SLR grammar ('' is ε):

```
(0) CODE -> DECLS
(1) DECLS -> DECL DECLS | epsilon
(2) DECL -> VDECL | FDECL | CDECL
(3) VDECL -> vtypeid semi | vtype ASSIGN semi
(4) ASSIGN -> id assign RHS
(5) RHS -> EXPR | literal | character | boolstr
(6) EXPR -> EXPR addsub TERM | TERM
(7) TERM -> TERM multdiv FACTOR | FACTOR
(8) FACTOR -> lparen EXPR rparen | id | num
(9) FDECL -> vtype id lparen ARGS rparen lbrace BLOCK RETURN rbrace
(10) ARGS -> vtype id MOREARGS | epsilon
(11) MOREARGS -> comma vtype id MOREARGS | epsilon
(12) BLOCK -> STMT BLOCK | epsilon
(13) STMT -> VDECL | ASSIGN semi | IFSTMT | WHILESTMT
(14) IFSTMT -> if lparen COND rparen lbrace BLOCK rbrace ELSE
(15) ELSE -> else lbrace BLOCK rbrace | epsilon
(16) WHILESTMT -> while lparen COND rparen lbrace BLOCK rbrace
(17) COND -> EXPR comp EXPR | boolstr
(18) RETURN -> return RHS semi
(19) CDECL -> class id lbrace ODECL rbrace
(20) ODECL -> VDECL ODECL | FDECL ODECL | epsilon
```

FIRST / FOLLOW table

Nonterminal	FIRST	FOLLOW
CODE	{vtypeid}	{$}
DECLS	{vtypeid}	{$,|}
DECL	{vtypeid}	{vtypeid}
VDECL	{vtypeid}	{|,vtypeid}
ASSIGN	{id}	{semi}
RHS	{}	{semi}
EXPR	{}	{|,addsub,rparen,comp}
TERM	{}	{|,addsub,multdiv,rparen,comp}
FACTOR	{lparen}	{|,addsub,multdiv,rparen,comp}
FDECL	{vtype}	{|,vtypeid}
ARGS	{vtype}	{rparen}
MOREARGS	{comma}	{|}
BLOCK	{vtypeid}	{return,|,rbrace}
STMT	{vtypeid}	{vtypeid}
IFSTMT	{if}	{|}
ELSE	{else}	{|}
WHILESTMT	{while}	{vtypeid}
COND	{}	{rparen}
RETURN	{return}	{rbrace}
CDECL	{class}	{vtypeid}
ODECL	{vtypeid}	{rbrace,|}

