CS 419 Compiler Projects Form

1)Instructions to be Followed (for the Hard Copy):

I. Fill page **NO 4** with the required Fields:

- **I.** Project Idea: The **Idea** will be assigned to you .
- **II.** Team Members NO#: Number of team members **7**
- **III.** Table:
  - ID: Your FCIH ID.
  - Name: Your **Full Name** as registered on College's Database.
  - Level & Department: Your **(Current)** level and department.
  - Section(Day-from-to): Your Section Day and time slot.
  - Role: Your role in project (**Team leader** OR **Member**).
  - Fill Page **NO 5 & 6** with your (**Regular Expressions**, **Finite automata**, **Parse trees** and **abstract syntax tree**) respectively.

2) Minus Policies:

I. **Project Policy:** affects all projects members including team leader.

II. **Member Policy:** affects a member of project's members.

3) General Notes:

I. Total grade of Project is 15

II. **Deadline** to register yourself and your team on online form **Tuesday 05/04/2022 at 11:59 PM** after that **-2 Project Policy** will be applied**.**

III. Once you Registered, **NO modifications** will be done.

IV. Allowed only on registration for team in form, duplication will got **-2 Project Policy.**

V. Each group will be assigned **an Idea, ID and time slot** for **discussion**.

VI. Each team member and team leader in a team **must work in project's coding phase** (including implementation of **finite automata and parse trees**).

4) Discussion Notes:

I. **Copied Code** will be got ZERO Without Discussion.

II. By references to Section 3 (General notes) Point V , **-5 Member policy** will be applied to each team member (including team leader) who does not participate in project coding phase **as well as the team leader who does not report this case**.

III. Each team member must have **a complete knowledge** about the whole project

IV. **Evaluation** will be **Individual Evaluation** not project Evaluation.

V. **-2 Project Policy** will be applied in case of being late for assigned discussion time slot

VI. **NO discussion will be repeated under any circumstances.**

VII. At Discussion day, in case of offline discussions, each team must have **Hard Copy form** including (**Finite automata and parse trees of team's project**).

VIII. Discussion Day will be **sent later .**

5) Notes about Implementation:

I. **.Net or PHP** are only allowed.

II. The project must be a Web (use latest technologies).

III. Your code must be uploaded to github before discussion.

IV. **- 5 Project policy** will be applied in case of using **Built-in Method** within implementation of the scanner or parser, you must create your **own methods to match** for ex (your regular Expressions).

V. Each Project must contain a full functional editor (comment, uncomment, put red line under wrong words, auto complete, navigation to function or class, line NO).

VI. Each Project must contain **two buttons** , one button called “**Scan**” to run scanner and other called “**Parse**” to run parser –parser must take output of scanner to do it’s task.

VII. Each project must contain a button named “**Browse**” that allows us to choose a **file from a disk** that allows us to parse or scan this file **Without Showing what is inside the file** and shows the output.

VIII. – 3 Project policy will be applied if the content of the file that is mentioned in point V is opened or viewed.

5) Notes about Discussion Testing:

- There will be two types of Testing :

  I. **White Box Testing:** This will be from **Editor**.

  II. **Black Box Testing:** This will be from “**Browse**” Button

Thanks,

CS 419 Compiler

Project Form

Project Idea: #2

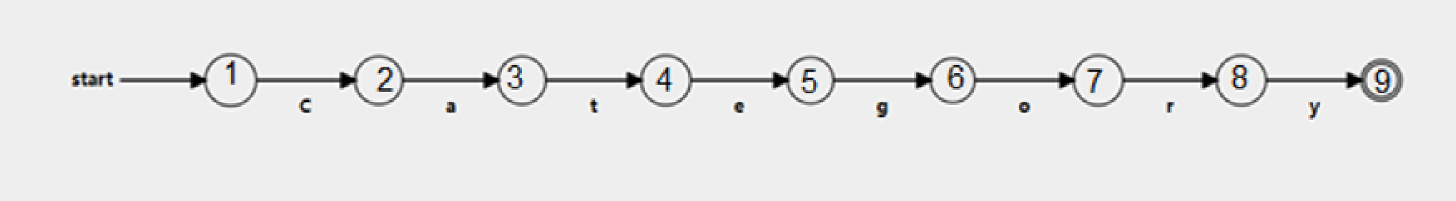
Team Members NO#:7

ID	Name	Level& Department	Section(Day-from-to)	Role (Lead/Member)	Grade
201900547	غيداء الطاهر احمد محمد	3	4-6- Wednesday	**Lead**	
201900573	كاترين حبيب جورج ناداب	3	4-6 Wednesday	**Member**	
201900568	فريدة أحمد توكو			**Member**	
201900554	فاروق ابراهيم فاروق عبداللطيف		4-6 Wednesday	**Member**	
201900534	عمر نبيل صابر ندا		4-6 Wednesday	**Member**	
201900386	طارق ايمن الدسوقي طلبة		Thursday 8-10	**Member**	
20170433	محمد حسين عبدالعال قابيل		**-**	**Member**	

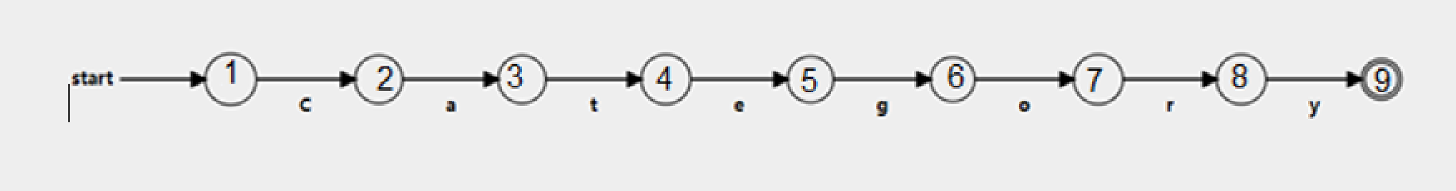
Regular Expression, Finite automata and Conversion from RegX to NFA, NFA to DFA

RE : Category -> Category

NFA :

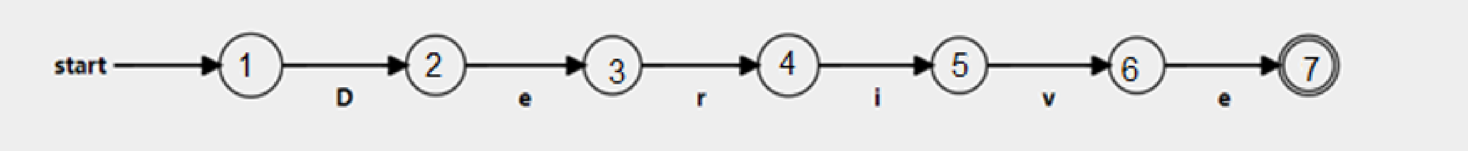


DFA :

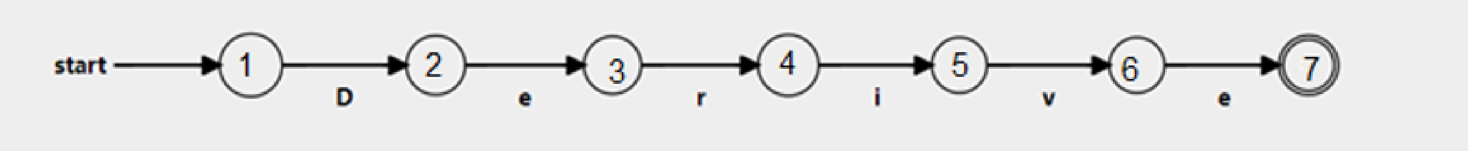


RE : Derive -> Derive

NFA :

