i) b) Bottom Up parser

ii) c) the lexical analysis of the program

iii) a) LR grammer

iv) b) left - most derivation

v) c) Canonical LR

④ Viable Prefix :-
part
II

The prefixes of right sentential forms that
can appear on the stack of a shift-reduce
parser are called viable prefixes.

By definition, a viable prefix is a prefix of
a right sentential form that does not continue
past the right end of the rightmost handle
of that sentential form.

Example :-
Let : $S \rightarrow x_1 \, x_2 \, x_3 \, x_4$

$A \rightarrow x_1 \, x_2$

Let $w = x_1 \, x_2 \, x_3$

Stack	Input
$	$x_1 \, x_2 \, x_3$
$ x_1	$x_2 \, x_3$
$ $x_1 \, x_2$	x_3

Stack	Input
$ A	x_3
$ A x_3	$
⋮	

As we see, x_1, x_2, x_3 will never appear on the stack.

So, it is not a viable prefix.

④
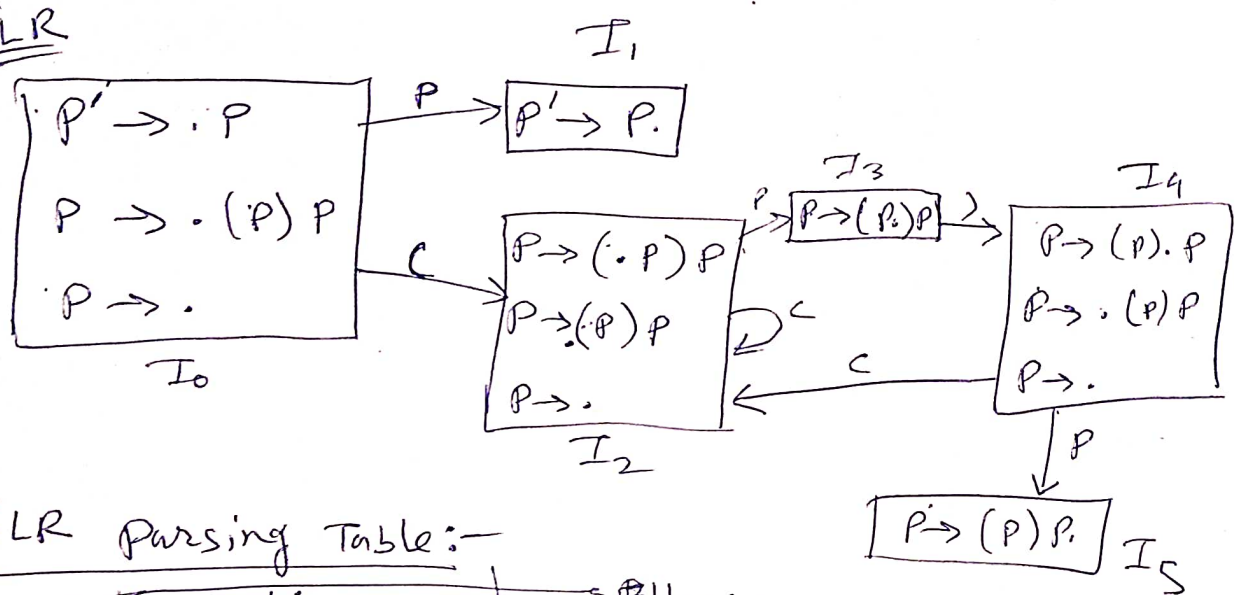
$$P \rightarrow P(P) \quad \text{———①}$$

$$P \rightarrow \varepsilon \quad \text{———②}$$

SLR

SLR Parsing Table:-

Follow $(') = \{ >, \$ \}$

State	Action			goto
	(	)	$	P
0	S_2	r_2	r_2	1
1			Accept	
2	S_2	r_2	r_2	3
3		S_4	.	
4	S_2	r_2	r_2	5
5		r_1	r_1	

② R.F. : $(a | b)^* ba$.

DFA :
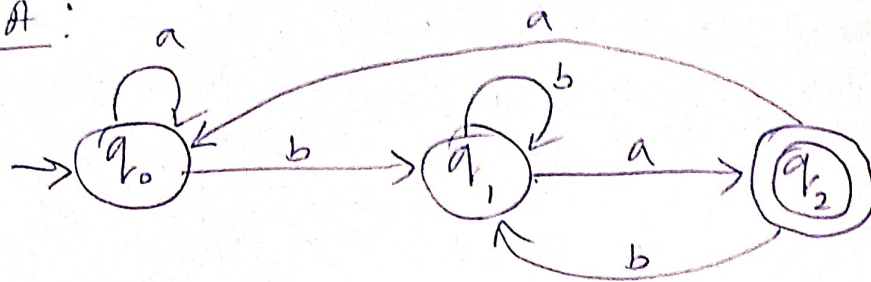
Grammar corresponding to the DFA :

$$q_0 \to a\, q_0 / b\, q_1 \qquad (q_0, q_1, q_2) \in V$$

$$q_1 \to a\, q_2 / b\, q_1 \qquad\qquad (a, b) \in T$$

$$q_2 \to a\, q_0 / b\, q_1 / \varepsilon$$

	First	Follow
$q_0 \to a\,q_0 / b\,q_1$	$\{a, b\}$	$\{\$\}$
q_1	$\{a, b\}$	$\{\$\}$
q_2	$\{a, b, \varepsilon\}$	$\{\$\}$

Last pos

$$Last\, (q_0) = \{\$\}$$
$$Last\, (q_1) = \{\$\}$$
$$Last\, (q_2) = \{\$\}$$

(3) Symbol table :-

Symbol table is an important ds created and maintained by compilers in order to store information about the occurence of various entities such as variable names, function names, objects, classes, interfaces, etc. Symbol table is used by both the analysis and the synthesis parts of a compiler.

A symbol table may serve the following purposes depending upon the language in hand :-

- To store the names of all entities in a structured form at one place.
- To verify if a variable has been declared.
- To implement type checking, by verifying assignments and expressions in the source code are semantically correct.
- To determine the scope of a name.