SLR closure table

Goto	Kernel	State	Closure
	{CODE -> .DECLS}	0	{CODE -> .DECLS; DECLS -> .DECL DECLS | epsilon; DECL -> .VDECL | FDECL | CDECL; VDECL -> .vtypeid semi | vtype ASSIGN semi}
goto(0, DECLS)	{CODE -> DECLS.}	1	{CODE -> DECLS.}
goto(0, DECL)	{DECLS -> DECL.DECLS | epsilon}	2	{DECLS -> DECL.DECLS | epsilon; DECLS -> .DECL DECLS | epsilon; DECL -> .VDECL | FDECL | CDECL; VDECL -> .vtypeid semi | vtype ASSIGN semi}
goto(0, VDECL)	{DECL -> VDECL.| FDECL | CDECL}	3	{DECL -> VDECL.| FDECL | CDECL}
goto(0, vtypeid)	{VDECL -> vtypeid.semi | vtype ASSIGN semi}	4	{VDECL -> vtypeid.semi | vtype ASSIGN semi}
goto(2, DECLS)	{DECLS -> DECL DECLS.| epsilon}	5	{DECLS -> DECL DECLS.| epsilon}
goto(2, DECL)	{DECLS -> DECL.DECLS | epsilon}	2	
goto(2, VDECL)	{DECL -> VDECL.| FDECL | CDECL}	3	
goto(2, vtypeid)	{VDECL -> vtypeid.semi | vtype ASSIGN semi}	4	
goto(3, |)	{DECL -> VDECL |.FDECL | CDECL}	6	{DECL -> VDECL |.FDECL | CDECL; FDECL -> .vtype id lparen ARGS rparen lbrace BLOCK RETURN rbrace}
goto(4, semi)	{VDECL -> vtypeid semi.| vtype ASSIGN semi}	7	{VDECL -> vtypeid semi.| vtype ASSIGN semi}
goto(5, |)	{DECLS -> DECL DECLS |.epsilon}	8	{DECLS -> DECL DECLS |.epsilon}
goto(6, FDECL)	{DECL -> VDECL | FDECL.| CDECL}	9	{DECL -> VDECL | FDECL.| CDECL}
goto(6, vtype)	{FDECL -> vtype.id lparen ARGS rparen lbrace BLOCK RETURN rbrace}	10	{FDECL -> vtype.id lparen ARGS rparen lbrace BLOCK RETURN rbrace}
goto(7, |)	{VDECL -> vtypeid semi |.vtype ASSIGN semi}	11	{VDECL -> vtypeid semi |.vtype ASSIGN semi}
goto(8, epsilon)	{DECLS -> DECL DECLS | epsilon.}	12	{DECLS -> DECL DECLS | epsilon.}
goto(9, |)	{DECL -> VDECL | FDECL |.CDECL}	13	{DECL -> VDECL | FDECL |.CDECL; CDECL -> .class id lbrace ODECL rbrace}
goto(10, id)	{FDECL -> vtype id.lparen ARGS rparen lbrace BLOCK RETURN rbrace}	14	{FDECL -> vtype id.lparen ARGS rparen lbrace BLOCK RETURN rbrace}
goto(11, vtype)	{VDECL -> vtypeid semi | vtype.ASSIGN semi}	15	{VDECL -> vtypeid semi | vtype.ASSIGN semi; ASSIGN -> .id assign RHS}
goto(13, CDECL)	{DECL -> VDECL | FDECL | CDECL.}	16	{DECL -> VDECL | FDECL | CDECL.}
goto(13, class)	{CDECL -> class.id lbrace ODECL rbrace}	17	{CDECL -> class.id lbrace ODECL rbrace}
goto(14, lparen)	{FDECL -> vtype id lparen.ARGS rparen lbrace BLOCK RETURN rbrace}	18	{FDECL -> vtype id lparen.ARGS rparen lbrace BLOCK RETURN rbrace; ARGS -> .vtype id MOREARGS | epsilon}
goto(15, ASSIGN)	{VDECL -> vtypeid semi | vtype ASSIGN.semi}	19	{VDECL -> vtypeid semi | vtype ASSIGN.semi}
goto(15, id)	{ASSIGN -> id.assign RHS}	20	{ASSIGN -> id.assign RHS}
goto(17, id)	{CDECL -> class id.lbrace ODECL rbrace}	21	{CDECL -> class id.lbrace ODECL rbrace}
goto(18, ARGS)	{FDECL -> vtype id lparen ARGS.rparen lbrace BLOCK RETURN rbrace}	22	{FDECL -> vtype id lparen ARGS.rparen lbrace BLOCK RETURN rbrace}
goto(18, vtype)	{ARGS -> vtype.id MOREARGS | epsilon}	23	{ARGS -> vtype.id MOREARGS | epsilon}
goto(19, semi)	{VDECL -> vtypeid semi | vtype ASSIGN semi.}	24	{VDECL -> vtypeid semi | vtype ASSIGN semi.}
