AIUB

SOLUTION

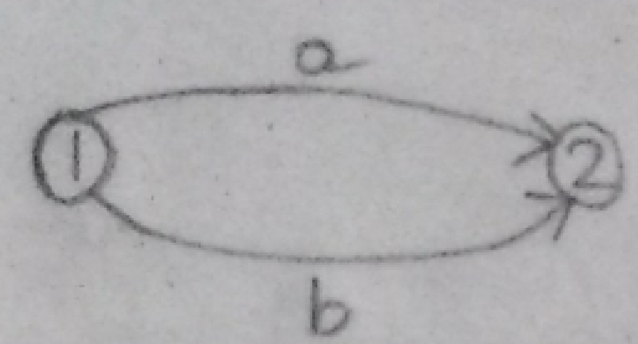
THEORY OF COMPUTATION FINAL TERM HAND NOTES

SOLVED BY-

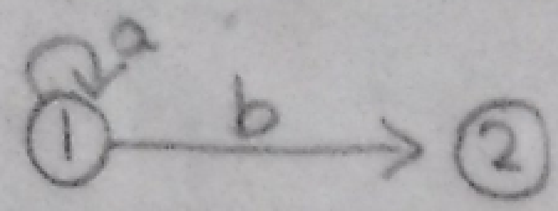
MUSTAFIZUR RAHMAN

DFA - Regular Expression

** $1 \xrightarrow{a} 2 \xrightarrow{b} 3 \Rightarrow 1 \xrightarrow{ab} 3$

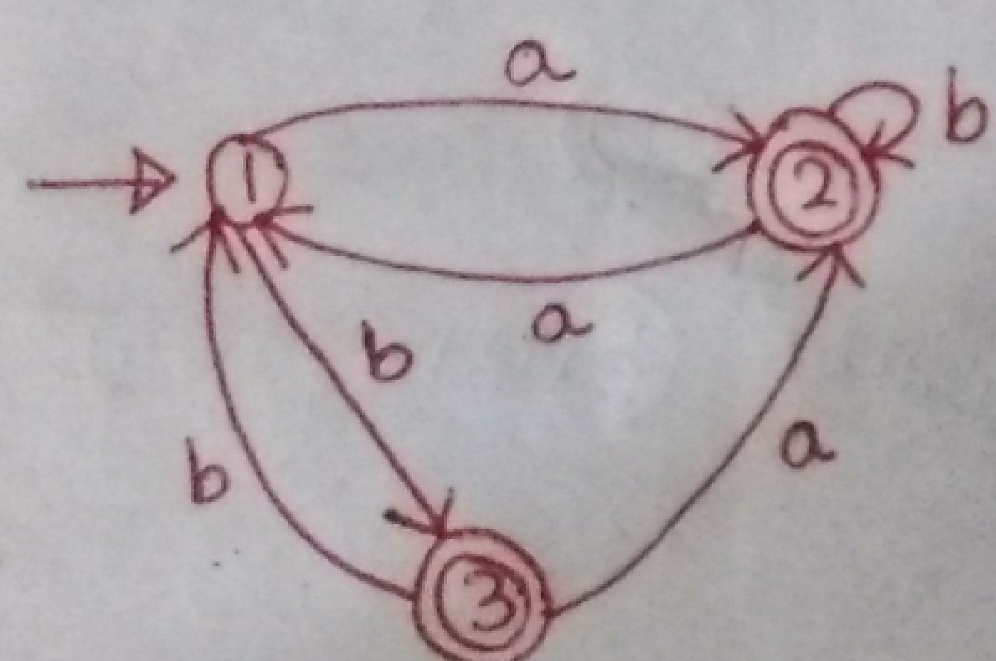


$\Rightarrow 1 \xrightarrow{a \cup b} 2$

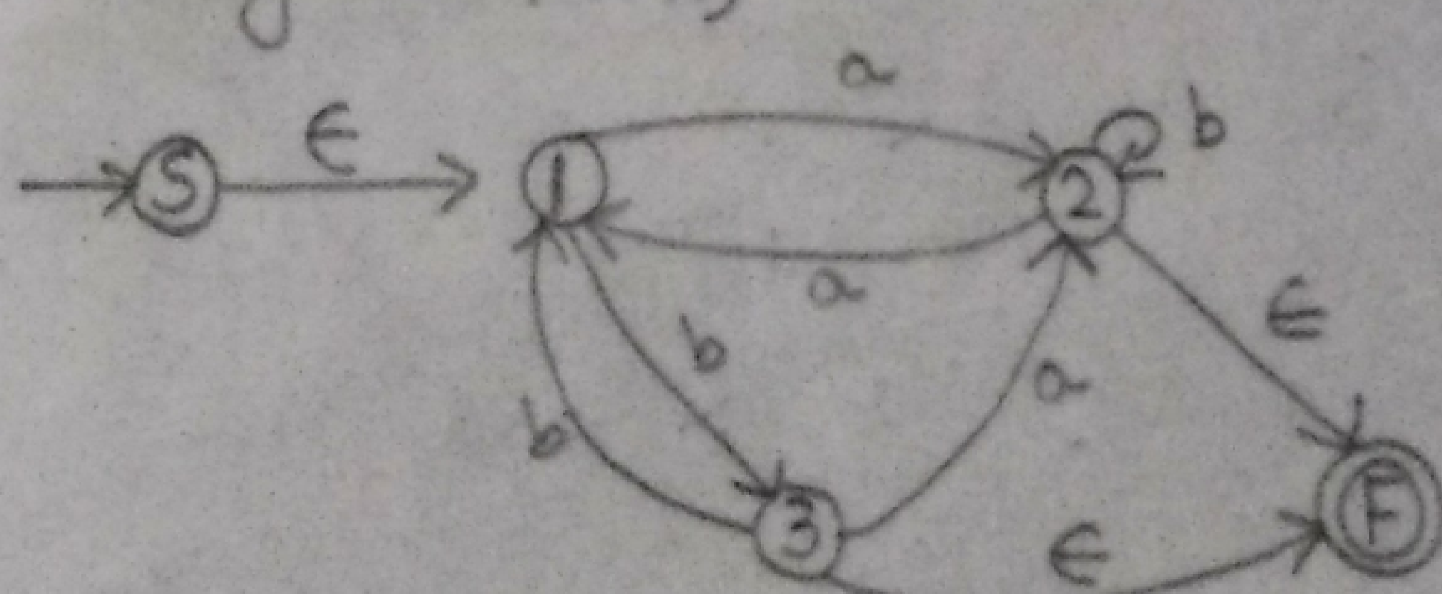


$\Rightarrow 1 \xrightarrow{a^* b} 2$

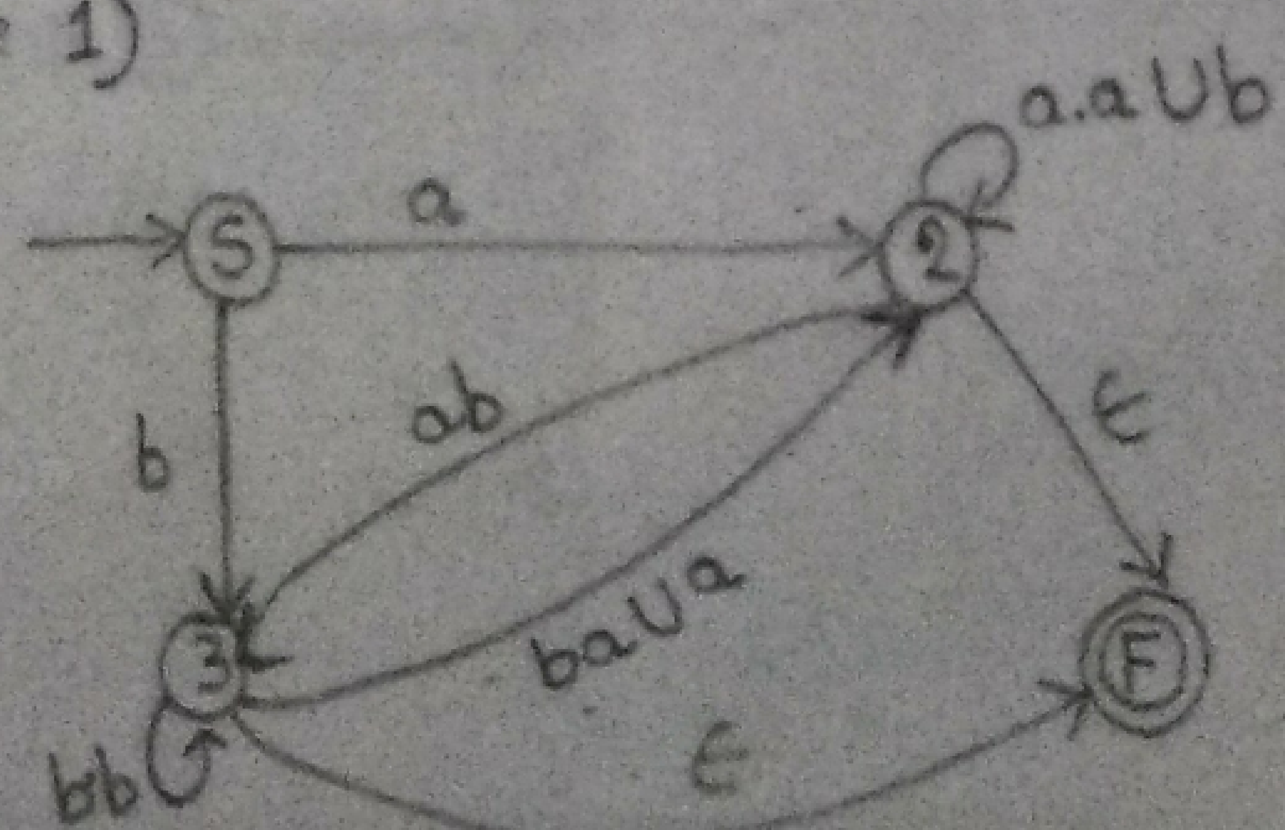
- Convert DFA to GNFA by adding start and final state.
- Remove middle order states by ascending order.



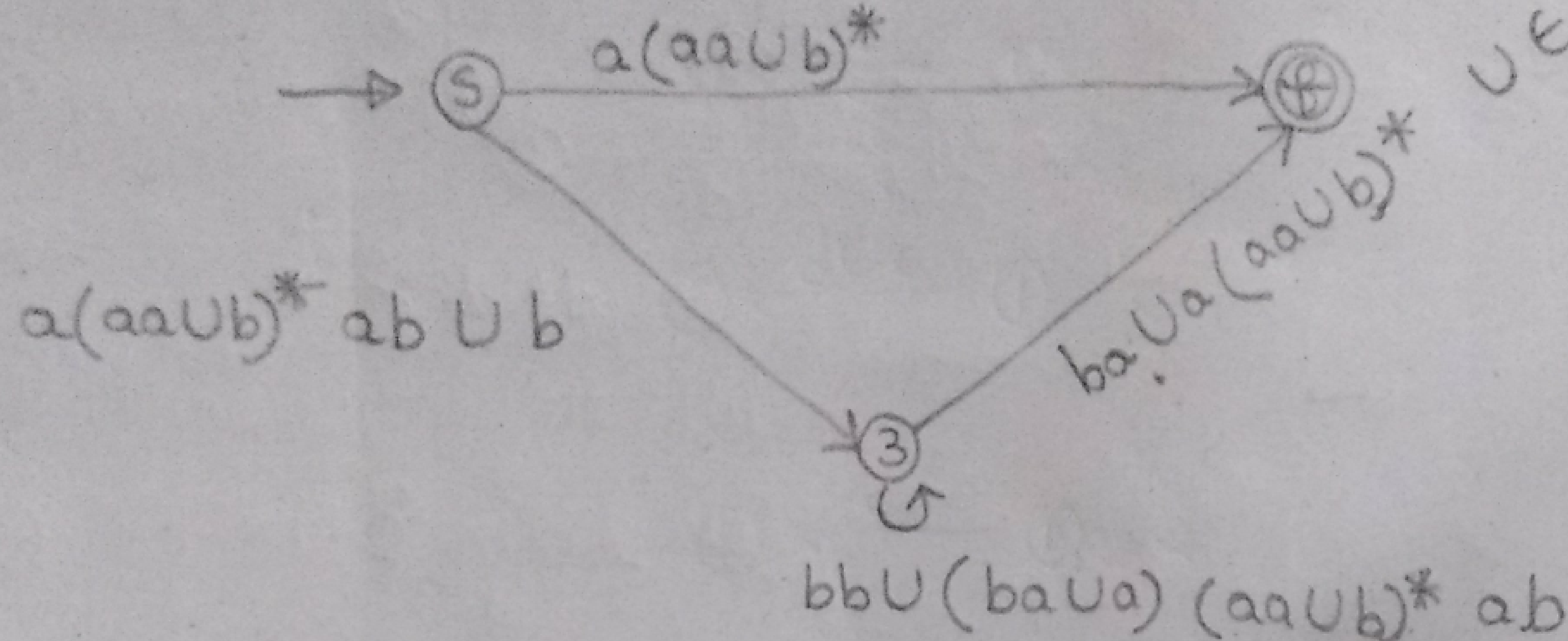
Step-1 (converting to GNFA)



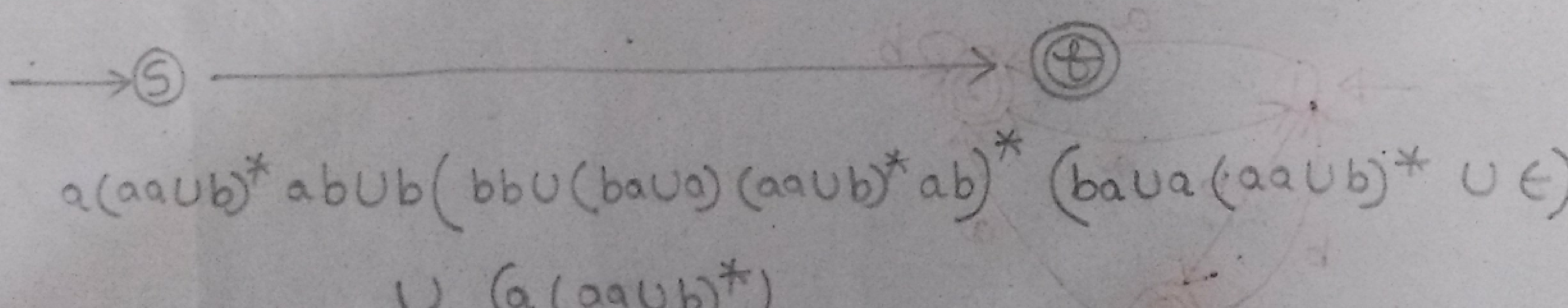
Step-2 (remove 1)

