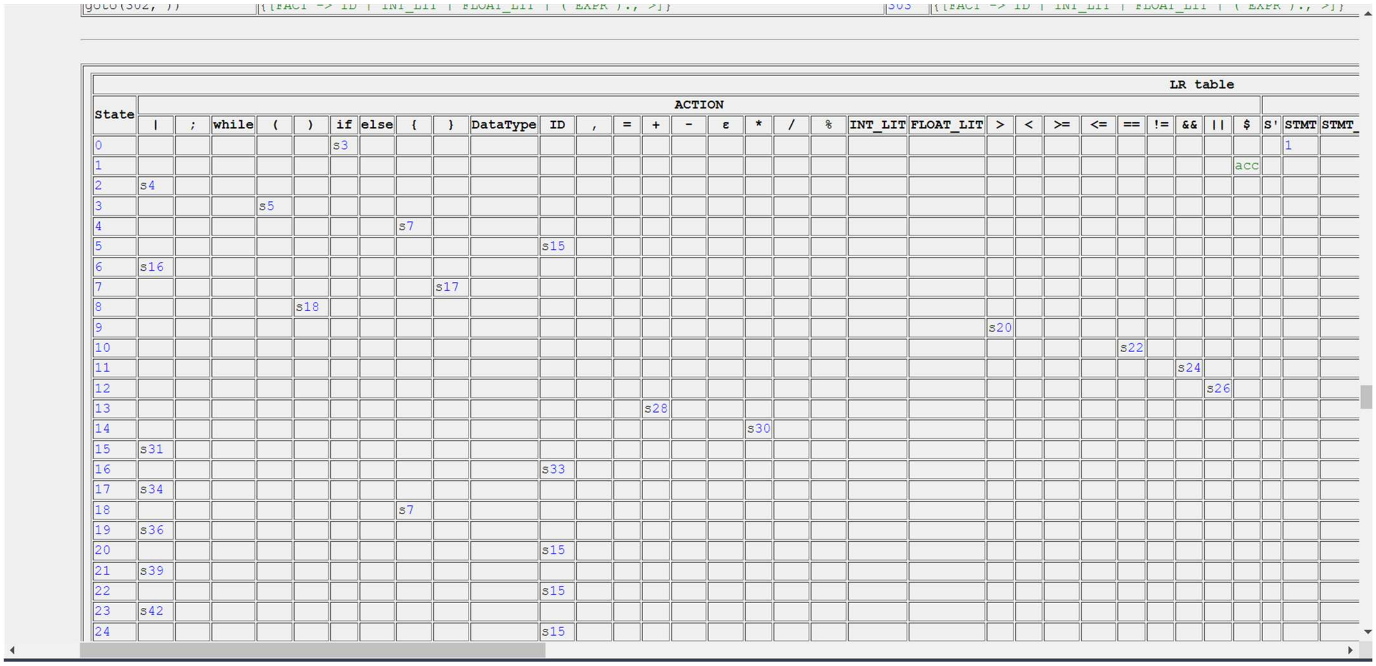
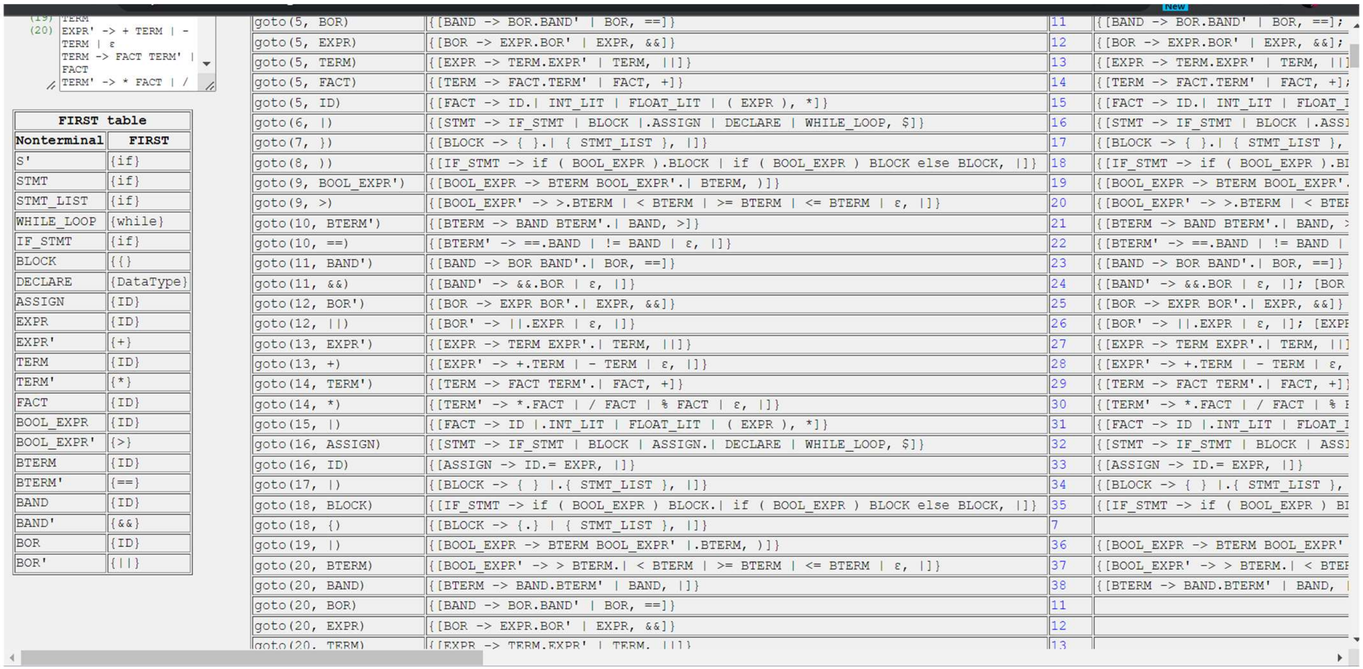


