

Internal Assessment Test I - Mar 2017
Sub: Compiler Design (10CS63)
Sem: VI, Branch: CSE

1. Phases of a Compiler

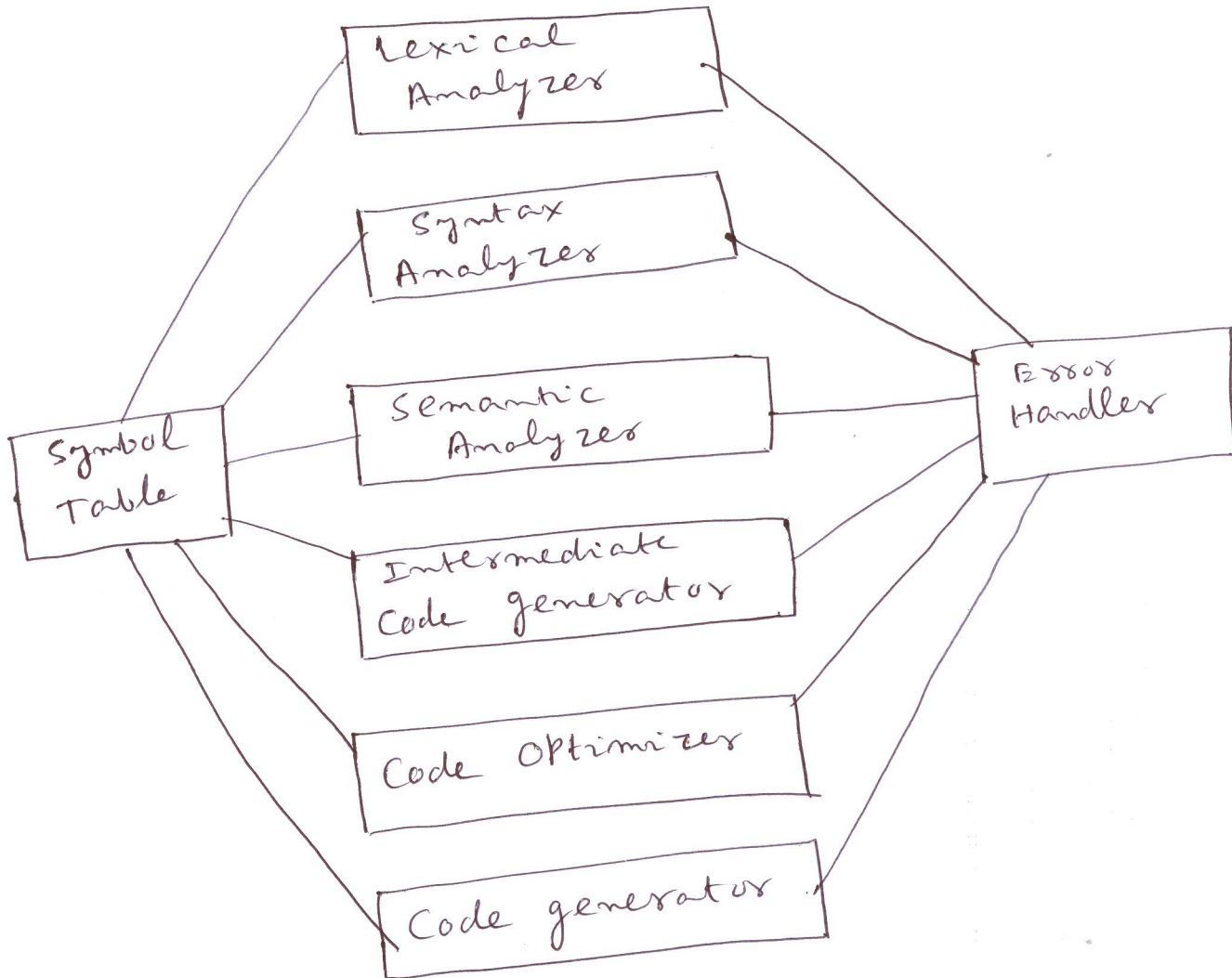


Diagram with explanation → 5 mark

Translation for assignment statement

$A = B * 5 + C - 10$ after each phase - 5M

~~code~~

2.(a) Algorithm for left recursion \rightarrow 2M

$$\text{or } A \rightarrow Ad | \beta$$

After removing left recursion

~~$$A \rightarrow \beta A' | \beta$$~~

$$A \rightarrow \beta A'$$

$$A' \rightarrow \alpha A' | \epsilon$$

$$S \rightarrow Aa | b$$