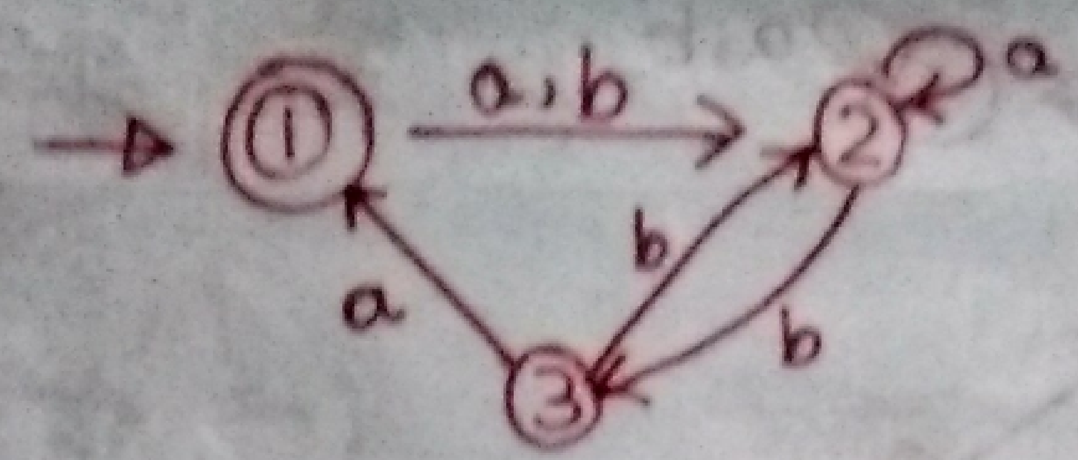


Step-3 (remove 2)

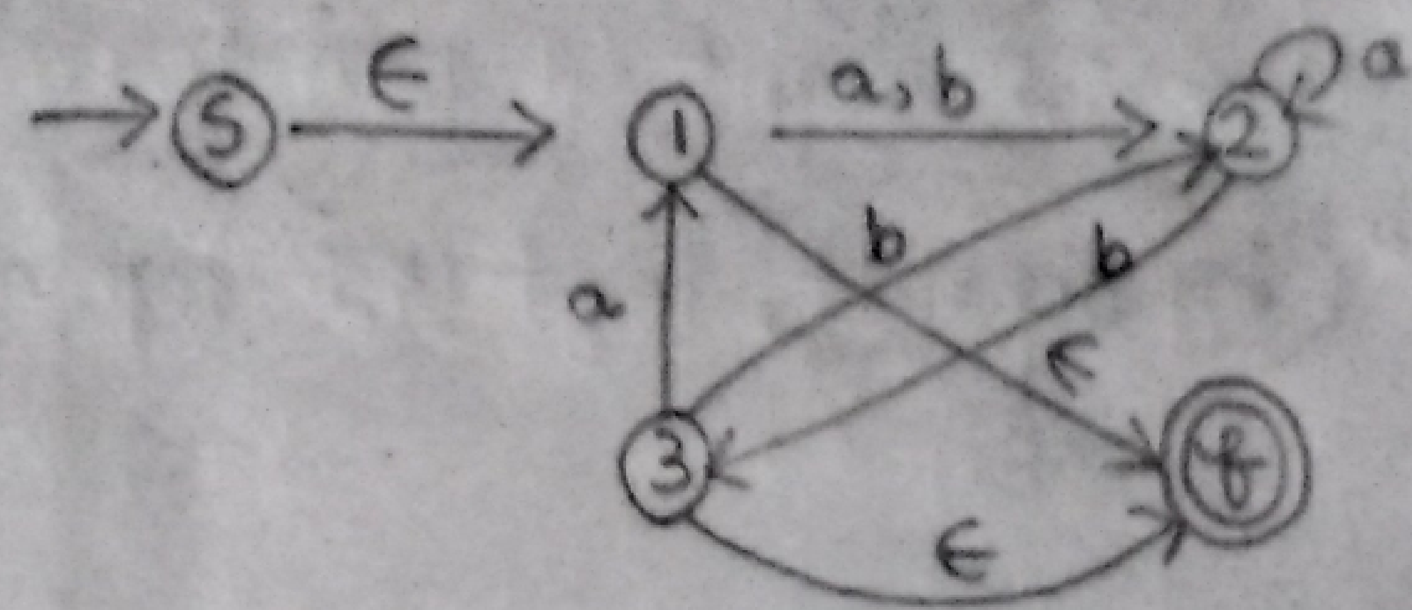


Step4 (remove 3)