goto(20, assign)	{ASSIGN -> id assign.RHS}	25	{ASSIGN -> id assign.RHS; RHS -> .EXPR | literal | character | boolstr; EXPR -> .EXPR addsub TERM | TERM}
goto(21, lbrace)	{CDECL -> class id lbrace.ODECL rbrace}	26	{CDECL -> class id lbrace.ODECL rbrace; ODECL -> .VDECL ODECL | FDECL ODECL | epsilon; VDECL -> .vtypeid semi | vtype ASSIGN semi}
goto(22, rparen)	{FDECL -> vtype id lparen ARGS rparen.lbrace BLOCK RETURN rbrace}	27	{FDECL -> vtype id lparen ARGS rparen.lbrace BLOCK RETURN rbrace}
goto(23, id)	{ARGS -> vtype id.MOREARGS | epsilon}	28	{ARGS -> vtype id.MOREARGS | epsilon; MOREARGS -> .comma vtype id MOREARGS | epsilon}
goto(25, RHS)	{ASSIGN -> id assign RHS.}	29	{ASSIGN -> id assign RHS.}
goto(25, EXPR)	{RHS -> EXPR.| literal | character | boolstr; EXPR -> EXPR.addsub TERM | TERM}	30	{RHS -> EXPR.| literal | character | boolstr; EXPR -> EXPR.addsub TERM | TERM}
goto(26, ODECL)	{CDECL -> class id lbrace ODECL.rbrace}	31	{CDECL -> class id lbrace ODECL.rbrace}
goto(26, VDECL)	{ODECL -> VDECL.ODECL | FDECL ODECL | epsilon}	32	{ODECL -> VDECL.ODECL | FDECL ODECL | epsilon; ODECL -> .VDECL ODECL | FDECL ODECL | epsilon; VDECL -> .vtypeid semi | vtype ASSIGN semi}
goto(26, vtypeid)	{VDECL -> vtypeid.semi | vtype ASSIGN semi}	4	
goto(27, lbrace)	{FDECL -> vtype id lparen ARGS rparen lbrace.BLOCK RETURN rbrace}	33	{FDECL -> vtype id lparen ARGS rparen lbrace.BLOCK RETURN rbrace; BLOCK -> .STMT BLOCK | epsilon; STMT -> .VDECL | ASSIGN semi | IFSTMT | WHILESTMT; VDECL -> .vtypeid semi | vtype ASSIGN semi}
goto(28, MOREARGS)	{ARGS -> vtype id MOREARGS.| epsilon}	34	{ARGS -> vtype id MOREARGS.| epsilon}
goto(28, comma)	{MOREARGS -> comma.vtype id MOREARGS | epsilon}	35	{MOREARGS -> comma.vtype id MOREARGS | epsilon}
goto(30, |)	{RHS -> EXPR |.literal | character | boolstr}	36	{RHS -> EXPR |.literal | character | boolstr}
goto(30, addsub)	{EXPR -> EXPR addsub.TERM | TERM}	37	{EXPR -> EXPR addsub.TERM | TERM; TERM -> .TERM multdiv FACTOR | FACTOR}
goto(31, rbrace)	{CDECL -> class id lbrace ODECL rbrace.}	38	{CDECL -> class id lbrace ODECL rbrace.}
goto(32, ODECL)	{ODECL -> VDECL ODECL.| FDECL ODECL | epsilon}	39	{ODECL -> VDECL ODECL.| FDECL ODECL | epsilon}
goto(32, VDECL)	{ODECL -> VDECL.ODECL | FDECL ODECL | epsilon}	32	
goto(32, vtypeid)	{VDECL -> vtypeid.semi | vtype ASSIGN semi}	4	
goto(33, BLOCK)	{FDECL -> vtype id lparen ARGS rparen lbrace BLOCK.RETURN rbrace}	40	{FDECL -> vtype id lparen ARGS rparen lbrace BLOCK.RETURN rbrace; RETURN -> .return RHS semi}
goto(33, STMT)	{BLOCK -> STMT.BLOCK | epsilon}	41	{BLOCK -> STMT.BLOCK | epsilon; BLOCK -> .STMT BLOCK | epsilon; STMT -> .VDECL | ASSIGN semi | IFSTMT | WHILESTMT; VDECL -> .vtypeid semi | vtype ASSIGN semi}
goto(33, VDECL)	{STMT -> VDECL.| ASSIGN semi | IFSTMT | WHILESTMT}	42	{STMT -> VDECL.| ASSIGN semi | IFSTMT | WHILESTMT}
goto(33, vtypeid)	{VDECL -> vtypeid.semi | vtype ASSIGN semi}	4	
goto(34, |)	{ARGS -> vtype id MOREARGS |.epsilon}	43	{ARGS -> vtype id MOREARGS |.epsilon}