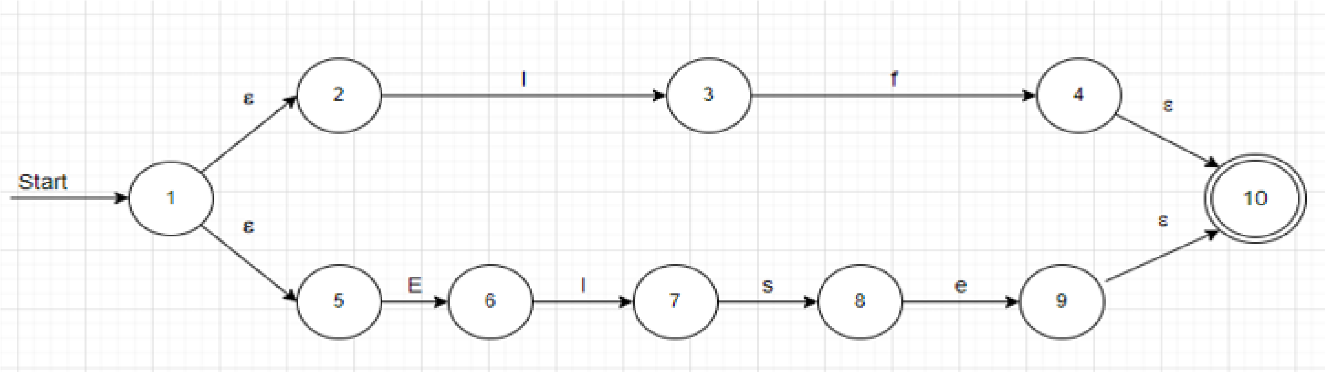


DFA :

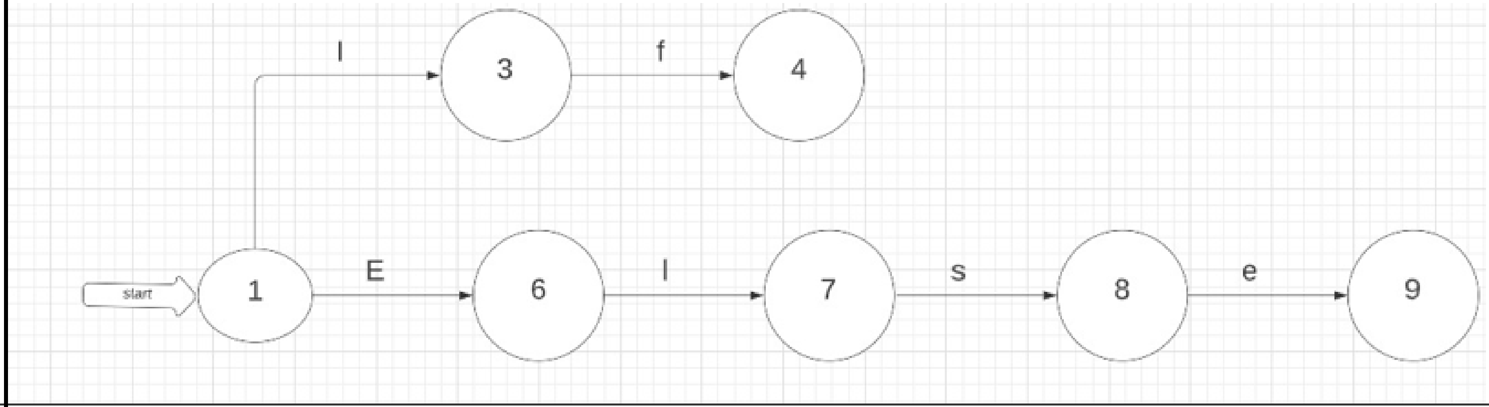


RE : Condtion -> If | Else

NFA :

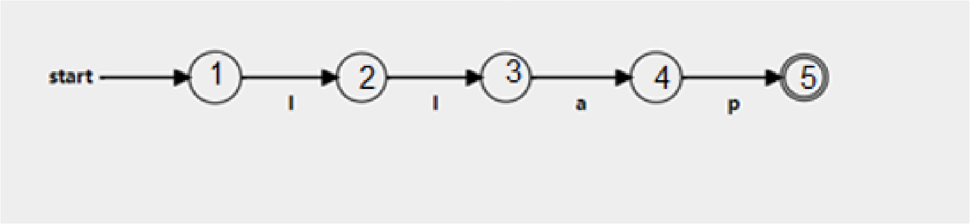


DFA :

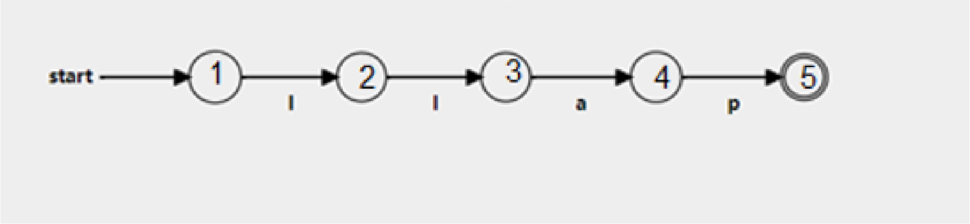


RE : llap -> llap

NFA :

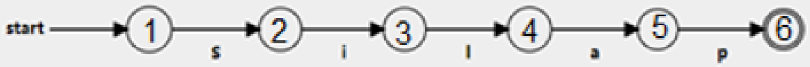


DFA :

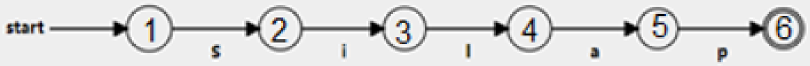


RE : Silap -> Silap

NFA :

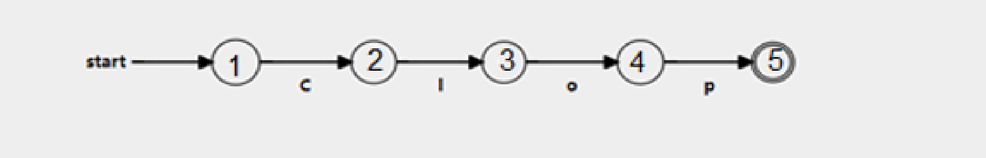


DFA :

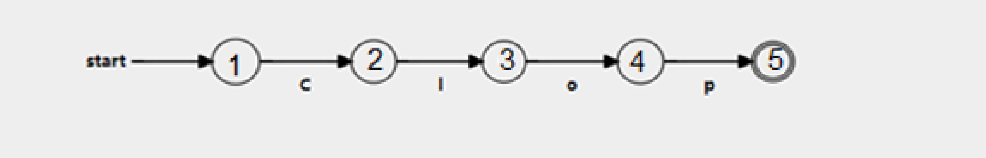


RE : Clop -> Clop

NFA :

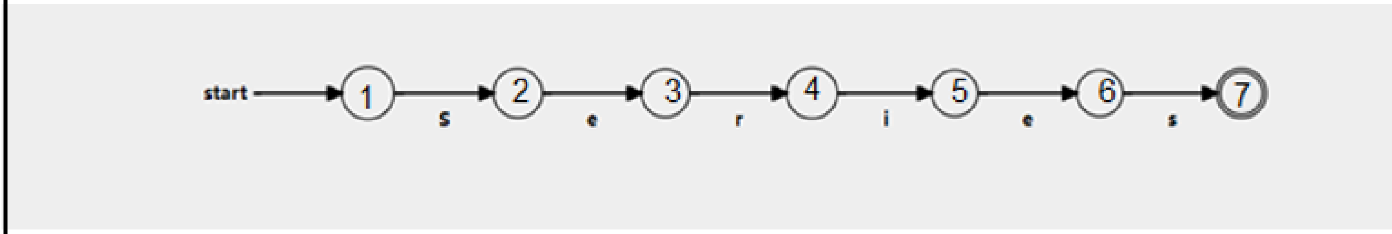


DFA :

