

CS 419 Compiler

Project Form

Project Idea: **2**

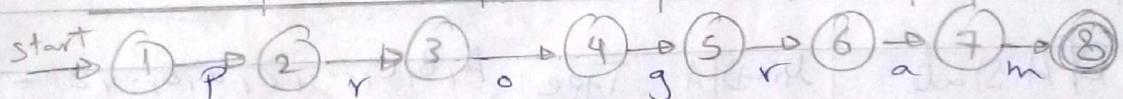
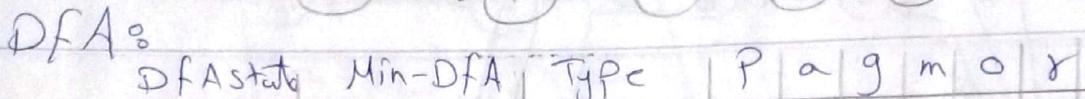
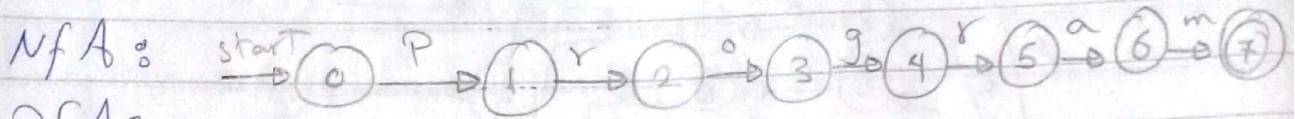
Team Members NO#: **7**

ID	Name	Level& Department	Section (Day- from-to)	Role (Lead/Member)	Grade
201900404	Abdelrahman Ashraf Aman Allah	3 / CS	Thursday From 8 to 10	Leader	
201900364	Saif Eldeen Reda Abdelfattah	3 / CS	Wednesday From 12 to 2	Member	
201900359	Sayed Anwar Sayed Shaban	3 / CS	Wednesday From 12 to 2	Member	
201900378	Shimaa Mohammed Ahmed Fouad	3 / CS	Wednesday From 12 to 2	Member	
201900360	Sayed Hamdy Sayed Hemeda	3 / CS	Wednesday From 12 to 2	Member	
201900493	Ali Ramadan Ali Hamed	3 / CS	Thursday From 8 to 10	Member	
201900384	Doha Ali Mostafa Abdelrahman	3 / CS	Thursday From 8 to 10	Member	

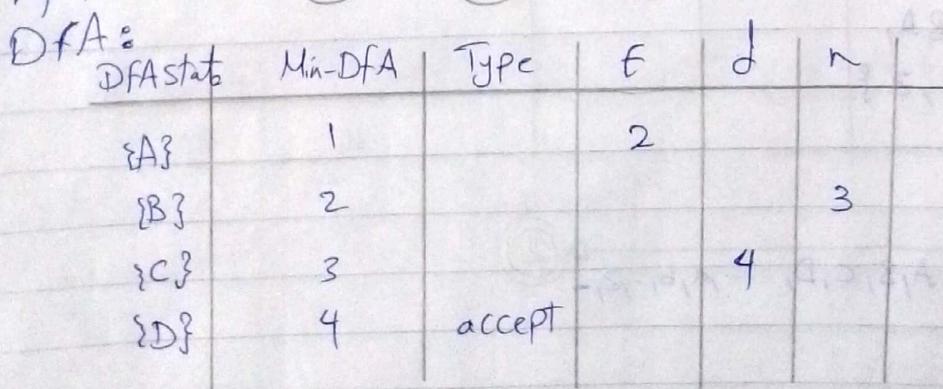
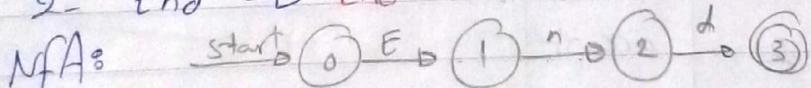
①

Regular Expression & NFA & DFA

1- Start \rightarrow Program

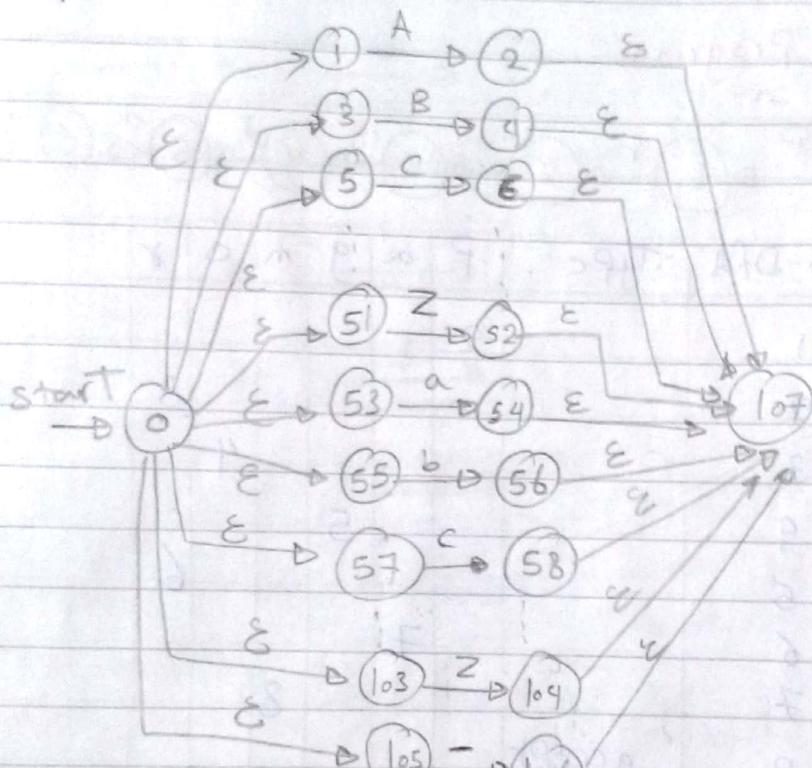


2- End \rightarrow End



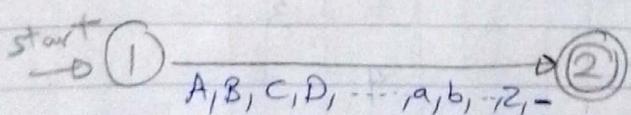
(2)

3-letter $\rightarrow A|B| \dots |Z| a|b| \dots |z| -$
 NFAs



DFA:

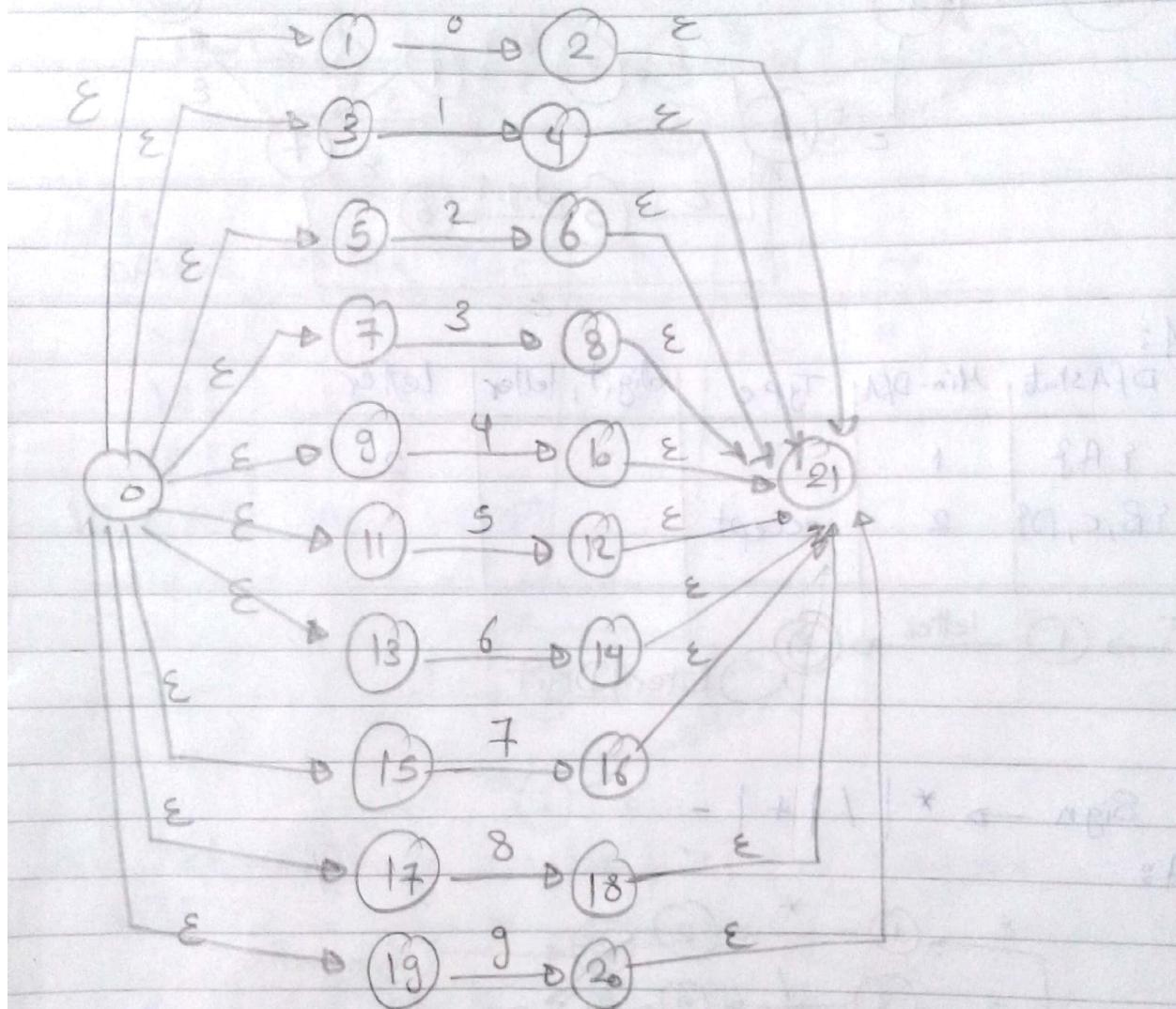
DFA State	Min-DFA	Type	A, B, C, ..., Z, a, b, ..., z, -
$\{A\}$	1		2
$\{AA, AB, AC, AD, AE, \dots, B, BA, C, D, E, \dots, Z\}$	2	accept	