goto(302,))	{[FACT -> ID | INT_LIT | FLOAT_LIT | (EXPR)., >]}	303	{[FACT -> ID | INT_LIT | FLOAT_LIT | (EXPR)., >]}

LR table

State	ACTION																																
	|	;	while	(	)	if	else	{	}	DataType	ID	,	=	+	-	ε	*	/	%	INT_LIT	FLOAT_LIT	>	<	>=	<=	==	!=	&&	||	$	S'	STMT	STMT_
0						s3																										1	
1																														acc			
2	s4																																
3				s5																													
4								s7																									
5											s15																						
6	s16																																
7									s17																								
8					s18																												
9																						s20											
10																										s22							
11																												s24					
12																													s26				
13														s28																			
14																	s30																
15	s31																																
16											s33																						
17	s34																																
18								s7																									
19	s36																																
20											s15																						
21	s39																																
22											s15																						
23	s42																																
24											s15																						

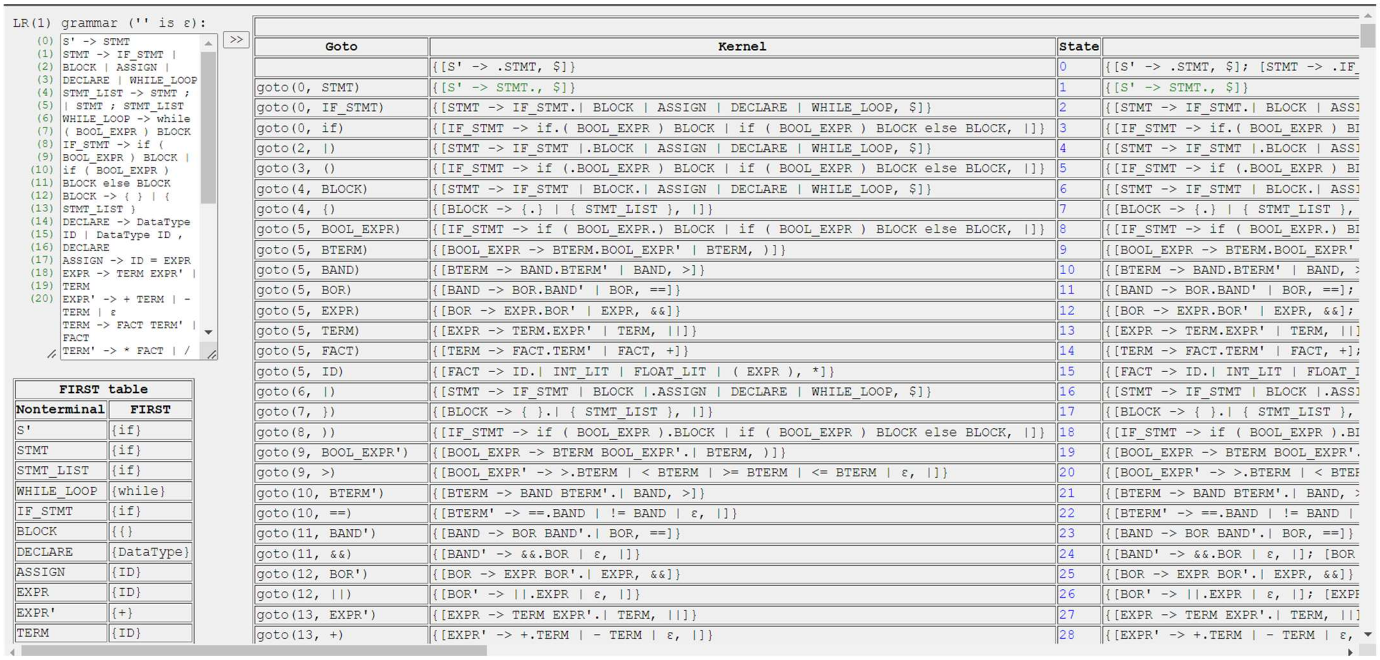


```
(19) TERM
(20) EXPR' -> + TERM | -
     TERM | ε
     TERM -> FACT TERM' |
     FACT
     TERM' -> * FACT | /
```