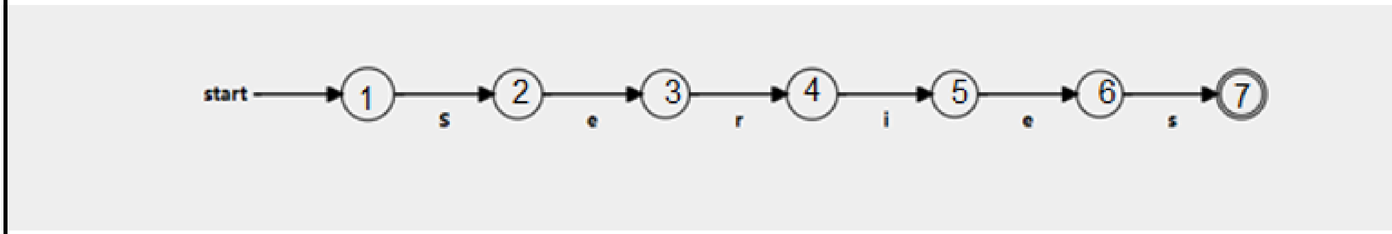


RE : Series -> Series

NFA :

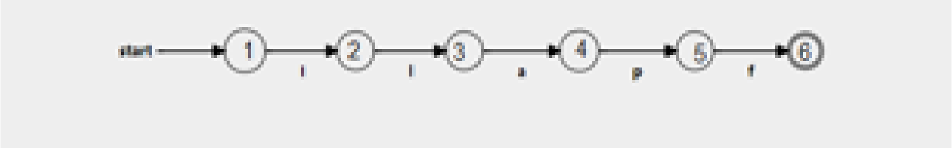


DFA :

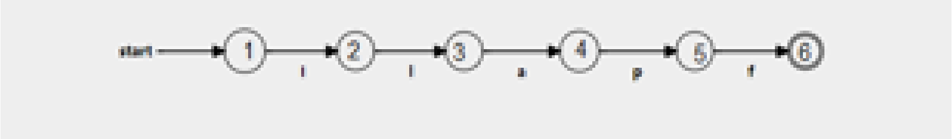


RE : llapf -> llapf

NFA :

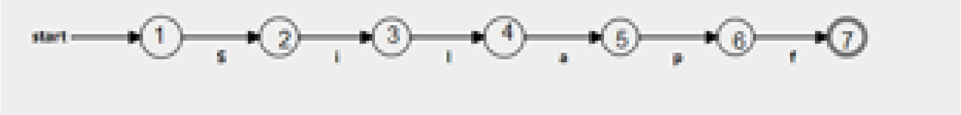


DFA :

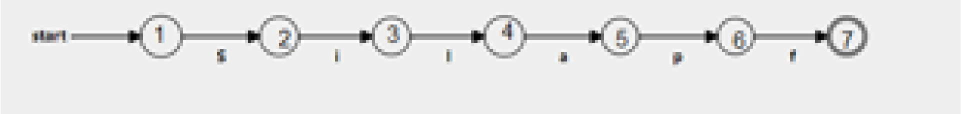


RE : Silapf - > Silapf

NFA :

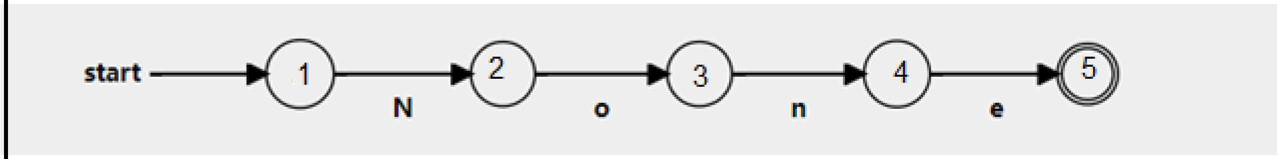


DFA :

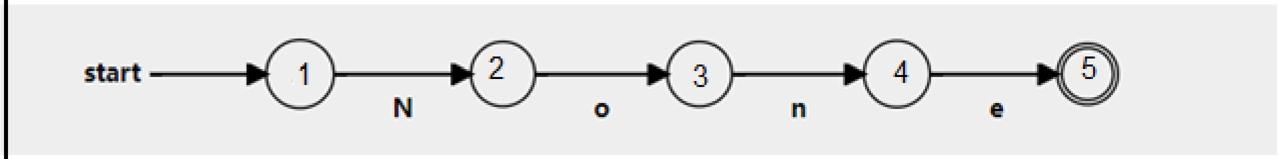


RE : None -> None

NFA :

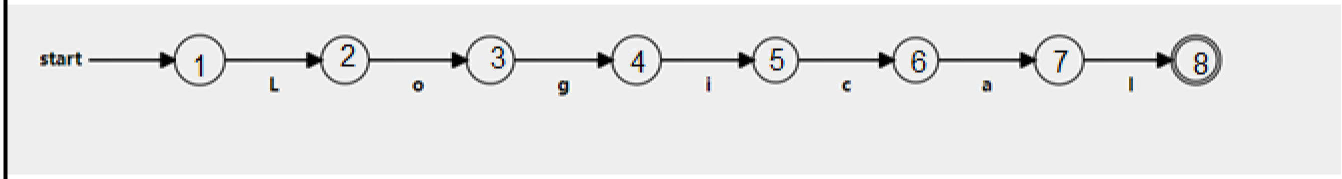


DFA :

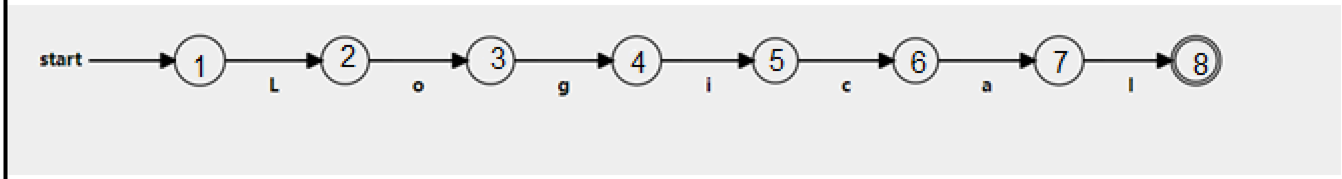


RE : Logical -> Logical

NFA :

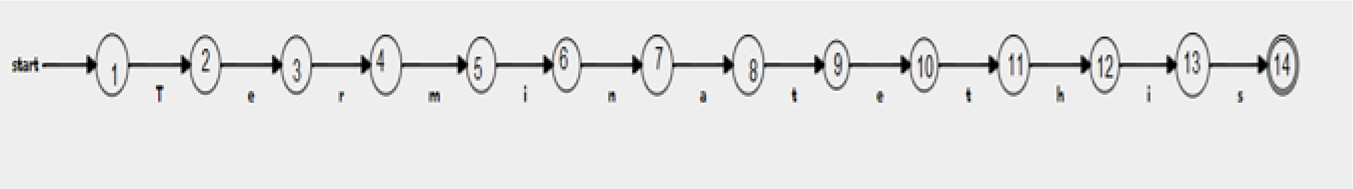


DFA :

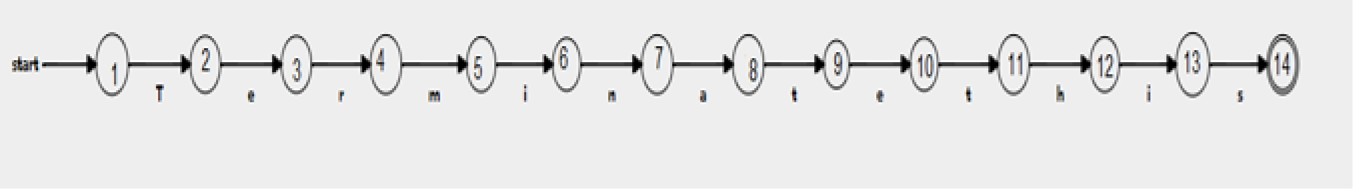


RE : Terminatethis -> Terminatethis

NFA :