(3)

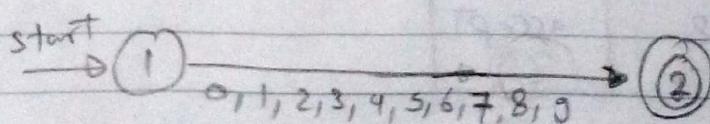
4-digit \rightarrow 0|1|2|3|4|5|6|7|8|9

NFA:



DFA:

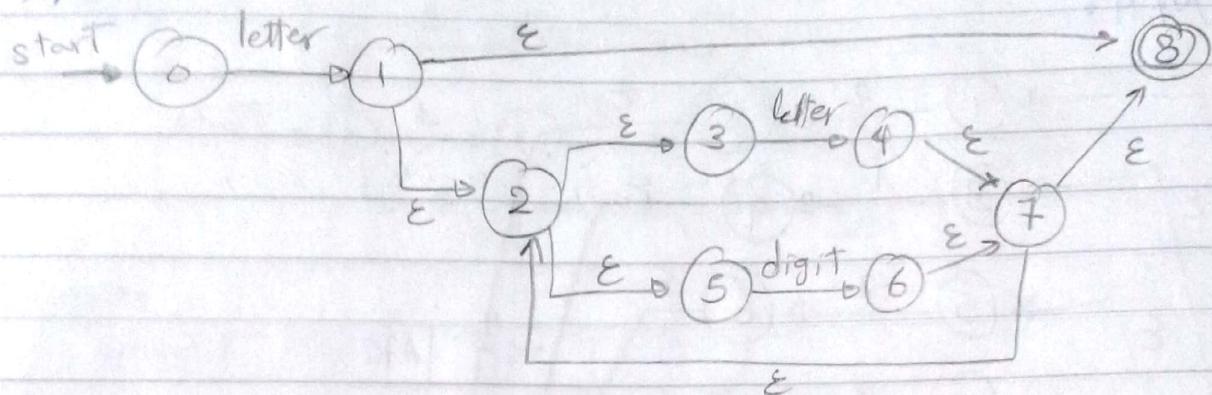
DFA-state	μ_{in} -DFA	Type	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
{A}	1		2
{B, C, D, E}	2	accept	
F, G, H, I, J, K	-		



(4)

5- $ID \rightarrow \text{letter} (\text{letter})^* \text{ digit}^*$

NFA:

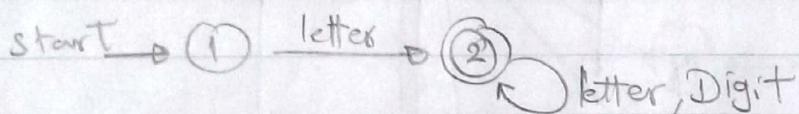


DFA:

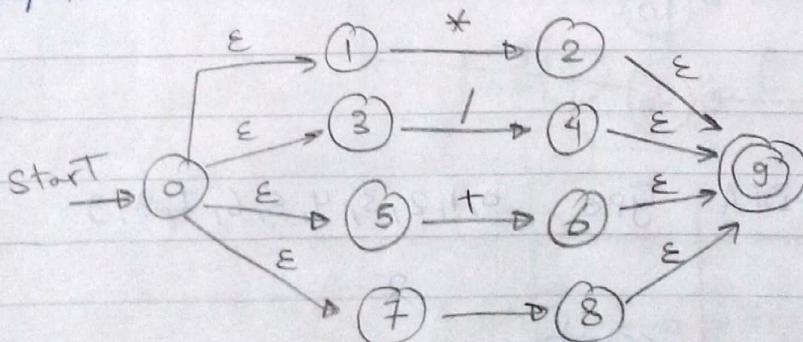
DFA-state	Min-DFA	Type
-----------	---------	------

$\{\text{A}\}$	1	digit, letter
----------------	---	---------------

$\{\text{B}, \text{C}, \text{D}\}$	2	accept
------------------------------------	---	--------

6- $\text{Sign} \rightarrow * | / | + | -$

NFA:

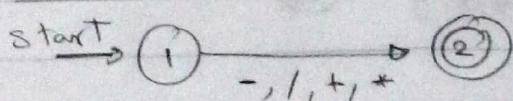


DFA:

DFA-state	Min-DFA	Type
-----------	---------	------

$\{\text{A}\}$	1	-, /, +, *
----------------	---	------------

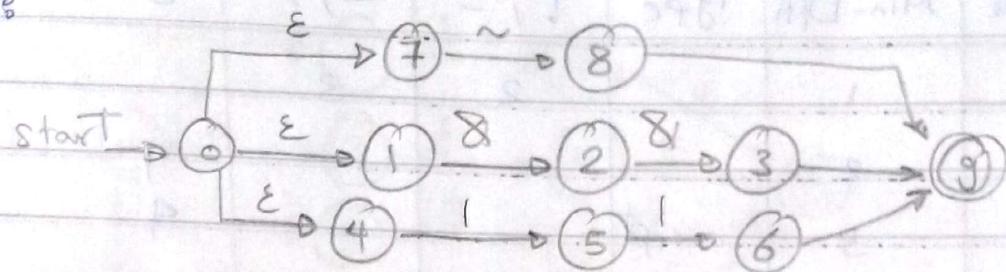
$\{\text{B}, \text{C}, \text{D}\}$	2	accept
------------------------------------	---	--------



(5)

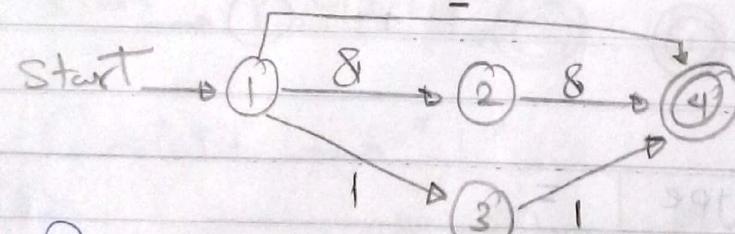
7 - logic of $\rightarrow 8, 8 \mid 1 \mid \sim$

NFA:



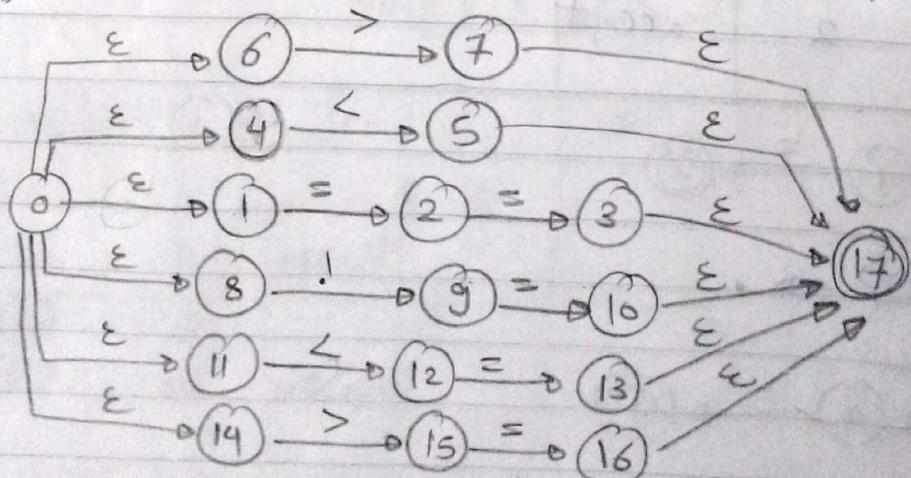
DFA:

DFA state	Min-DFA	Type	8.	1	\sim
{A}	1		2	3	4
{B}	2		4		
{C}	3			4	
{D, E, F}	4	accept			



8 - RelationsOp $\rightarrow == \mid < \mid > \mid != \mid \leq \mid \geq$

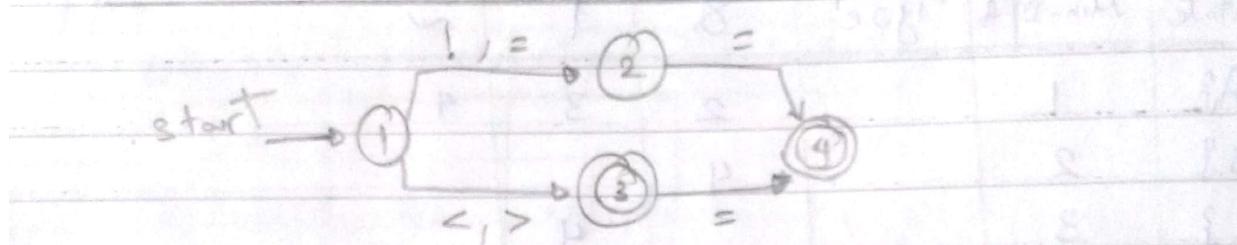
NFA:



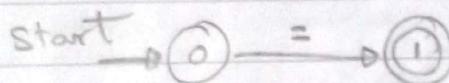
(6)

DFA:

DFA state	Min-DFA	Type	!, =	<, >	=
{A}	1		2	3	
{B, D}	2				4
{C, E}	3	accept			4
{F, G, H, I}	4	accept			

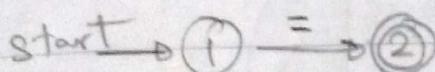
8- Assignment OP $\rightarrow =$

NFA:

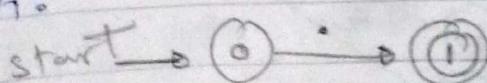


DFA:

DFA-state	Min-DFA	Type	=
{A}	1		2
{B}	2	accept	

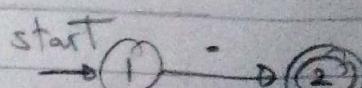
10- Access OP $\rightarrow .$

NFA:



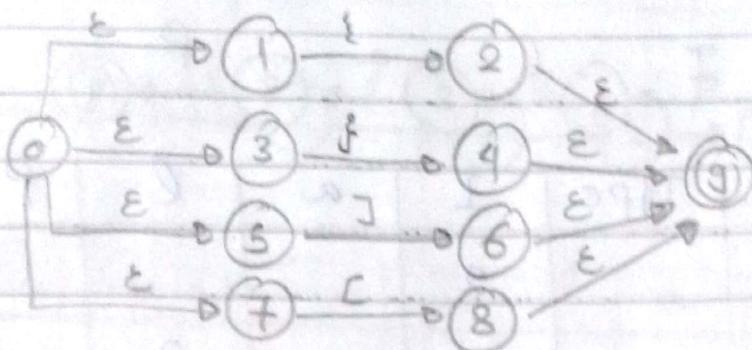
DFA:

DFA-state	Min-DFA	Type	.	
{A}	1		2	
{B}	2	accept		



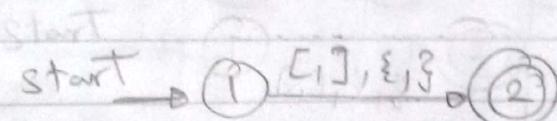
11- Braces $\rightarrow \{\mid\} \mid [\mid]$

NFA:



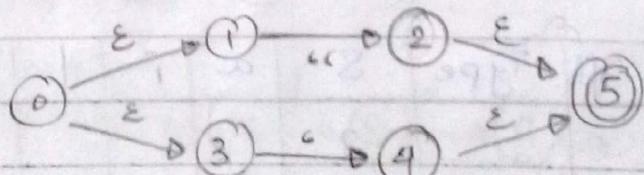
DFA:

DFA state	Min-DFA	Type	$\{, \}, [,]$
$\{A\}$	1		2
$\{B, C, D, E\}$	2	accept	



12- Quotation $\rightarrow " | "$

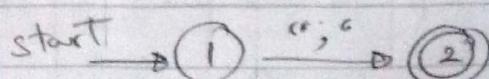
NFA:



DFA:

DFA-state	Min-DFA	Type	" , "	"
$\{A\}$	1		2	
$\{B\}$	2	accept		

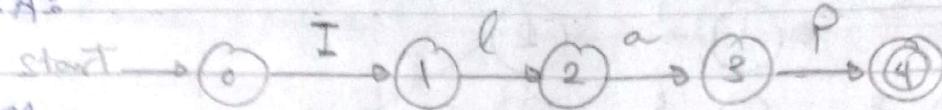
$\{C\}$ 3
 $\{D, E\}$ 4 accept



(8)

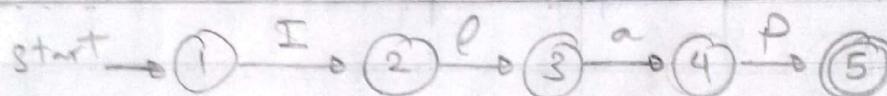
11- Integer \rightarrow Ilap

NFA:

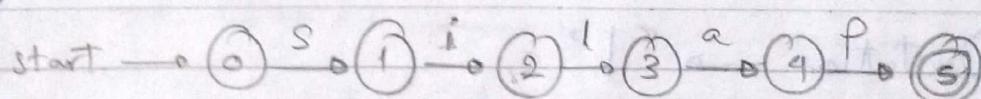


DFA:

DFA-state	DFA-min	Type	I	a	l	p
$\{A\}$	1			2		
$\{B\}$	2					3
$\{C\}$	3				4	
$\{D\}$	4					5
$\{E\}$	5	accept				

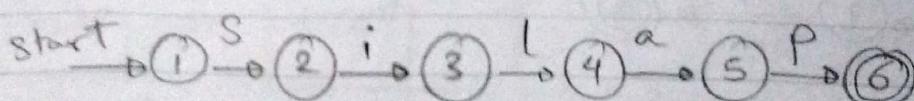
12- SInteger \rightarrow Silap

NFA:



DFA:

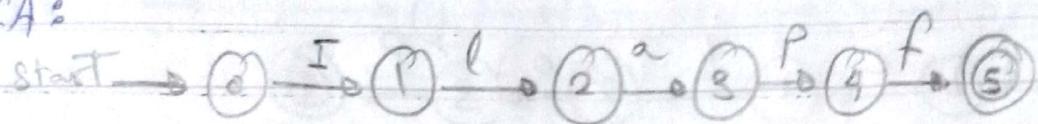
DFA-state	Min-DFA	Type	S	a	i	l	p
$\{A\}$	1			2			
$\{B\}$	2					3	
$\{C\}$	3					4	
$\{D\}$	4				5		
$\{E\}$	5					6	
$\{F\}$	6						7
$\{G\}$	6	accept					



(9)

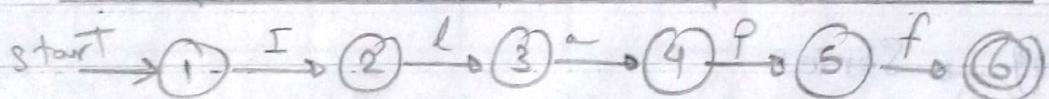
13- float \rightarrow Ilapf

NFA:

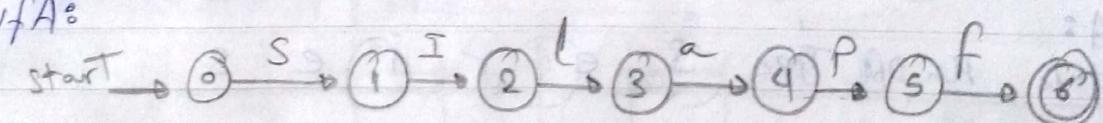


DFA:

DFA-state	Min-DFA	Type	I	l	a	p	f
{A}	1		2				
{B}	2			3			
{C}	3				4		
{D}	4					5	
{E}	5						6
{F}	6	accept					

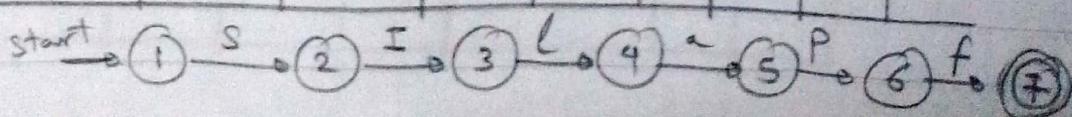
14- Sfloat \rightarrow SILapf

NFA:



DFA:

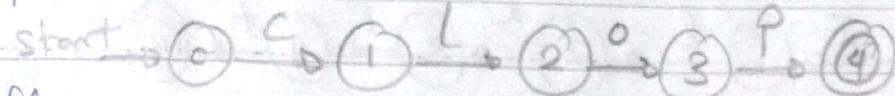
DFA-state	MinDFA	Type	S	I	l	a	p	f
{A}	1		2					
{B}	2			3				
{C}	3				4			
{D}	4					5		
{E}	5						6	
{F}	6							7
{G}	7	accept						



(10)

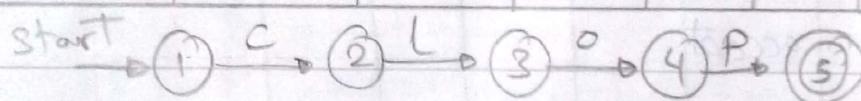
15- Character → Clop

NFA:



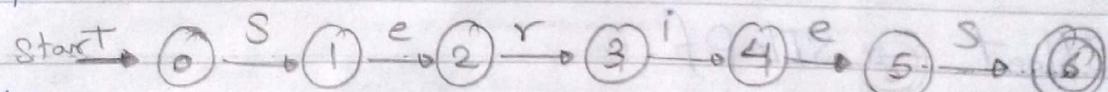
DFA:

DFA-state	Min-DFA	Type	C	L	O	P
{A}	1		2			
{B}	2			3		
{C}	3				4	5
{D}	4					5
{E}	5	accept				



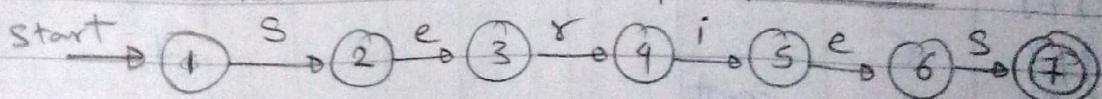
16- String → Series

NFA:

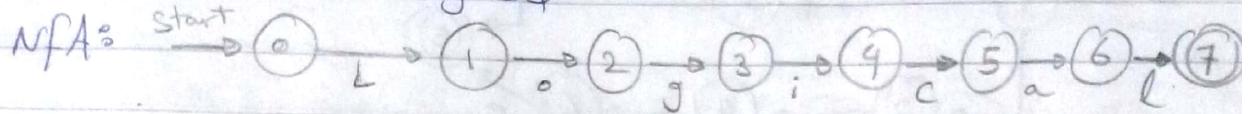


DFA:

DFAstate	Min-DFA	Type	S	e	r	i	e	s
{A}	1		2					
{B}	2			3				
{C}	3				4			
{D}	4					5		
{E}	5						6	
{F}	6							7
{G}	7	accept						



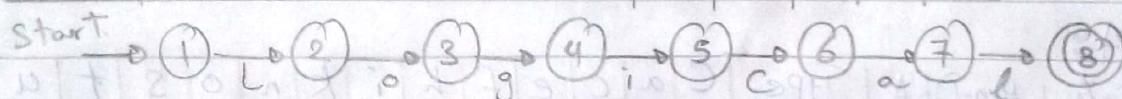
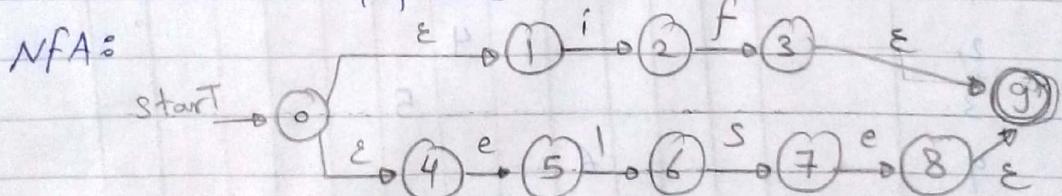
(11)

17 - Boolean \rightarrow Logical

DFA:

DFA state Min-DFA Type L o g i c a l

$\{A\}$	1	2					
$\{B\}$	2		3				
$\{C\}$	3			4			
$\{D\}$	4				5		
$\{E\}$	5					6	
$\{F\}$	6						7
$\{G\}$	7						8
$\{H\}$	8	accept					

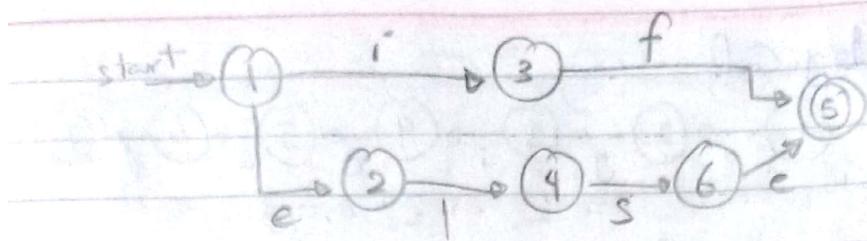
18 - Condition \rightarrow if | else

DFA:

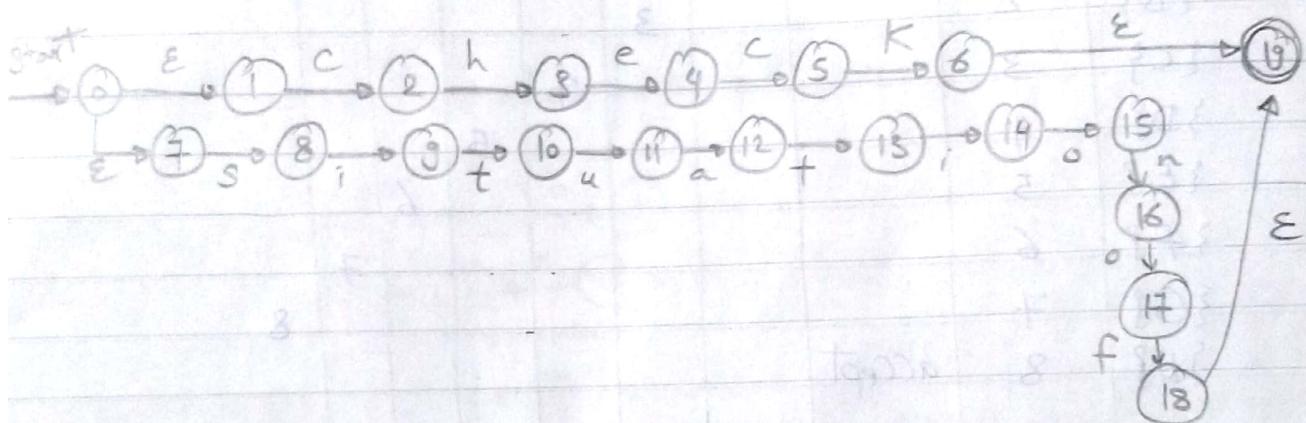
DFA state Min-DFA Type e i l s i f

$\{A\}$	1	2		3		
$\{B\}$	2		4	11		
$\{C\}$	3				5	
$\{D\}$	4			6		
$\{E, G\}$	5					
$\{F\}$	6	accept				

(12)