FIRST table

Nonterminal	FIRST
S'	{if}
STMT	{if}
STMT_LIST	{if}
WHILE_LOOP	{while}
IF_STMT	{if}
BLOCK	{{}
DECLARE	{DataType}
ASSIGN	{ID}
EXPR	{ID}
EXPR'	{+}
TERM	{ID}
TERM'	{*}
FACT	{ID}
BOOL_EXPR	{ID}
BOOL_EXPR'	{>}
BTERM	{ID}
BTERM'	{==}
BAND	{ID}
BAND'	{&&}
BOR	{ID}
BOR'	{||}

goto(5, BOR)	{[BAND -> BOR.BAND' | BOR, ==]}	11	{[BAND -> BOR.BAND' | BOR, ==];
goto(5, EXPR)	{[BOR -> EXPR.BOR' | EXPR, &&]}	12	{[BOR -> EXPR.BOR' | EXPR, &&];
goto(5, TERM)	{[EXPR -> TERM.EXPR' | TERM, ||]}	13	{[EXPR -> TERM.EXPR' | TERM, ||]
goto(5, FACT)	{[TERM -> FACT.TERM' | FACT, +]}	14	{[TERM -> FACT.TERM' | FACT, +];
goto(5, ID)	{[FACT -> ID.| INT_LIT | FLOAT_LIT | (EXPR), *]}	15	{[FACT -> ID.| INT_LIT | FLOAT_I
goto(6, |)	{[STMT -> IF_STMT | BLOCK |.ASSIGN | DECLARE | WHILE_LOOP, $]}	16	{[STMT -> IF_STMT | BLOCK |.ASSI
goto(7, })	{[BLOCK -> { }.| { STMT_LIST }, |]}	17	{[BLOCK -> { }.| { STMT_LIST },
goto(8,))	{[IF_STMT -> if (BOOL_EXPR).BLOCK | if (BOOL_EXPR) BLOCK else BLOCK, |]}	18	{[IF_STMT -> if (BOOL_EXPR).BI
goto(9, BOOL_EXPR')	{[BOOL_EXPR -> BTERM BOOL_EXPR'.| BTERM,)]}	19	{[BOOL_EXPR -> BTERM BOOL_EXPR'.
goto(9, >)	{[BOOL_EXPR' -> >.BTERM | < BTERM | >= BTERM | <= BTERM | ε, |]}	20	{[BOOL_EXPR' -> >.BTERM | < BTEI
goto(10, BTERM')	{[BTERM -> BAND BTERM'.| BAND, >]}	21	{[BTERM -> BAND BTERM'.| BAND, >
goto(10, ==)	{[BTERM' -> ==.BAND | != BAND | ε, |]}	22	{[BTERM' -> ==.BAND | != BAND |
goto(11, BAND')	{[BAND -> BOR BAND'.| BOR, ==]}	23	{[BAND -> BOR BAND'.| BOR, ==]}
goto(11, &&)	{[BAND' -> &&.BOR | ε, |]}	24	{[BAND' -> &&.BOR | ε, |]; [BOR
goto(12, BOR')	{[BOR -> EXPR BOR'.| EXPR, &&]}	25	{[BOR -> EXPR BOR'.| EXPR, &&]}
goto(12, ||)	{[BOR' -> ||.EXPR | ε, |]}	26	{[BOR' -> ||.EXPR | ε, |]; [EXPI
goto(13, EXPR')	{[EXPR -> TERM EXPR'.| TERM, ||]}	27	{[EXPR -> TERM EXPR'.| TERM, ||]
goto(13, +)	{[EXPR' -> +.TERM | - TERM | ε, |]}	28	{[EXPR' -> +.TERM | - TERM | ε,
goto(14, TERM')	{[TERM -> FACT TERM'.| FACT, +]}	29	{[TERM -> FACT TERM'.| FACT, +]}
goto(14, *)	{[TERM' -> *.FACT | / FACT | % FACT | ε, |]}	30	{[TERM' -> *.FACT | / FACT | % I
goto(15, |)	{[FACT -> ID |.INT_LIT | FLOAT_LIT | (EXPR), *]}	31	{[FACT -> ID |.INT_LIT | FLOAT_I
goto(16, ASSIGN)	{[STMT -> IF_STMT | BLOCK | ASSIGN.| DECLARE | WHILE_LOOP, $]}	32	{[STMT -> IF_STMT | BLOCK | ASSI
goto(16, ID)	{[ASSIGN -> ID.= EXPR, |]}	33	{[ASSIGN -> ID.= EXPR, |]}
goto(17, |)	{[BLOCK -> { } |.{ STMT_LIST }, |]}	34	{[BLOCK -> { } |.{ STMT_LIST },
goto(18, BLOCK)	{[IF_STMT -> if (BOOL_EXPR) BLOCK.| if (BOOL_EXPR) BLOCK else BLOCK, |]}	35	{[IF_STMT -> if (BOOL_EXPR) BI
goto(18, {)	{[BLOCK -> {.} | { STMT_LIST }, |]}	7	
goto(19, |)	{[BOOL_EXPR -> BTERM BOOL_EXPR' |.BTERM,)]}	36	{[BOOL_EXPR -> BTERM BOOL_EXPR'
goto(20, BTERM)	{[BOOL_EXPR' -> > BTERM.| < BTERM | >= BTERM | <= BTERM | ε, |]}	37	{[BOOL_EXPR' -> > BTERM.| < BTEI
goto(20, BAND)	{[BTERM -> BAND.BTERM' | BAND, |]}	38	{[BTERM -> BAND.BTERM' | BAND, |
goto(20, BOR)	{[BAND -> BOR.BAND' | BOR, ==]}	11	
goto(20, EXPR)	{[BOR -> EXPR.BOR' | EXPR, &&]}	12	
goto(20, TERM)	{[EXPR -> TERM.EXPR' | TERM, ||]}	13	



