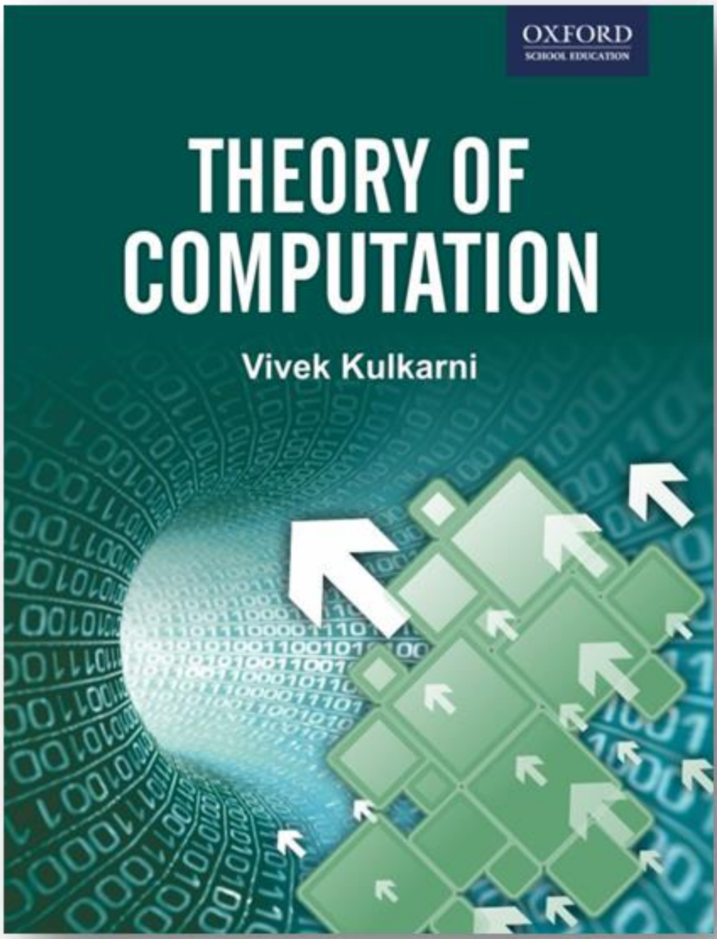


Author: Vivek Kulkarni
(vivek_kulkarni@yahoo.com)

Solution for

Model Question Paper 1

(From Appendix B)

Q.1 a) Let, S = {1, 2, 3, 4} and R = {(1, 2), (2, 3), (1, 4), (4, 2), (3, 4)}.

The transitive closure of a relation R, which is denoted by R^+, is defined as follows:

If $(a, b) \in R$, then (a, b) is in R^+

If $(a, b) \in R^+$ and $(b, c) \in R^+$, then (a, c) is in R^+

Hence, for given set S and relation R,

R^+ = {(1, 2), (2, 3), (1, 4), (4, 2), (3, 4), (1, 3), (2, 4), (4, 3), (3, 2)}

Reflexive and transitive closure of a relation R, which is denoted by R^*, is defined as follows:

$R^* = R^+ \cup \{(a, a) \mid \forall a \in S\}$, where R is a relation defined over set S.

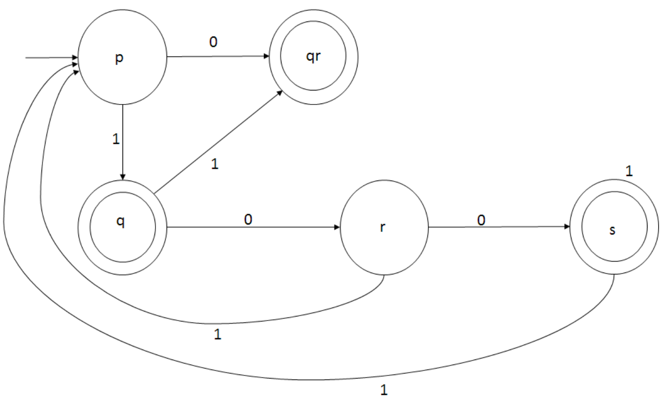
Hence, for given set S and relation R,

R^* = {(1, 2), (2, 3), (1, 4), (4, 2), (3, 4), (1, 1), (2, 2), (3, 3), (4, 4)}

Q.1 b) Refer to the example 2.20 from the book.