$$A \rightarrow Ac | Sd | a$$

After eliminating left recursion
we will get

Since here indirect left recursion
is existing remove that first

$$S \rightarrow Aa | b$$

$$A \rightarrow Ac | Aad | bd | a$$

After left recursion removal

$$S \rightarrow Aa | b$$

$$A \rightarrow bdA' | aA'$$

$$A' \rightarrow cA' | adA' | \epsilon$$

\rightarrow 3M

2. (b) Regular definition for an unsigned no.

digit $\rightarrow [0-9]$

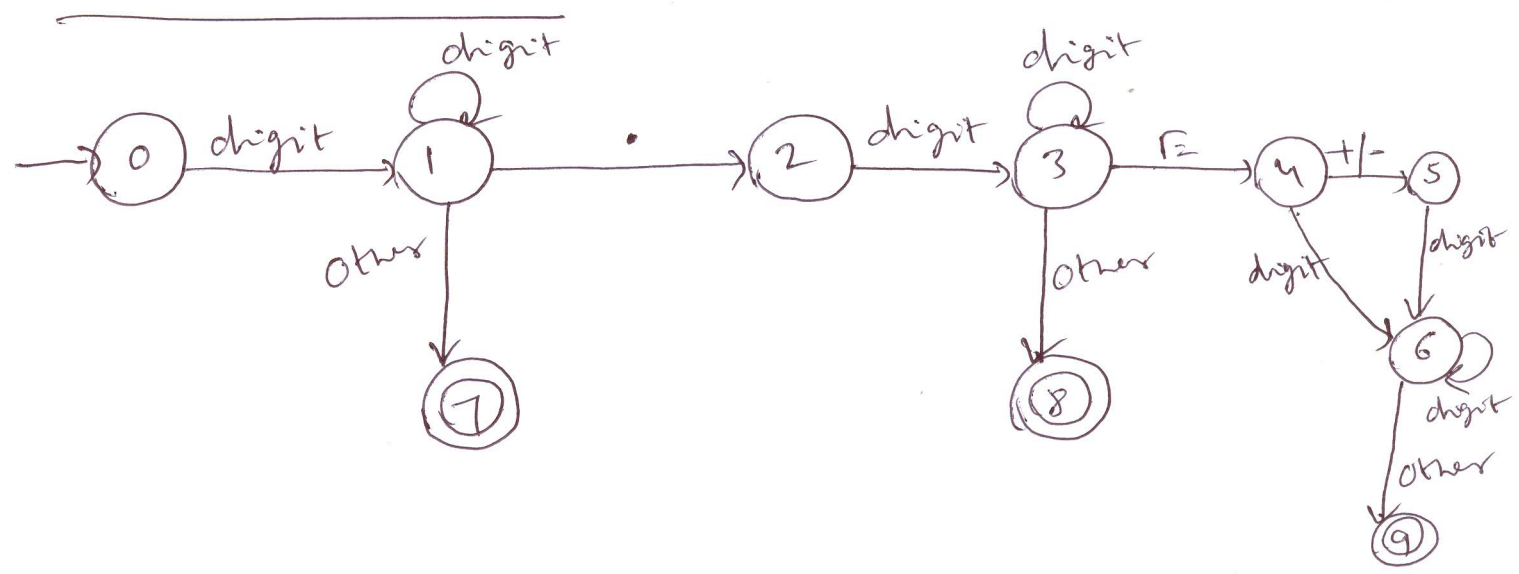
\rightarrow 2M

digits \rightarrow digit⁺

number \rightarrow digits(\cdot digits)? ($[+-]$? digits)?