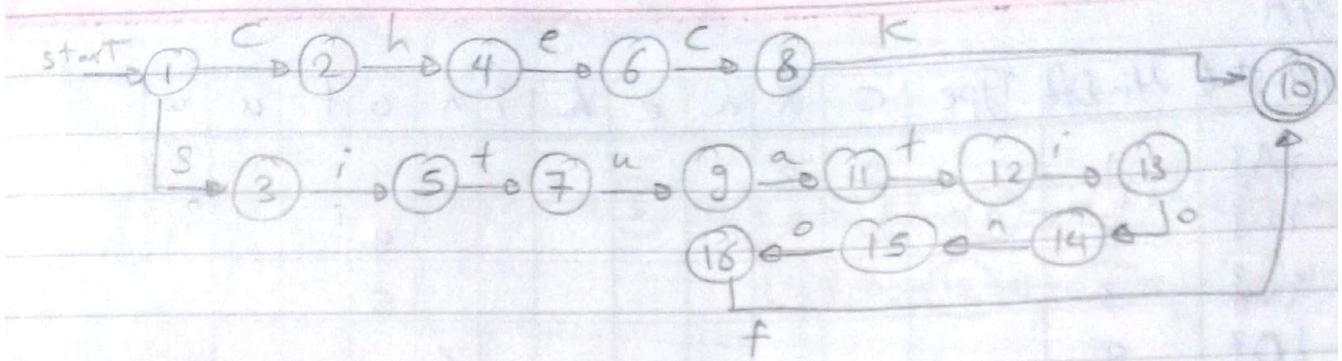
18- switch → check | situation of
NFA:



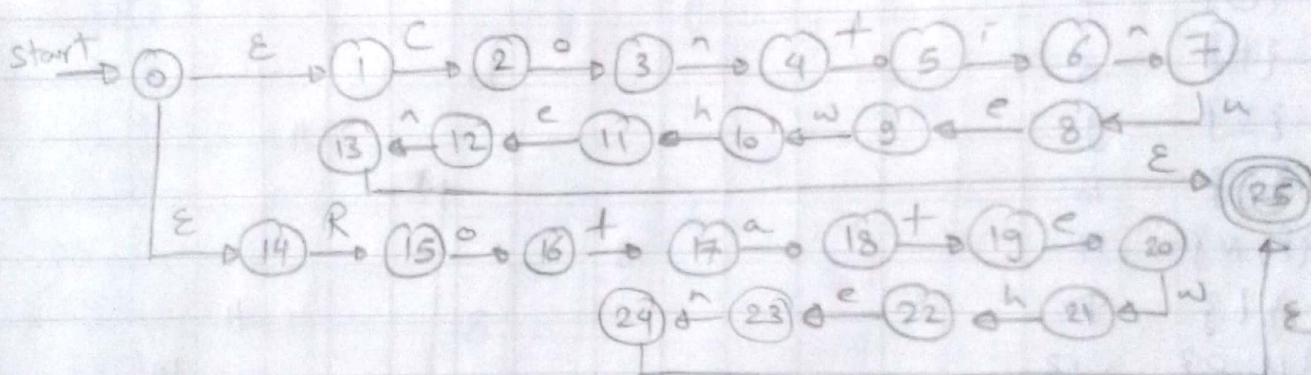
DFA:

DFA state	Min-DFA	Type	C	a	c	e	f	h	i	k	n	o	s	t	u
$\{A\}$	1				2									3	
$\{B\}$	2											4			
$\{C\}$	3											5			
$\{D\}$	4										6				
$\{E\}$	5														7
$\{F\}$	6								8						
$\{G\}$	7														9
$\{H\}$	8											10			
$\{I\}$	9														
$\{J, P\}$	10	accept													
$\{K\}$	11														12
$\{L\}$	12											13			
$\{M\}$	13													14	
$\{N\}$	14											15			
$\{O\}$	15													16	
$\{P\}$	16										10				

(13)



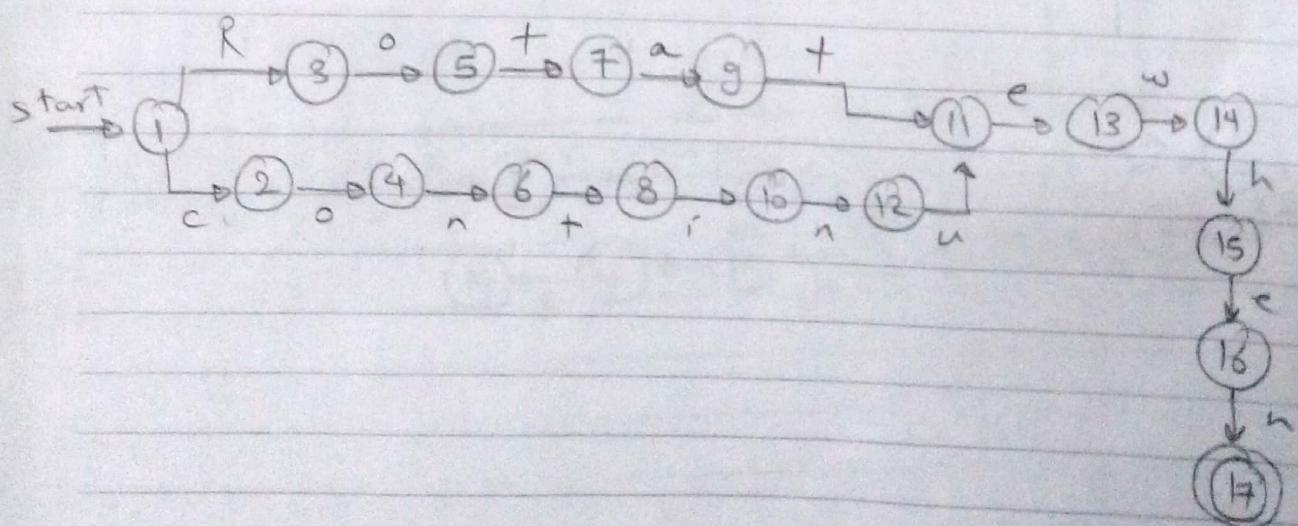
2o - loop \rightarrow Continue when | Rotate when



(14)

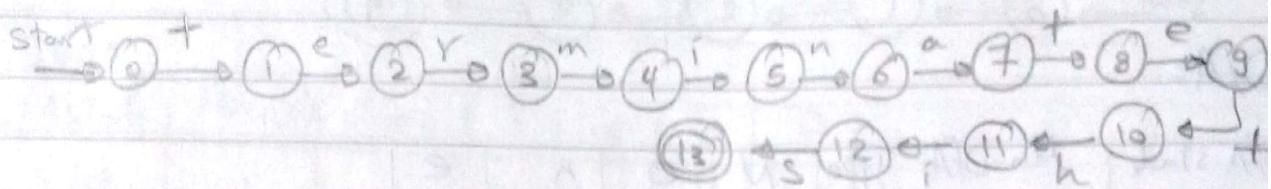
DFA

DFA-state	Min-DFA	Type	c	R	a	e	h	i	n	o	t	u	w
{A}	1			2	3								
{B}	2												4
{C}	3												5
{D}	4												6
{E}	5												
{F}	6												7
{G}	7												8
{H}	8												
{I}	9												
{J}	10												
{K, N}	11												
{L}	12												
{M, P}	13												
{O, R}	14												14
{Q, T}	15												
{S, V}	16												
{U, W}	17	accept											



(15)

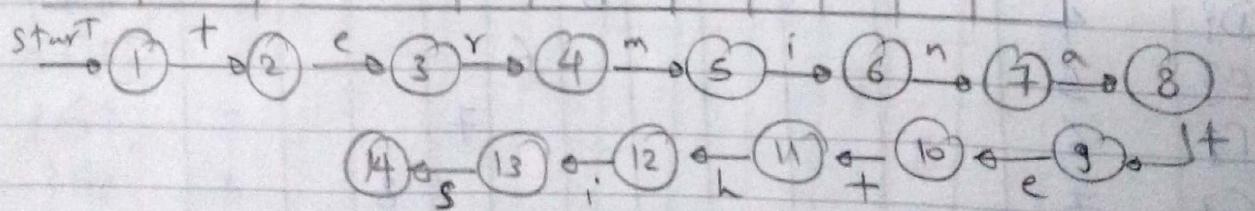
21- Break \rightarrow terminate this
NFA:



DFA:

DFA-state Min-DFA Type a e h i n r s +

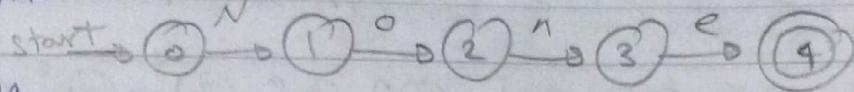
$\{A\}$	1								2	
$\{B\}$	2								3	
$\{C\}$	3								4	
$\{D\}$	4								5	
$\{E\}$	5								6	
$\{F\}$	6								7	
$\{G\}$	7								8	
$\{H\}$	8								9	
$\{I\}$	9								10	
$\{J\}$	10								11	
$\{K\}$	11								12	
$\{L\}$	12								13	
$\{M\}$	13								14	
$\{N\}$	14	accept								