Q.1 c) The transitions from states p, q, r and s can be easily obtained from table.

δ'(p, 0) = qr,	δ'(p, 1) = q,
δ'(q, 0) = r,	δ'(q, 1) = qr,
δ'(r, 0) = s,	δ'(r, 1) = p,
δ'(s, 0) = ϕ,	δ'(s, 1) = p,



Find transition for new state qr

δ'(qr, 0) = δ (q, 0) ∪ δ (r, 0)

= {r} ∪ {s}

= rs (new state)

δ'(qr, 1) = δ (q, 1) ∪ δ (r, 1)

= {qr} ∪ {p}

= pqr (new state)

δ'(rs, 0) = δ (r, 0) ∪ δ (s, 0)

$= \{s\} \cup \{\phi\}$

$= s$ (existing state)

$\delta'(rs, 1) = \delta(r, 1) \cup \delta(s, 1)$

$= \{p\} \cup \{p\}$

$= p$ (existing state)

$\delta'(pqr, 0) = \delta(p, 0) \cup \delta(q, 0) \cup \delta(r, 0)$

$= \{qr\} \cup \{r\} \cup \{s\}$

$= qrs$ (new state)

$\delta'(pqr, 1) = \delta(p, 1) \cup \delta(q, 1) \cup \delta(r, 1)$

$= \{q\} \cup \{qr\} \cup \{p\}$

$= pqr$ (existing state)

$\delta'(qrs, 0) = \delta(q, 0) \cup \delta(r, 0) \cup \delta(s, 0)$

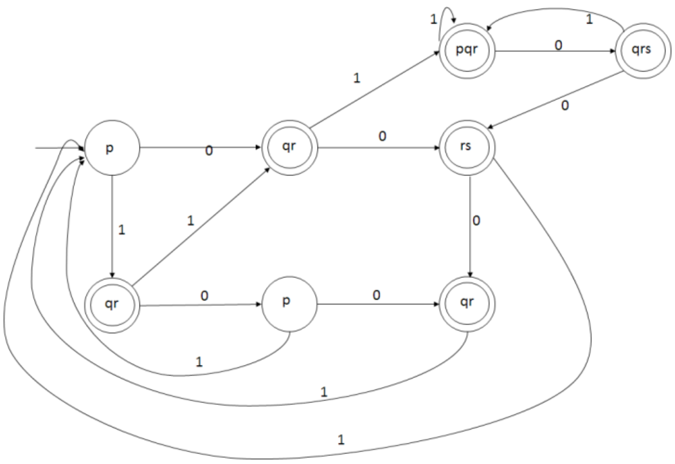
$= \{r\} \cup \{s\} \cup \{\phi\}$

$= rs$ (existing state)

$\delta'(qrs, 1) = \delta(q, 1) \cup \delta(r, 1) \cup \delta(s, 1)$

$= \{qr\} \cup \{p\} \cup \{p\}$

$= pqr$ (existing state)



Σ / Q	0	1
p	qr	q
* q	r	qr
r	s	p
* s	-	p

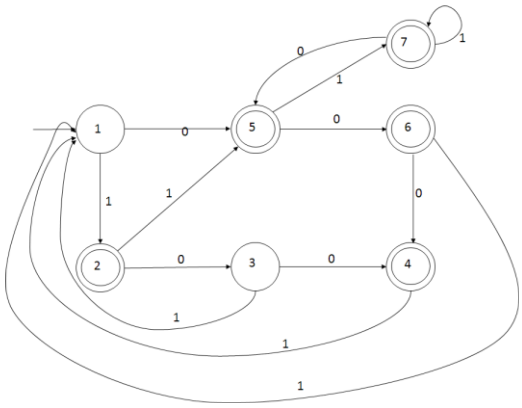
* qr	rs	pqr
* rs	s	p
* pqr	qrs	pqr
* qrs	rs	pqr

We can see ‘r’ and ‘rs’ are equivalent but ‘r’ is non-final state and ‘rs’ is final state, therefore we cannot replace ‘rs’ by ‘r’.