Transition diagram

\rightarrow 3M



3. (a) Handle

\rightarrow 1 Mark

\rightarrow Handle is the RHS of a prodⁿ whose rightmost symbol is at the top of stack & leftmost is inside the stack. Handle is replaced by it's LHS nonterminal.

Stack	Input	Action
\$	000111 \$	Shift
\$0	00111 \$	Shift
\$00	0111 \$	Shift
\$000	111 \$	Shift
\$0001	11 \$	Reduce by $S \rightarrow 01$
\$00S	1 \$	Shift
\$00S1	1 \$	Reduce by $S \rightarrow 0S1$
\$0S	\$ \$	Shift
\$0S1	\$ \$	Reduce by $S \rightarrow 0S1$
\$S	\$ \$	Accept

\rightarrow 4 Mark

3.(b) Key problems with top down parse

- ① ~~Shift-reduce conflict~~
- ② ~~Reduce-reduce conflict~~
- ① Backtracking \rightarrow | Mark
- ② Recursion

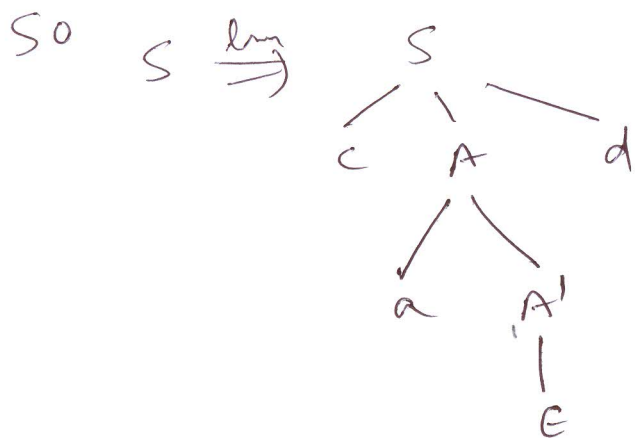
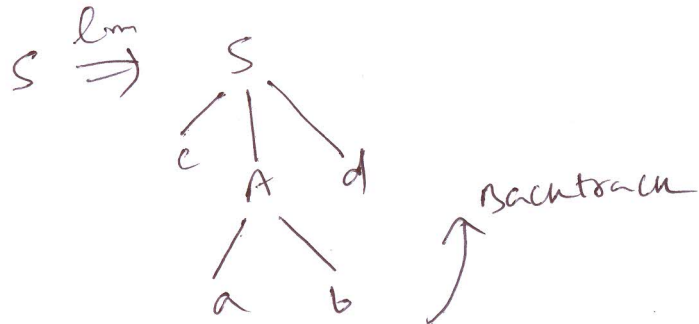
Recursive descent parser for

$$S \rightarrow cAd \quad A \rightarrow ab|a$$

I/b string $w = cad$

Remove left recursion

$$\begin{aligned} S &\rightarrow cAd \\ A &\rightarrow aA' \\ A' &\rightarrow bA' | \epsilon \end{aligned}$$



\rightarrow LM

Q. 4. $S \rightarrow aA \mid b$ $A \rightarrow c \mid e$ $B \rightarrow d \mid e$ \rightarrow 5M

Predictive Parsing table

	a	b	c	d	\$
S	$S \rightarrow aA \mid b$				
A		$A \rightarrow e$	$A \rightarrow c$	$A \rightarrow e$	
B		$B \rightarrow e$		$B \rightarrow d$	

