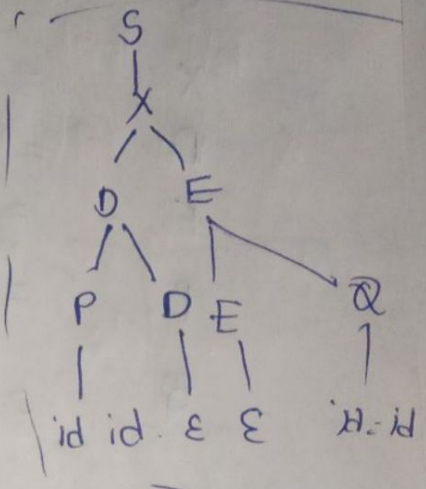
Compilers Midsem.

M. Sanjay
190419.

(1)

a) The grammar is **not** ambiguous because for any word, there is a unique parse tree possible.



b)

FIRST(S) = {ε, id}

FIRST(X) = {ε, id}

FIRST(D) = {ε, id}

FIRST(E) = {ε, id}

FIRST(P) = {id}

FIRST(Q) = {id}

[6

b)

Follow's

S = {$}
X = {$}
D = {$, id}
E = {$}
P = {$, id}
Q = {$, id}

Rough:-

S → X
→ DE
→ PDE

DQE
PDQE

c) To build LR(0) Automaton.
First let's Augment Gramm

① S' → S
② S → X
③ X → DE
④ D → PD
⑤ D → ε
⑥ E → QE
⑦ E → ε
⑧ P → id id
⑨ Q → id = id