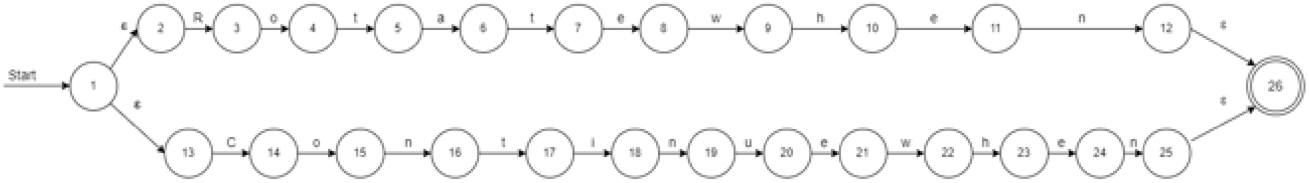


DFA :

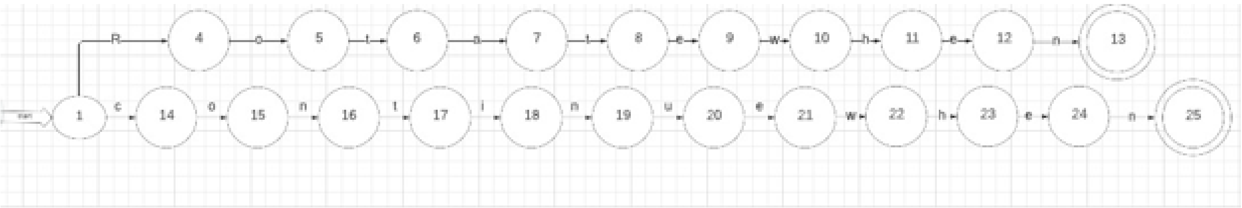


RE : Repeat -> Rotatewhen | Continuewhen

NFA :

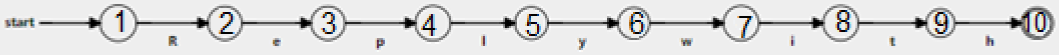


NFA :

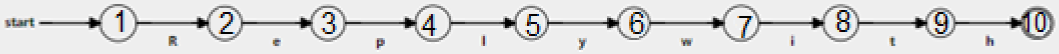


RE : Replywith -> Replywith

NFA :

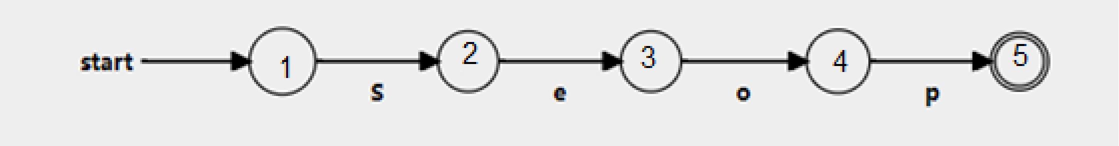


DFA :

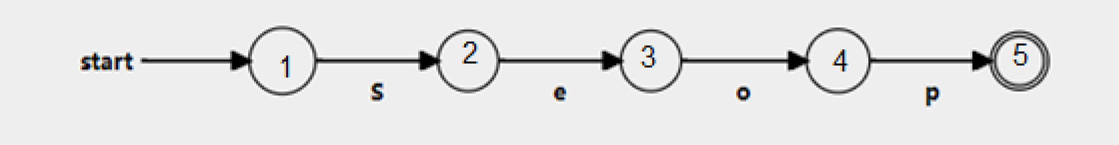


RE : Seop -> Seop

NFA :

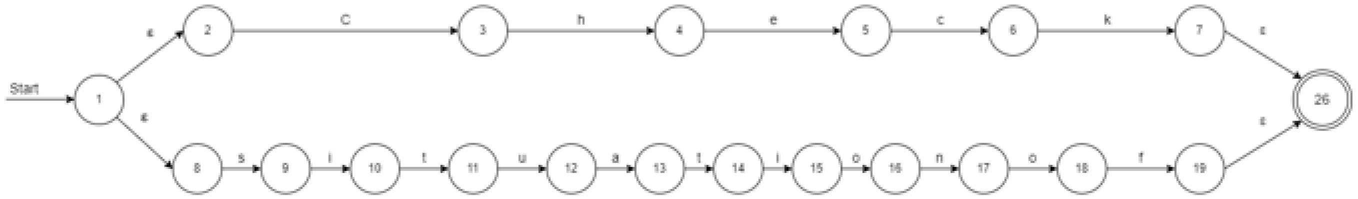


DFA :

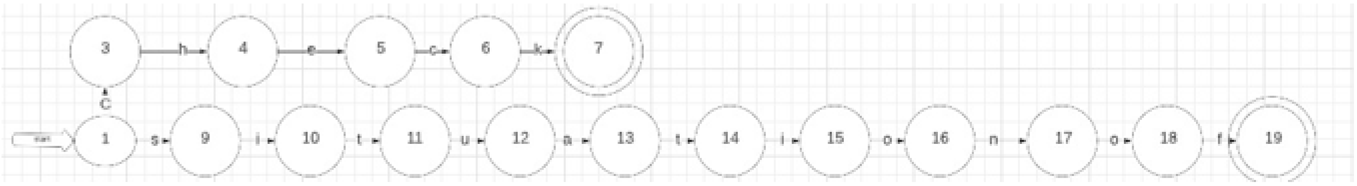


RE : Switch - > Check | situationof

NFA :

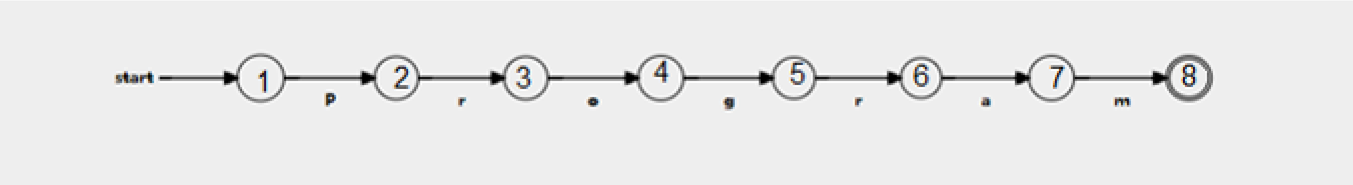


DFA :

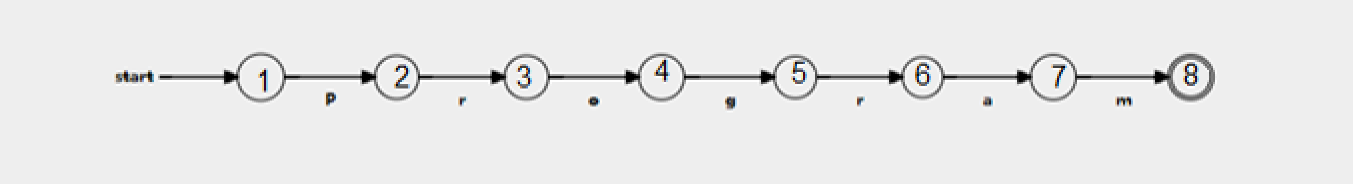


RE : Program -> Program

NFA :

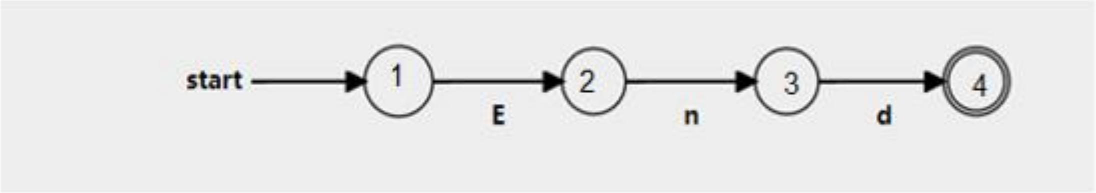


DFA :