But, we can replace ‘qrs’ by ‘qr’ as these are equivalent and both are final states. Replacing ‘qrs’ by ‘qr’, the modified state table will look as seen below.

Q \ Σ	0	1
p	qr	q
* q	r	qr
r	s	p
* s	-	p
* qr	rs	pqr
* rs	s	p
* pqr	qrs	pqr

We can now draw the TG which is the final equivalent DFA.



You may also refer to the example 2.10 from the book.

Q.2 a) To perform the induction step you proceed as follows:

Assume that $k^4 - 4k^2$ is divisible by 3.

Then consider $(k+1)^4 - 4(k+1)^2$ and expand it.

$$(k+1)^4 - 4(k+1)^2$$
$$= k^4 + 4k^3 + 6k^2 + 4k + 1 - 4(k^2 + 2k + 1)$$
$$= k^4 + 4k^3 + 6k^2 + 4k + 1 - 4k^2 - 8k - 4$$

We assumed that $k^4 - 4k^2$ is divisible by 3, so subtract that out, leaving:

$$4k^3 + 6k^2 + 4k + 1 - 8k - 4$$
$$= 4k^3 + 6k^2 - 4k - 3$$

$6k^2$ and -3 are divisible by 3, so leaving those out too, we get,

$$4k^3 - 4k$$
$$= 4k(k^2 - 1)$$

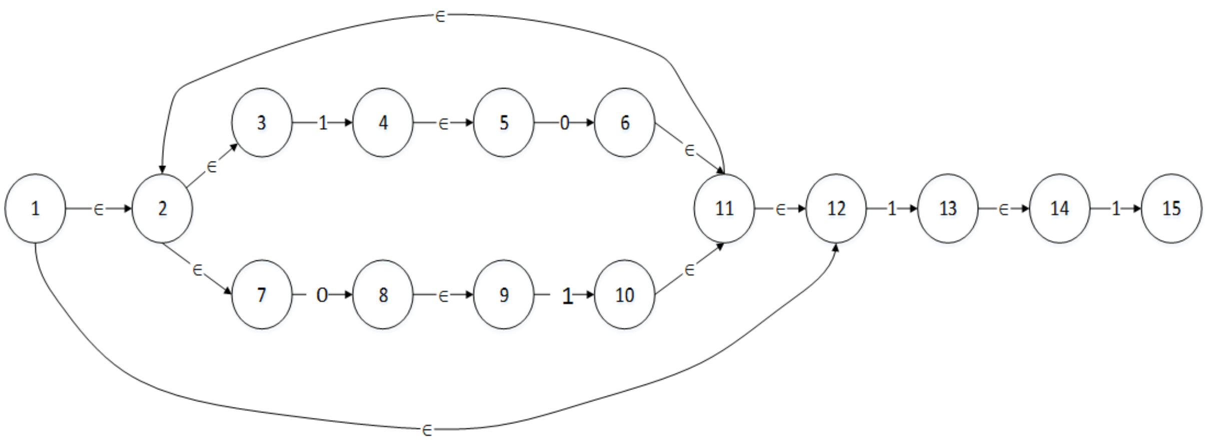
If k mod 3 = 0, 4k is divisible by 3
If k mod 3 = 1, $k^2 - 1$ is divisible by 3
If k mod 3 = 2, $k^2 - 1$ is divisible by 3

So all the parts of $(k+1)^4 - 4(k+1)^2$ are divisible by 3.

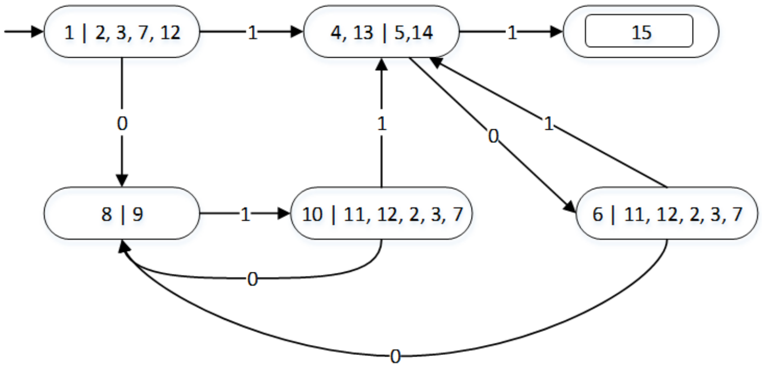
Therefore the whole thing is divisible by 3, if $k^4 - 4k^2$ is divisible by 3.