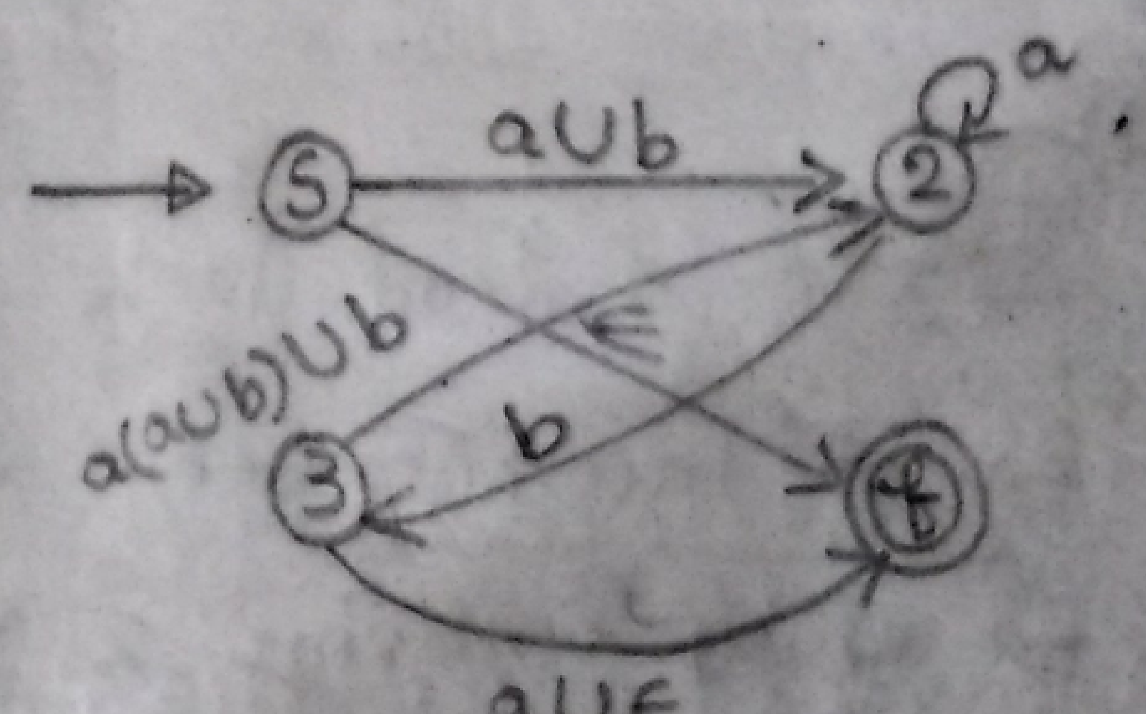




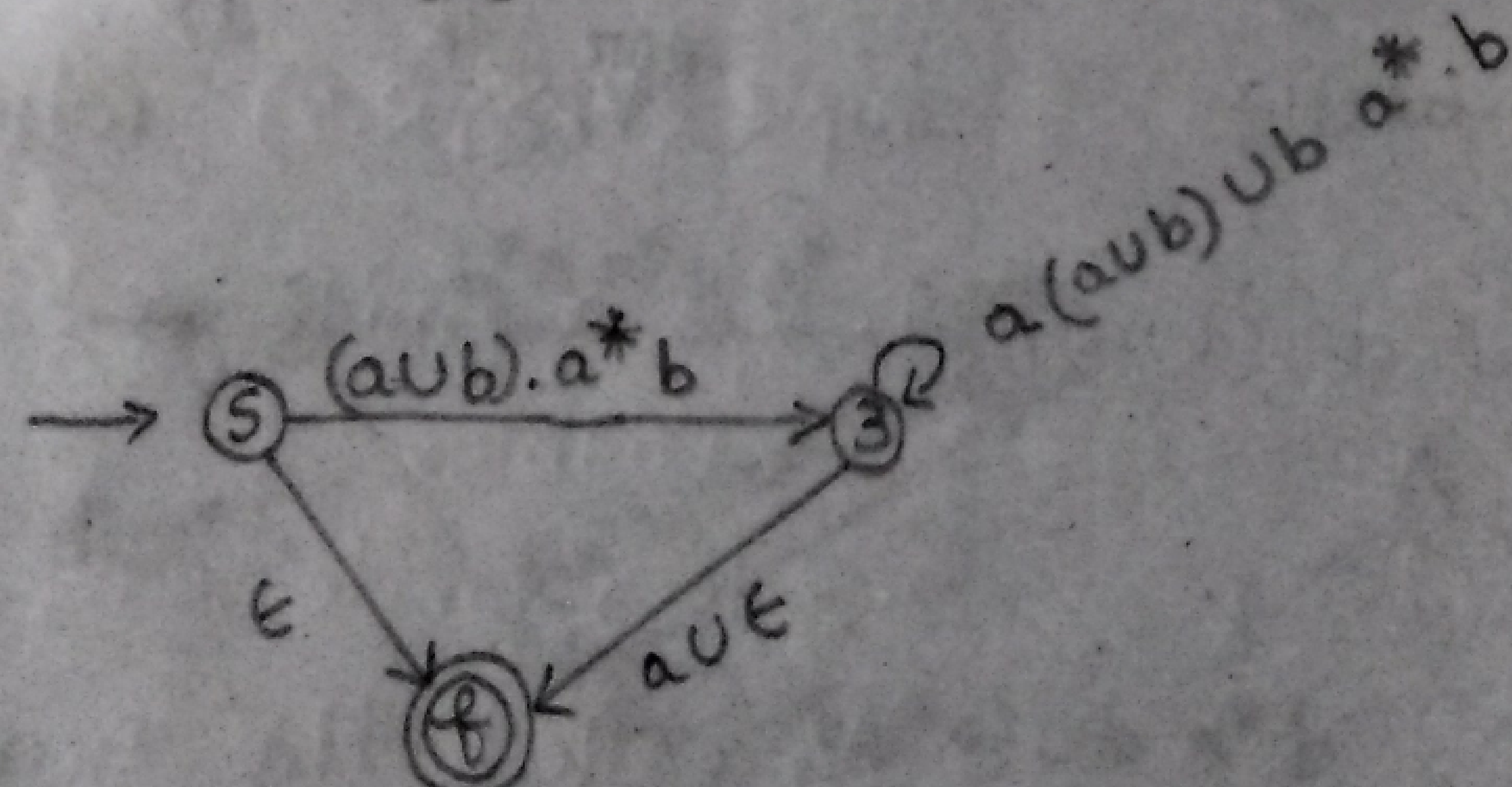
Step 1



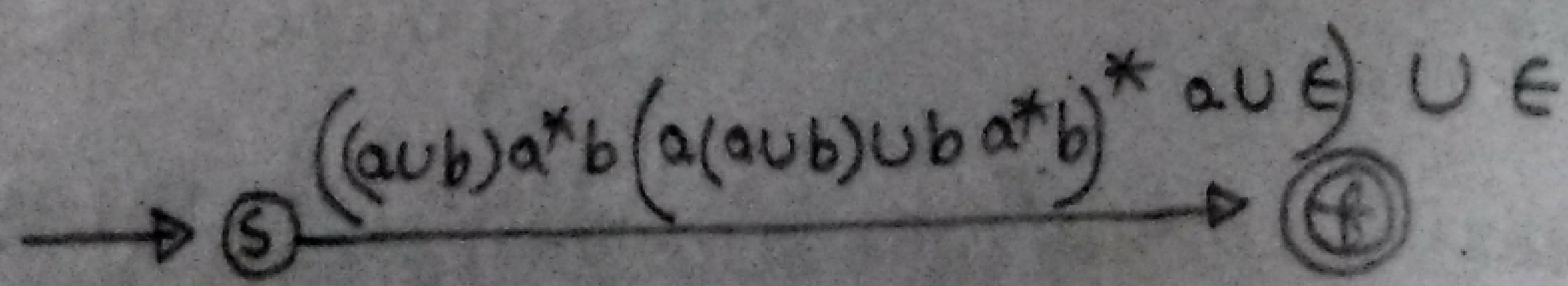
Step 2

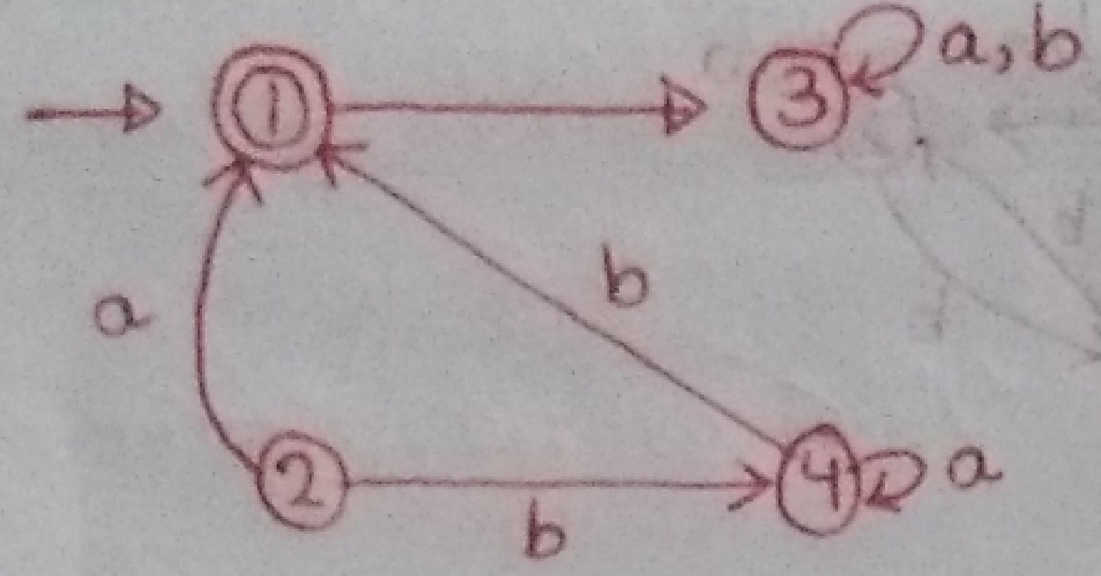


Step 3

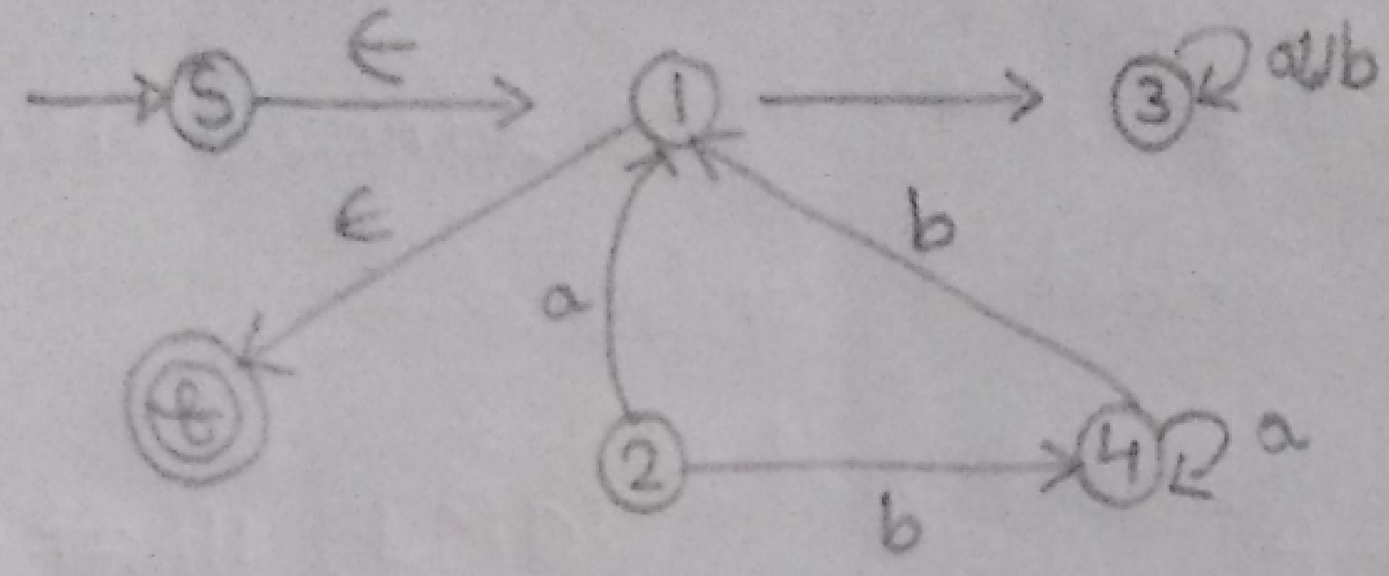


Step 4

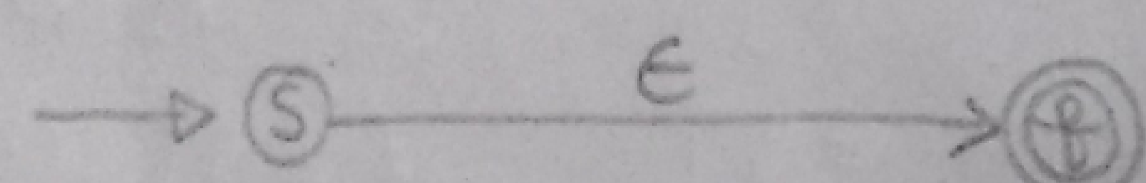




Step-1



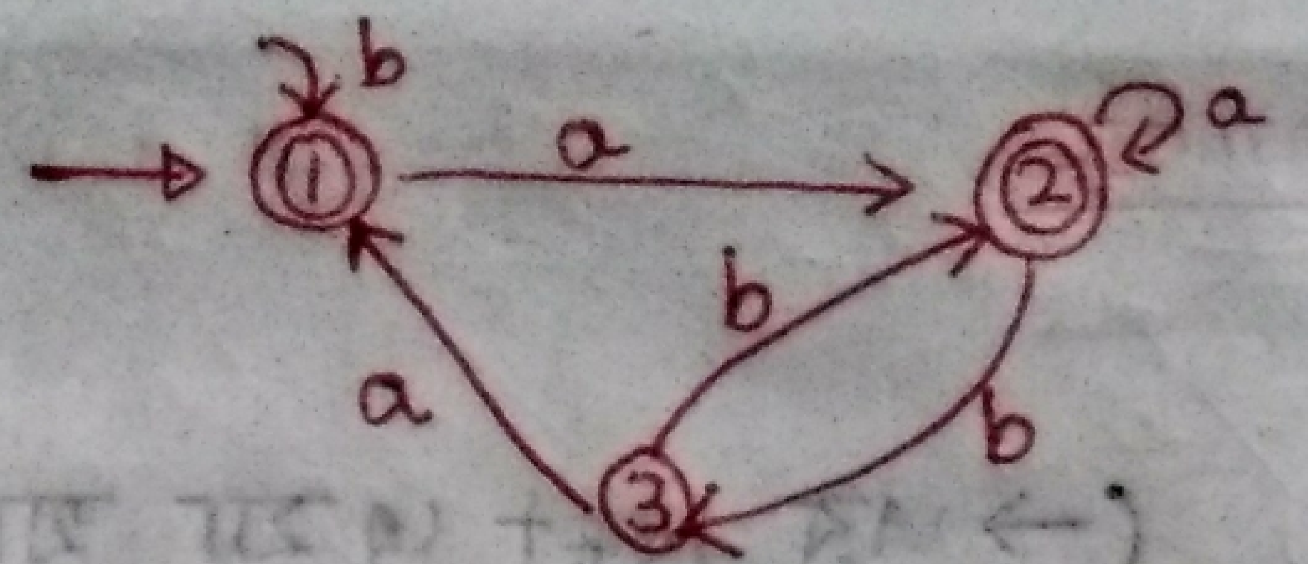
Step-1



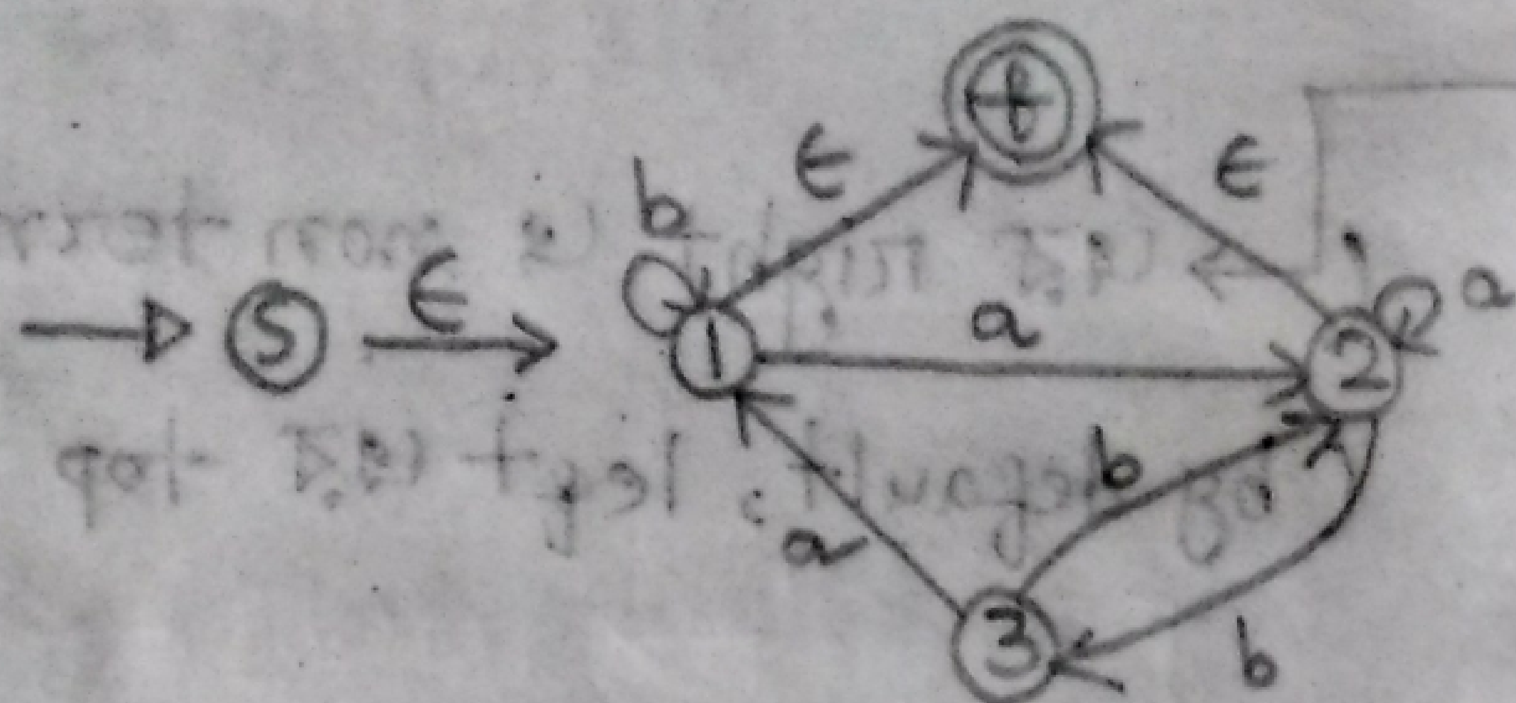
→ 3 is a dead lock

→ no incoming edge for 2

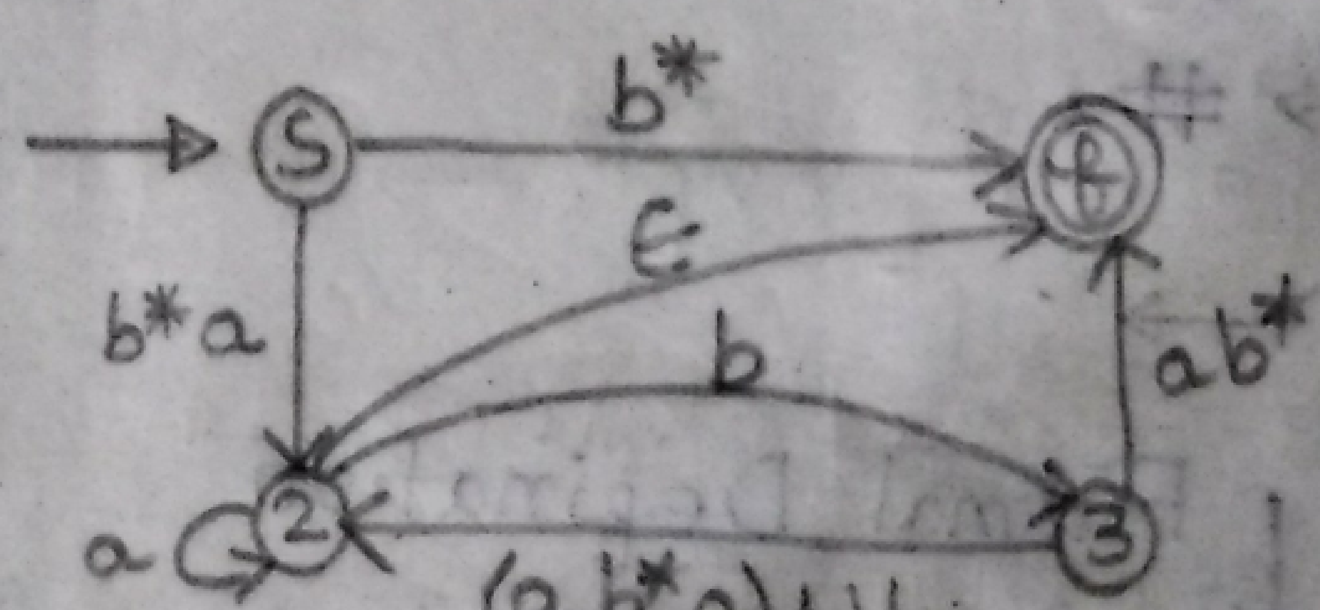
→ after removing 2 no incoming for 4



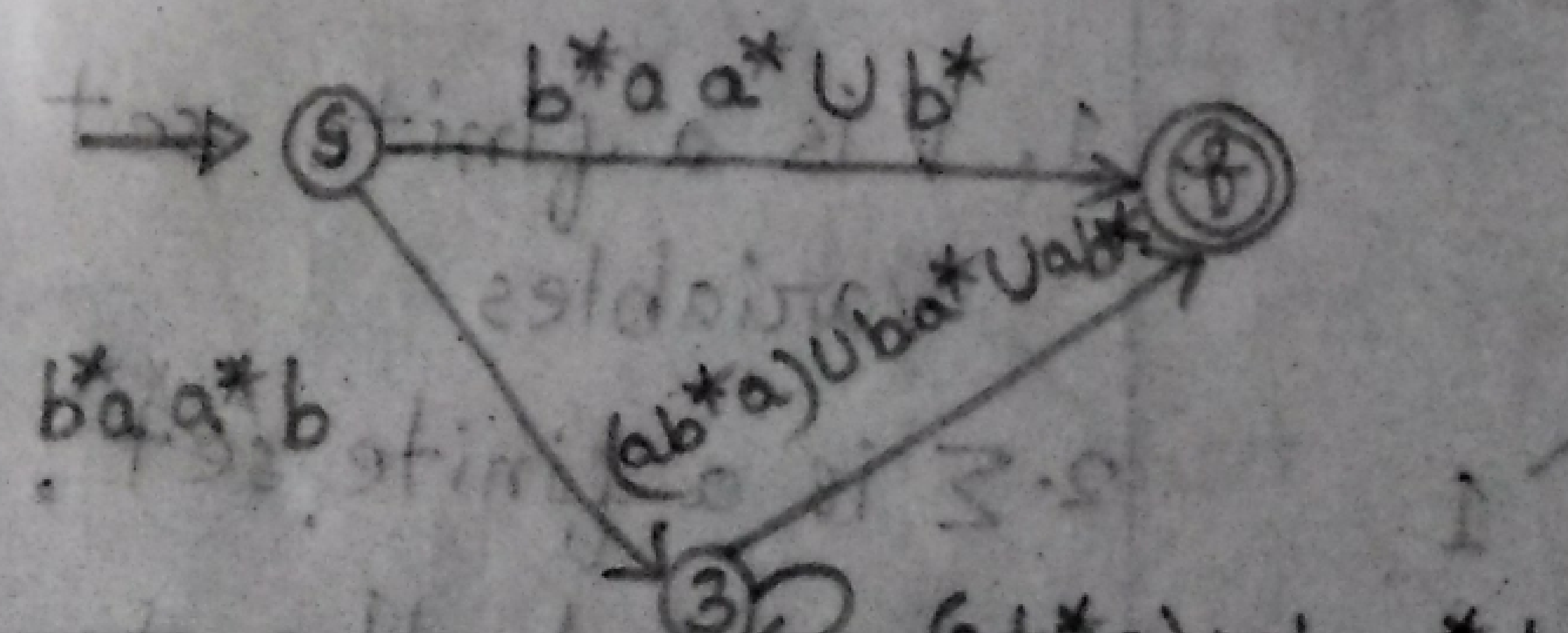
Step-1



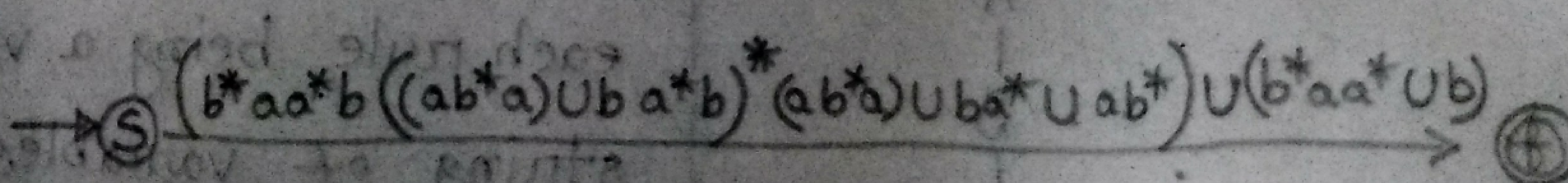
Step-2



Step-3



Step-4



Context Free Grammar

Four Components -

i) Non terminals or variable (→ এর left এ যা থাকবে)

ii) Terminals (→ এর right এ non terminal বাদে সব)

iii) Productions or rules

iv) Start variable (by default, left এর top most variable)