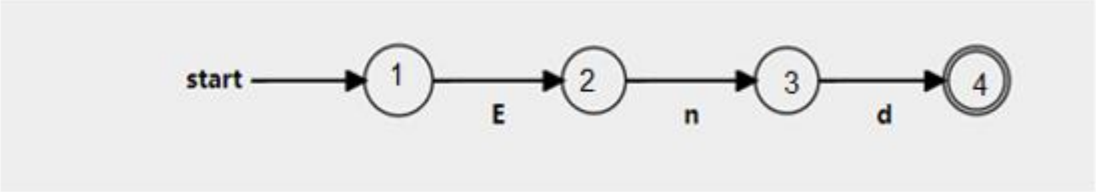


RE : End -> End

NFA :

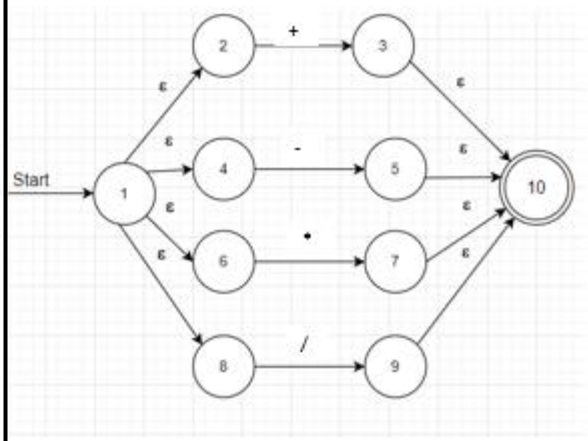


DFA :

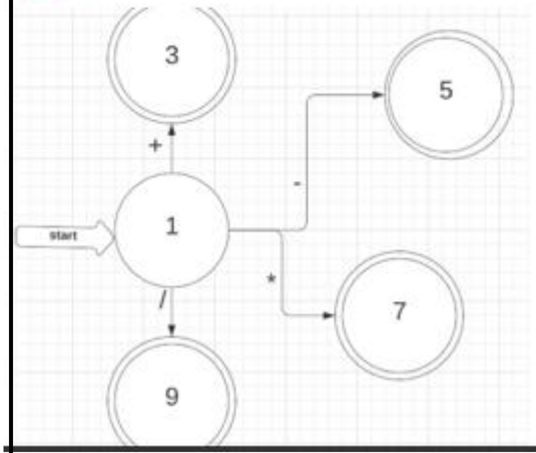


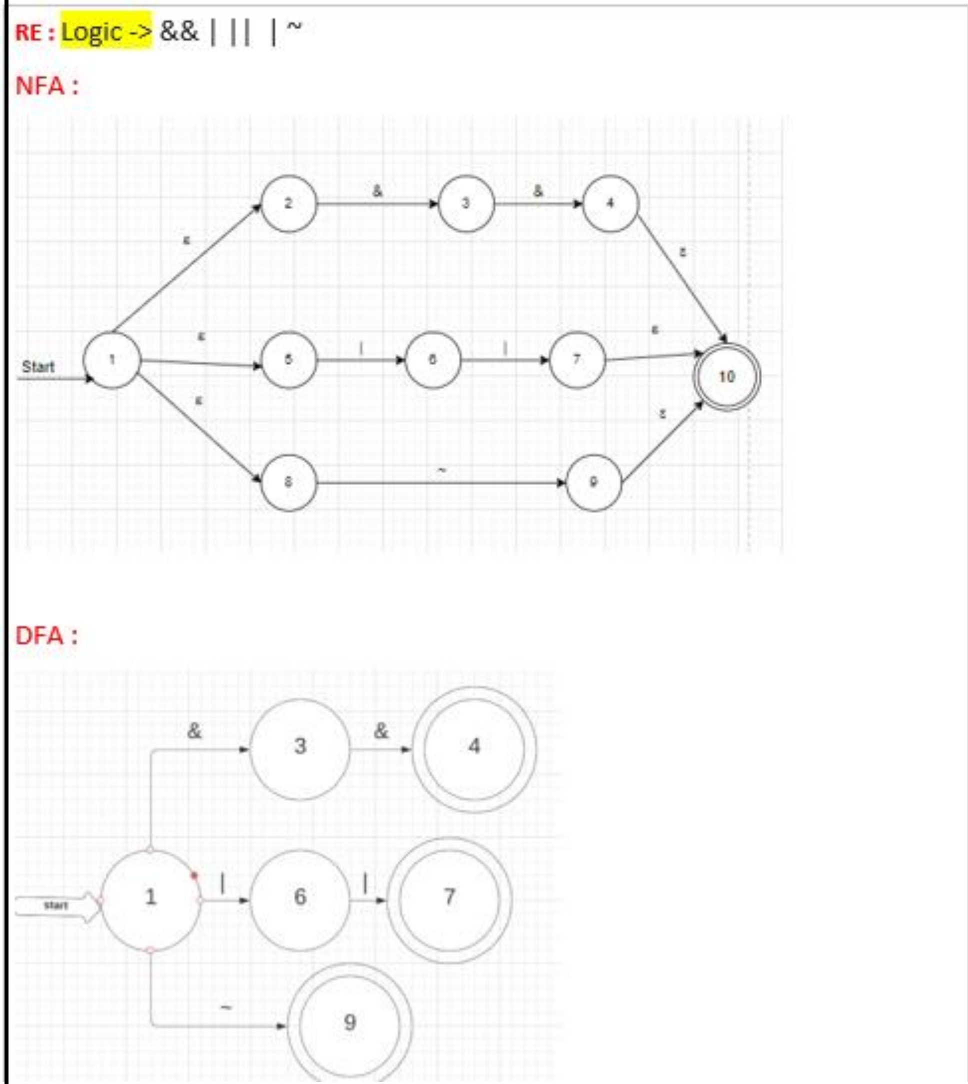
RE : Arithmetic -> + | - | * | /

NFA :



DFA :





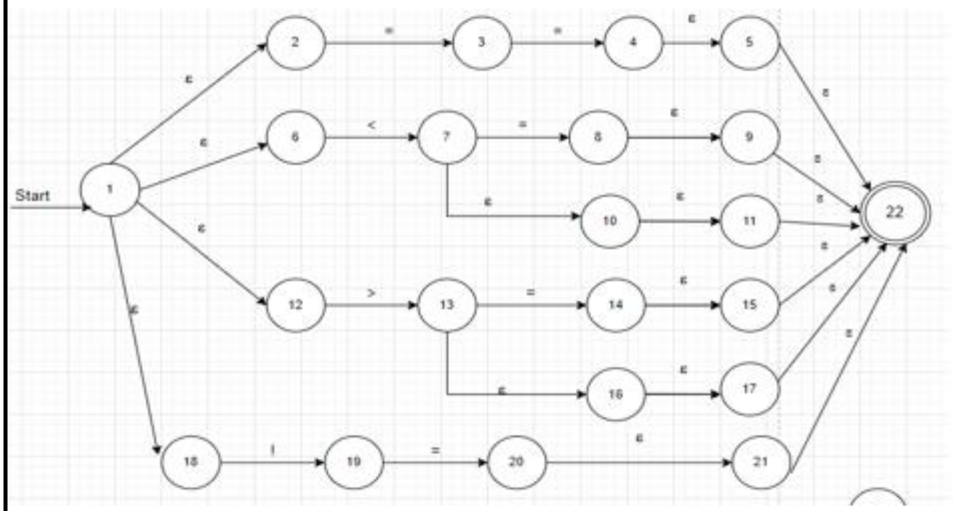
RE : Logic -> && | || | ~

NFA :