goto(35, vtype)	{MOREARGS -> comma vtype.id MOREARGS | epsilon}	44	{MOREARGS -> comma vtype.id MOREARGS | epsilon}
goto(36, literal)	{RHS -> EXPR | literal.| character | boolstr}	45	{RHS -> EXPR | literal.| character | boolstr}
goto(37, TERM)	{EXPR -> EXPR addsub TERM.| TERM; TERM -> TERM.multdiv FACTOR | FACTOR}	46	{EXPR -> EXPR addsub TERM.| TERM; TERM -> TERM.multdiv FACTOR | FACTOR}
goto(39, |)	{ODECL -> VDECL ODECL |.FDECL ODECL | epsilon}	47	{ODECL -> VDECL ODECL |.FDECL ODECL | epsilon; FDECL -> .vtype id lparen ARGS rparen lbrace BLOCK RETURN rbrace}
goto(40, RETURN)	{FDECL -> vtype id lparen ARGS rparen lbrace BLOCK RETURN.rbrace}	48	{FDECL -> vtype id lparen ARGS rparen lbrace BLOCK RETURN.rbrace}
goto(40, return)	{RETURN -> return.RHS semi}	49	{RETURN -> return.RHS semi; RHS -> .EXPR | literal | character | boolstr; EXPR -> .EXPR addsub TERM | TERM}
goto(41, BLOCK)	{BLOCK -> STMT BLOCK.| epsilon}	50	{BLOCK -> STMT BLOCK.| epsilon}
goto(41, STMT)	{BLOCK -> STMT.BLOCK | epsilon}	41	
goto(41, VDECL)	{STMT -> VDECL.| ASSIGN semi | IFSTMT | WHILESTMT}	42	
goto(41, vtypeid)	{VDECL -> vtypeid.semi | vtype ASSIGN semi}	4	
goto(42, |)	{STMT -> VDECL |.ASSIGN semi | IFSTMT | WHILESTMT}	51	{STMT -> VDECL |.ASSIGN semi | IFSTMT | WHILESTMT; ASSIGN -> .id assign RHS}
goto(43, epsilon)	{ARGS -> vtype id MOREARGS | epsilon.}	52	{ARGS -> vtype id MOREARGS | epsilon.}
goto(44, id)	{MOREARGS -> comma vtype id.MOREARGS | epsilon}	53	{MOREARGS -> comma vtype id.MOREARGS | epsilon; MOREARGS -> .comma vtype id MOREARGS | epsilon}
goto(45, |)	{RHS -> EXPR | literal |.character | boolstr}	54	{RHS -> EXPR | literal |.character | boolstr}
goto(46, |)	{EXPR -> EXPR addsub TERM |.TERM}	55	{EXPR -> EXPR addsub TERM |.TERM; TERM -> .TERM multdiv FACTOR | FACTOR}
goto(46, multdiv)	{TERM -> TERM multdiv.FACTOR | FACTOR}	56	{TERM -> TERM multdiv.FACTOR | FACTOR; FACTOR -> .lparen EXPR rparen | id | num}
goto(47, FDECL)	{ODECL -> VDECL ODECL | FDECL.ODECL | epsilon}	57	{ODECL -> VDECL ODECL | FDECL.ODECL | epsilon; ODECL -> .VDECL ODECL | FDECL ODECL | epsilon; VDECL -> .vtypeid semi | vtype ASSIGN semi}
goto(47, vtype)	{FDECL -> vtype.id lparen ARGS rparen lbrace BLOCK RETURN rbrace}	10	
goto(48, rbrace)	{FDECL -> vtype id lparen ARGS rparen lbrace BLOCK RETURN rbrace.}	58	{FDECL -> vtype id lparen ARGS rparen lbrace BLOCK RETURN rbrace.}
goto(49, RHS)	{RETURN -> return RHS.semi}	59	{RETURN -> return RHS.semi}
goto(49, EXPR)	{RHS -> EXPR.| literal | character | boolstr; EXPR -> EXPR.addsub TERM | TERM}	30	
goto(50, |)	{BLOCK -> STMT BLOCK |.epsilon}	60	{BLOCK -> STMT BLOCK |.epsilon}
goto(51, ASSIGN)	{STMT -> VDECL | ASSIGN.semi | IFSTMT | WHILESTMT}	61	{STMT -> VDECL | ASSIGN.semi | IFSTMT | WHILESTMT}
goto(51, id)	{ASSIGN -> id.assign RHS}	20	
goto(53, MOREARGS)	{MOREARGS -> comma vtype id MOREARGS.| epsilon}	62	{MOREARGS -> comma vtype id MOREARGS.| epsilon}
goto(53, comma)	{MOREARGS -> comma.vtype id MOREARGS | epsilon}	35	
goto(54, character)	{RHS -> EXPR | literal | character.| boolstr}	63	{RHS -> EXPR | literal | character.| boolstr}