$000\#111$ (LMD)

$A \to 0A1$

$A \to B$

$B \to \#$

$A \to 0A1$

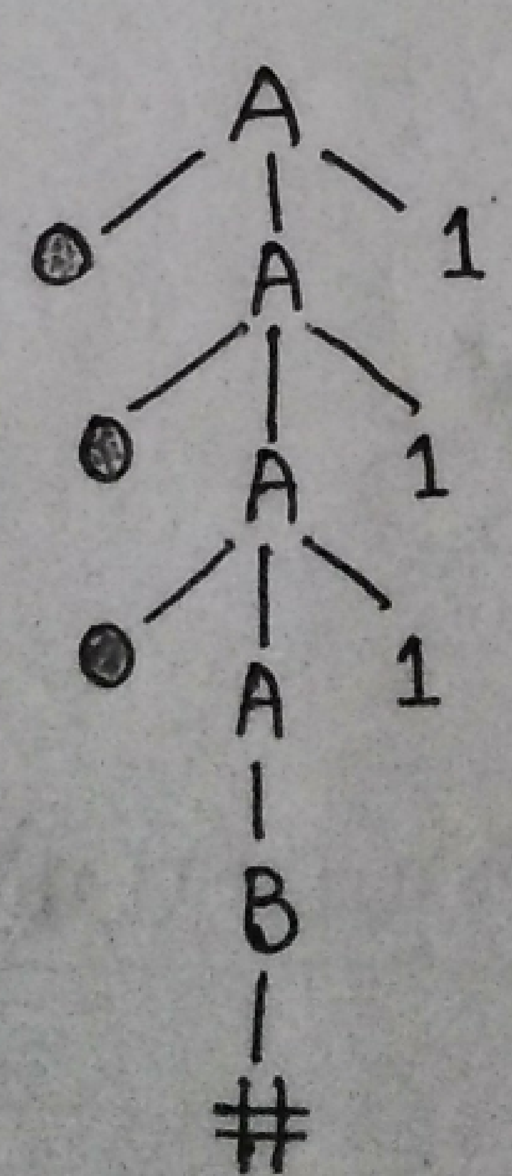
$\to 00A11$ $[A \to 0A1]$

$\to 000A111$ $[A \to 0A1]$

$\to 000B111$ $[A \to B]$

$\to 000\#111$ $[B \to \#]$

Parse tree



Formal Defination

A context free grammar is a tuple (V, Σ, R, S) where

1. V is a finite set called the variables
2. Σ is a finite set, disjoint from V, called the terminals
3. R is a finite set of rules, with each rule being a variable and a string of variables and terminals
4. $S \in V$ is the start variable.

list $\rightarrow$ list + list

list $\rightarrow$ list - list

list $\rightarrow$ digit

digit $\rightarrow$ 0|1|2|3|5

3+2-1

list $\rightarrow$ list + list

$\rightarrow$ digit + list [list $\rightarrow$ digit]

$\rightarrow$ 3+ list [digit $\rightarrow$ 3]

$\rightarrow$ 3+ list - list [list $\rightarrow$ list - list]

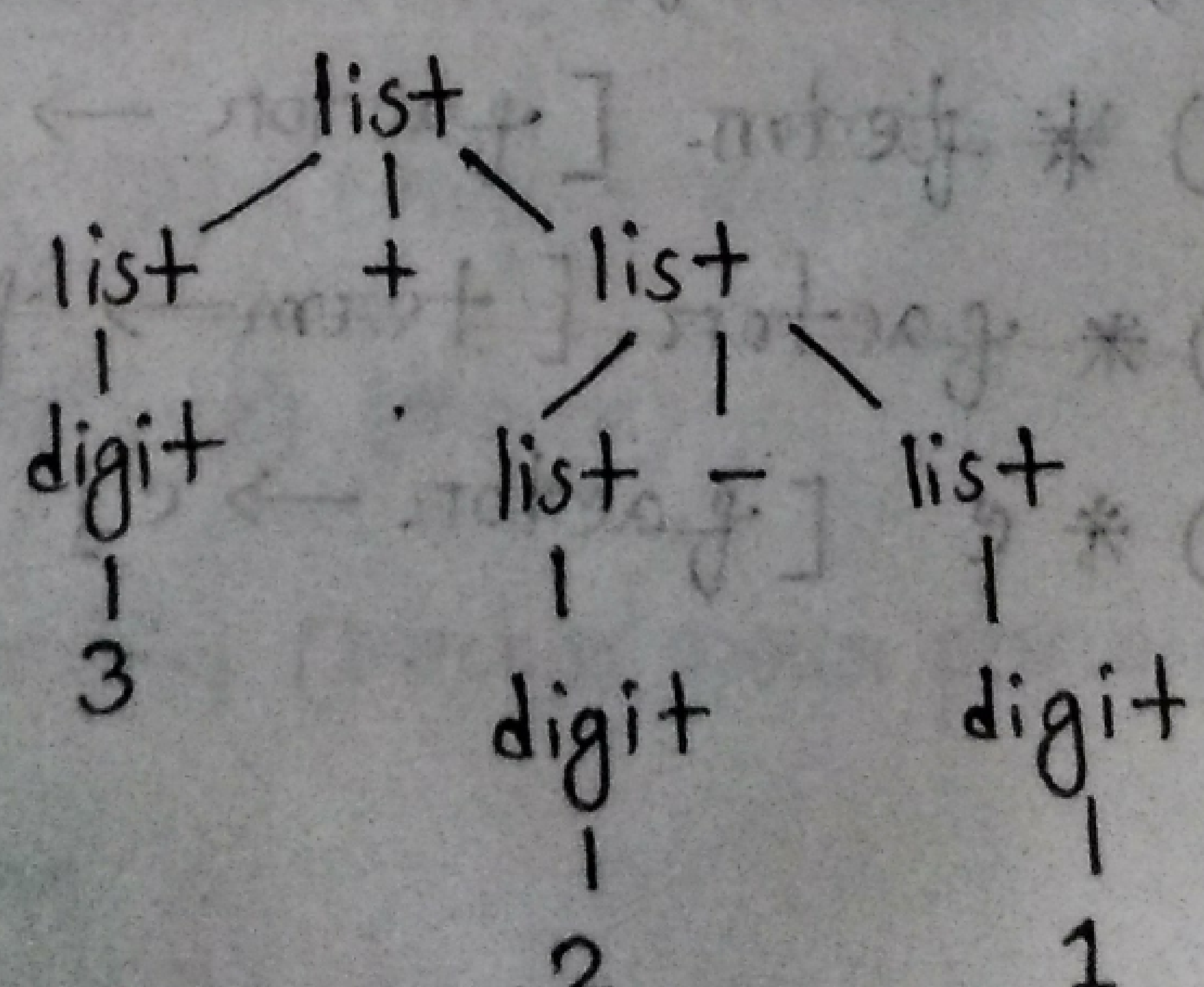
$\rightarrow$ 3+ digit - list [list $\rightarrow$ digit]

$\rightarrow$ 3+2- list [digit $\rightarrow$ 2]

$\rightarrow$ 3+2- digit [lis $\rightarrow$ digit]

$\rightarrow$ 3+2-1 [digit $\rightarrow$ 1]

Parse tree



expr → expr + term | term

term → term * factor | factor

factor → (expr) | a | b | c

(a+b) * c

expr → term

→ term * factor [~~expr~~ term → term * factor]

→ factor * factor [term → factor]

→ (expr) * factor [factor → (expr)]

→ (expr + term) * factor [expr → expr + term]

→ (term + term) * factor [expr → term]

→ (factor + term) * factor [term → factor]

→ (a + term) * factor [factor → a]

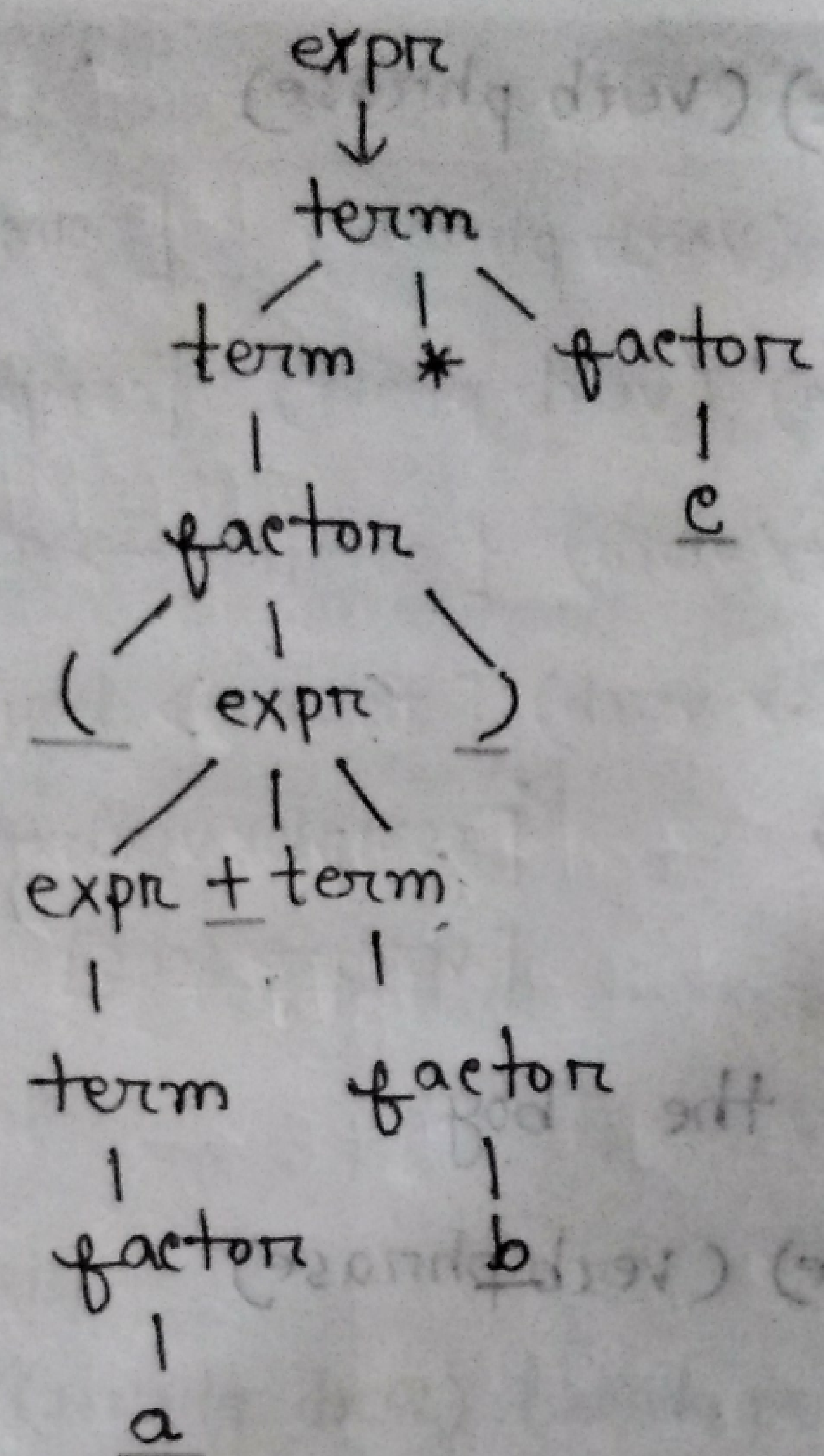
→ (a + factor) * factor [term → factor]

→ (a+b) * factor [factor → b]

→ ~~(a+b) * factor [term → factor]~~

→ (a+b) * c [factor → c]

parse tree



Sentence → (noun phrase) (verb phrase)

Noun phrase → (complex-noun) | (complex-noun) (prep-phrase)

verb phrase → (complex-verb) | (complex-verb) (prep-phrase)

prep phrase → (prep) (compx-noun)

complx noun → (article) (noun)

complx verb → (verb) | (verb) (noun phrase)

article → a | the

noun → boy | girl | flower

verb → touches | likes | sees

prep → with

a boy sees

Sentenee → (noun phrase) (verb phrase)

→ (complx-noun) (verb phrase) [noun phrase → comp noun]

→ (article) (noun) (verb phase) [compl noun → (arti) (noun)]

→ a (noun) (complex verb) [article → a]

→ a boy (complex verb) [noun → boy]

→ a boy (verb) [complex verb → verb]

→ a boy sees [verb → sees]

a girl with a flower likes the boy

Sentenee → (nounphrase) (verb phrase)

→ (complx noun) (Prep phrase) (verb phrase) [NP → (CN) (PP)]

→ (article) (noun) (Prep phrase) (verb phrase) [CN → (a) (n)]

→ a (noun) (Prep phrase) (verb phrase)

→ a girl (prep phrase) (verb phrase)

→ a girl (prep) (complx noun) (verb phrase)

→ a girl with (complx noun) (verb phrase)

→ a girl with (article) (noun) (verb phrase)

→ a girl with a (noun) (verb phrase)

→ a girl with a flower (complex verb)

→ a girl with a flower (verb) (noun phrase)

→ a girl with a flower likes (noun phrase)

→ a girl with a flower likes (complx noun)

→ a girl with a flower likes (article) (noun)

→ a girl with a flower likes the (noun)

→ a girl with a flower likes the boy.

Ambiguity

A string w is derived ambiguity in context free grammar G if it has two or more different left most derivations. Grammar G is ambigious if it generates some string ambigously.