Q.2 **b)** Refer to the example 3.44 from the book.

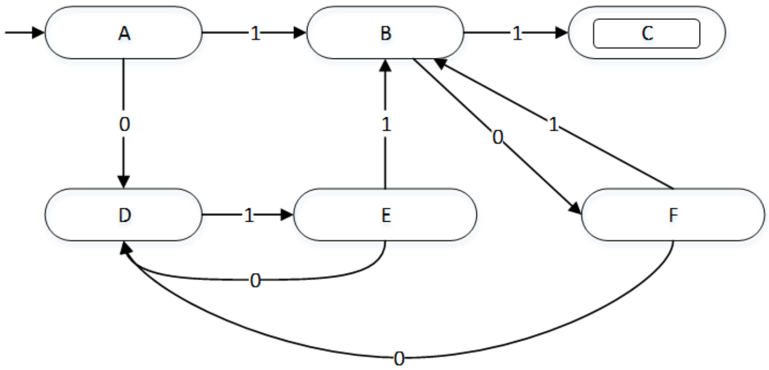
Q.2 **c)** NFA with ∈- transitions can be drawn as below,



DFA can be obtained by the direct method as,

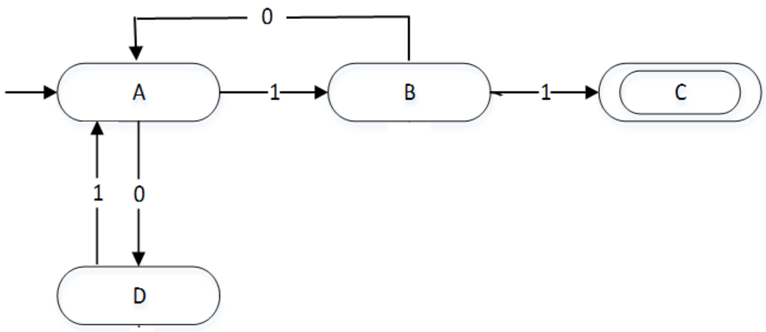


Relabeling the states we would get,



Q \ Σ	0	1
A	D	B
B	F	C
* C	-	-
D	-	E
E	D	B
F	D	B

As we can see A, E and F are the equivalent states, we can get rid of E and F and only keep state A. We need to replace all occurrences of E and F by A while reducing. The reduced DFA can be drawn as below,



Q.3 a) The language given is as follows

$(a + b)^*\ bbb\ (a + b)^*$

Hence the context free grammar denoted by it is

$S \rightarrow AbbbA$ (1)

$A \rightarrow aA \mid bA \mid \in$

(2) (3) (4)

Consider the example string, "bbb"

$S \rightarrow AbbbA$, rule (1)

$S \rightarrow bbb$, rule (4)

Q.3 b) $S \rightarrow S + S \mid S * S \mid 4$

Let us derive a string "4 + 4 * 4"

Leftmost derivation:

$S \rightarrow \underline{S} + S$

$\rightarrow 4 + \underline{S}$

$\rightarrow 4 + \underline{S} * S$

$\rightarrow 4 + 4 * \underline{S}$

$\rightarrow 4 + 4 * 4$

OR

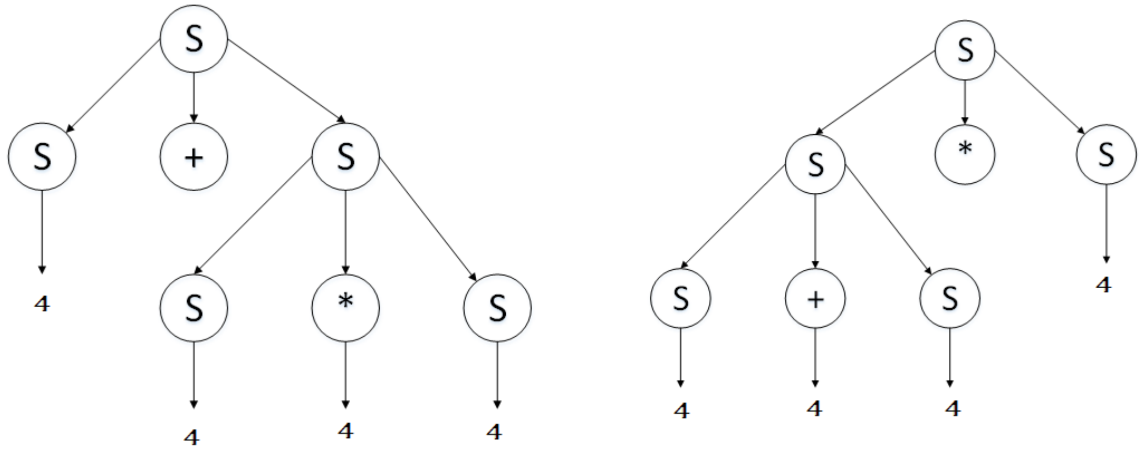
$S \rightarrow \underline{S} * S$

$\rightarrow \underline{S} + S * S$

$\rightarrow 4 + \underline{S} * S$

$\rightarrow 4 + 4 * \underline{S}$

$\rightarrow 4 + 4 * 4$



There can be two derivation trees as shown above as there are two different ways of deriving the same string, one using start production as '$S \to S + S$' and other using '$S \to S * S$'

Thus, the grammar given is ambiguous.
An equivalent unambiguous grammar can be written as,

$S \to S + T \mid T$

$T \to T * F \mid F$

$F \to 4$

Q.3 **c)** $S \to a\,B \mid b\,A$

$A \to a \mid a\,S \mid b\,A\,A$

$B \to b \mid b\,S \mid a\,B\,B$

Above grammar is not in CNF as CNF requires productions of the form,

Nonterminal → String of 2 nonterminal.

Or

Nonterminal → One terminal

Converting above to CNF,

Step 1: $S \to X\,B \mid Y\,A$

$A \to a \mid X\,S \mid Y\,A\,A$

$B \to b \mid Y\,S \mid X\,B\,B$

$X \to a$

$Y \to b$

Step 2: $A \to Y\,R_1$

$B \rightarrow X\ R_2$

$R_1 \rightarrow A\ A$

$R_2 \rightarrow B\ B$

Therefore, equivalent grammar in CNF can be written as,

$S \rightarrow X\ B \mid Y\ A$

$A \rightarrow a \mid X\ S \mid Y\ R_1$

$B \rightarrow b \mid Y\ S \mid X\ R_2$

$R_1 \rightarrow A\ A$

$R_2 \rightarrow B\ B$

$X \rightarrow a$

$Y \rightarrow b$

Q.3 d) Left most derivation:

$S \rightarrow 0\ \underline{A}\ S$

$S \rightarrow 0\ \underline{S}\ 1\ A\ S$

$S \rightarrow 0\ 0\ 1\ \underline{A}\ S$

$S \rightarrow 0\ 0\ 1\ 1\ 0\ \underline{S}$

$S \rightarrow 0\ 0\ 1\ 1\ 0\ 0$

Right most derivation:

$S \rightarrow 0\ A\ \underline{S}$

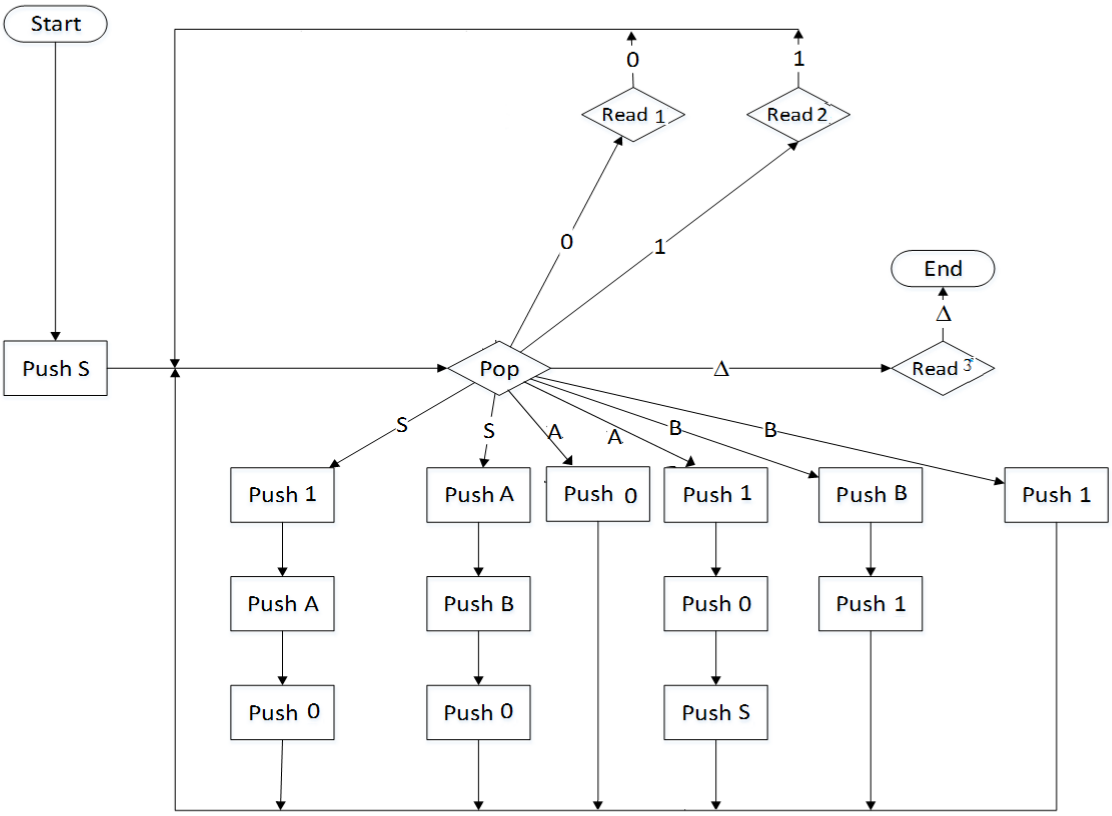
$S \rightarrow 0\ \underline{A}\ 0$

$S \rightarrow 0\ S\ 1\ \underline{A}\ 0$

$S \rightarrow 0\ \underline{S}\ 1\ 1\ 0\ 0$

$S \rightarrow 0\ 0\ 1\ 1\ 0\ 0$

Q.4 a) The required PDA can be drawn as below,



Q.4 b)

(1) Refer to the section 6.6.

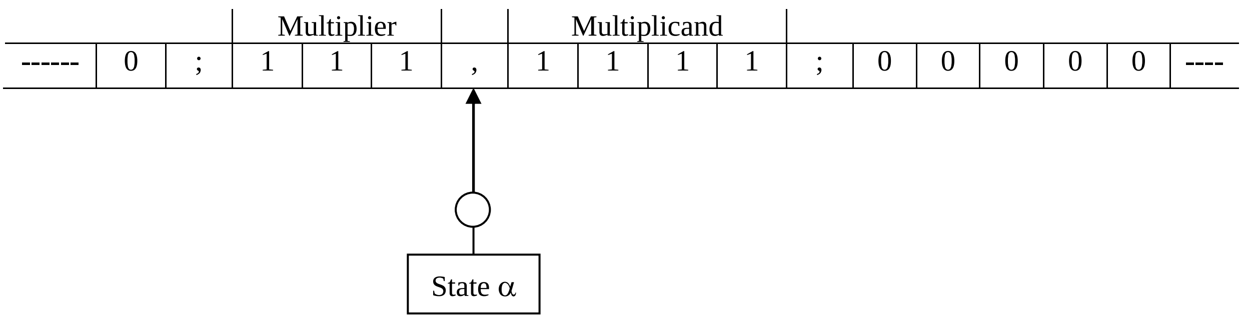
(2) For the set of all strings over alphabet $\{a, b\}$ with exactly twice many a's as b's, the grammar can be written as,

$$S \rightarrow a\,a\,S\,b \mid a\,b\,a\,S \mid b\,S\,a\,a \mid a\,a\,b \mid b\,a\,a \mid a\,b\,a$$

Now, one can draw the PDA using the above grammar by the algorithm described in section 6.7.

