

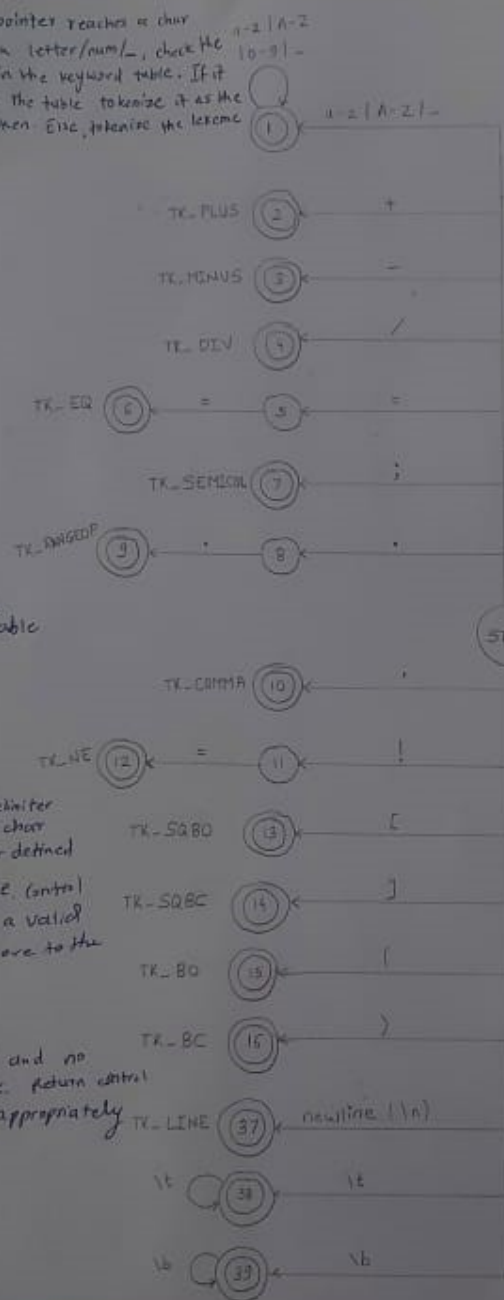
PATTERN	TOKEN
integer	INTEGER
real	REAL
boolean	BOOLEAN
of	OF
array	ARRAY
start	START
end	END
declare	DECLARE
module	MODULE
driver	DRIVER
program	PROGRAM
get-value	GET_VALUE
print	PRINT
use	USE
with	WITH
parameters	PARAMETERS
takes	TAKES
input	INPUT
returns	RETURNS
for	FOR
in	IN
switch	SWITCH
case	CASE
break	BREAK
default	DEFAULT
while	WHILE
AND	AND
OR	OR
true	TRUE
false	FALSE

If forward pointer reaches a char other than a letter/num/_, check the read lexeme in the keyword table. If it is present in the table, tokenize it as the respective token. Else, tokenize the lexeme as TK_ID.

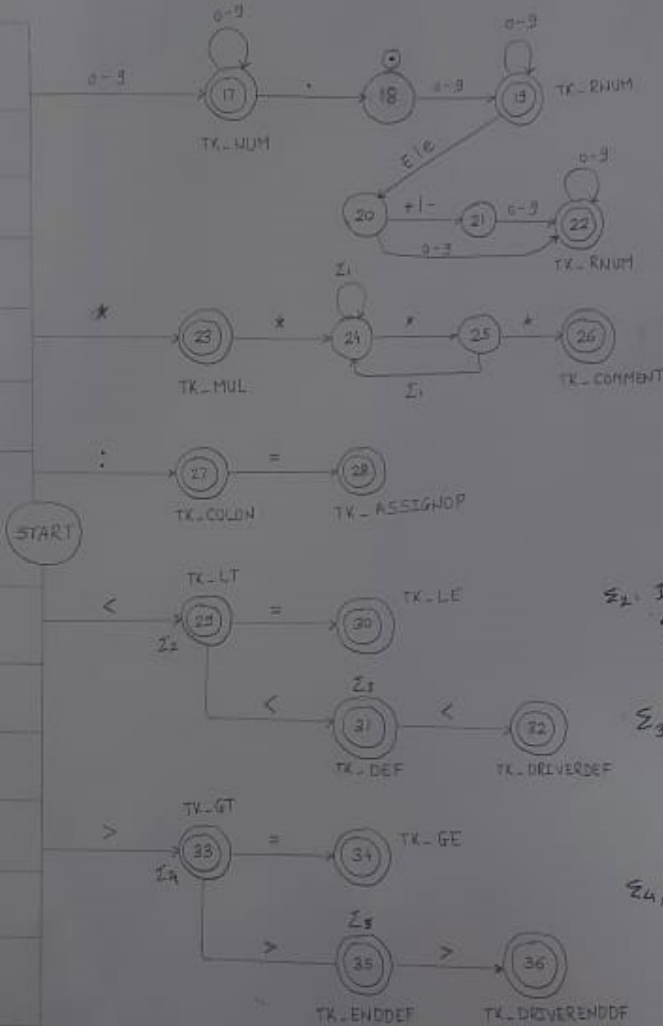
Included in Keyword table

①: If forward pointer reaches a delimiter (space, newline, tab, or any other char for which a transition is not defined from the final state) tokenize. Control reaches START. If there exists a valid transition from final state, move to the next state.

Trap state: A state which is non-final and no further transitions are possible. Return control to START and handle error appropriately.



② If the lookahead reads a ' ' retract to the previous state and tokenize as NUM. Start simulation from the start state.



CSF363 COMPILER DFA

GROUP:

- 1) Nachiket Kotalwar
2020A7PS0024P
- 2) Parth Patel
2020A7PS0026P
- 3) Parth Jyoti Purkayastha
2020A7PS0043P
- 4) Laksh Abhinav
2020A7PS0045P
- 5) Anish Kulkarni
2020A7PS0075P

Z₁ = Any char other than *

Z₂: If forward pointer encounters a '<' or '=' proceed to next state else tokenize as TK_LT

Z₃: If forward pointer encounters a '<' proceed to next state else tokenize as TK_DEF

Z₄, Z₅: Similar to Z₂ and Z₃ respectively for '>'.