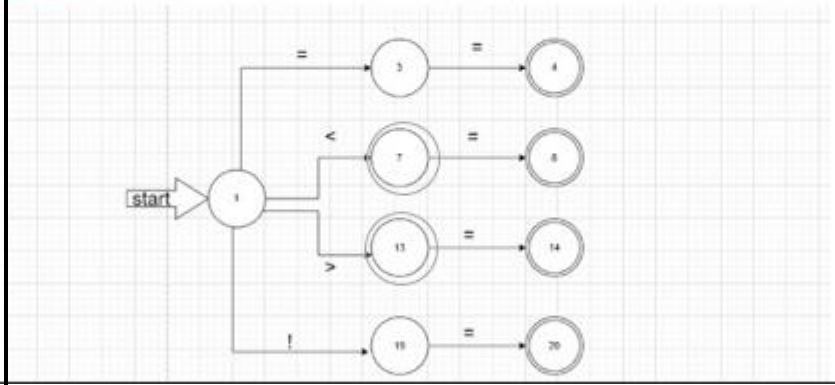
DFA :

RE : Relational -> == | < | > | != |<= | >=

NFA :

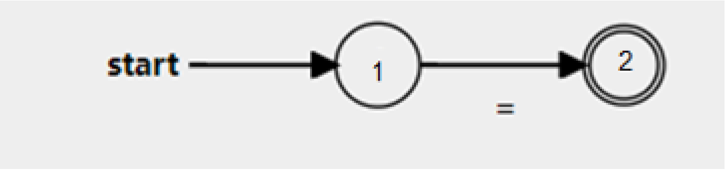


DFA :

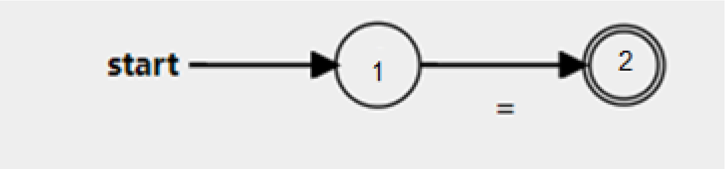


RE : Assignment -> =

NFA :

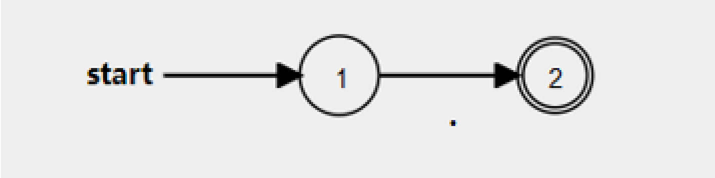


DFA :

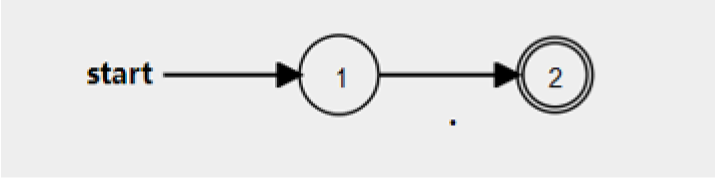


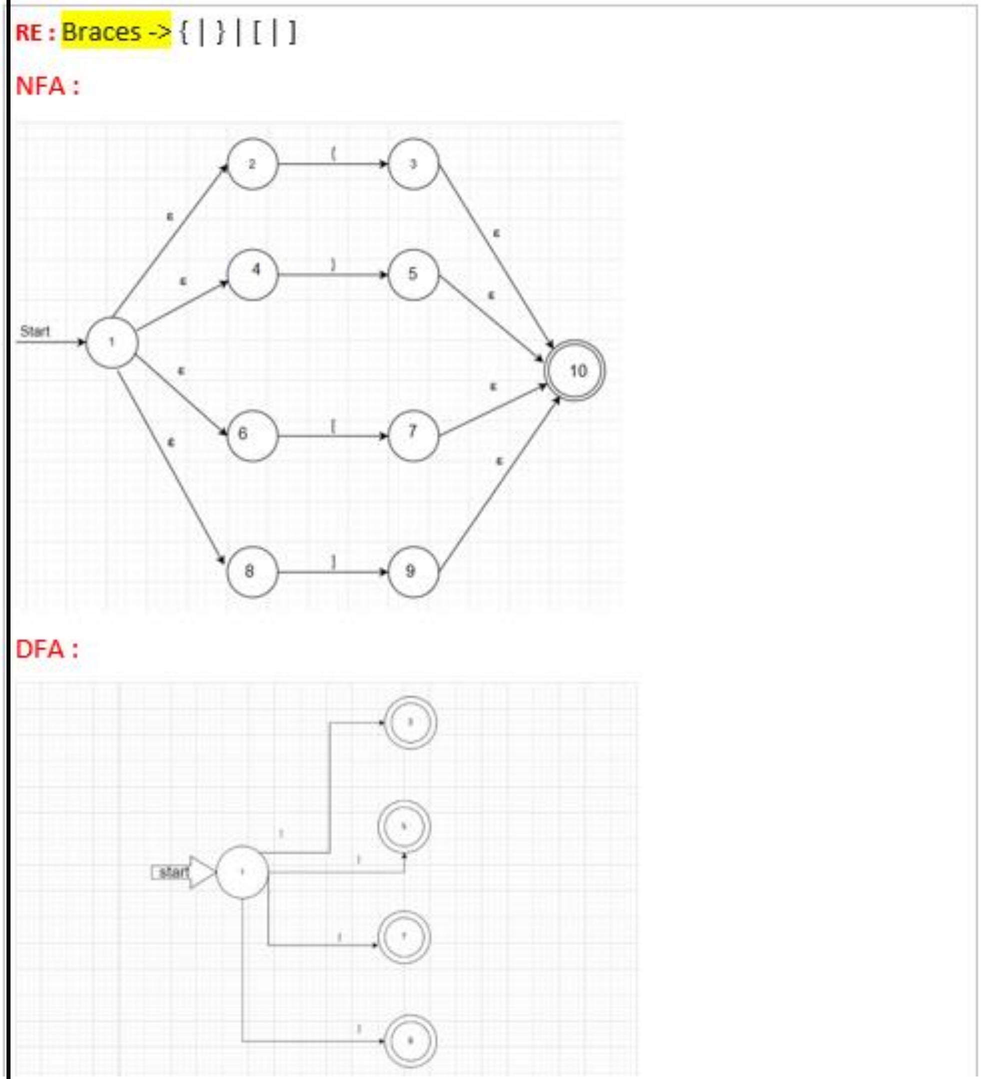
RE : Access -> .

NFA :



DFA :





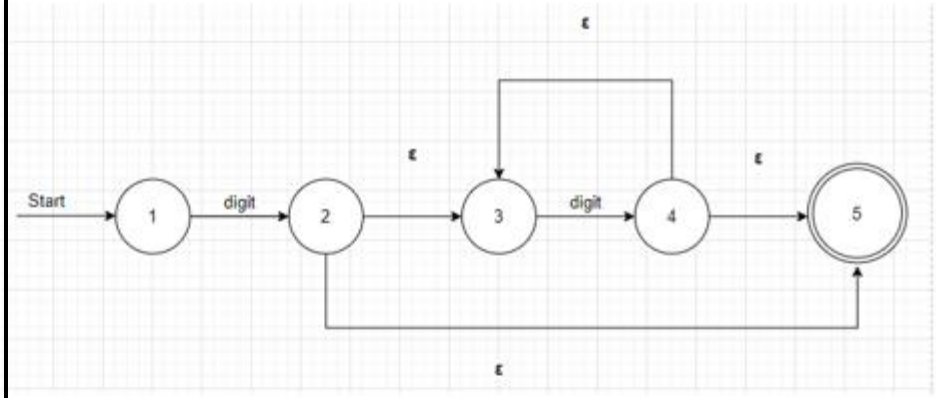
RE : Braces -> { | } | [|]

NFA :