Prove this is ambiguous 3*2/4

list → list * list

list → list / list

list → digit

digit → 0-9

list → list * list

→ ~~list * list * list~~ ~~[list → list * list]~~

→ digit * list [list → digit]

→ 3* list [digit → 3]

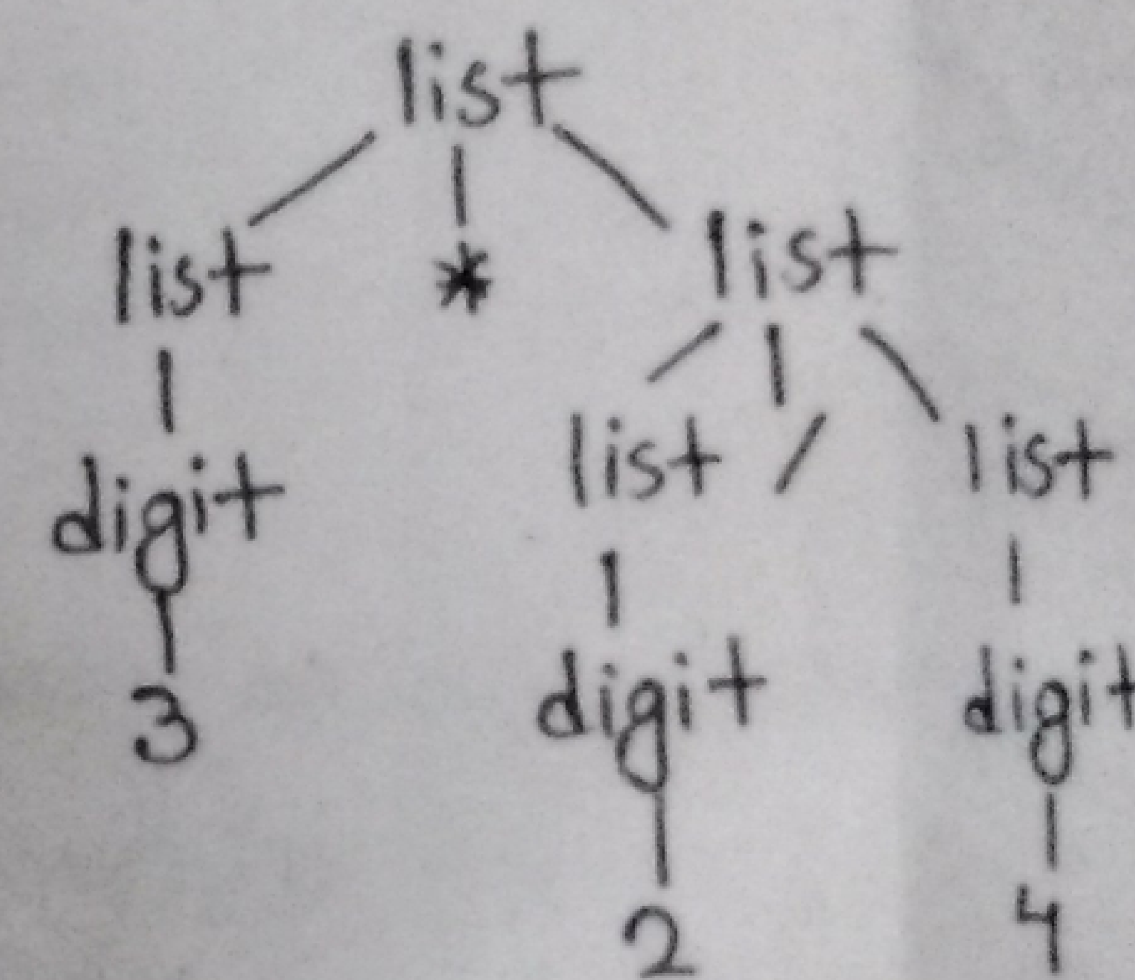
→ 3* list/list [list → list/list]

→ 3* digit/list [list → digit]

→ 3* 2/list [digit → 2]

→ 3*2/ digit [list → digit]

→ 3*2/4 [digit → 4]



list → list/list

→ list * list /list [list → list*list]

→ digit * list / list [list → digit]

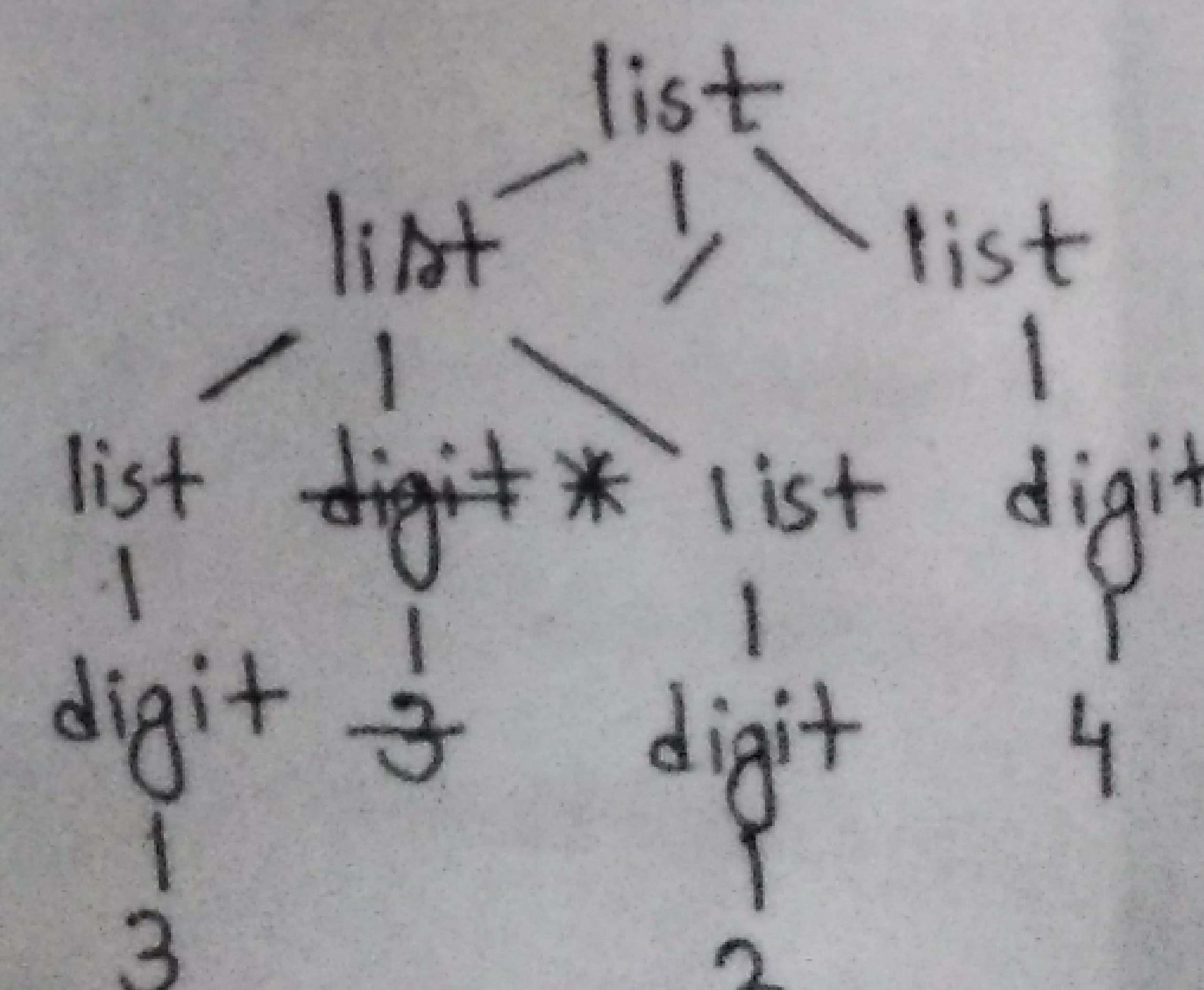
→ 3 * list/ list [digit → 3]

→ 3* digit/ list [list → digit]

→ 3*2/ list [digit → 2]

→ 3*2/ digit [list → digit]

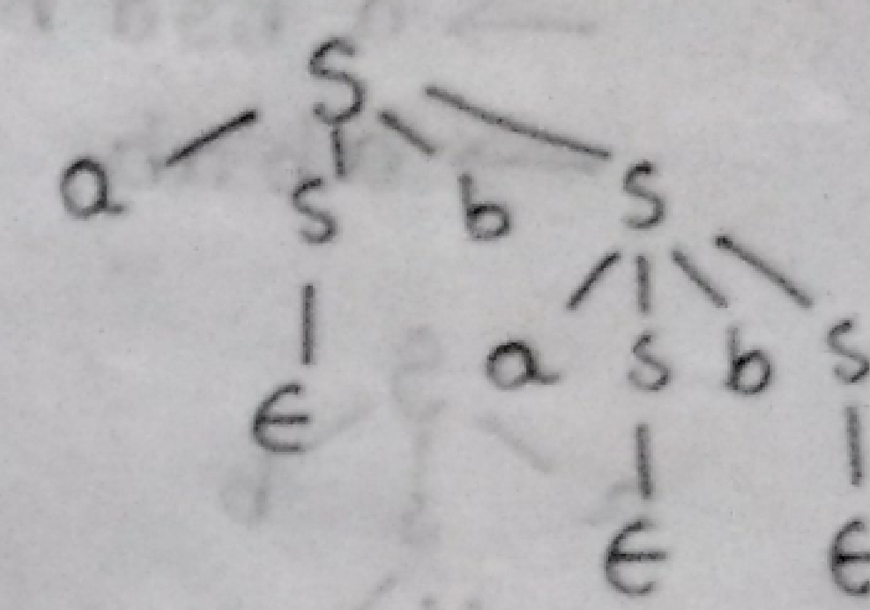
→ 3*2/4 [digit → 4]



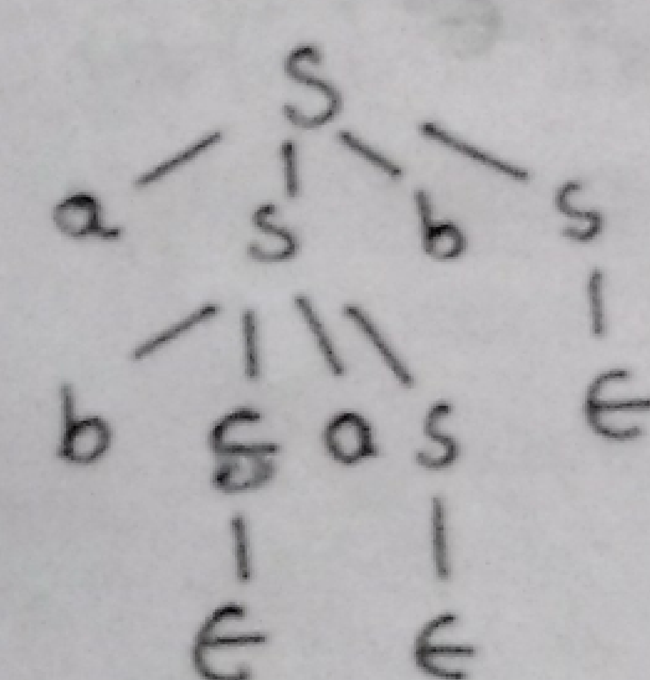
$S \rightarrow aSbS \mid bSaS \mid \epsilon$

input = abab

$S \rightarrow aSbS$
$\rightarrow abS \; [S \rightarrow \epsilon]$
$\rightarrow ab\,aSbS \; [S \rightarrow aSbS]$
$\rightarrow ababS \; [S \rightarrow \epsilon]$
$\rightarrow abab \; [S \rightarrow \epsilon]$



$S \rightarrow aSbS$
$\rightarrow a\; bSaS\; bS \; [S \rightarrow bSaS]$
$\rightarrow abaS\, bS \; [S \rightarrow \epsilon]$
$\rightarrow ababS \; [S \rightarrow \epsilon]$
$\rightarrow abab \; [S \rightarrow \epsilon]$

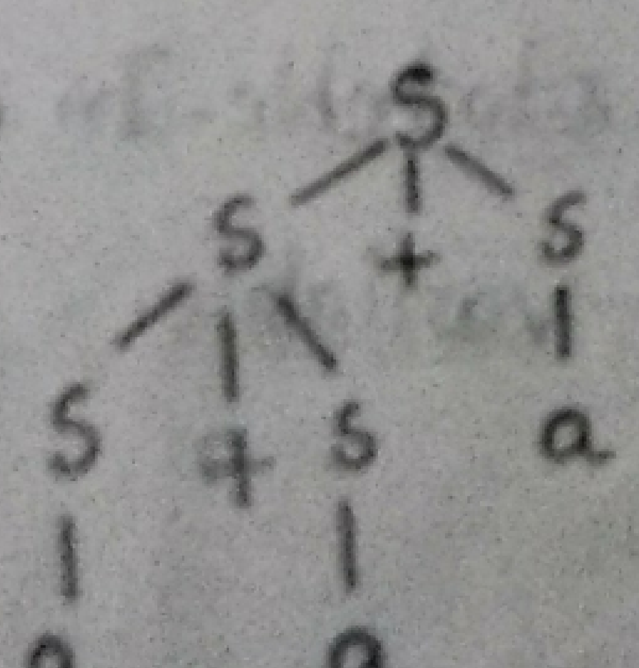


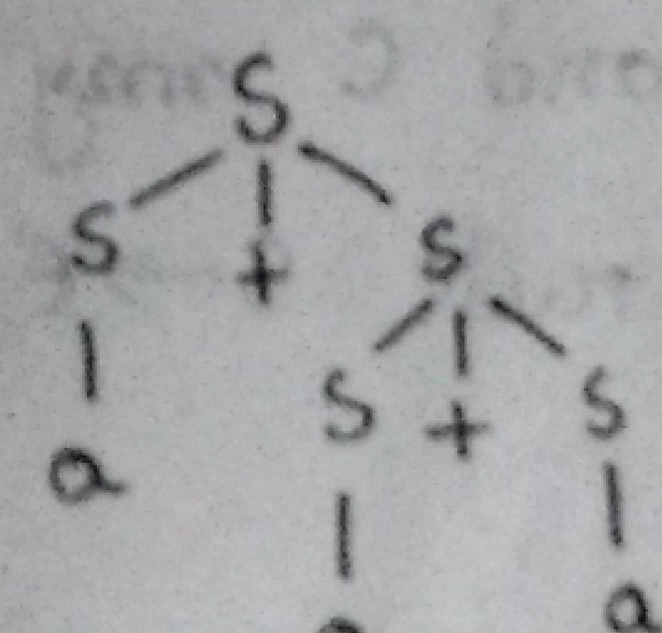
$S \rightarrow S+S \mid (S) \mid a$

a+a+a

$S \rightarrow S+S$
$\rightarrow S+S+S \; [S \rightarrow S+S]$
$\rightarrow a+S+S \; [S \rightarrow a]$
$\rightarrow a+a+S \; [S \rightarrow a]$
$\rightarrow a+a+a \; [S \rightarrow a]$