(16)

22- void \rightarrow None

NFA:



DFA:

DFA-state Min-DFA Type

{A}

{B}

{C}

{D}

{E}

2

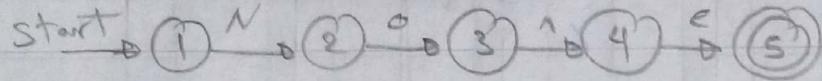
3

4

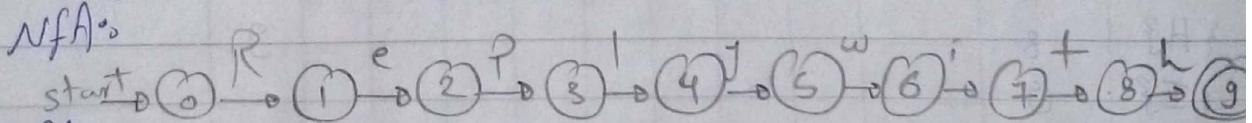
5

6

accept

23- Return \rightarrow Reply with.

NFA:



DFA:

DFA-state Min-DFA Type

{A}

{B}

{C}

{D}

{E}

{F}

{G}

{H}

{I}

{J}

2

3

4

5

6

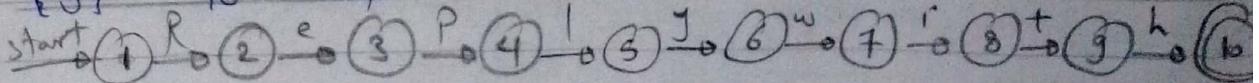
7

8

9

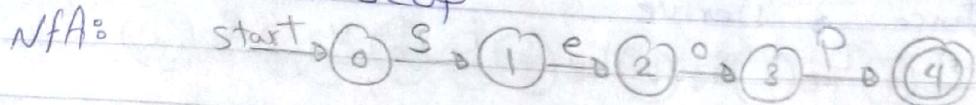
10

accept



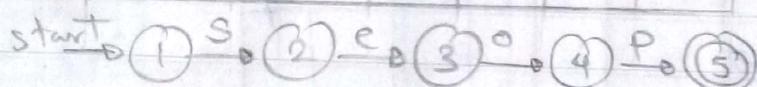
(17)

24- struct → Scop

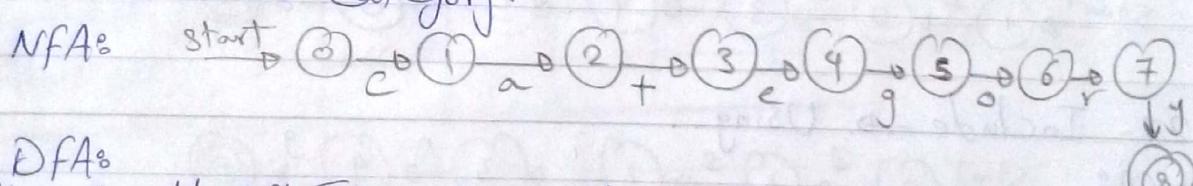


DFA:

DFA-state	Min-DFA	Type	S	e	o	p
{A}	1		2			
{B}	2			3		
{C}	3				4	
{D}	4					5
{E}	5	accept				

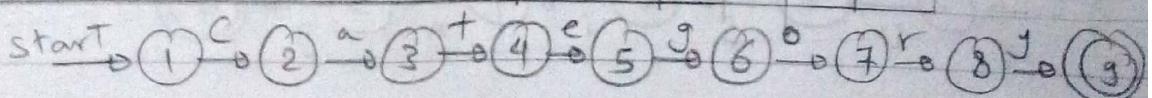


25- Class → Category.

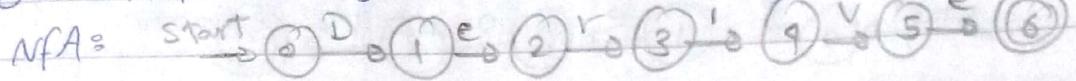


DFA:

DFA-state	Min-DFA	Type	C	a	+	e	g	o	r	j
{A}	1		2							
{B}	2			3						
{C}	3				4					
{D}	4					5				
{E}	5						6			
{F}	6							7		
{G}	7								8	
{H}	8									9
{I}	9	accept								

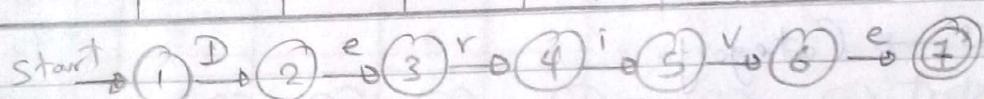
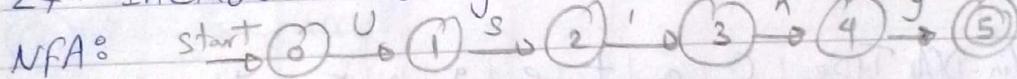


(18)

26- Inheritance \rightarrow Derive

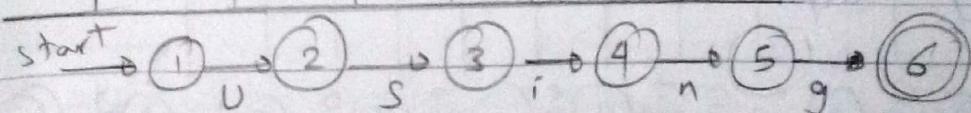
DFA:

DFA-state	Min-DFA	Type	D	e	r	i	v	e
$\{A\}$	1		2					
$\{B\}$	2			3				
$\{C\}$	3				4			
$\{D\}$	4					5		
$\{E\}$	5						6	
$\{F\}$	6							7
$\{G\}$	7	accept						

27- Include \rightarrow Using.

DFA:

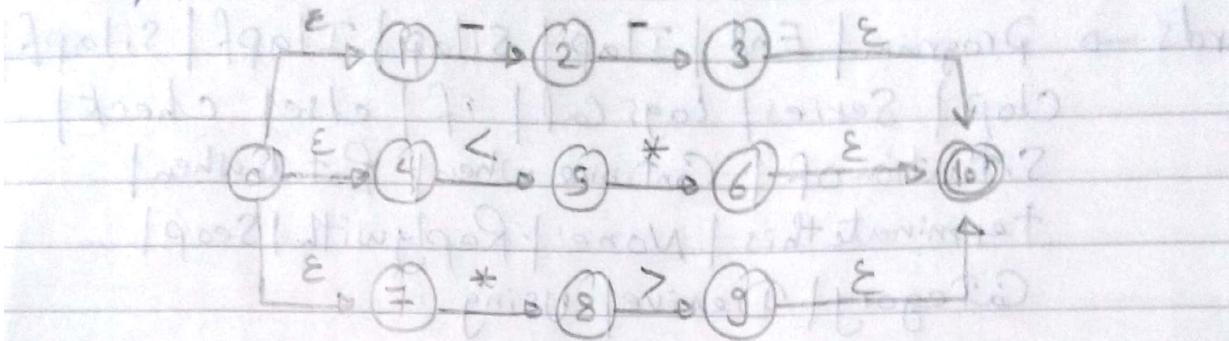
DFA-state	Min-DFA	Type	U	S	i	n	g
$\{A\}$	1		2				
$\{B\}$	2			3			
$\{C\}$	3				4		
$\{D\}$	4					5	
$\{E\}$	5						6
$\{F\}$	6	accept					



(19)

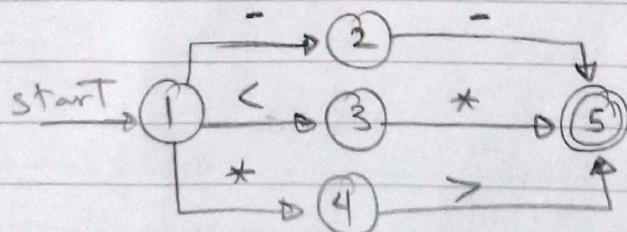
28- Comment \rightarrow -- | <*| * >

NFA:



DFA:

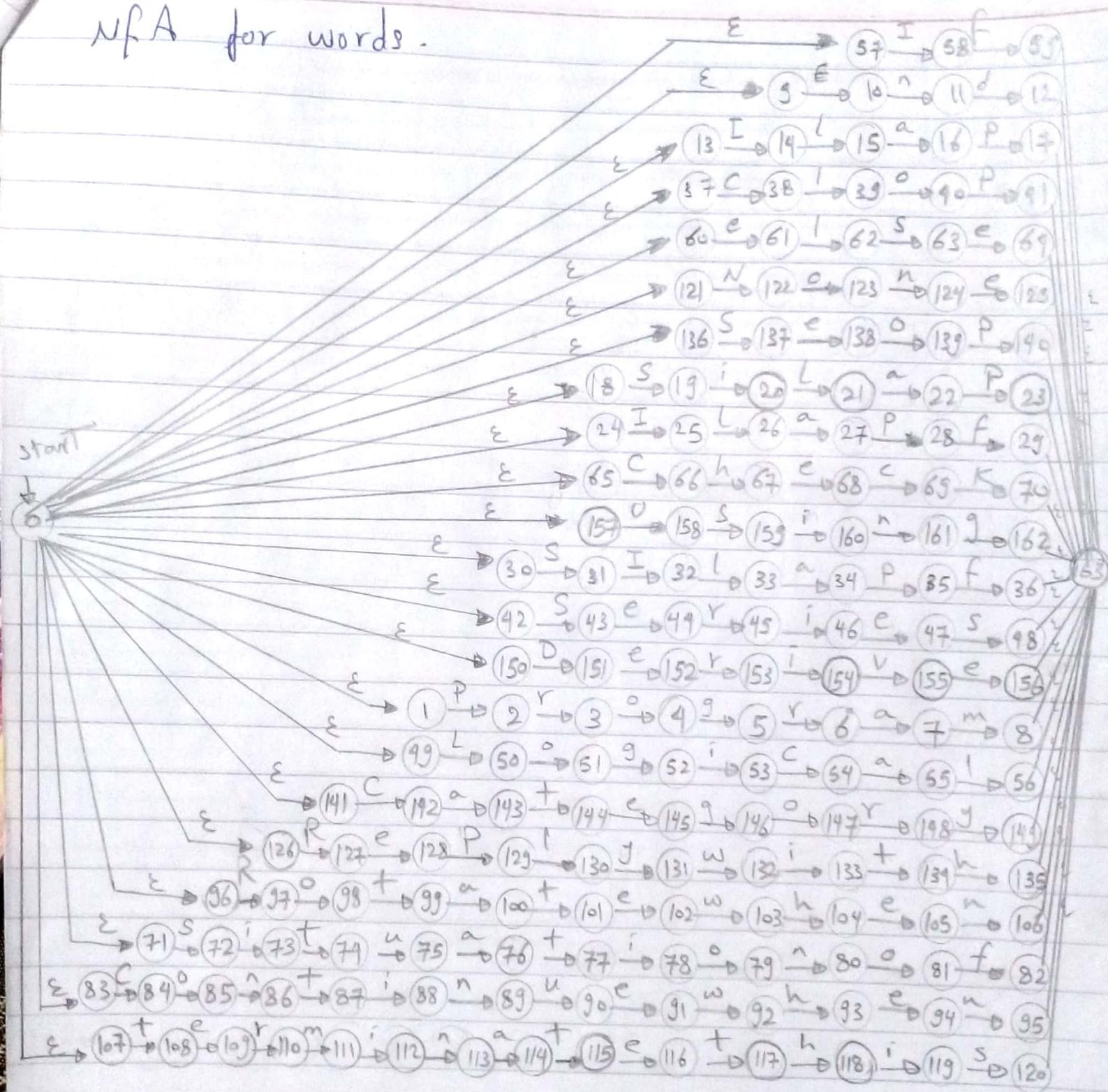
DFA-state	Min-DFA	Type	-	<	>	*
{A}	1		2	3		4
{B}	2			5		
{C}	3					5
{D}	4				5	
{E, F, G}	5	accept				



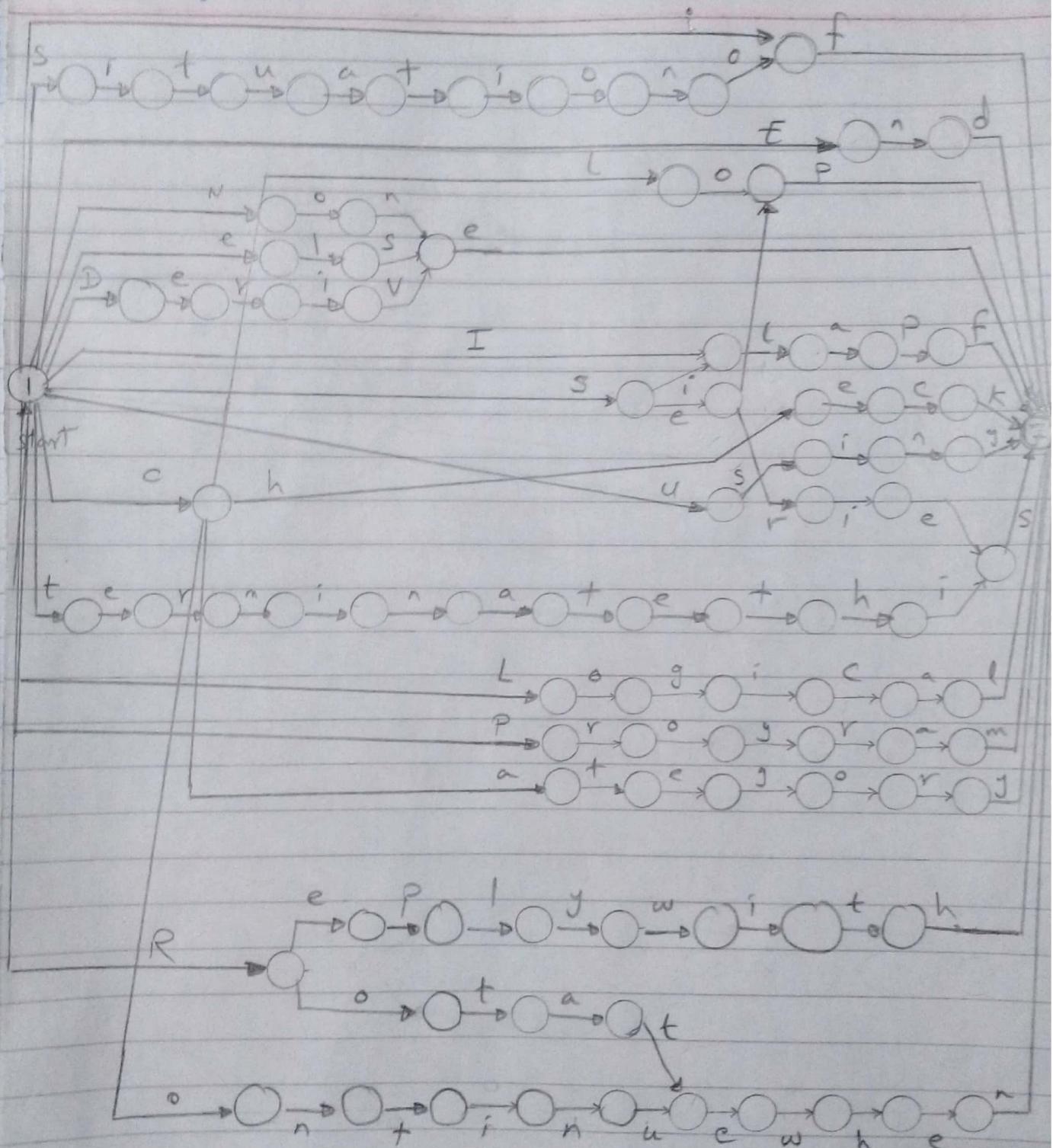
\Rightarrow Regular expression for all words

words \rightarrow Program | End | Iflap | Silap | Iflapf | Silappf |
CloP | Series | logical | if | else | check |
Situation of | Continue when | Rotate when |
Terminate this | None | Reply with | Sleep |
Category | Derived | Using -

