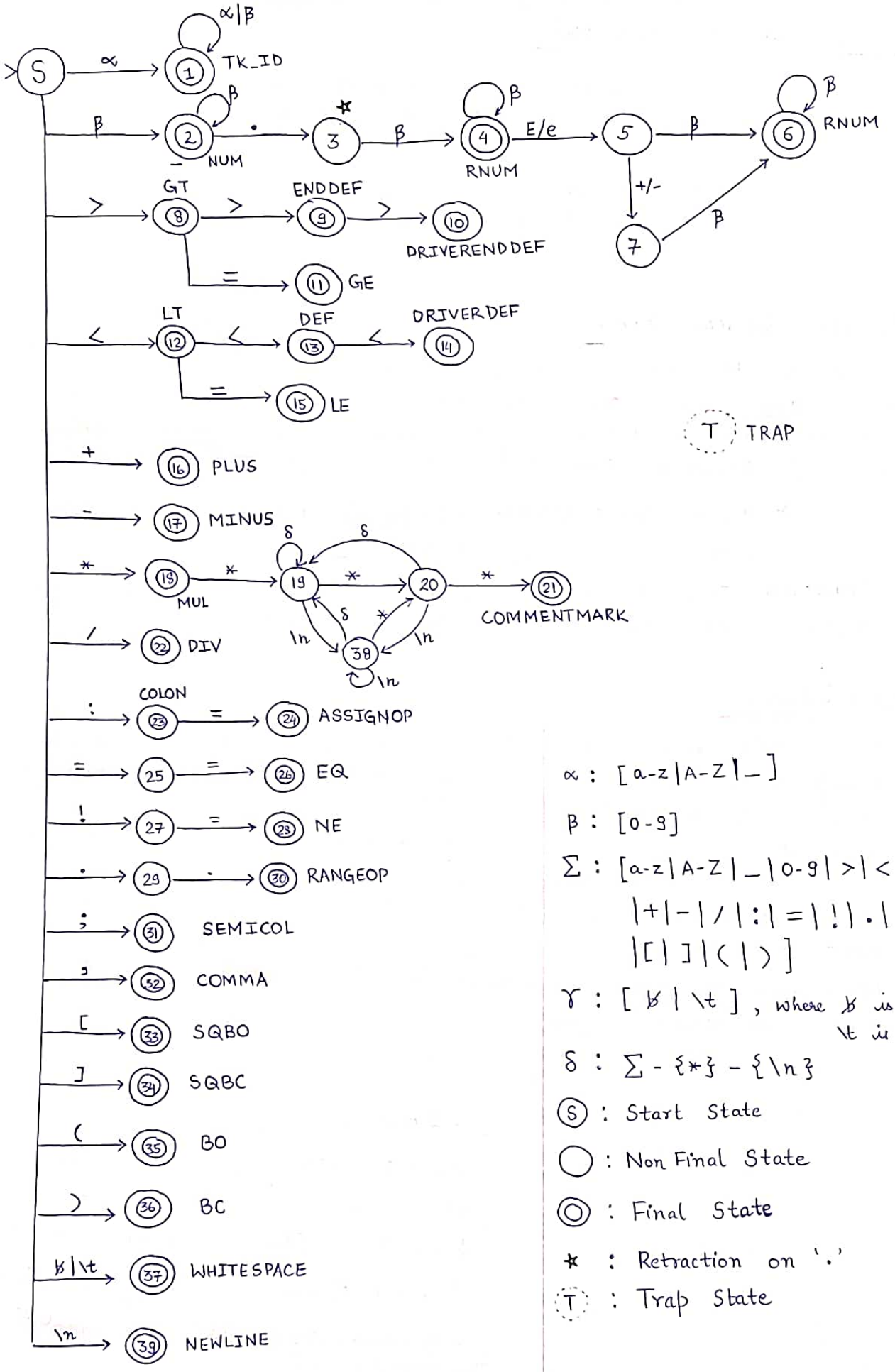

α : [a-z|A-Z|_]

β : [0-9]

Σ : [a-z|A-Z|_|0-9|>|<|ϐ|\t|\n |+|-|/|:|=|!|.|;|,|* |[|]|(|)]

γ : [ϐ | \t] , where ϐ is blank, \t is tab

δ : Σ - {*} - {\n}

(S) : Start State

◯ : Non Final State

◎ : Final State

⋆ : Retraction on '.'

(T) : Trap State

Passing Control to Start State

σ represents the characters on outgoing transistions from any particular State. Whenever any character other than that from σ is read from a final state we pass control to the start state i.e.

$\forall\ f \in F$, on transistion (f, x, g), $x \in \sigma$
Pass control to 'S', if input $\in \Sigma - \sigma$

Reporting Lexical Error

σ represents the characters on outgoing transistions from any particular State. Whenever any character other than that from σ is read from a non-final state, we pass control to Trap State (T) and report lexical Error. (Except for conditional trap - State 3)

$\forall\ q \notin F$, on transistion (q, x, r), $x \in \sigma$
Pass control to 'T', if input $\in \Sigma - \sigma$

CONDITIONAL TRAP : For State 3 if input is '.', then we donot move to trap. Instead, retract and pass control to start state.

Miscellaneous

COMMENTMARK, WHITESPACE & NEWLINE are ignored & not tokenised.
NEWLINE (State 39) increments the line Number & like any other final State, control is passed to Start State
When in State 19, & a newline character (\n) is encountered, move to State 38, where line no is incremented.
Whenever a lexeme is generated, it is first checked in the lookup table to search for reserved keywords.
AND, OR, true, false are inserted in the lookup table but will be treated as tokens (lexemes)

TEAM :

Akshat Gupta	2020A7PS0096P
Anish Ghule	2020A7PS0129P
P.V.S Tarak Shree Vallabha	2020A7PS1513P
Shadan Hussain	2020A7PS0134P
Shashank Agrawal	2020A7PS0073P