goto(55, TERM)	{EXPR -> EXPR addsub TERM | TERM.; TERM -> TERM.multdiv FACTOR | FACTOR}	64	{EXPR -> EXPR addsub TERM | TERM.; TERM -> TERM.multdiv FACTOR | FACTOR}
goto(56, FACTOR)	{TERM -> TERM multdiv FACTOR.| FACTOR}	65	{TERM -> TERM multdiv FACTOR.| FACTOR}
goto(56, lparen)	{FACTOR -> lparen.EXPR rparen | id | num}	66	{FACTOR -> lparen.EXPR rparen | id | num; EXPR -> .EXPR addsub TERM | TERM}
goto(57, ODECL)	{ODECL -> VDECL ODECL | FDECL ODECL.| epsilon}	67	{ODECL -> VDECL ODECL | FDECL ODECL.| epsilon}
goto(57, VDECL)	{ODECL -> VDECL.ODECL | FDECL ODECL | epsilon}	32	
goto(57, vtypeid)	{VDECL -> vtypeid.semi | vtype ASSIGN semi}	4	
goto(59, semi)	{RETURN -> return RHS semi.}	68	{RETURN -> return RHS semi.}
goto(60, epsilon)	{BLOCK -> STMT BLOCK | epsilon.}	69	{BLOCK -> STMT BLOCK | epsilon.}
goto(61, semi)	{STMT -> VDECL | ASSIGN semi.| IFSTMT | WHILESTMT}	70	{STMT -> VDECL | ASSIGN semi.| IFSTMT | WHILESTMT}
goto(62, |)	{MOREARGS -> comma vtype id MOREARGS |.epsilon}	71	{MOREARGS -> comma vtype id MOREARGS |.epsilon}
goto(63, |)	{RHS -> EXPR | literal | character |.boolstr}	72	{RHS -> EXPR | literal | character |.boolstr}
goto(64, multdiv)	{TERM -> TERM multdiv.FACTOR | FACTOR}	56	
goto(65, |)	{TERM -> TERM multdiv FACTOR |.FACTOR}	73	{TERM -> TERM multdiv FACTOR |.FACTOR; FACTOR -> .lparen EXPR rparen | id | num}
goto(66, EXPR)	{FACTOR -> lparen EXPR.rparen | id | num; EXPR -> EXPR.addsub TERM | TERM}	74	{FACTOR -> lparen EXPR.rparen | id | num; EXPR -> EXPR.addsub TERM | TERM}
goto(67, |)	{ODECL -> VDECL ODECL | FDECL ODECL |.epsilon}	75	{ODECL -> VDECL ODECL | FDECL ODECL |.epsilon}
goto(70, |)	{STMT -> VDECL | ASSIGN semi |.IFSTMT | WHILESTMT}	76	{STMT -> VDECL | ASSIGN semi |.IFSTMT | WHILESTMT; IFSTMT -> .if lparen COND rparen lbrace BLOCK rbrace ELSE}
goto(71, epsilon)	{MOREARGS -> comma vtype id MOREARGS | epsilon.}	77	{MOREARGS -> comma vtype id MOREARGS | epsilon.}
goto(72, boolstr)	{RHS -> EXPR | literal | character | boolstr.}	78	{RHS -> EXPR | literal | character | boolstr.}
goto(73, FACTOR)	{TERM -> TERM multdiv FACTOR | FACTOR.}	79	{TERM -> TERM multdiv FACTOR | FACTOR.}
goto(73, lparen)	{FACTOR -> lparen.EXPR rparen | id | num}	66	
goto(74, rparen)	{FACTOR -> lparen EXPR rparen.| id | num}	80	{FACTOR -> lparen EXPR rparen.| id | num}
goto(74, addsub)	{EXPR -> EXPR addsub.TERM | TERM}	37	
goto(75, epsilon)	{ODECL -> VDECL ODECL | FDECL ODECL | epsilon.}	81	{ODECL -> VDECL ODECL | FDECL ODECL | epsilon.}
goto(76, IFSTMT)	{STMT -> VDECL | ASSIGN semi | IFSTMT.| WHILESTMT}	82	{STMT -> VDECL | ASSIGN semi | IFSTMT.| WHILESTMT}
goto(76, if)	{IFSTMT -> if.lparen COND rparen lbrace BLOCK rbrace ELSE}	83	{IFSTMT -> if.lparen COND rparen lbrace BLOCK rbrace ELSE}
goto(80, |)	{FACTOR -> lparen EXPR rparen |.id | num}	84	{FACTOR -> lparen EXPR rparen |.id | num}
goto(82, |)	{STMT -> VDECL | ASSIGN semi | IFSTMT |.WHILESTMT}	85	{STMT -> VDECL | ASSIGN semi | IFSTMT |.WHILESTMT; WHILESTMT -> .while lparen COND rparen lbrace BLOCK rbrace}