FOLLOW(A)

$$= \{b, d\}$$

FOLLOW(B)

$$= \{b\}$$

Moves made by the parser for $w = acdb$

Stack	ILP	Action
\$ S	acdb \$	$S \rightarrow aA \mid b$
\$ bSA a	acdb \$	Match
\$ bSA	cdb \$	$A \rightarrow c$
\$ bSA	cdb \$	Match
\$ bSA	db \$	$B \rightarrow d$
\$ bSA	db \$	Match
\$ bSA	b \$	Match
\$	\$	Accept

Q. 5 (a) Alg. for FIRST \rightarrow 3M
 " " FOLLOW \rightarrow 3M

(b) $S \rightarrow iC \mid S S' \mid a$ $S' \rightarrow eS \mid e$ $C \rightarrow b$ (2+2 marks)

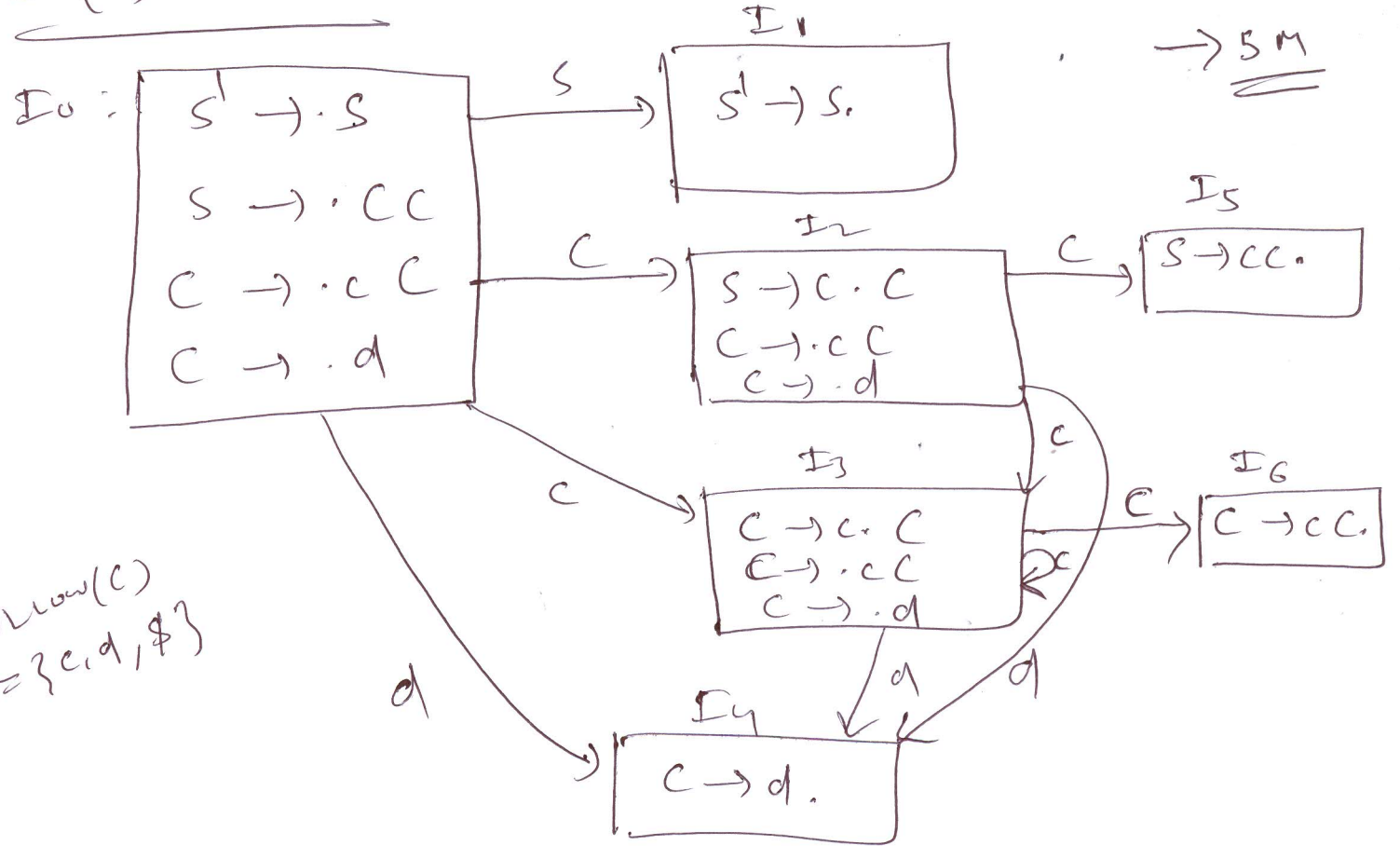
FIRST(S) = $\{i, a\}$ FIRST(S') = $\{e, e\}$ FIRST(C) = $\{b\}$

