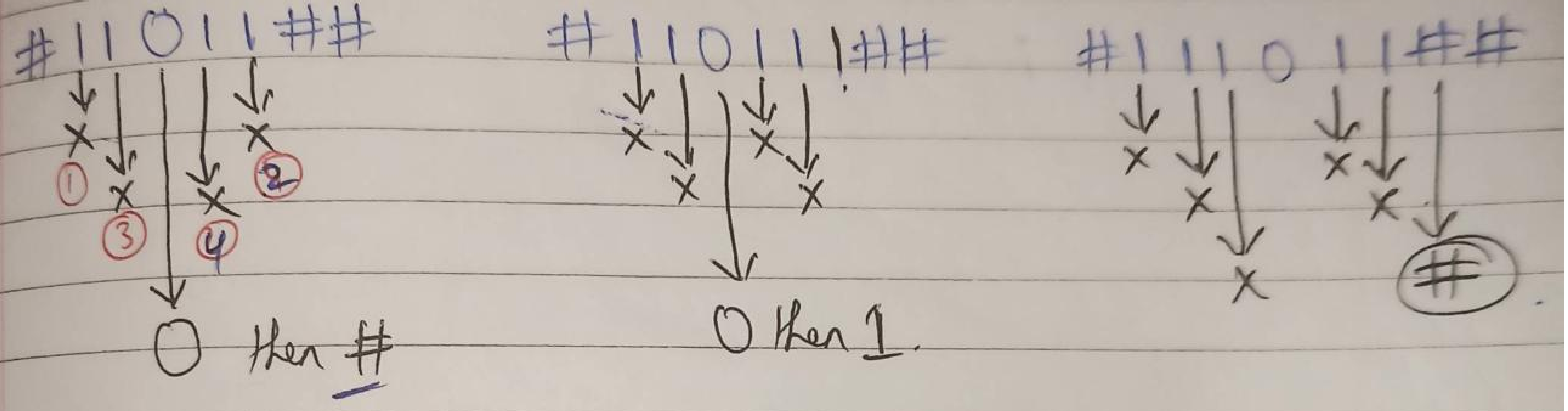
```
11 01110###
#1 0 1 1 1 0###
#1 0 Y 1 1 0###
#1 0 Y 1 1 0 1###
#1 0 Y Y 1 0 1####
#1 0 Y Y Y 0 1 1###
#1 0 Y Y Y 0 1 1 1###
#1 0 Y Y 1 0 1 1 1###
#1 0 Y Y 1 0 1 1 1###
#1 0 1 1 1 0 1 1 1###
## 0 1 1 1 0 1 1 1###
## 0 Y 1 1 0 1 1 1###
## 0 Y 1 1 0 1 1 1 1###
## 0 Y Y 1 0 1 1 1 1###
## 0 Y Y Y 0 1 1 1 1 1###
## 0 Y Y Y 0 1 1 1 1 1###
## 0 Y Y Y 0 1 1 1 1 1 1###
## 0 Y Y 1 0 1 1 1 1 1 1###
## 0 Y 1 1 0 1 1 1 1 1 1###
## 0 1 1 1 0 1 1 1 1 1 1###
## # 1 1 1 0 1 1 1 1 1 1###
### ### 0 1 1 1 1 1 1###
### ### # 1 1 1 1 1 1###
```

↑

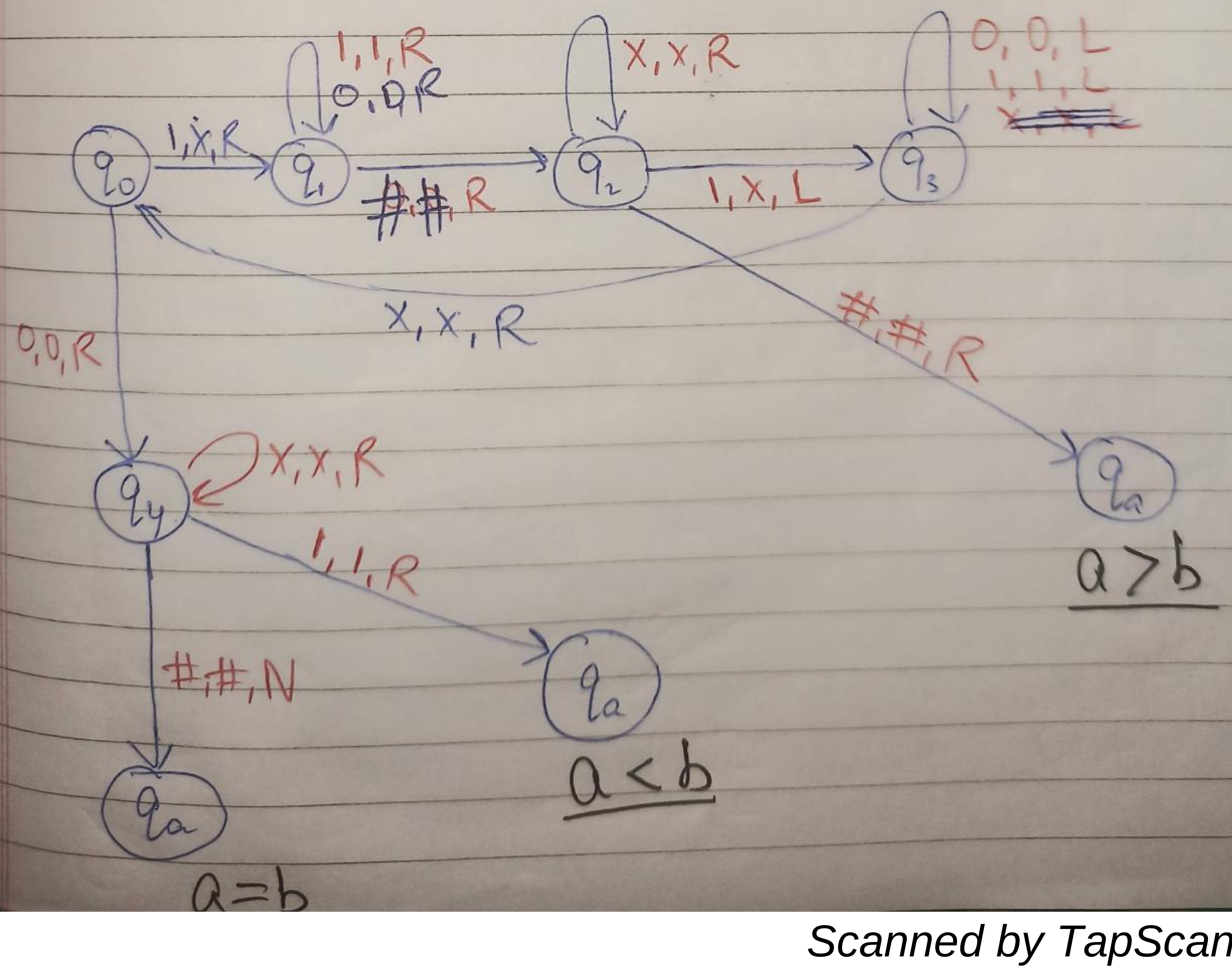
Output

Turing Machine as Comparator :-

1) a > b → 111011
2) a < b → 110111
3) a = b → 11011



#11011##
①
②
③
④
0 then #
#110111##
0 then 1
#111011##
#



q0
q1
q2
q3
q4
qa
qa
qa
1,X,R
1,1,R
0,0,R
#,#,R
X,X,R
1,X,L
0,0,L
1,1,L
~~X,X,L~~
X,X,R
0,0,R
#,#,R
X,X,R
1,1,R
#,#,N
a>b
a<b
a=b