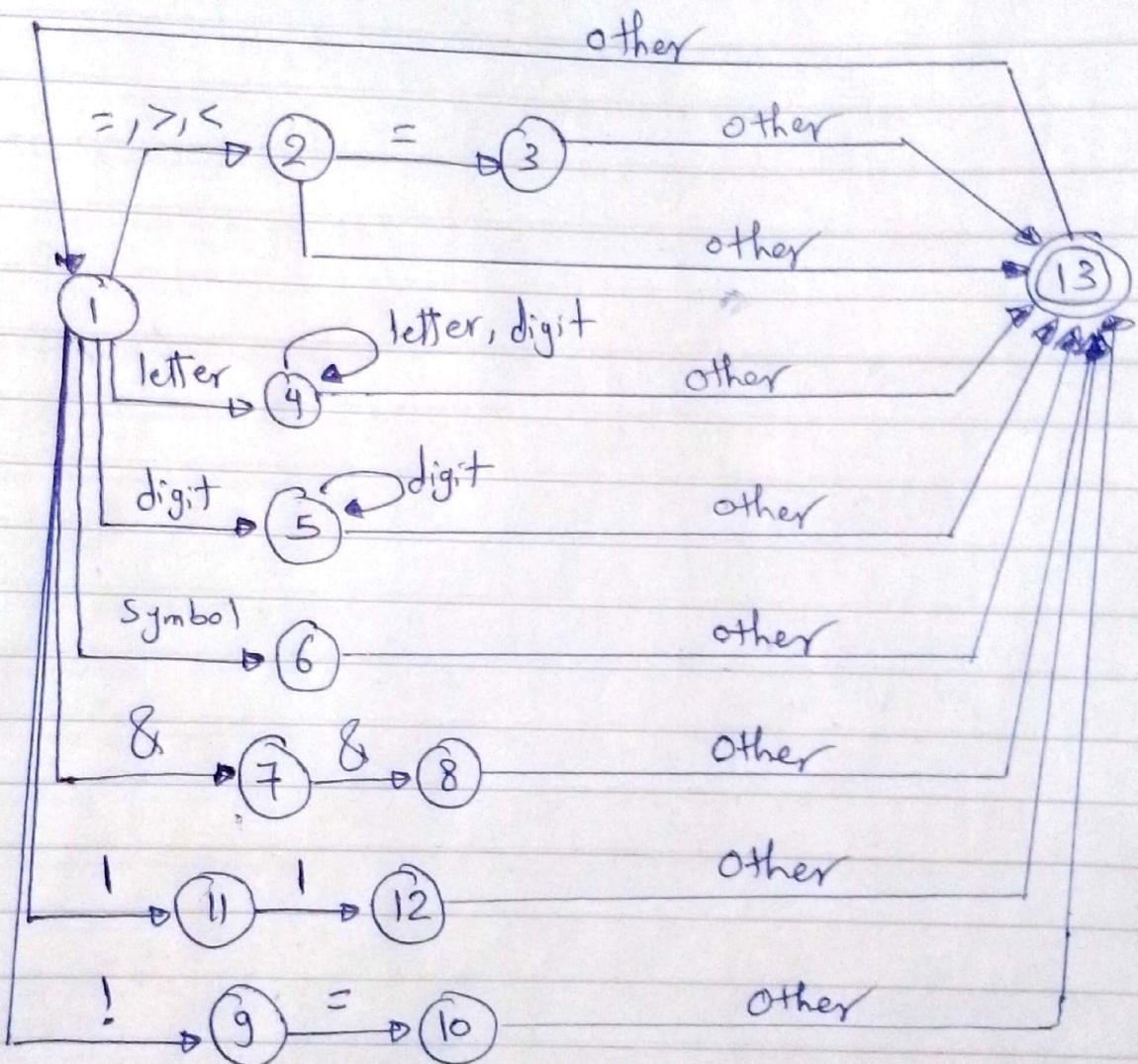
NFA for words.



DFA for words



final DFA



State	(<,>)	Letter	Digit	(+,-,*,/ ,~,.,',{,},[,],(,))	&		=	!	Other (White Space , ;)	Accept
1	2	4	5	6	7	11	2	9	1	NO
2							3		13	NO
3									13	NO
4		4	4						13	NO
5			5						13	NO
6									13	NO
7					8					NO
8									13	NO
9						10				NO
10									13	NO
11						12				NO
12									13	NO
13								1		YES

Special Cases:

(-) if (--) => single line comment , move the pointer until line delimiter then return to start state .

(<) if (<*) => multi line comment , move the pointer until you get (*>) then return to start state .

Grammar Parser LL1

1. Program → Program ClassDeclaration End
2. ClassDeclaration → Category ID CD'
3. CD' → { Class_Implementation }
4. CD' → Derive { Class_Implementation }
5. Class_Implementation → Type ID DD
6. DD → (ParameterList) MD' Class_Implementation
7. DD → D'; VarDeclaration Class_Implementation
8. Class_Implementation → Comment Class_Implementation
9. Class_Implementation → using_command Class_Implementation
10. Class_Implementation → Func_Call Class_Implementation
11. Class_Implementation → ϵ
12. MethodDeclaration → Func Decl MD'
13. MD' → ;
14. MD' → { VarDeclaration Statements }
15. Func Decl → Type ID (ParameterList)
16. Type → Ilap
17. Type → Silap
18. Type → Clop
19. Type → Series
20. Type → Ilapf
21. Type → Silapf
22. Type → None
23. Type → Logical
24. ParameterList → ϵ
25. ParameterList → None
26. ParameterList → Non-Empty List
27. Non-Empty List → Type ID NO'
28. NO' → , Type ID NO'
29. NO' → ϵ

- 30. $\text{VarDeclaration} \rightarrow \epsilon$
- 31. $\text{VarDeclaration} \rightarrow \text{Type ID_List ; VarDeclaration}$
- 32. $\text{ID_List} \rightarrow \text{ID D'}$
- 33. $\text{D'} \rightarrow , \text{ID D'}$
- 34. $\text{D'} \rightarrow \epsilon$
- 35. $\text{Statements} \rightarrow \epsilon$
- 36. $\text{Statements} \rightarrow \text{Statement Statements}$
- 37. $\text{Statement} \rightarrow \text{Assignment}$
- 38. $\text{Statement} \rightarrow \text{If_Statement}$
- 39. $\text{Statement} \rightarrow \text{Rotatewhen_Statement}$
- 40. $\text{Statement} \rightarrow \text{Continuewhen_Statement}$
- 41. $\text{Statement} \rightarrow \text{Replywith_Statement}$
- 42. $\text{Statement} \rightarrow \text{terminatethis_Statement}$
- 43. $\text{Statement} \rightarrow \text{read (ID);}$
- 44. $\text{Statement} \rightarrow \text{write (Expression);}$
- 45. $\text{Assignment} \rightarrow \text{VarDeclaration} = \text{Expression};$
- 46. $\text{Func_Call} \rightarrow \text{ID (Argument_List) ;}$
- 47. $\text{Argument_List} \rightarrow \epsilon$
- 48. $\text{Argument_List} \rightarrow \text{NonEmpty_Argument_List}$
- 49. $\text{NonEmpty_Argument_List} \rightarrow \text{Expression E'}$
- 50. $\text{E'} \rightarrow , \text{Expression E'}$
- 51. $\text{E'} \rightarrow \epsilon$
- 52. $\text{Block Statements} \rightarrow \{ \text{statements} \}$
- 53. $\text{If_Statement} \rightarrow \text{if (Condition_Expression) Block Statements}$
- 54. $\text{Condition_Expression} \rightarrow \text{Condition CO'}$
- 55. $\text{CO'} \rightarrow \epsilon$
- 56. $\text{CO'} \rightarrow \text{Condition_Op Condition}$
- 57. $\text{Condition_Op} \rightarrow \&&$
- 58. $\text{Condition_Op} \rightarrow ||$
- 59. $\text{Condition} \rightarrow \text{Expression Comparison_Op Expression}$

- 60. Comparison _Op → ==
 - 61. Comparison _Op → !=
 - 62. Comparison _Op → >
 - 63. Comparison _Op → >=
 - 64. Comparison _Op → <
 - 65. Comparison _Op → <=
 - 66. Rotatewhen _Statement → Rotatewhen(Condition _Expression)
 Block Statements
 - 67. Continuewhen _Statement → Continuewhen (expression ;
 expression ; expression) Block Statements
 - 68. Replywith _Statement → Replywith Expression ;
 - 69. Replywith _Statement → return ID ;
 - 70. terminatethis _Statement → terminatethis;
 - 71. Expression → Term EX'
 - 72. EX' → Add Op Term EX'
 - 73. EX' → ε
 - 74. Add Op → +
 - 75. Add Op → -
 - 76. Term → Factor T'
 - 77. T' → Mul_Op Factor T'
 - 78. T' → ε
 - 79. Mul_Op → *
 - 80. Mul_Op → /
 - 81. Factor → ID
 - 82. Factor → Number
 - 83. Comment → <* STR *>
 - 84. Comment → -- STR
 - 85. using_command → using(F_name.txt);
 - 86. F_name → STR
-