$S \rightarrow S+S$
$\rightarrow a+S \; [S \rightarrow a]$
$\rightarrow a+S+S \; [S \rightarrow S+S]$
$\rightarrow a+a+S \; [S \rightarrow a]$
$\rightarrow a+a+a \; [S \rightarrow a]$

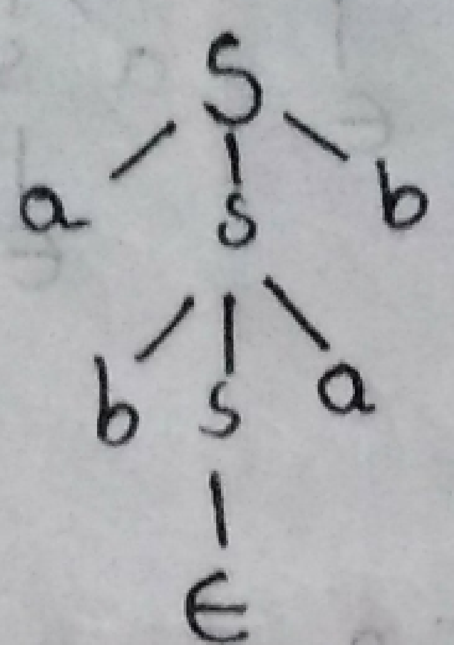




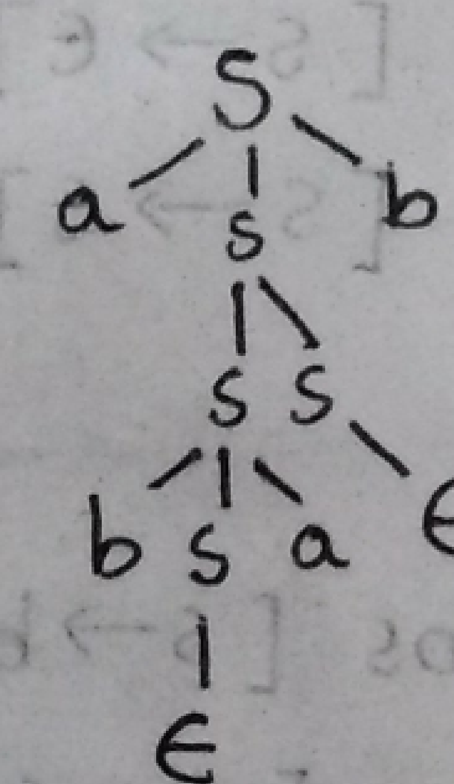
$S \rightarrow aSb \mid bSa \mid SS \mid \epsilon$

abab

$S \rightarrow aSb$
$\rightarrow abSab$ [$S \rightarrow bSa$]
$\rightarrow abab$ [$S \rightarrow \epsilon$]



$S \rightarrow aSb$
$\rightarrow aSSb$ [$S \rightarrow SS$]
$\rightarrow abSaSb$ [$S \rightarrow bSa$]
$\rightarrow abaSb$ [$S \rightarrow \epsilon$]
$\rightarrow abab$ [$S \rightarrow \epsilon$]



Chomsky Normal Form

Formal defination of CNF

A context free grammar is in Chomskey Normal Form if every rule is of the form

$A \rightarrow BC$
$A \rightarrow a$

where a is any terminal and A, B, C are any variable except that B and C may not be the start variable. In addition we permit the rule $S \rightarrow \epsilon$ where S is start variable.

Chomskey Normal Form (CNF)

A context free grammar is in Chomskey normal form if every rule is of the form

$A \rightarrow BC$

$A \rightarrow a$

where a is any terminal and A, B, C are any variables except that B and C may not be the start variable. In addition we permit the rule $S \rightarrow \epsilon$ where S is the start variable.

CFG → CNF

$S \rightarrow ASA \mid aB$

$A \rightarrow B \mid \epsilon$

$B \rightarrow b \mid \epsilon$

Step 1: add a new start variable

$S_0 \rightarrow S$

$S \rightarrow ASA \mid aB$

$A \rightarrow B \mid \epsilon$

$B \rightarrow b \mid \epsilon$

Step 2: removing ϵ rules $A \rightarrow \epsilon$; $B \rightarrow \epsilon$, $A \rightarrow \epsilon$

$S_0 \rightarrow S$

$S \rightarrow ASA \mid aB \mid SA \mid AS \mid S$

$A \rightarrow B$

$B \rightarrow b \mid \epsilon$

$S_0 \to S$

$S \to ASA \mid aB \mid SA \mid AS \mid S \mid a$

$A \to B \mid \epsilon$

$B \to b$

* ϵ remove করার সময় তা start variable এ আসলে remove করবনা।

* right এ শুধু একটি non terminal থাকলে remove (S-3)

$S_0 \to S$

$S \to ASA \mid aB \mid SA \mid AS \mid S \mid a$

$A \to B$

$B \to b$

Step3 : remove unit rules $S \to S$, $S_0 \to S$, $A \to B$

$S_0 \to S$

$S \to ASA \mid aB \mid SA \mid AS \mid a$

$A \to B$

$B \to b$

$S_0 \to ASA \mid aB \mid SA \mid AS \mid a$

$S \to ASA \mid aB \mid SA \mid AS \mid a$

$A \to B$

$B \to b$

$S_0 \to ASA \mid aB \mid SA \mid AS \mid a$

$S \to ASA \mid aB \mid SA \mid AS \mid a$

$A \to b$

$B \to b$

$A \to B$

$B \to \alpha$

$\therefore A \to \alpha$

Step 4: Convert remaining rules into proper form:

$S_0 \rightarrow V_1 A \mid V_2 B \mid SA \mid AS \mid a$

$S \rightarrow V_1 A \mid V_2 B \mid SA \mid AS \mid a$

$A \rightarrow b$

$B \rightarrow b$

$V_1 \rightarrow AS$

$V_2 \rightarrow a$

$S \rightarrow aXbX$

$X \rightarrow aY \mid bY \mid \epsilon$

$Y \rightarrow X \mid d$

Step 1:

$S_0 \rightarrow S$

$S \rightarrow aXbX$

$X \rightarrow aY \mid bY \mid \epsilon$

$Y \rightarrow X \mid d$

Step 2: $X \rightarrow \epsilon$, $Y \rightarrow \epsilon$

$S_0 \rightarrow S$

$S \rightarrow aXbX \mid abX \mid aXb \mid ab$

$X \rightarrow aY \mid bY$

$Y \rightarrow X \mid d \mid \epsilon$

$S_0 \rightarrow S$

$S \rightarrow aXbX \mid abX \mid aXb \mid ab$

$X \rightarrow aY \mid bY \mid a \mid b$

$Y \rightarrow X \mid d$

Step3: $S_0 \to S$, $Y \to X$

$S_0 \to aXbX \mid abX \mid aXb \mid ab$

$S \to aXbX \mid abX \mid aXb \mid ab$

$X \to aY \mid bY \mid a \mid b$

$Y \to X \mid d$

$S_0 \to aXbX \mid abX \mid aXb \mid ab$

$S \to aXbX \mid abX \mid aXb \mid ab$

$X \to aY \mid bY \mid a \mid b$

$Y \to aY \mid bY \mid a \mid b \mid d$

Step4:

$S_0 \to V_1V_2 \mid V_3X \mid V_1V_4 \mid V_5V_4$

$V_1 \to aX$

$V_2 \to bX$

$V_3 \to ab$

$V_4 \to b$

$V_5 \to a$

$S \to V_1V_2 \mid V_3X \mid V_1V_4 \mid V_5V_4$

$X \to V_5Y \mid V_4Y \mid a \mid b$

$Y \to V_5Y \mid V_4Y \mid a \mid b \mid d$

$S \rightarrow bS \mid aT \mid \epsilon$

$T \rightarrow aT \mid bR \mid \epsilon$

$R \rightarrow bS \mid \epsilon$

Step 1 :

$S_0 \rightarrow S$

$S \rightarrow bS \mid aT \mid \epsilon$

$T \rightarrow aT \mid bR \mid \epsilon$

$R \rightarrow bS \mid \epsilon$

Step 2 : $S \rightarrow \epsilon,\ T \rightarrow \epsilon,\ R \rightarrow \epsilon$

$S_0 \rightarrow S \mid \epsilon$

$S \rightarrow bS \mid aT \mid b$

$T \rightarrow aT \mid bR \mid \epsilon$

$R \rightarrow bS \mid \epsilon \mid b$

$S_0 \rightarrow S \mid \epsilon$

$S \rightarrow bS \mid aT \mid b \mid a$

$T \rightarrow aT \mid bR \mid a$

$R \rightarrow bS \mid \epsilon \mid b$

$S_0 \rightarrow S \mid \epsilon$

$S \rightarrow bS \mid aT \mid b \mid a$

$T \rightarrow aT \mid bR \mid \cancel{a} \mid b$

$R \rightarrow bS \mid b$

Step 3 : $S_0 \rightarrow S$

$S_0 \rightarrow bS \mid aT \mid b \mid a \mid \epsilon$

$S \rightarrow bS \mid aT \mid b \mid a$

$T \rightarrow aT \mid bR \mid a \mid b$

$R \rightarrow bS \mid b$

Step 4 : $S_0 \rightarrow V_1 S \mid V_2 T \mid b \mid a \mid \epsilon$

$V_1 \rightarrow b$

$V_2 \rightarrow a$

$S \rightarrow V_1 S \mid V_2 T \mid b \mid a$

$T \rightarrow V_2 T \mid V_1 R \mid a \mid b$

$R \rightarrow V_1 S \mid b$

Designing a Context Free Grammar

[any num of 0/1 replaced by a non Terminal]

* $L = \{w / w \text{ contains at least two 1's}\}$

$S \rightarrow T1T1T$

$T \rightarrow 0T \mid 1T \mid \epsilon$

* contains exactly two 1's

$S \rightarrow T1T1T$

$T \rightarrow 0T \mid \epsilon$

* atmost two 1's

$S \rightarrow T1T \mid T1T1T \mid T$

$T \rightarrow 0T \mid \epsilon$

* Starts and ends with same symbol

$S \rightarrow 0T0 \mid 1T1 \mid 0 \mid 1$

$T \rightarrow 0T \mid 1T \mid \epsilon$

* each a is followed by at least One b

$S \rightarrow TabTB \mid T$

$T \rightarrow bT \mid \epsilon$

$B \rightarrow abTB \mid \epsilon$

* begins with a and ends with b

$S \rightarrow aTb$

$T \rightarrow aT \mid bT \mid \epsilon$

* begins with a or ends with b

$S \rightarrow aT \mid Tb$

$T \rightarrow aT \mid bT \mid \epsilon$

* odd number of a and ends with b

* starts and ends with same symbol

$S \rightarrow aTa \mid bTb$

$T \rightarrow aT \mid bT \mid \epsilon$

* length of w is odd

$S \rightarrow aT \mid bT$

$T \rightarrow aaT \mid bbT \mid abT \mid baT \mid \epsilon$

* length of w is odd and middle symbol is a

$S \rightarrow a \mid aSa \mid bSa \mid aSb \mid bSa$

* w is a palindrome number

$S \rightarrow a \mid b \mid aSa \mid bSb \mid \epsilon$

* contains even num of a

$S \rightarrow BT \mid \epsilon$

$B \rightarrow bB \mid \epsilon$

$T \rightarrow aBaBT \mid \epsilon$

* contains aab as substring

$S \rightarrow TaabT$

$T \rightarrow aT \mid bT \mid \epsilon$

* every a is followed by exactly 2 b

$S \rightarrow TabbB$

$T \rightarrow bT \mid \epsilon$

$B \rightarrow abbB \mid \epsilon$

* string length atmost two

$S \rightarrow aT \mid bT \mid \epsilon$

$T \rightarrow a \mid b \mid \epsilon$

* contains a single a

$S \rightarrow BaB$

$B \rightarrow bB \mid \epsilon$

* Start with a and has odd length

$S \rightarrow aT$

$T \rightarrow abT \mid baT \mid aaT \mid bbT \mid \epsilon$

Push Down Automata (PDA)

→ It is used to identify non regular language.