LR(1) grammar ('' is ε):

```
S' -> STMT
STMT -> IF_STMT |
BLOCK | ASSIGN |
DECLARE | WHILE_LOOP
STMT_LIST -> STMT ;
| STMT ; STMT_LIST
WHILE_LOOP -> while
( BOOL_EXPR ) BLOCK
IF_STMT -> if (
BOOL_EXPR ) BLOCK |
if ( BOOL_EXPR )
BLOCK else BLOCK
BLOCK -> { } | {
STMT_LIST }
DECLARE -> DataType
ID | DataType ID ,
DECLARE
ASSIGN -> ID = EXPR
EXPR -> TERM EXPR' |
TERM
EXPR' -> + TERM | -
TERM | ε
TERM -> FACT TERM' |
FACT
TERM' -> * FACT | /
```

FIRST table	
Nonterminal	**FIRST**
S'	{if}
STMT	{if}
STMT_LIST	{if}
WHILE_LOOP	{while}
IF_STMT	{if}
BLOCK	{{}
DECLARE	{DataType}
ASSIGN	{ID}
EXPR	{ID}
EXPR'	{+}
TERM	{ID}

Goto	Kernel	State	
	{[S' -> .STMT, $]}	0	{[S' -> .STMT, $]; [STMT -> .IF_
goto(0, STMT)	{[S' -> STMT., $]}	1	{[S' -> STMT., $]}
goto(0, IF_STMT)	{[STMT -> IF_STMT.| BLOCK | ASSIGN | DECLARE | WHILE_LOOP, $]}	2	{[STMT -> IF_STMT.| BLOCK | ASSI
goto(0, if)	{[IF_STMT -> if.(BOOL_EXPR) BLOCK | if (BOOL_EXPR) BLOCK else BLOCK, |]}	3	{[IF_STMT -> if.(BOOL_EXPR) BI
goto(2, |)	{[STMT -> IF_STMT |.BLOCK | ASSIGN | DECLARE | WHILE_LOOP, $]}	4	{[STMT -> IF_STMT |.BLOCK | ASSI
goto(3, ()	{[IF_STMT -> if (.BOOL_EXPR) BLOCK | if (BOOL_EXPR) BLOCK else BLOCK, |]}	5	{[IF_STMT -> if (.BOOL_EXPR) BI
goto(4, BLOCK)	{[STMT -> IF_STMT | BLOCK.| ASSIGN | DECLARE | WHILE_LOOP, $]}	6	{[STMT -> IF_STMT | BLOCK.| ASSI
goto(4, {)	{[BLOCK -> {.} | { STMT_LIST }, |]}	7	{[BLOCK -> {.} | { STMT_LIST },
goto(5, BOOL_EXPR)	{[IF_STMT -> if (BOOL_EXPR.) BLOCK | if (BOOL_EXPR) BLOCK else BLOCK, |]}	8	{[IF_STMT -> if (BOOL_EXPR.) BI
goto(5, BTERM)	{[BOOL_EXPR -> BTERM.BOOL_EXPR' | BTERM,)]}	9	{[BOOL_EXPR -> BTERM.BOOL_EXPR'