Q.5 a) Refer to the example 4.10 from the book.

The initial configuration of the TM is assumed as,



The simplified functional matrix is,

S \ I	0	1	a	b	;	,
α	L	$0\beta R$	-	-	ϕN	L
β	R	aR	-	-	γL	R
γ	-	-	$1\delta R$	-	R	L
δ	$b\in L$	R	-	R	R	-
$\in$	L	L	$1\delta R$	L	L	αN
ϕ	-	-	-	-	-	-

(i) 111 X 1111

--	0	;	1	1	1	,	1	1	1	1	;	0	0	0	0	0	--	Initial configuration
						↑												
--	0	;	1	1	1	,	1	1	1	1	;	0	0	0	0	0	--	$\delta(\alpha, ,) = (L)$
					↑													
--	0	;	1	1	0	,	1	1	1	1	;	0	0	0	0	0	--	$\delta(\alpha, 1) = (0\ \beta\ R)$
						↑												
--	0	;	1	1	0	,	1	1	1	1	;	0	0	0	0	0	--	$\delta(\beta, ,) = (R)$
							↑											
--	0	;	1	1	0	,	a	1	1	1	;	0	0	0	0	0	--	$\delta(\beta, 1) = (a\ R)$ Replace 1's by a's till one finds semicolon (;)
								↑										
--	0	;	1	1	0	,	a	a	a	a	;	0	0	0	0	0	--	$\delta(\beta, 1) = (a\ R)$
											↑							
--	0	;	1	1	0	,	a	a	a	a	;	0	0	0	0	0	--	$\delta(\beta, ;) = (\gamma\ L)$
										↑								

--	0	;	1	1	0	,	a	a	a	1	;	0	0	0	0	0	--
											↑						

$\delta(\gamma, \alpha) = (1\delta\ R)$

--	0	;	1	1	0	,	a	a	a	1	;	0	0	0	0	0	--
												↑					

$\delta(\delta, ;) = (R)$

--	0	;	1	1	0	,	a	a	a	1	;	b	0	0	0	0	--
											↑						

$\delta(\delta, 0) = (\ b \in L\ )$

--	0	;	1	1	0	,	a	a	1	1	;	b	b	0	0	0	--
												↑					

$\delta(\delta, 0) = (\ b \in L)$

The process continues till the entire multiplicand is copied to the result area

--	0	;	1	1	0	,	a	1	1	1	;	b	b	b	0	0	--
														↑			

$\delta(\delta, 0) = (\ b \in L)$

--	0	;	1	1	0	,	1	1	1	1	;	b	b	b	b	0	--
														↑			

$\delta(\delta, 0) = (\ b \in L)$

--	0	;	1	1	0	,	1	1	1	1	;	b	b	b	b	0	--
						↑											

$\delta(\in, ,) = (\ \alpha N\ )$

Move left till it finds another 1 from the multiplier

--	0	;	1	0	0	,	1	1	1	1	;	b	b	b	b	b	--
					↑												

$\delta(\alpha, 1) = (0\ \beta\ R)$

Continue the process till all the 1's are exhausted from the multiplier area and we are left with 12 b's which is the final answer.

--	0	;	0	0	0	,	1	1	1	1	;	b	b	b	b	b	b	b	b	b	b	b	b	--

(ii) 111 X 11

The simulation can be done as above.

Q.5 b)

(i) Power of Turing machine over finite state machine: Refer to the section 4.1.

(ii) Universal Turing machine: Refer to the section 4.9.

Q.6 **a)** Refer to the section 4.18.

Q.6 **b)** Refer to the section 9.2.1, theorem 2.

Q.6 **c)** Refer to the section 10.3.3.2.