→ A, B → C

A ↓ input symbol

B ↓ Stack top symbol read

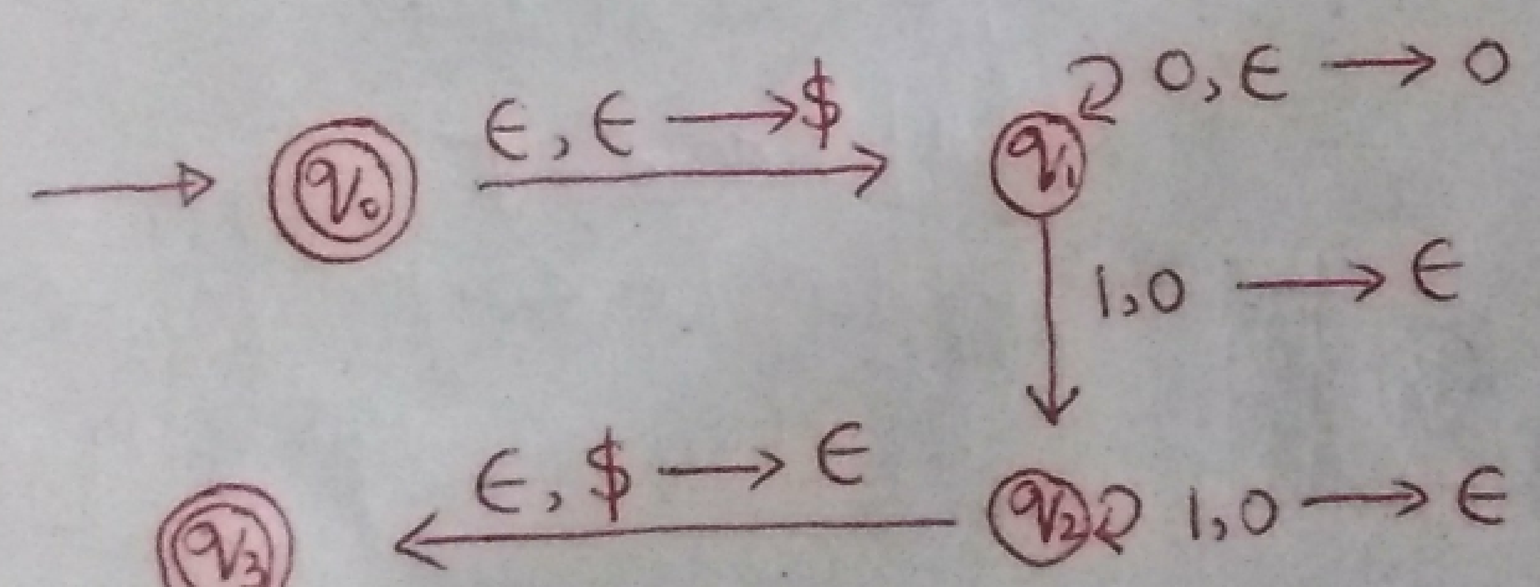
C → insert symbol on stack

→ write → push

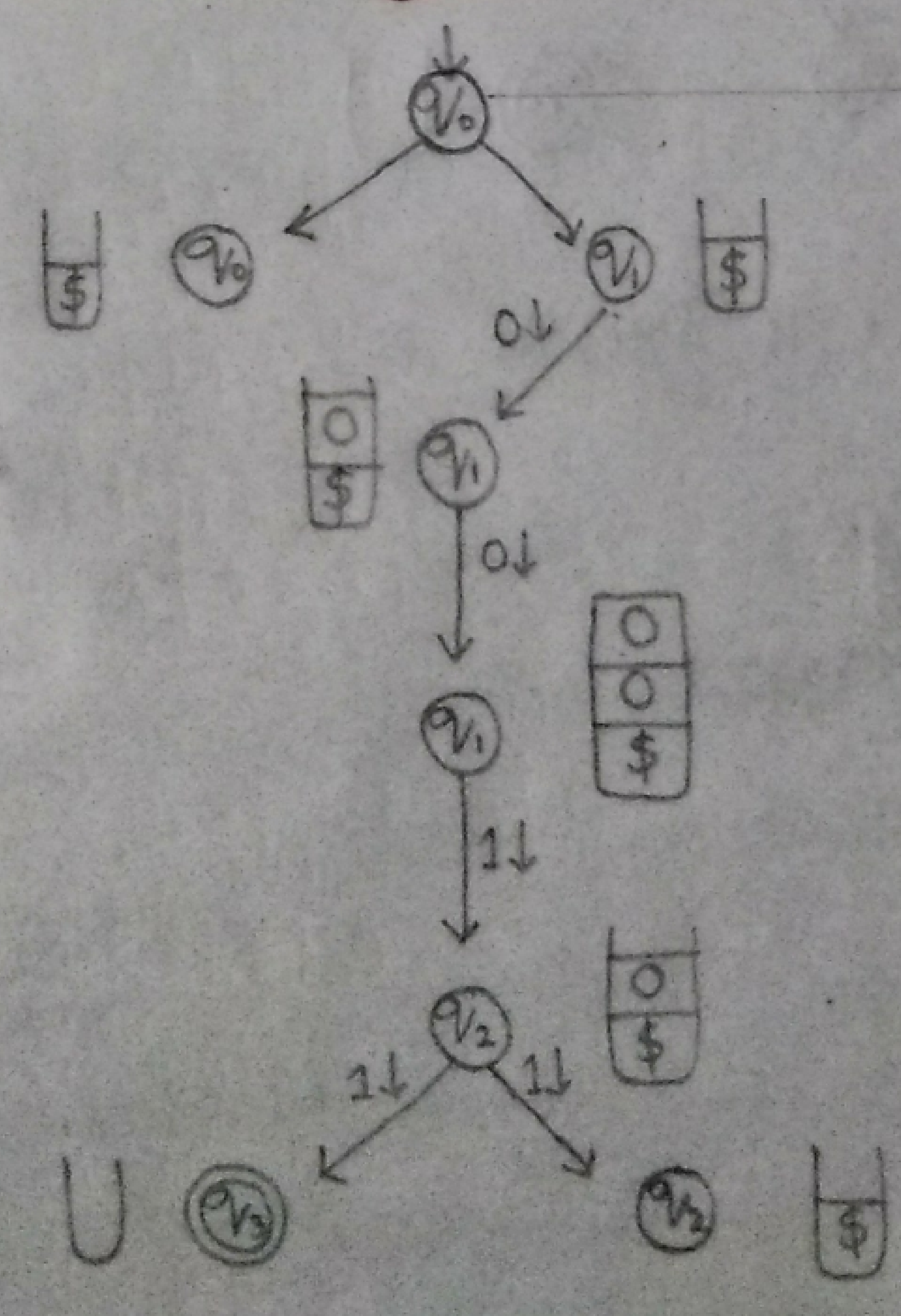
read → pop

→ final state এ যেয়ে stack empty হতে হবে। তাহলে PDA → accep

*



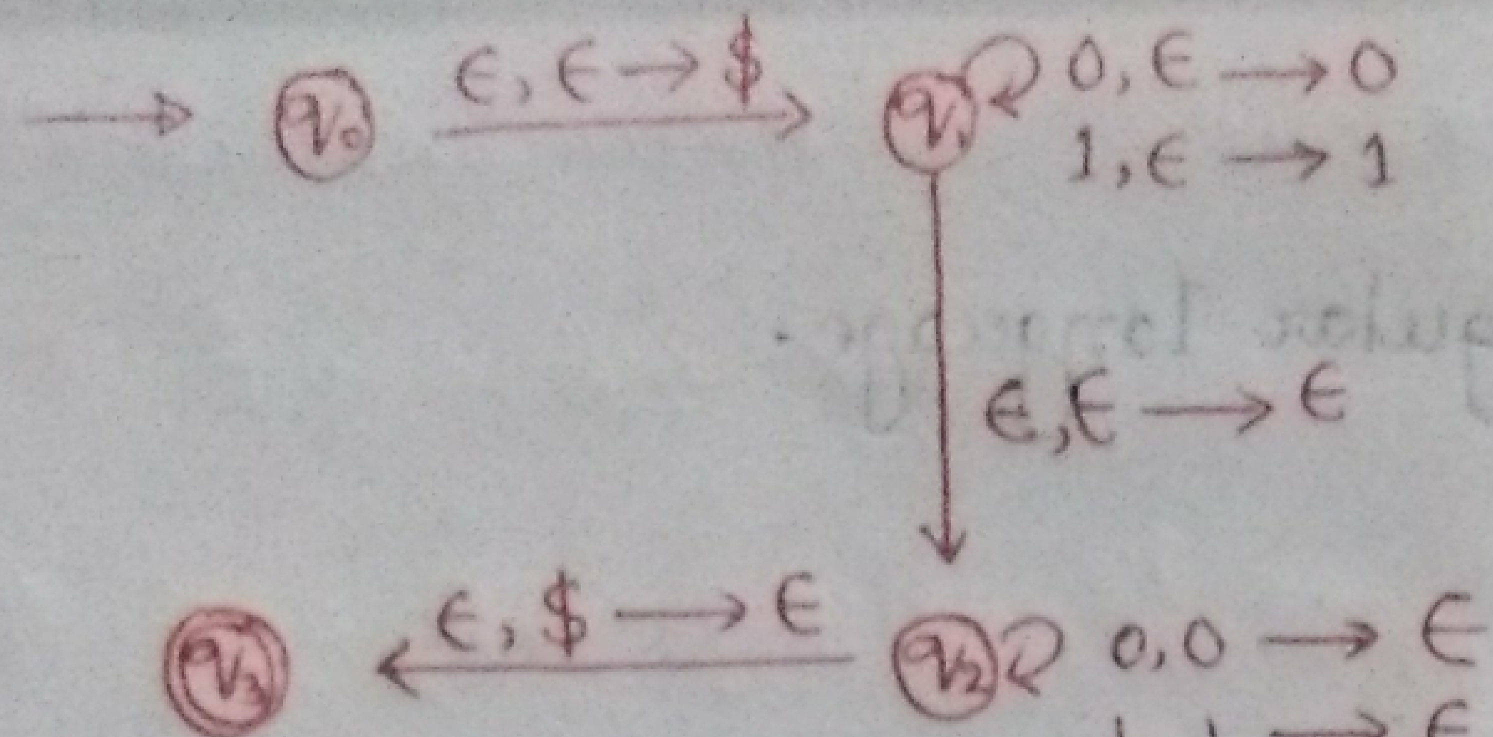
input → 0011



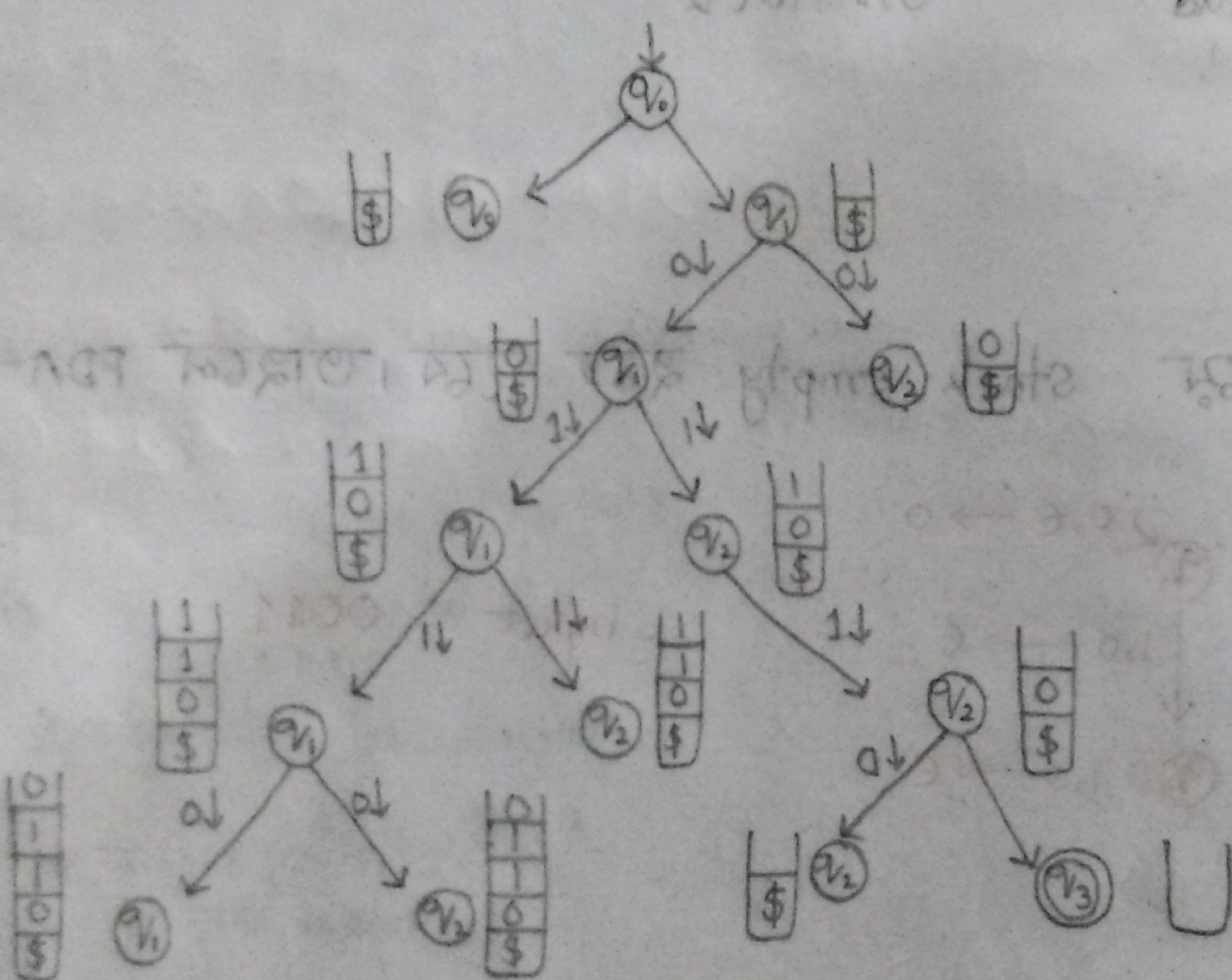
কোন কিছু read না করে q_0 আর ε read করে q_1

Stack empty. The input is accepted.