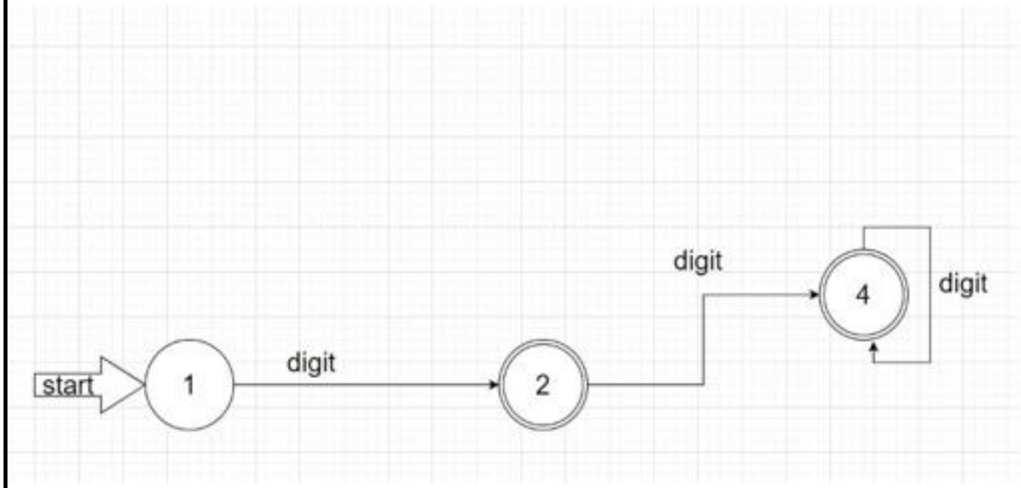
DFA :

Digits -> digit digit*

NFA :

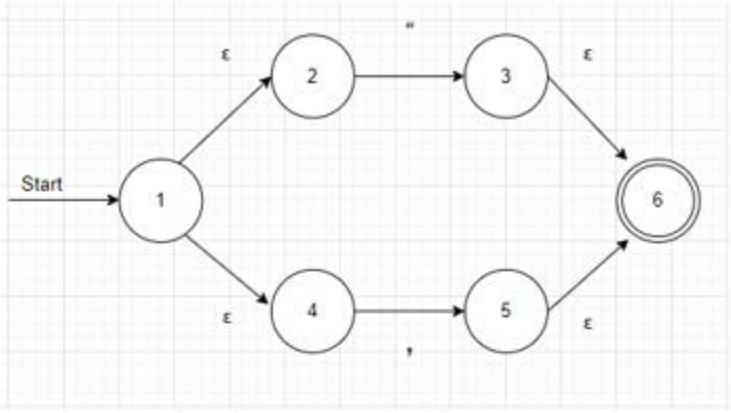


DFA :

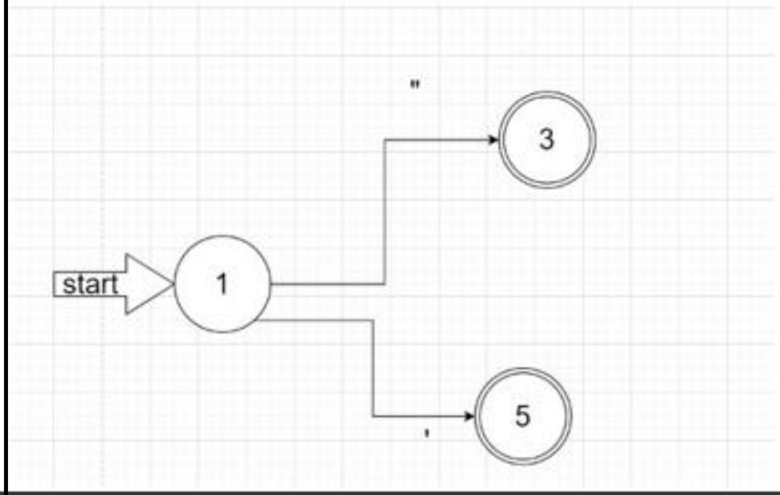


RE : Quotation -> “ | ’

NFA :

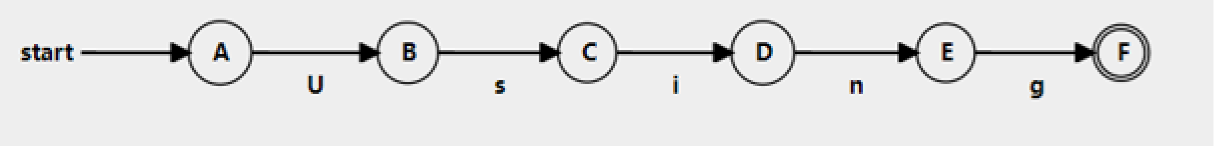


DFA :

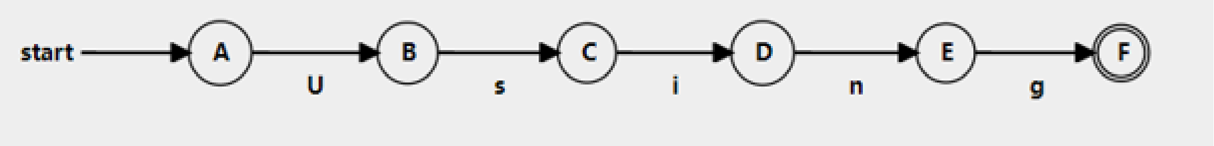


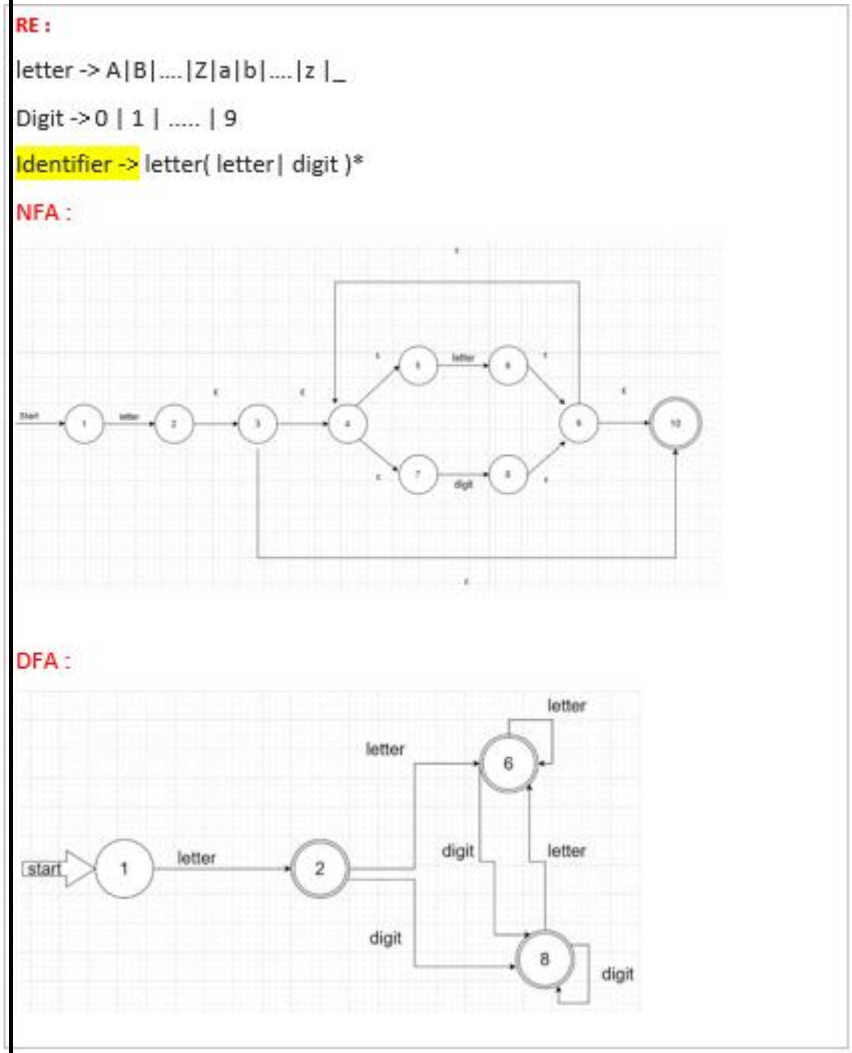
RE : Using -> Using

NFA :



