# Main

S-S (Class) = { Class }

S-S (Class-Body) = { Line , Num , bigNum , ID , Define , } }

S-S (Main) = { Main }

S-S (Main-Foll) = { Class , $ }

# Declaration

S-S ( DataType-Identifier ) = { Line , Num , bigNum , ID }

S-S (Option) = { = , [ , ID , Define , Repeat ,RepeatWhile ,Either, Line , Num , bigNum , ID , Break ,} , Return , > , ; }

S-S (Dec-Assign) = { = , ID , Define , Repeat ,RepeatWhile ,Either, Line , Num , bigNum , ID , Break ,} , Return , > , ; }

S-S (Init) = { = , ; , , }

S-S (Init1) = { ID , String-Const , Integer-Const , double-Const , Char-Const }

S-S (List) = { ; , , }

# Array

S-S (Array) = { [ }

S-S (Size) = { ID , String-Const , Integer-Const, Double-Const, Char-Const , ] }

S-S (Detail) = { = , ; }

S-S (ID-Const-More) = { , , } }

# Repeat

S-St (Repeat) = { Repeat }

S-St (F1) = { ID, ++ , -- , ) }

S-St (F2) = { += , -= , /= , %= , \*= , ++ , -- }

S-St (Repeat-While) = { RepeatWhile }

S-St ( Either ) = { Either }

S-St ( O-Or ) = { Or , ID , Define , Repeat ,RepeatWhile ,Either, Line , Num , bigNum , ID , Break, Return , } , > }

# Shift

S-St (Shift) = { Shift }

S-St (Default) = { Default: , } }

S-St (Bound) = { bound , Default , } }

# Function

S-St ( Fn-Def ) = { Define }

S-St ( Fn-Def-Parameter ) = { Line , Num , bigNum , ) }

S-St (Fn-Def-Parameter1) = { , , ) }

S-St (Fn-Init) = { = , , , Line , Num , bigNum , ), Line , Num , bigNum, ID }

S-St ( Fn-Body ) = { ID ,}, Define , Repeat ,RepeatWhile ,Either, Line , Num , bigNum , ID , Break }

S-St ( Fn-Return ) = { Return , } }

S-St (Fn-Call) = { ID }

S-St (Fn-Call-Param) = { Line , Num , bigNum, ID , ) }

S-St (Fn-Call-Param1) = { , , ) }

S-St (Fn-Call-Param2) = { Line , Num , bigNum , ID }

# Expression

S-St (OE) = { ID , String-Const , Integer-Const, Double-Const, Char-Const }

S-St (F) = { ID , String-Const , Integer-Const, Double-Const, Char-Const }

S-St ( T’ ) = {\* , / , % , + , - , < , > , <= , >= , != , == , ID , String-Const , Integer-Const, Double-Const, Char-Const , &&,: , ) , ; , , , Line , Num , bigNum , ] , || }

S-St (T) = { ID , String-Const , Integer-Const, Double-Const, Char-Const }

S-St ( E’ ) = { + , - , < , > , <= , >= , != , == , ID , String-Const , Integer-Const, Double-Const, Char-Const , &&,: , ) , ; , , , Line , Num , bigNum , ] , || }

S-St (E) = { ID , String-Const , Integer-Const, Double-Const, Char-Const }

S-St (RE’) = { < , > , <= , >= , != , == , ID , String-Const , Integer-Const, Double-Const, Char-Const , &&,: , ) , ; , , , Line , Num , bigNum , ] , || }

S-St (RE) = { ID , String-Const , Integer-Const, Double-Const, Char-Const }

S-St (AE’) = { && , : , ) , ; , , , Line , Num , bigNum , ID , ] , || }

S-St (AE) = { ID , String-Const , Integer-Const, Double-Const, Char-Const }

S-St (OE’) = { || , : , ) , ; , , , Line , Num , bigNum , ID , ] }

# Other

S-St (S-St)= {Shift , ID , Define , Repeat ,RepeatWhile ,Either, Line , Num , bigNum , ID , Break }

S-St (M-St) = { ID , Define , Repeat ,RepeatWhile ,Either, Line , Num , bigNum , ID , Break , Return , } , > }

S-St (Const) = { String-Const , Integer-Const , double-Const , Char-Const }

S-St (DT) = { Line , Num , bigNum }

S-St ( Inc-Dec )= { ++ , -- }

S-St (Assg-Opper) = { += , -= , /= , %= , \*= }