SLR closure table

Goto	Kernel	State	Closure
goto(83, lparen)	{IFSTMT -> if lparen.COND rparen lbrace BLOCK rbrace ELSE}	86	{IFSTMT -> if lparen.COND rparen lbrace BLOCK rbrace ELSE; COND -> .EXPR comp EXPR | boolstr; EXPR -> .EXPR addsub TERM | TERM}
goto(84, id)	{FACTOR -> lparen EXPR rparen | id.| num}	87	{FACTOR -> lparen EXPR rparen | id.| num}
goto(85, WHILESTMT)	{STMT -> VDECL | ASSIGN semi | IFSTMT | WHILESTMT.}	88	{STMT -> VDECL | ASSIGN semi | IFSTMT | WHILESTMT.}
goto(85, while)	{WHILESTMT -> while.lparen COND rparen lbrace BLOCK rbrace}	89	{WHILESTMT -> while.lparen COND rparen lbrace BLOCK rbrace}
goto(86, COND)	{IFSTMT -> if lparen COND.rparen lbrace BLOCK rbrace ELSE}	90	{IFSTMT -> if lparen COND.rparen lbrace BLOCK rbrace ELSE}
goto(86, EXPR)	{COND -> EXPR.comp EXPR | boolstr; EXPR -> EXPR.addsub TERM | TERM}	91	{COND -> EXPR.comp EXPR | boolstr; EXPR -> EXPR.addsub TERM | TERM}
goto(87, |)	{FACTOR -> lparen EXPR rparen | id |.num}	92	{FACTOR -> lparen EXPR rparen | id |.num}
goto(89, lparen)	{WHILESTMT -> while lparen.COND rparen lbrace BLOCK rbrace}	93	{WHILESTMT -> while lparen.COND rparen lbrace BLOCK rbrace; COND -> .EXPR comp EXPR | boolstr; EXPR -> .EXPR addsub TERM | TERM}
goto(90, rparen)	{IFSTMT -> if lparen COND rparen.lbrace BLOCK rbrace ELSE}	94	{IFSTMT -> if lparen COND rparen.lbrace BLOCK rbrace ELSE}
goto(91, comp)	{COND -> EXPR comp.EXPR | boolstr}	95	{COND -> EXPR comp.EXPR | boolstr; EXPR -> .EXPR addsub TERM | TERM}
goto(91, addsub)	{EXPR -> EXPR addsub.TERM | TERM}	37	
goto(92, num)	{FACTOR -> lparen EXPR rparen | id | num.}	96	{FACTOR -> lparen EXPR rparen | id | num.}
goto(93, COND)	{WHILESTMT -> while lparen COND.rparen lbrace BLOCK rbrace}	97	{WHILESTMT -> while lparen COND.rparen lbrace BLOCK rbrace}
goto(93, EXPR)	{COND -> EXPR.comp EXPR | boolstr; EXPR -> EXPR.addsub TERM | TERM}	91	
goto(94, lbrace)	{IFSTMT -> if lparen COND rparen lbrace.BLOCK rbrace ELSE}	98	{IFSTMT -> if lparen COND rparen lbrace.BLOCK rbrace ELSE; BLOCK -> .STMT BLOCK | epsilon; STMT -> .VDECL | ASSIGN semi | IFSTMT | WHILESTMT; VDECL -> .vtypeid semi | vtype ASSIGN semi}
goto(95, EXPR)	{COND -> EXPR comp EXPR.| boolstr; EXPR -> EXPR.addsub TERM | TERM}	99	{COND -> EXPR comp EXPR.| boolstr; EXPR -> EXPR.addsub TERM | TERM}
goto(97, rparen)	{WHILESTMT -> while lparen COND rparen.lbrace BLOCK rbrace}	100	{WHILESTMT -> while lparen COND rparen.lbrace BLOCK rbrace}
goto(98, BLOCK)	{IFSTMT -> if lparen COND rparen lbrace BLOCK.rbrace ELSE}	101	{IFSTMT -> if lparen COND rparen lbrace BLOCK.rbrace ELSE}