DFA :





RE :

letter -> A|B|….|Z|a|b|….|z |_

Digit -> 0 | 1 | ….. | 9

Identifier -> letter(letter| digit)*

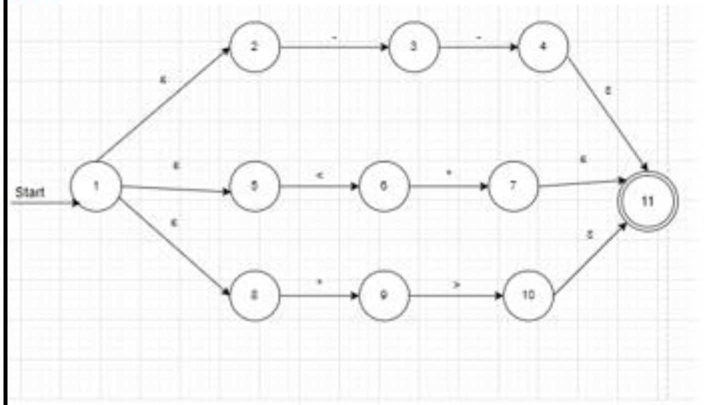
NFA :

DFA :

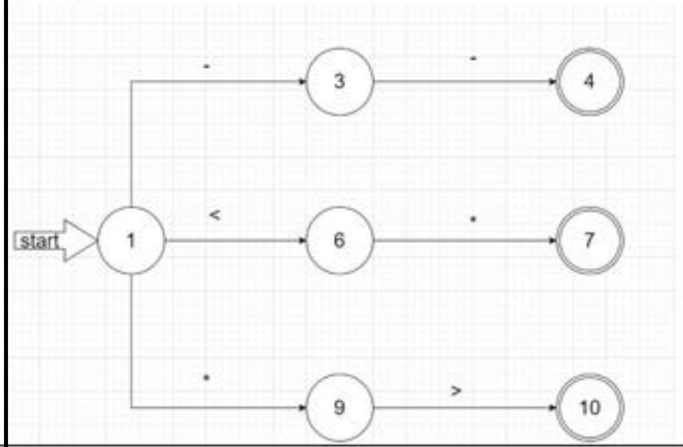
RE :

Comment -> -- | <* | *>

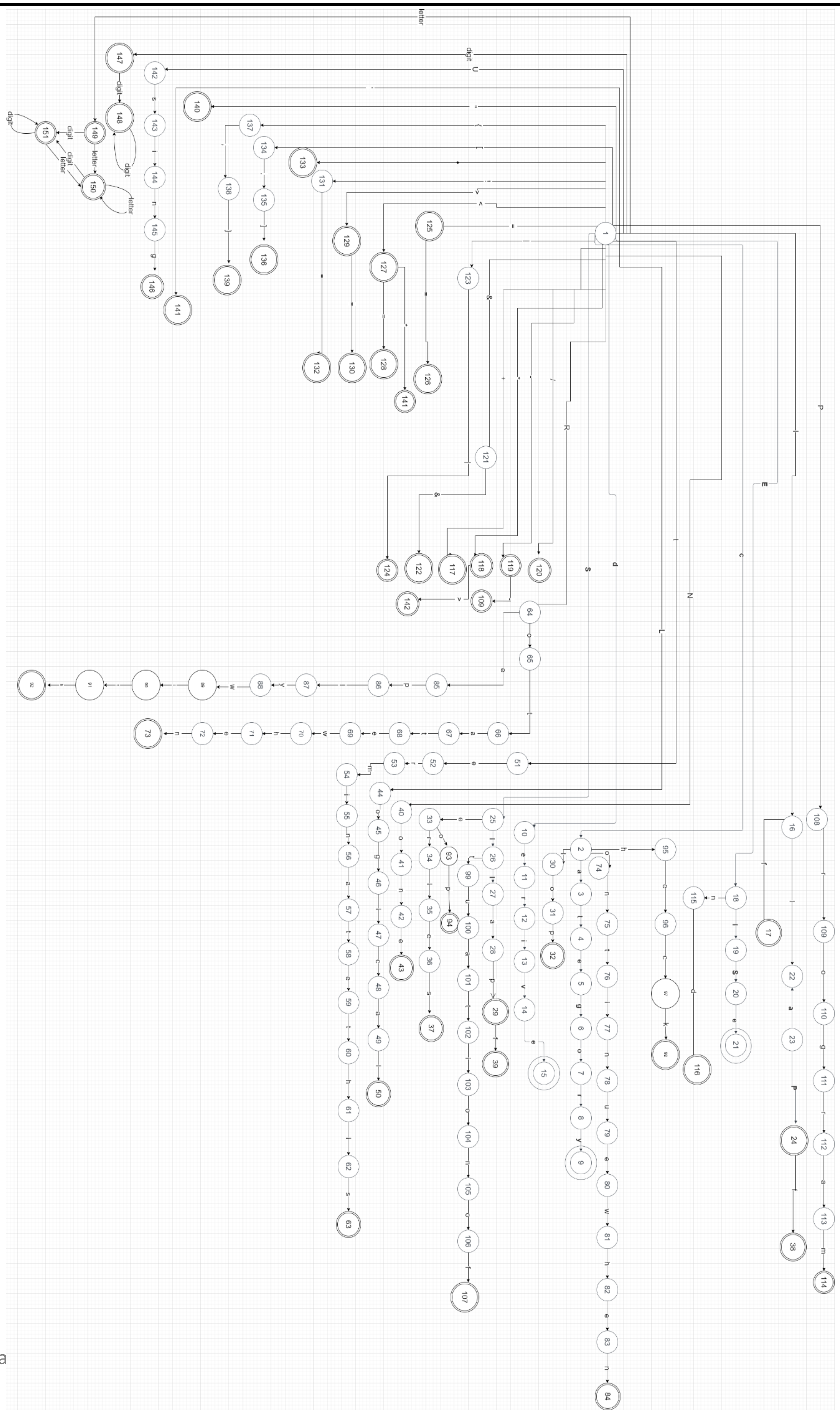
NFA :



DFA :



DFA :


letter
digit
U
147
148
149
150
151
digit
letter
142
s
143
i
144
n
145
g
146
141
140
137
134
133
131
138
139
135
136
129
127
125
126
128
130
132
141
123
&
121
&
124
122
117
118
119
120
142
109
1
P
E
R
S
d
N
L
c
64
o
65
e
85
p
86
87
y
88
w
89
90
91
92
t
66
a
67
68
e
69
w
70
h
71
e
72
n
73
51
e
52
r
53
m
54
55
n
56
a
57
t
58
e
59
t
60
h
61
62
s
63
44
o
45
g
46
47
c
48
a
49
50
40
o
41
n
42
e
43
33
r
34
35
e
36
s
37
o
93
p
94
25
e
26
27
a
28
p
29
39
99
u
100
a
101
102
103
o
104
n
105
o
106
f
107
10
e
11
r
12
13
v
14
e
15
2
h
o
30
o
31
p
32
a
3
t
4
e
5
g
6
o
7
r
8
y
9
74
n
75
t
76
77
n
78
u
79
e
80
w
81
h
82
e
83
n
84
95
e
96
c
97
k
98
18
n
115
d
116
19
S
20
e
21
16
17
22
a
23
P
24
38
108
109
o
110
g
111
r
112
a
113
m
114

Parse tree and Abstract syntax tree