**Language Description for CMM Lexical Analyser:**

|  |  |  |
| --- | --- | --- |
| **Token** | **Transition Diagram** | **Regular Definition** |
| Variable Name |  | R -> letter (letter ll number)\* |
| Arithmetic Operator |  | R-> \* ll + ll / ll - |
| Relational Operator |  | R-> >A ll <A ll =A ll <>  A-> = ll NULL |
| Numeric Constant |  | R-> 0 ll 1 ll 2 ll 3 ll 4 ll … ll 9 |
| Brackets |  | R-> ( ll { ll [ ll ] ll } ll } |
| Comment |  | R-> #A  A->number ll letter ll symbol |
| Multi Line Comment |  | R->##A##  A-> letter ll digit |
| Comma |  | R->, |
| Semi Colon |  | R->; |
| Assignment Operator |  | R-> <- |
| Datatypes |  | R -> int ll char |
| Keywords |  | R->def ll if ll else ll while ll ret ll print ll read |

**Grammar:**

\* P -> P D | ^

\* D -> V | F

\* V -> T id ;

\* T -> int | char

\* F -> def T id ( OPL ) { SL }

\* OPL -> PL | ^

\* PL -> PAR , PL | PAR

\* PAR -> T id

\* SL -> S SL | ^

\* S -> A | SS | I | PR | RT

\* SS -> if ( E ) S | if ( E ) S else S

\* A -> id <- E ;

\* I -> while ( E ) S

\* PR -> print L; | print id;

\* RT-> ret L; | ret ID;

\* L-> 'digit' | 'character' | num

\* E->R E'

\* E' -> == R E' | <> R E' | ^

\* R -> N R'

\* R' -> < N R' | > N R' | <= N R' | >= N R' | ^

\* N -> G Q N'

\* N' -> + Q N' | - Q N' | ^

\* Q -> \* QT' | / Q T' | ^

\* G -> id | num | ( E )

**Grammar:**