goto(5, BAND)	{[BTERM -> BAND.BTERM' | BAND, >]}	10	{[BTERM -> BAND.BTERM' | BAND, >
goto(5, BOR)	{[BAND -> BOR.BAND' | BOR, ==]}	11	{[BAND -> BOR.BAND' | BOR, ==];
goto(5, EXPR)	{[BOR -> EXPR.BOR' | EXPR, &&]}	12	{[BOR -> EXPR.BOR' | EXPR, &&];
goto(5, TERM)	{[EXPR -> TERM.EXPR' | TERM, ||]}	13	{[EXPR -> TERM.EXPR' | TERM, ||]
goto(5, FACT)	{[TERM -> FACT.TERM' | FACT, +]}	14	{[TERM -> FACT.TERM' | FACT, +];
goto(5, ID)	{[FACT -> ID.| INT_LIT | FLOAT_LIT | (EXPR), *]}	15	{[FACT -> ID.| INT_LIT | FLOAT_I
goto(6, |)	{[STMT -> IF_STMT | BLOCK |.ASSIGN | DECLARE | WHILE_LOOP, $]}	16	{[STMT -> IF_STMT | BLOCK |.ASSI
goto(7, })	{[BLOCK -> { }.| { STMT_LIST }, |]}	17	{[BLOCK -> { }.| { STMT_LIST },
goto(8,))	{[IF_STMT -> if (BOOL_EXPR).BLOCK | if (BOOL_EXPR) BLOCK else BLOCK, |]}	18	{[IF_STMT -> if (BOOL_EXPR).BI
goto(9, BOOL_EXPR')	{[BOOL_EXPR -> BTERM BOOL_EXPR'.| BTERM,)]}	19	{[BOOL_EXPR -> BTERM BOOL_EXPR'.
goto(9, >)	{[BOOL_EXPR' -> >.BTERM | < BTERM | >= BTERM | <= BTERM | ε, |]}	20	{[BOOL_EXPR' -> >.BTERM | < BTEF
goto(10, BTERM')	{[BTERM -> BAND BTERM'.| BAND, >]}	21	{[BTERM -> BAND BTERM'.| BAND, >
goto(10, ==)	{[BTERM' -> ==.BAND | != BAND | ε, |]}	22	{[BTERM' -> ==.BAND | != BAND |
goto(11, BAND')	{[BAND -> BOR BAND'.| BOR, ==]}	23	{[BAND -> BOR BAND'.| BOR, ==]}
goto(11, &&)	{[BAND' -> &&.BOR | ε, |]}	24	{[BAND' -> &&.BOR | ε, |]; [BOR
goto(12, BOR')	{[BOR -> EXPR BOR'.| EXPR, &&]}	25	{[BOR -> EXPR BOR'.| EXPR, &&]}
goto(12, ||)	{[BOR' -> ||.EXPR | ε, |]}	26	{[BOR' -> ||.EXPR | ε, |]; [EXPF
goto(13, EXPR')	{[EXPR -> TERM EXPR'.| TERM, ||]}	27	{[EXPR -> TERM EXPR'.| TERM, ||]
goto(13, +)	{[EXPR' -> +.TERM | - TERM | ε, |]}	28	{[EXPR' -> +.TERM | - TERM | ε,