FOLLOW(S) = $\{e, \$\}$ FOLLOW(S') = $\{e, \$\}$ FOLLOW(C) = $\{t\}$

6. $S \rightarrow CC$

$C \rightarrow cC \mid d$

LR(0) items



$FOLLOW(C) = \{c, d, \$\}$

SLR Table

	ACTION			GOTO	
	c	d	\$	S	C
0	S3	S4		1	2
1			ACC		
2	S3	S4			5
3	S3	S4			6
4	r3	r3	r3		
5			r1		
6	r2	r2	r2		

Yes it is SLR.

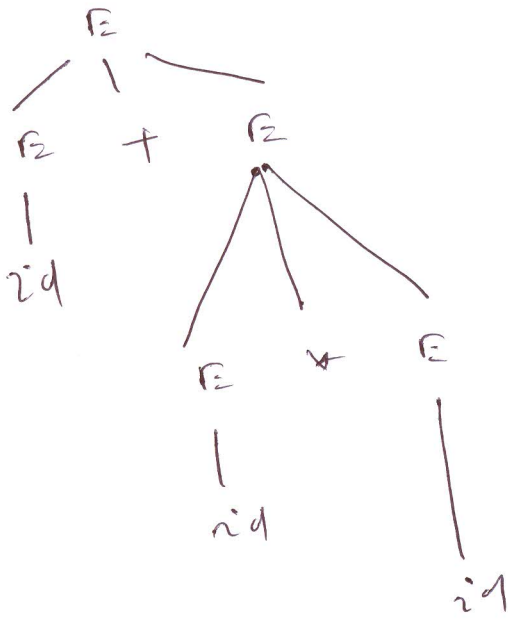
7. (a) Explanation of Input buffering strategy \rightarrow 4M

(b) $E \rightarrow E + E \mid E * E \mid E - E \mid (E) \mid id$

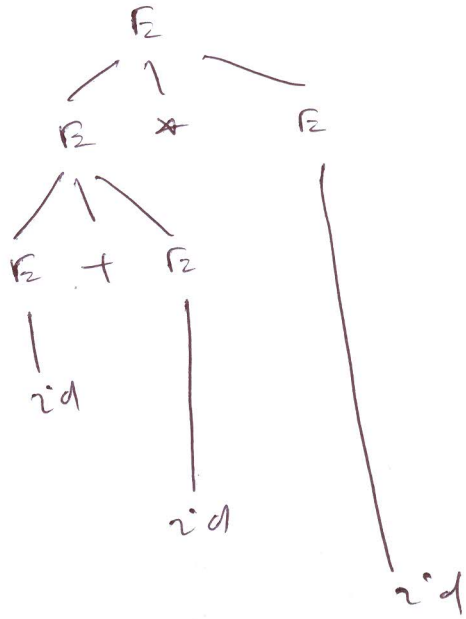
$w = id + id * id$

\rightarrow 3M

LMD 1



LMD 2



Since 2 LMD is existing, it is ambiguous.

Equivalent Unambiguous grammar

$E \rightarrow E + T \mid E - T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid id$

\rightarrow 3 Marks