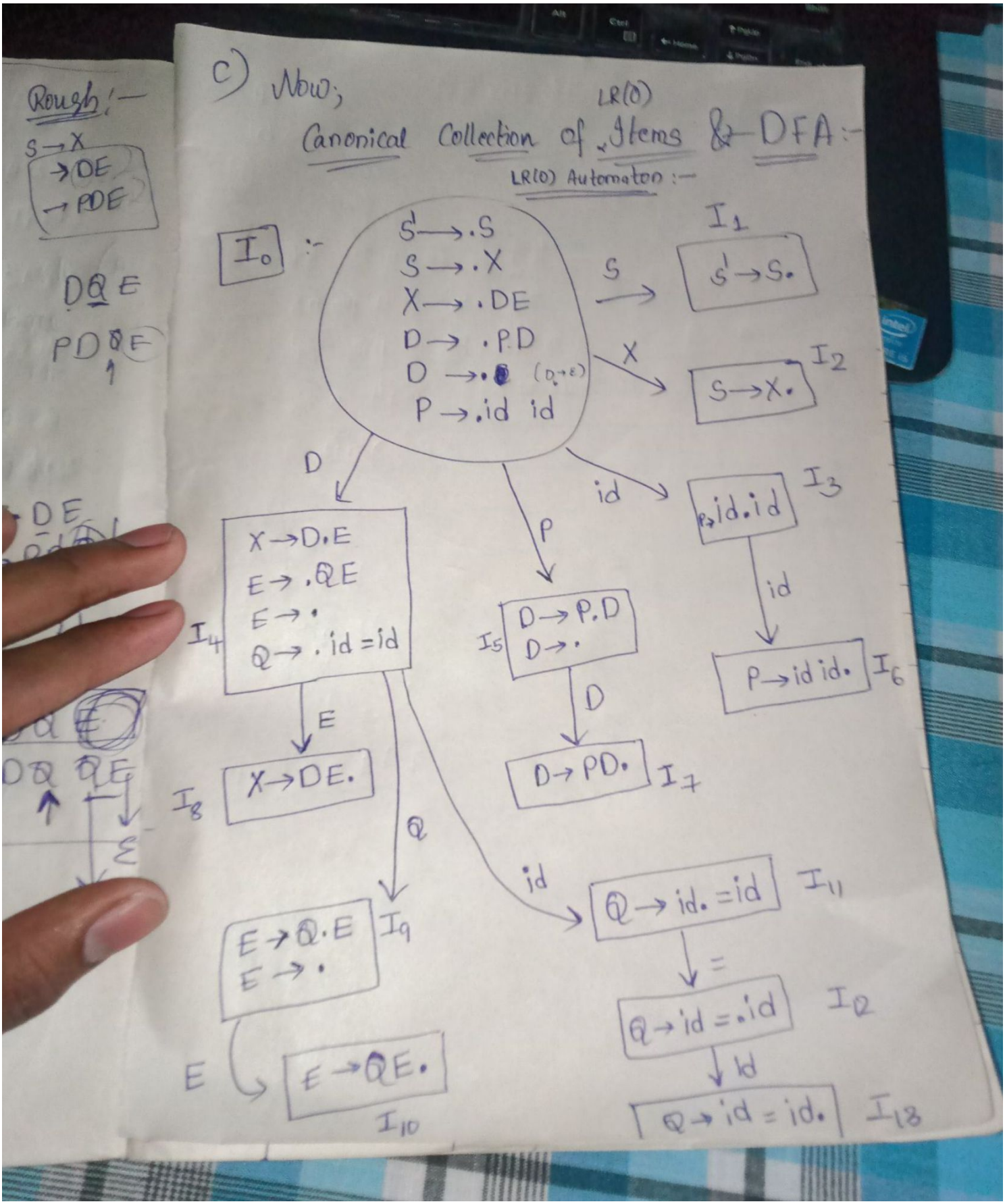
X → DE
X → PDQE
X → P id

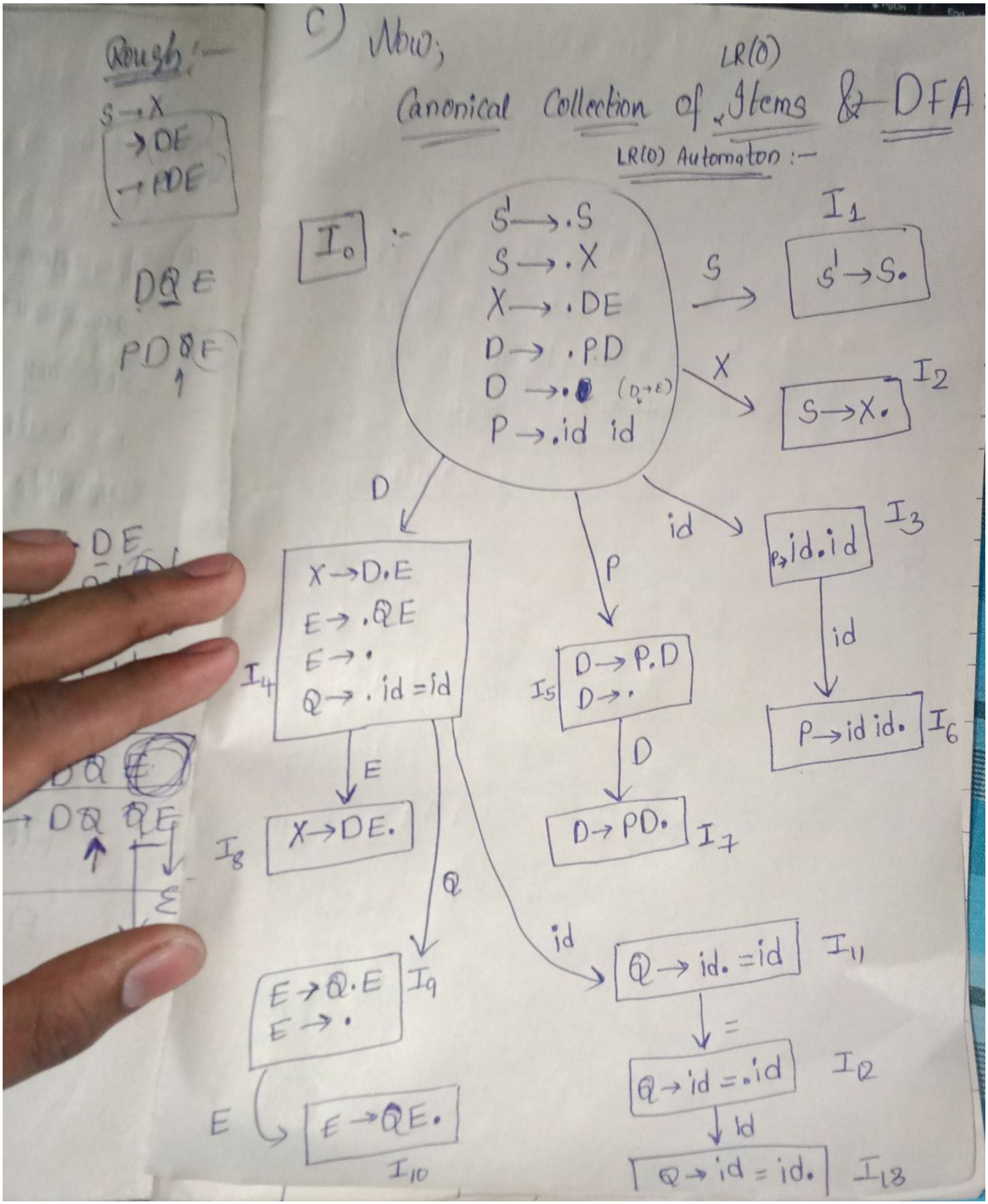
S → X → DE
S → X → DQE
→ DQQE

ε
id

1c)

	Actio
	id
0	S
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	


Rough :-
S → X
→ DE
→ PDE
DQE
PDQE
DE
DQ QE
E
c) Now;
LR(0)
Canonical Collection of Items & DFA :-
LR(0) Automaton :-
I0 :-
S' → .S
S → .X
X → .DE
D → .PD
D → . (D→ε)
P → .id id
S
I1
S' → S.
X
I2
S → X.
D
P
id
I3
P → id.id
id
P → id id.
I6
I4
X → D.E
E → .QE
E → .
Q → . id = id
I5
D → P.D
D → .
D
D → PD.
I7
E
I8
X → DE.
Q
id
I11
Q → id. = id
E → Q.E
E → .
I9
=
I12
Q → id = .id
id
Q → id = id.
I18
E
E → QE.
I10


Rough:-
S → X
→ DE
→ PDE
DQE
PDQE
c) Now;
LR(0)
Canonical Collection of Items & DFA
LR(0) Automaton :-
I0 :-
S' → .S
S → .X
X → .DE
D → .PD
D → . (D→ε)
P → .id id
S
I1
S' → S.
X
I2
S → X.
D
P
id
I3
P → id.id
id
P → id id.
I6
I4
X → D.E
E → .QE
E → .
Q → . id = id
I5
D → P.D
D → .
D
D → PD.
I7
E
I8
X → DE.
Q
E → Q.E
E → .
I9
E
E → QE.
I10
id
Q → id. = id
I11
=
Q → id = .id
I12
id
Q → id = id.
I13

(1) c) LR(0) automaton is drawn fully with states in boxes in earlier page.

Now,

SLR Parsing Table.

	Actions			GOTO					
	id	=	$	X	D	E	P	Q	S
0	S3			2	4		5		1
1			accept						
2			r2						
3	S6								
4	S11					8		9	
5					7				
6	r8		r8						
7	r4		r4						
8			r3						
9						10			
10			r6						
11		S12							
12	S13								
13	r9		r9						

8 (d) Yes; the Grammar is SLR(1) because in the above SLR parsing table that we got in (c); There is no cell with multiple actions.

i.e; There is NO Reduce - Reduce Conflicts & There is No SHIFT - REDUCE Conflicts in the Given Grammar.

(Q-2). No, There is No grammar that is RL(0) but not LR(0) because in case of bottom up parsing method, even left-recursive / Right recursive grammar's can be accepted without any transformations. It is just that we need to modify our parsing algorithm while moving from one method to other. like instead of using w$ in stack; we keep ^w in stack.