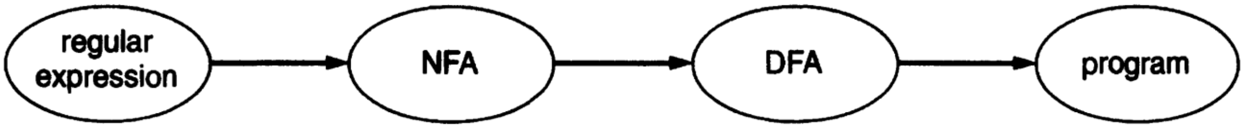
Compiler #1 Scanner

Team: 22

In our language we consider:
letter = [a-z] | [A-Z] | _
digit = [0-9]

We need specific patterns to represent identifiers, constants, relational operators, and comments.

Then, converting them to NFA and DFA to write the scanner program.

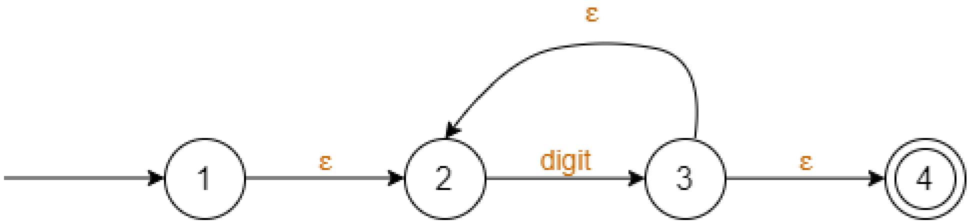


❖Regular Expressions:

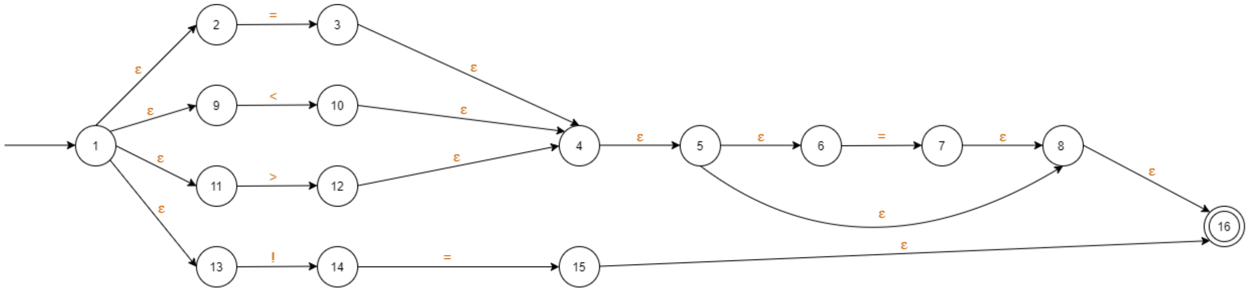
1- Constant -> digit$^+$
2- Relational Operator **-> ((=** | < | **>) =?) | !=**
3- Identifier **->** letter (letter | digit)*
4- Comment **->** / **(^ | @** (letter | digit)* **@/)**

❖NFAs:

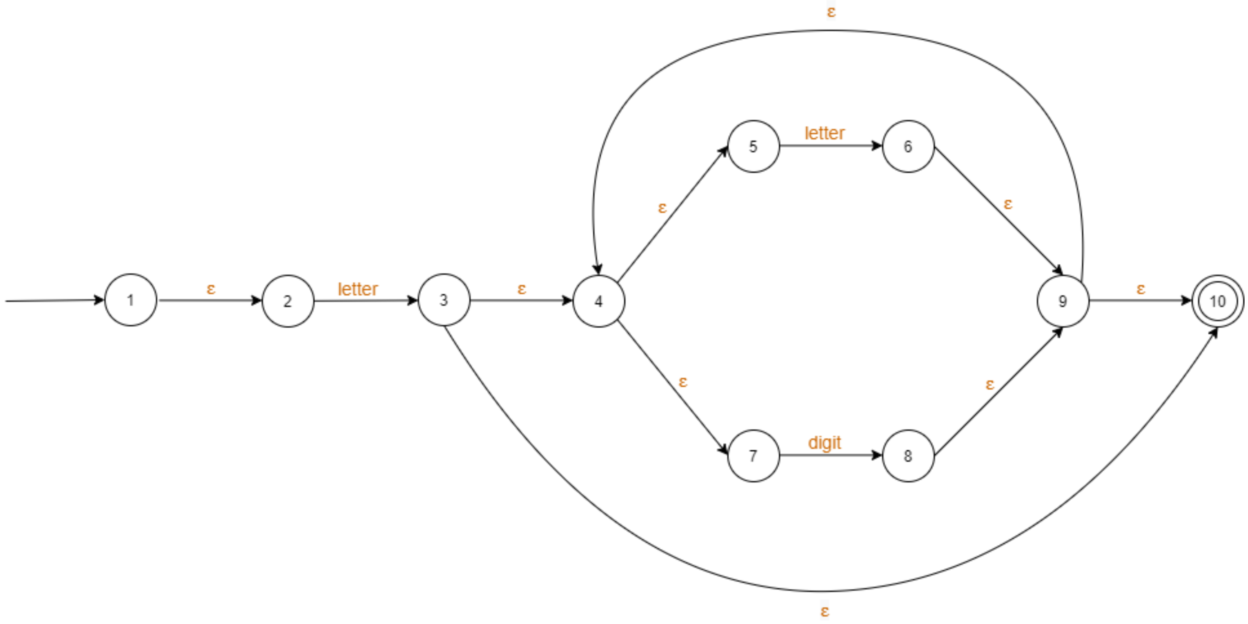
1- NFA for Constant:



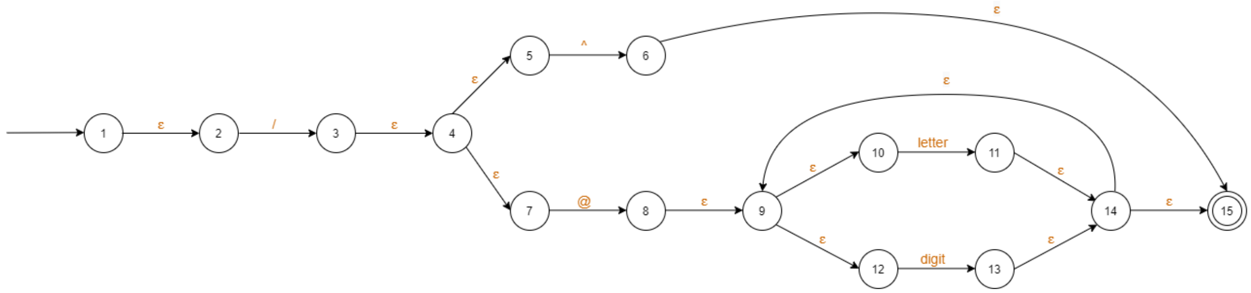
2- NFA for Relational Operators:



3- NFA for Identifiers:

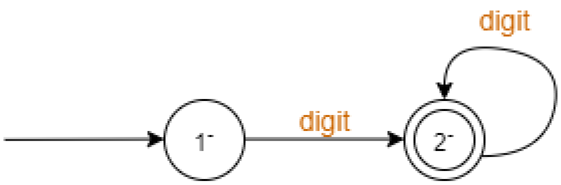


4- NFA for Comments:

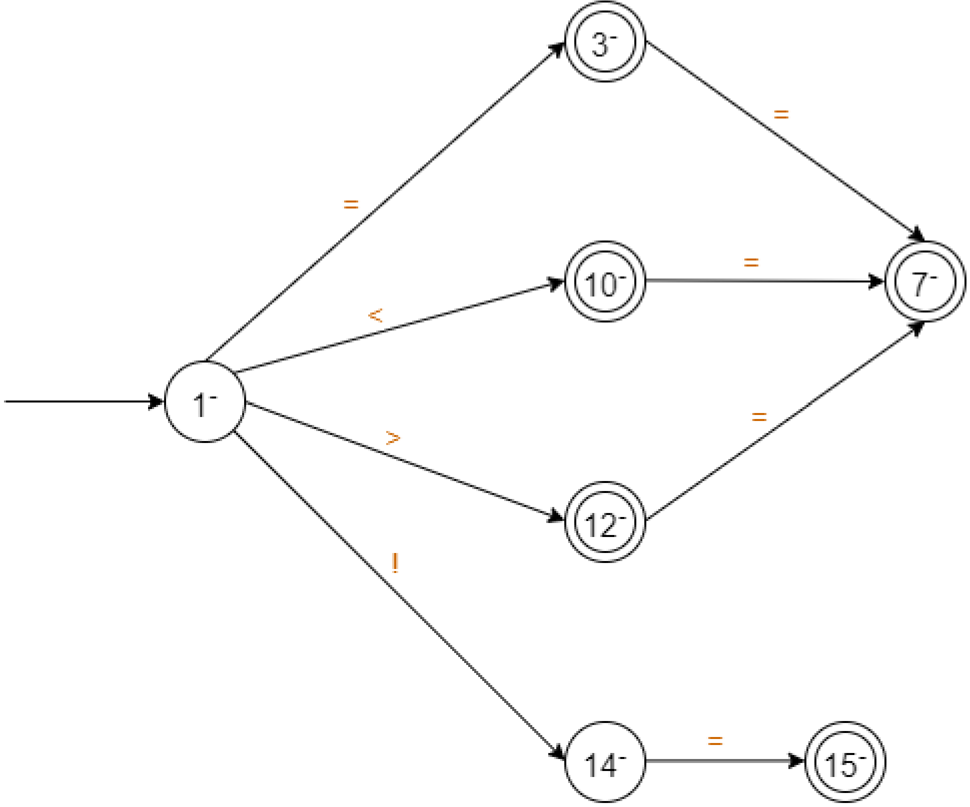


DFAs:

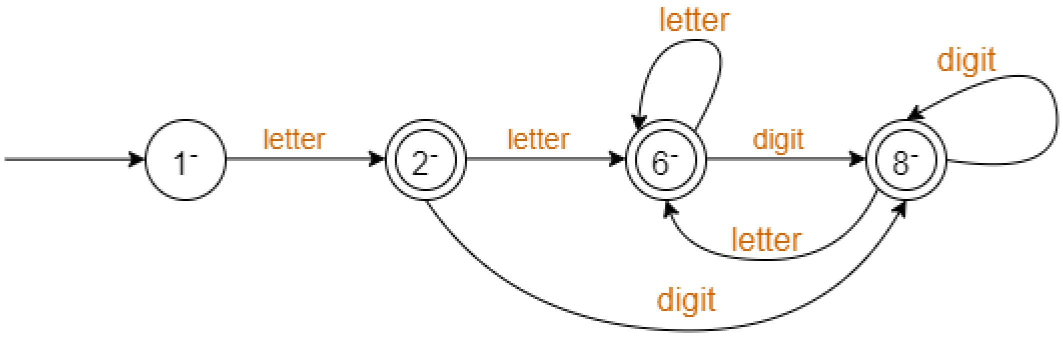
1- DFA for Constant:



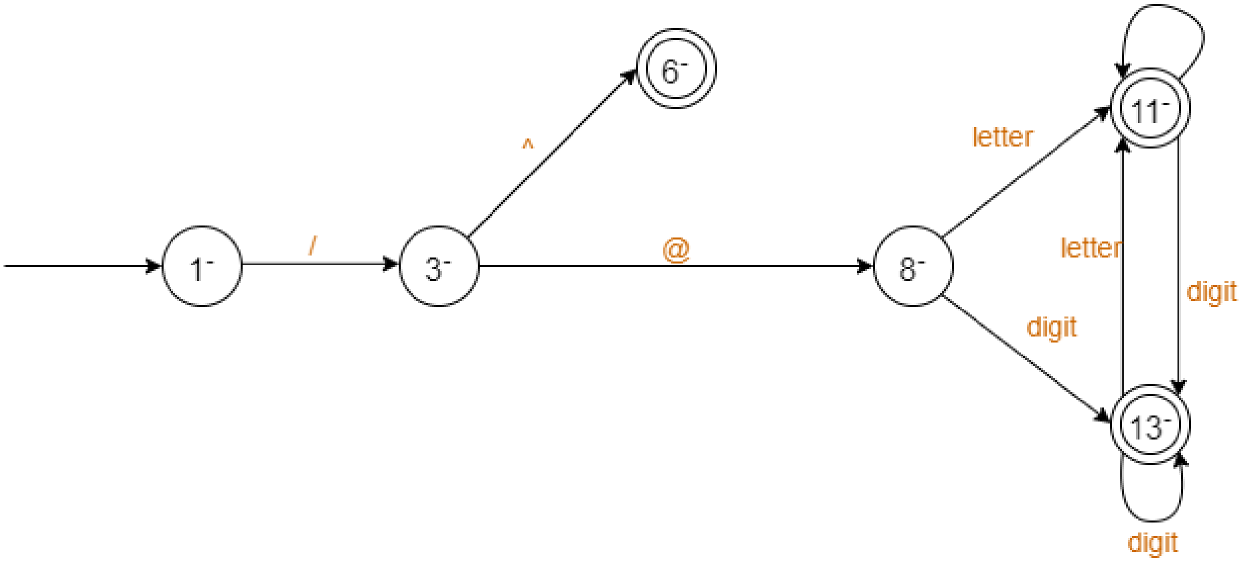
2- DFA for Relational Operators:



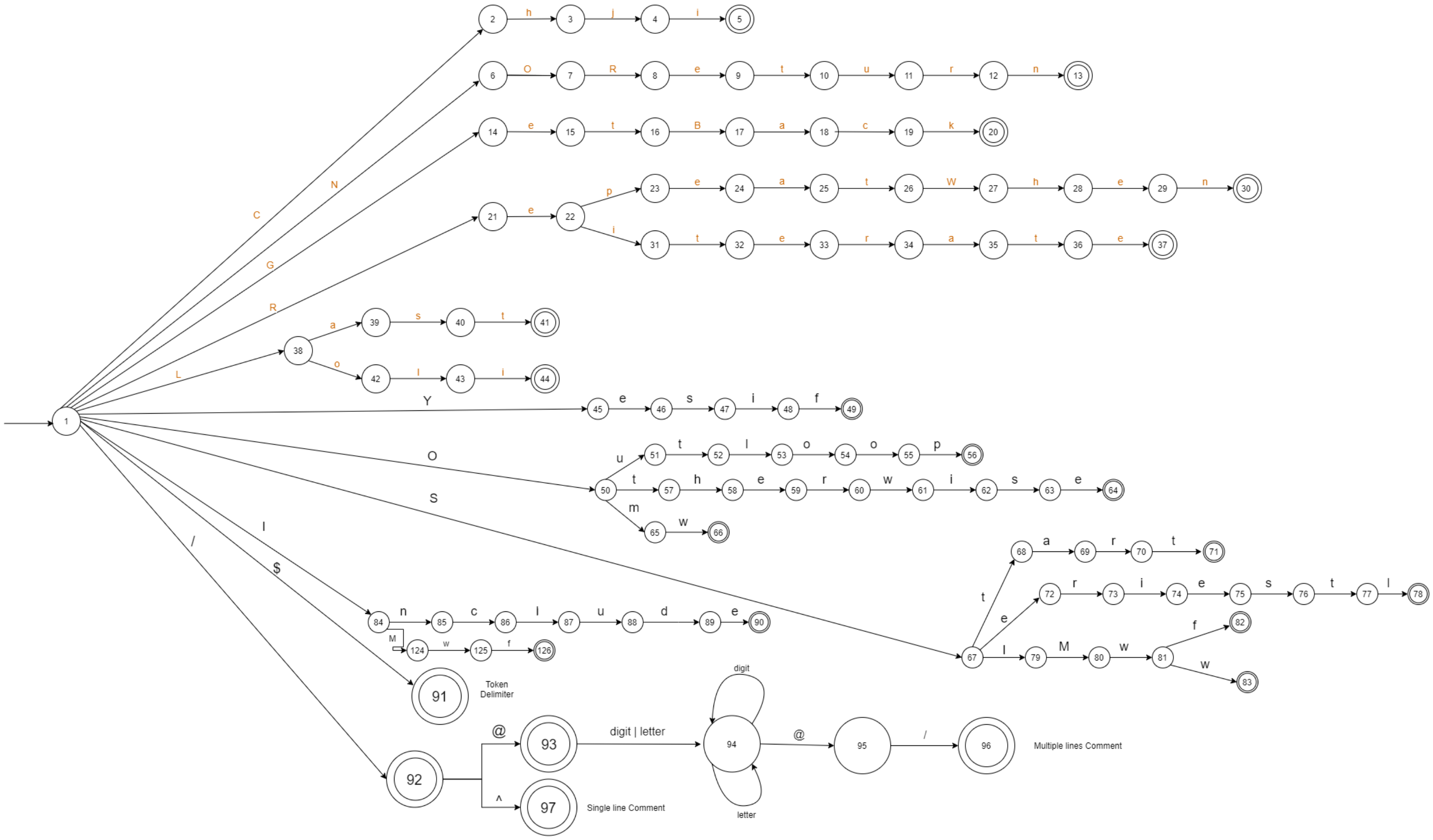
3- DFA for Identifiers:

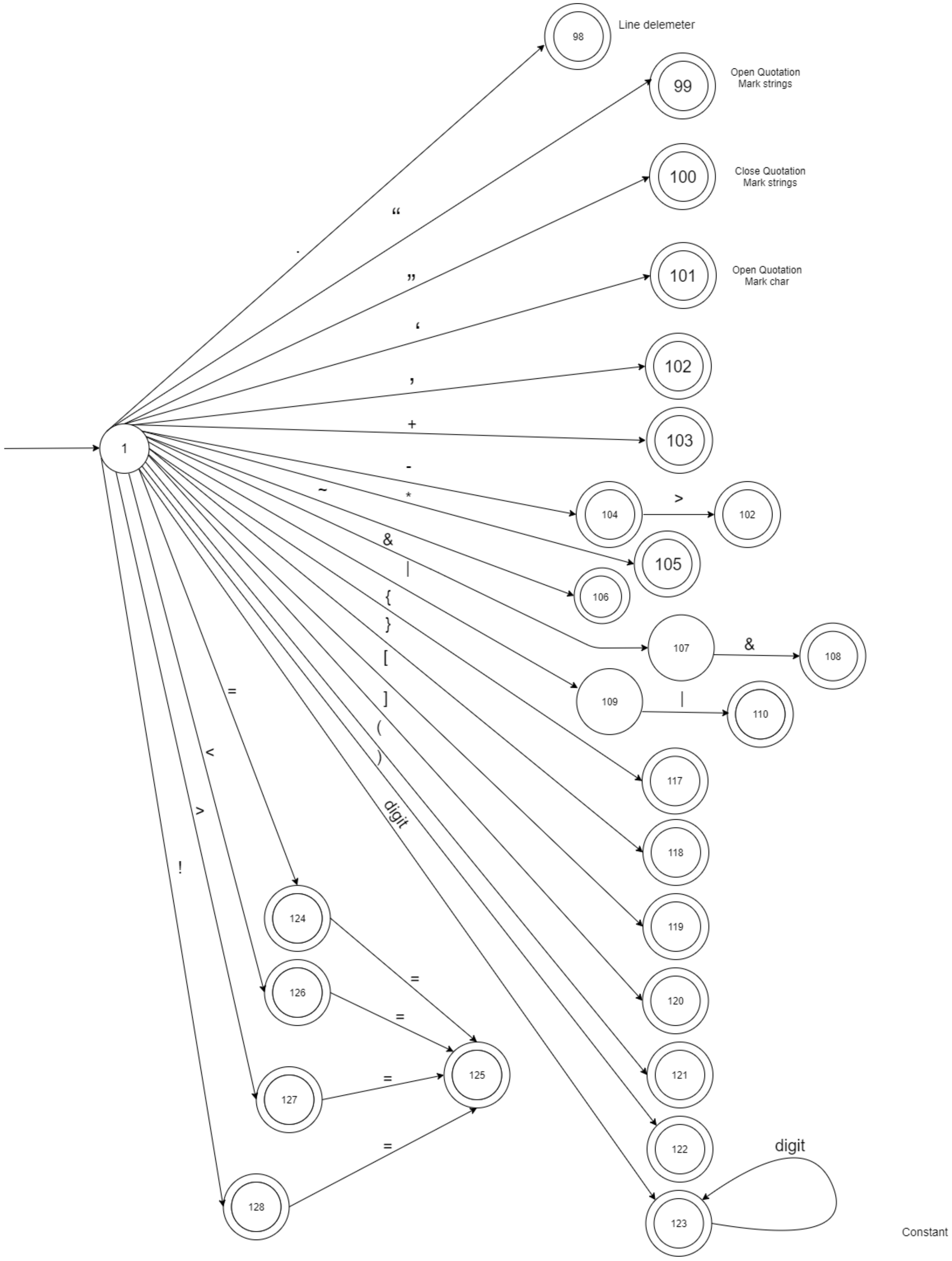


4- DFA for Comments:



5- DFA for all tokens:




98
Line delemeter
99
Open Quotation
Mark strings
100
Close Quotation
Mark strings
101
Open Quotation
Mark char
102
103
104
102
105
106
107
108
109
110
117
118
119
120
121
122
123
Constant
1
.
“
”
‘
,
+
-
~
*
>
&
|
{
}
[
]
(
)
&
|
digit
digit
=
<
>
!
124
126
127
128
125
=
=
=
=