goto(98, STMT)	{BLOCK -> STMT.BLOCK | epsilon}	41	
goto(98, VDECL)	{STMT -> VDECL.| ASSIGN semi | IFSTMT | WHILESTMT}	42	
goto(98, vtypeid)	{VDECL -> vtypeid.semi | vtype ASSIGN semi}	4	
goto(99, |)	{COND -> EXPR comp EXPR |.boolstr}	102	{COND -> EXPR comp EXPR |.boolstr}
goto(99, addsub)	{EXPR -> EXPR addsub.TERM | TERM}	37	
goto(100, lbrace)	{WHILESTMT -> while lparen COND rparen lbrace.BLOCK rbrace}	103	{WHILESTMT -> while lparen COND rparen lbrace.BLOCK rbrace; BLOCK -> .STMT BLOCK | epsilon; STMT -> .VDECL | ASSIGN semi | IFSTMT | WHILESTMT; VDECL -> .vtypeid semi | vtype ASSIGN semi}
goto(101, rbrace)	{IFSTMT -> if lparen COND rparen lbrace BLOCK rbrace.ELSE}	104	{IFSTMT -> if lparen COND rparen lbrace BLOCK rbrace.ELSE; ELSE -> .else lbrace BLOCK rbrace | epsilon}
goto(102, boolstr)	{COND -> EXPR comp EXPR | boolstr.}	105	{COND -> EXPR comp EXPR | boolstr.}
goto(103, BLOCK)	{WHILESTMT -> while lparen COND rparen lbrace BLOCK.rbrace}	106	{WHILESTMT -> while lparen COND rparen lbrace BLOCK.rbrace}
goto(103, STMT)	{BLOCK -> STMT.BLOCK | epsilon}	41	
goto(103, VDECL)	{STMT -> VDECL.| ASSIGN semi | IFSTMT | WHILESTMT}	42	
goto(103, vtypeid)	{VDECL -> vtypeid.semi | vtype ASSIGN semi}	4	
goto(104, ELSE)	{IFSTMT -> if lparen COND rparen lbrace BLOCK rbrace ELSE.}	107	{IFSTMT -> if lparen COND rparen lbrace BLOCK rbrace ELSE.}
goto(104, else)	{ELSE -> else.lbrace BLOCK rbrace | epsilon}	108	{ELSE -> else.lbrace BLOCK rbrace | epsilon}
goto(106, rbrace)	{WHILESTMT -> while lparen COND rparen lbrace BLOCK rbrace.}	109	{WHILESTMT -> while lparen COND rparen lbrace BLOCK rbrace.}
goto(108, lbrace)	{ELSE -> else lbrace.BLOCK rbrace | epsilon}	110	{ELSE -> else lbrace.BLOCK rbrace | epsilon; BLOCK -> .STMT BLOCK | epsilon; STMT -> .VDECL | ASSIGN semi | IFSTMT | WHILESTMT; VDECL -> .vtypeid semi | vtype ASSIGN semi}
goto(110, BLOCK)	{ELSE -> else lbrace BLOCK.rbrace | epsilon}	111	{ELSE -> else lbrace BLOCK.rbrace | epsilon}
goto(110, STMT)	{BLOCK -> STMT.BLOCK | epsilon}	41	
goto(110, VDECL)	{STMT -> VDECL.| ASSIGN semi | IFSTMT | WHILESTMT}	42	
goto(110, vtypeid)	{VDECL -> vtypeid.semi | vtype ASSIGN semi}	4	
goto(111, rbrace)	{ELSE -> else lbrace BLOCK rbrace.| epsilon}	112	{ELSE -> else lbrace BLOCK rbrace.| epsilon}
goto(112, |)	{ELSE -> else lbrace BLOCK rbrace |.epsilon}	113	{ELSE -> else lbrace BLOCK rbrace |.epsilon}
goto(113, epsilon)	{ELSE -> else lbrace BLOCK rbrace | epsilon.}	114	{ELSE -> else lbrace BLOCK rbrace | epsilon.}