*

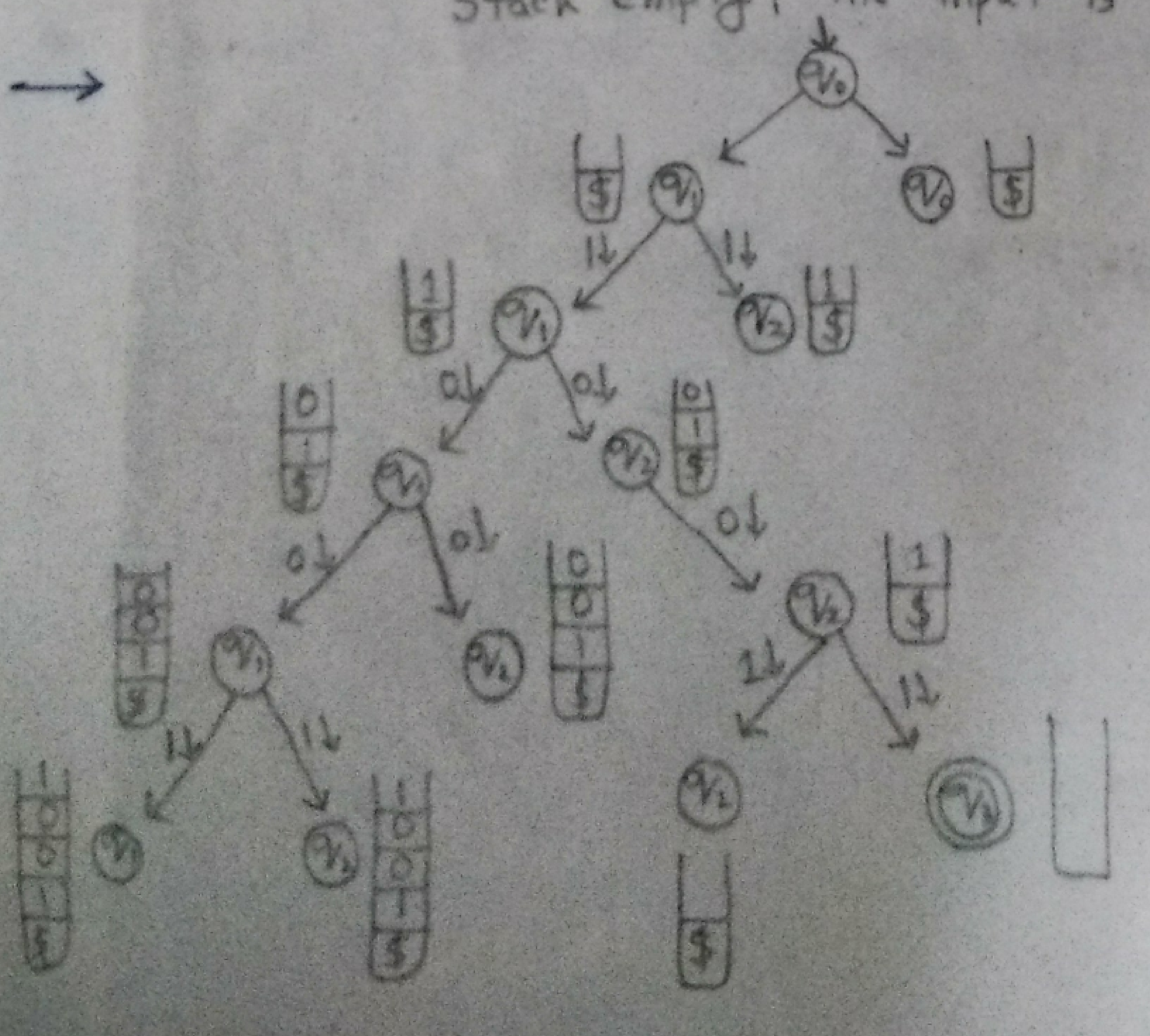


input 0110



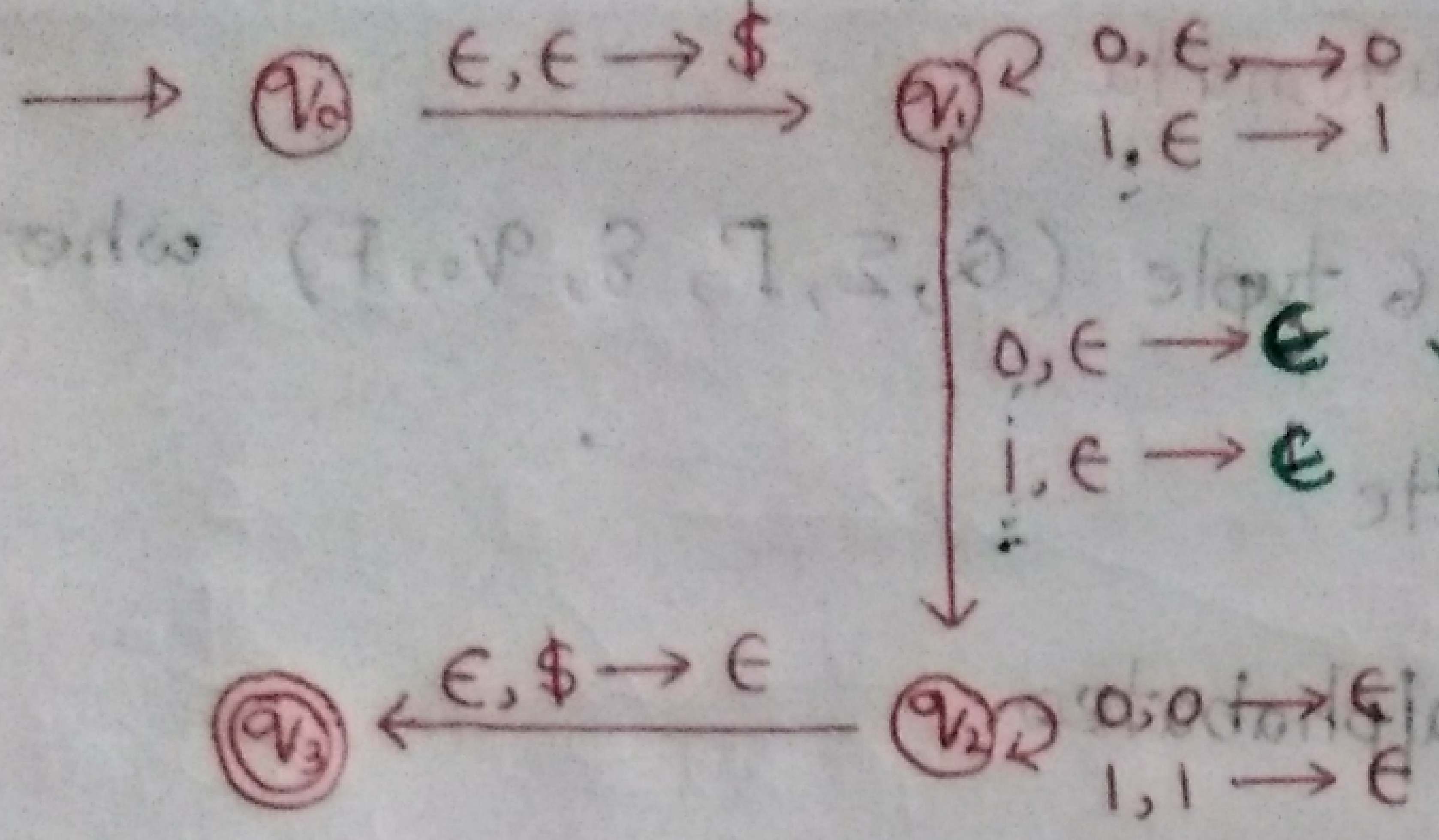
Stack empty. The input is accepted.

input 1001

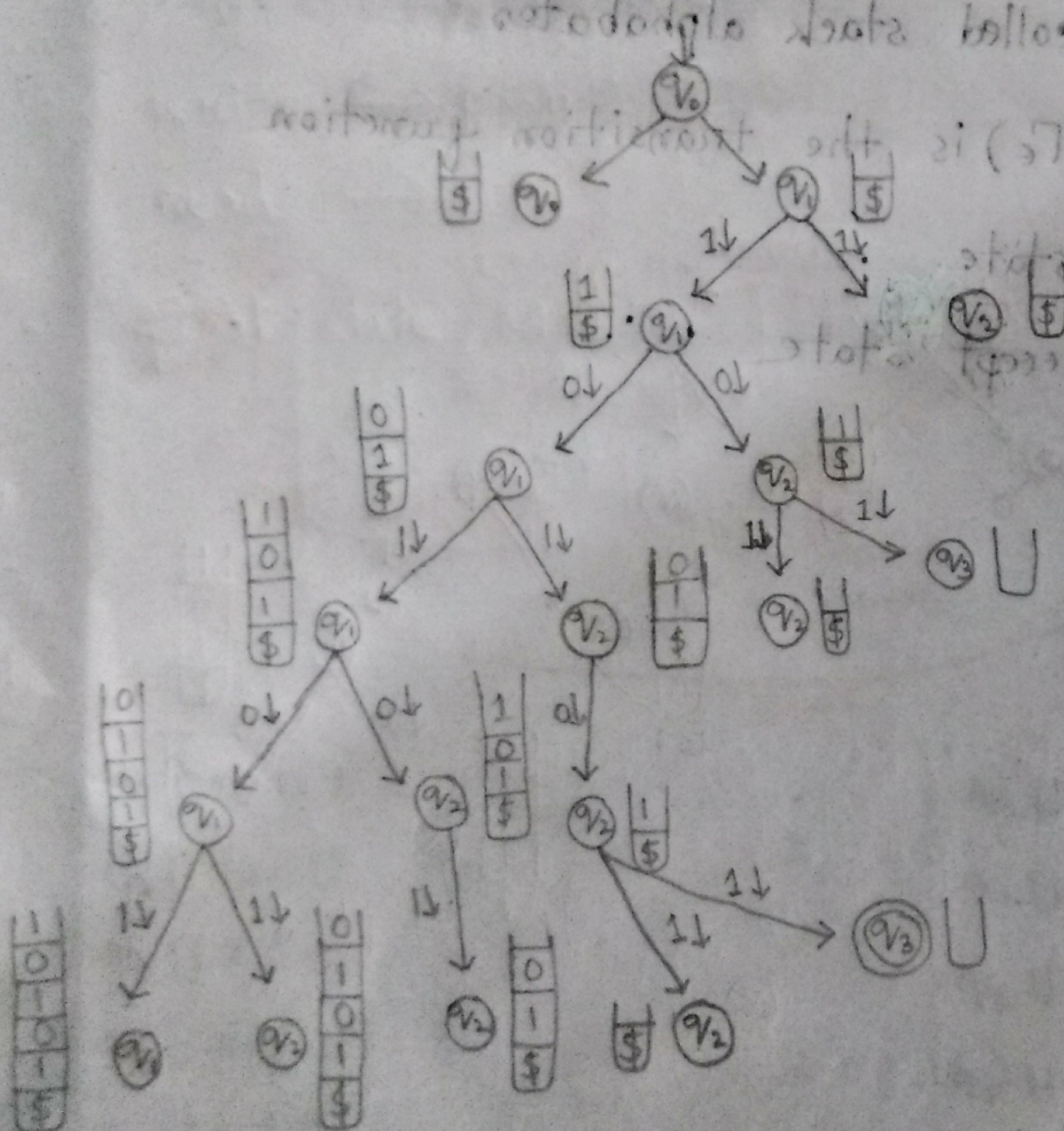


Stack empty, input accepted.

*



input 10101



accepted

Formal Defination of Push down Automata

A pushdown automata is a 6 tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$ where

i) Q is the finite set of state

ii) Σ is the finite set called alphabates

iii) Γ is the finite set called stack alphabates

iv) $\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \longrightarrow P(Q \times \Gamma_\epsilon)$ is the transition function

v) $q_0 \in Q$ is the start state

vi) $F \subseteq Q$ is the set of accept state

Turing Machine

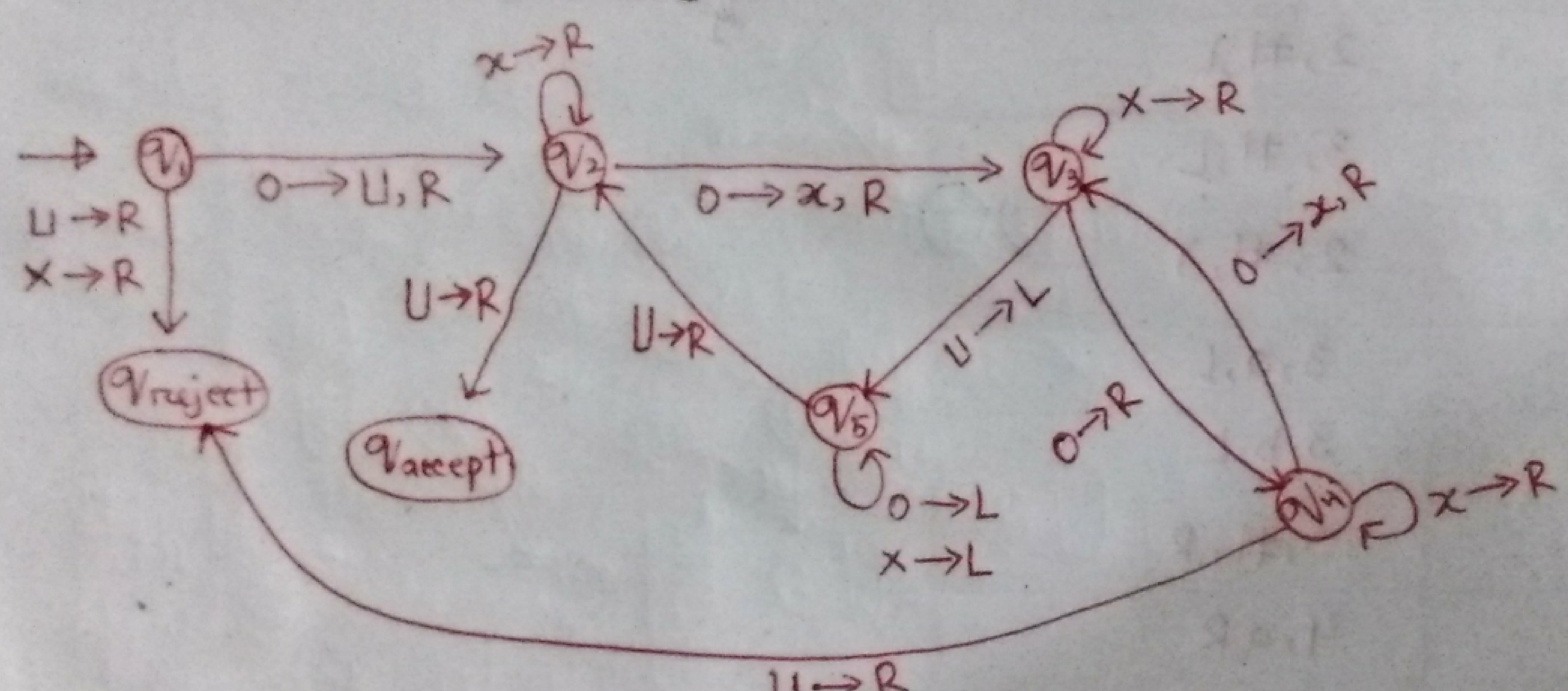
* Transition function

$$A \longrightarrow B, C$$

$A \rightarrow$ input symbol

$B \rightarrow$ replace symbol

$C \rightarrow$ move symbol



For input 0000

$q_1\ \overset{\downarrow}{0}000\sqcup$

$q_2\ \sqcup\overset{\downarrow}{0}00\sqcup$

$q_3\ \sqcup x\overset{\downarrow}{0}0\sqcup$

$q_4\ \sqcup x0\overset{\downarrow}{0}\sqcup$

$q_3\ \sqcup x0x\overset{\downarrow}{\sqcup}$

$q_5\ \sqcup x0\overset{\downarrow}{x}\sqcup$

$q_5\ \sqcup x\overset{\downarrow}{0}x\sqcup$

$q_5\ \sqcup\overset{\downarrow}{x}0x\sqcup$

$q_5\ \overset{\downarrow}{\sqcup}x0x\sqcup$

$q_2\ \sqcup\overset{\downarrow}{x}0x\sqcup$

$q_2\ \sqcup x\overset{\downarrow}{0}x\sqcup$

$q_3\ \sqcup xx\overset{\downarrow}{x}\sqcup$

$q_3\ \sqcup xxx\overset{\downarrow}{\sqcup}$

$q_5\ \sqcup xx\overset{\downarrow}{x}\sqcup$

$q_5\ \sqcup x\overset{\downarrow}{x}x\sqcup$

$q_5\ \sqcup\overset{\downarrow}{x}xx\sqcup$

$q_6\ \overset{\downarrow}{\sqcup}xxx\sqcup$

$q_2\ \sqcup\overset{\downarrow}{x}xx\sqcup$

$q_2\ \sqcup x\overset{\downarrow}{x}x\sqcup$

$q_2\ \sqcup xx\overset{\downarrow}{x}\sqcup$

$q_2\ \sqcup xxx\overset{\downarrow}{\sqcup}$

$q_{accept}\ \sqcup xxx\sqcup$

*

State	Symbol	δ(State, Symbol)
0	a	1, #, R
0	b	4, #, R
0	#	0, #, Y
1	a	1, a, R
1	b	1, b, R
1	#	2, #, L
2	a	3, #, L
2	#	2, #, Y
3	a	3, a, L
3	b	3, b, L
3	#	0, #, R
4	a	4, a, R
4	b	4, b, R
4	#	5, #, L
5	b	3, #, L
5	#	5, #, Y

→ Y is accept state

→ input baab#

→ 0 start state

0 baab#
4 # aab#
4 # aab#
4 # aab #
4 # aab #
5 # aab #
3 # aa # #
3 # aa ##
3 # aa ##

0 # aa # #
1 # # a # #
1 # # a # #
2 # # a # #
3 # # # # #
0 # # # # #
accepted