LR table

ACTION

State	|	epsilon	vtypeid	semi	vtype	id	assign	literal	character	boolstr	addsub	multdiv	lparen	rparen	num	lbrace	rbrace	comma	if	else	while	comp	return	class	$
0			s4																						
1																									acc
2			s4																						
3	s6																								
4				s7																					
5	s8																								
6					s10																				
7	s11																								
8		s12																							
9	s13																								
10						s14																			
11					s15																				
12	r_1																								r_1
13																								s17	
14													s18												
15						s20																			
16			r_2																						
17						s21																			
18					s23																				
19				s24																					
20							s25																		
21																s26									
22														s27											
23						s28																			
24	r_3		r_3																						
25																									
26			s4																						
27																s33									
28																		s35							
29				r_4																					
30	s36										s37														
31																	s38								
32			s4																						
33			s4																						
34	s43																								
35					s44																				
36								s45																	
37																									
38			r_{19}																						
39	s47																								
40																							s49		
41			s4																						
42	s51																								
43		s52																							
44						s53																			
45	s54																								
46	s55											s56													
47					s10																				
48																	s58								
49																									
50	s60																								
51						s20																			
52														r_{10}											
53																		s35							
54									s63																
55																									
56													s66												
57			s4																						

GOTO

State	CODE	DECLS	DECL	VDECL	ASSIGN	RHS	EXPR	TERM	FACTOR	FDECL	ARGS	MOREARGS	BLOCK	STMT	IFSTMT	ELSE	WHILESTMT	COND	RETURN	CDECL	ODECL
0		1	2	3																	
1																					
2		5	2	3																	
3																					
4																					
5																					
6										9											
7																					
8																					
9																					
10																					
11																					
12																					
13																				16	
14																					
15					19																
16																					
17																					
18											22										
19																					
20																					
21																					
22																					
23																					
24																					
25						29	30														
26				32																	31
27																					
28												34									
29																					
30																					
31																					
32				32																	39
33				42									40	41							
34																					
35																					
36																					
37								46													
38																					
39																					
40																			48		
41				42									50	41							
42																					
43																					
44																					
45																					
46																					
47										57											
48																					
49						59	30														
50																					
51					61																
52																					
53												62									
54																					
55								64													
56									65												
57				32																	67

Input (tokens): id + id * id

Maximum number of steps: 100

PARSE

Step	Stack	Input	Action	Tree
1	0	id + id * id $		

(Trace)

LR table

ACTION

State	|	epsilon	vtypeid	semi	vtype	id	assign	literal	character	boolstr	addsub	multdiv	lparen	rparen	num	lbrace	rbrace	comma	if	else	while	comp	return	class	$
58	r_9		r_9																						
59				s68																					
60		s69																							
61				s70																					
62	s71																								
63	s72																								
64	r_6										r_6	s56		r_6								r_6			
65	s73																								
66																									
67	s75																								
68																	r_{18}								
69	r_{12}																r_{12}						r_{12}		
70	s76																								
71		s77																							
72										s78															
73													s66												
74											s37			s80											
75		s81																							
76																			s83						
77	r_{11}																								
78				r_5																					
79	r_7										r_7	r_7		r_7								r_7			
80	s84																								
81	r_{20}																r_{20}								
82	s85																								
83													s86												
84						s87																			
85																					s89				
86																									
87	s92																								
88			r_{13}																						
89													s93												
90														s94											
91											s37											s95			
92															s96										
93																									
94																s98									
95																									
96	r_8										r_8	r_8		r_8								r_8			
97														s100											
98			s4																						
99	s102										s37														
100																s103									
101																	s104								
102										s105															
103			s4																						
104																				s108					
105														r_{17}											
106																	s109								
107	r_{14}																								
108																s110									
109			r_{16}																						
110			s4																						
111																	s112								
112	s113																								
113		s114																							
114	r_{15}																								

GOTO

State	CODE	DECLS	DECL	VDECL	ASSIGN	RHS	EXPR	TERM	FACTOR	FDECL	ARGS	MOREARGS	BLOCK	STMT	IFSTMT	ELSE	WHILESTMT	COND	RETURN	CDECL	ODECL
58																					
59																					
60																					
61																					
62																					
63																					
64																					
65																					
66							74														
67																					
68																					
69																					
70																					
71																					
72																					
73									79												
74																					
75																					
76															82						
77																					
78																					
79																					
80																					
81																					
82																					
83																					
84																					
85																	88				
86							91											90			
87																					
88																					
89																					
90																					
91																					
92																					
93							91											97			
94																					
95							99														
96																					
97																					
98				42									101	41							
99																					
100																					
101																					
102																					
103				42									106	41							
104																107					
105																					
106																					
107																					
108																					
109																					
110				42									111	41							
111																					
112																					
113																					
114																					