SLR(1)-автомат

№ сост	Конфигурация	Символ перехода (свертки)	Состояние-преемник	Свертка {действие}
1	S = • "prog" "id" A ";" Block "eof"	prog	2	
2	S = "prog" • "id" A ";" Block "eof"	id	3	
3	S = "prog" "id" • A ";" Block "eof"	A	4	
	A = {A1} •	;		R45 {A1}
4	S = "prog" "id" A • ";" Block "eof"	;	5	
5	S = "prog" "id" A ";" • Block "eof"	Block	6	
	Block = • "var" LstVar "start" LstStmt "stop"	var	7	
6	S = "prog" "id" A ";" Block • "eof"	eof	stop	
7	Block = "var" • LstVar "start" LstStmt "stop"	LstVar	8	
	LstVar = • DeclVar ";"	DeclVar	9	
	LstVar = • LstVar DeclVar ";"	LstVar	8	
	DeclVar = • "id" B LstId {A3}	id	10	
8	Block = "var" LstVar • "start" LstStmt "stop"	start	11	
	LstVar = LstVar • DeclVar ";"	DeclVar	12	
	DeclVar = • "id" B LstId {A3}	id	10	
9	LstVar = DeclVar • ";"	;	13	
10	DeclVar = "id" • B LstId {A3}	B	14	
	B = {A2} •	,, :		R46 {A2}
11	Block = "var" LstVar "start" • LstStmt "stop"	LstStmt	15	
	LstStmt = • Stmt ";"	Stmt	16	
	LstStmt = • LstStmt Stmt ";"	LstStmt	15	
	Stmt = • "id" C AssignmentTarget "ass" Expr {A7}	id	17	
	Stmt = • "repeat" LstStmt "until" LogicalExpr {A26}	repeat	18	
12	LstVar = LstVar DeclVar • ";"	;	19	
13	LstVar = DeclVar ";" •	id, start		R3
14	DeclVar = "id" B • LstId {A3}	LstId	20	
	LstId = • "," "id" B LstId {A4}	,	21	
	LstId = • ":" TypeSpec {A5}	:	22	
15	Block = "var" LstVar "start" LstStmt • "stop"	stop	23	
	LstStmt = LstStmt • Stmt ";"	Stmt	24	
	Stmt = • "id" C AssignmentTarget "ass" Expr {A7}	id	17	
	Stmt = • "repeat" LstStmt "until" LogicalExpr {A26}	repeat	18	
16	LstStmt = Stmt • ";"	;	25	
17	Stmt = "id" • C AssignmentTarget "ass" Expr {A7}	C	26	
	C = {A6} •	[, ass		R47 {A6}
18	Stmt = "repeat" • LstStmt "until" LogicalExpr {A26}	LstStmt	27	
	LstStmt = • Stmt ";"	Stmt	16	
	LstStmt = • LstStmt Stmt ";"	LstStmt	27	
	Stmt = • "id" C AssignmentTarget "ass" Expr {A7}	id	17	
	Stmt = • "repeat" LstStmt "until" LogicalExpr {A26}	repeat	18	
19	LstVar = LstVar DeclVar ";" •	id, start		R4
20	DeclVar = "id" B LstId {A3} •	;		R5 {A3}
21	LstId = "," • "id" B LstId {A4}	id	28	
22	LstId = ":" • TypeSpec {A5}	TypeSpec	29	
	TypeSpec = • "integer" {A21}	integer	30	
	TypeSpec = • "real" {A22}	real	31	
	TypeSpec = • "string" {A23}	string	32	
	TypeSpec = • "bool" {A24}	bool	33	

№ сост	Конфигурация	Символ перехода (свертки)	Состояние-преемник	Свертка {действие}
	TypeSpec = • "array" "[" "num_int" "]" "of" TypeSpec {A25}	array	34	
23	Block = "var" LstVar "start" LstStmt "stop" •	eof		R2
24	LstStmt = LstStmt Stmt • ";"	;	35	
25	LstStmt = Stmt ";" •	id, stop, repeat, until		R13
26	Stmt = "id" C • AssignmentTarget "ass" Expr {A7}	AssignmentTarget	36	
	AssignmentTarget = • D	D	37	
	AssignmentTarget = • "[" Expr "]" {A28}	[	38	
	D = {A27} •	ass		R48 {A27}
27	Stmt = "repeat" LstStmt • "until" LogicalExpr {A26}	until	39	
	LstStmt = LstStmt • Stmt ";"	Stmt	24	
	Stmt = • "id" C AssignmentTarget "ass" Expr {A7}	id	17	
	Stmt = • "repeat" LstStmt "until" LogicalExpr {A26}	repeat	18	
28	LstId = "," "id" • B LstId {A4}	B	40	
	B = {A2} •	,, :		R46 {A2}
29	LstId = ":" TypeSpec {A5} •	;		R7 {A5}
30	TypeSpec = "integer" {A21} •	;		R8 {A21}
31	TypeSpec = "real" {A22} •	;		R9 {A22}
32	TypeSpec = "string" {A23} •	;		R10 {A23}
33	TypeSpec = "bool" {A24} •	;		R11 {A24}
34	TypeSpec = "array" • "[" "num_int" "]" "of" TypeSpec {A25}	[	41	
35	LstStmt = LstStmt Stmt ";" •	id, stop, repeat, until		R14
36	Stmt = "id" C AssignmentTarget • "ass" Expr {A7}	ass	42	
37	AssignmentTarget = D •	ass		R17
38	AssignmentTarget = "[" • Expr "]" {A28}	Expr	43	
	Expr = • SimpleExpr	SimpleExpr	44	
	Expr = • Expr "||" E SimpleExpr {A9}	Expr	43	
	SimpleExpr = • Term	Term	45	
	SimpleExpr = • SimpleExpr "+" F Term {A11}	SimpleExpr	44	
	SimpleExpr = • SimpleExpr "-" F Term {A30}	SimpleExpr	44	
	Term = • Factor	Factor	46	
	Term = • Term "*" G Factor {A13}	Term	45	
	Term = • Term "/" G Factor {A31}	Term	45	
	Factor = • "num_real" {A14}	num_real	47	
	Factor = • "num_int" {A15}	num_int	48	
	Factor = • "string_literal" {A17}	string_literal	49	
	Factor = • "bool_literal" {A18}	bool_literal	50	
	Factor = • "id" H FactorTail {A16}	id	51	
	Factor = • "(" LogicalExpr ")" {A32}	(	52	
39	Stmt = "repeat" LstStmt "until" • LogicalExpr {A26}	LogicalExpr	53	
	LogicalExpr = • SimpleLogicalExpr	SimpleLogicalExpr	54	
	LogicalExpr = • LogicalExpr RelOp M SimpleLogicalExpr {A39}	LogicalExpr	53	
	SimpleLogicalExpr = • Expr {A40}	Expr	55	
	Expr = • SimpleExpr	SimpleExpr	44	
	Expr = • Expr "||" E SimpleExpr {A9}	Expr	55	
	SimpleExpr = • Term	Term	45	
	SimpleExpr = • SimpleExpr "+" F Term {A11}	SimpleExpr	44	
	SimpleExpr = • SimpleExpr "-" F Term {A30}	SimpleExpr	44	
	Term = • Factor	Factor	46	

№ сост	Конфигурация	Символ перехода (свертки)	Состояние-преемник	Свертка {действие}
	Term = • Term "*" G Factor {A13}	Term	45	
	Term = • Term "/" G Factor {A31}	Term	45	
	Factor = • "num_real" {A14}	num_real	47	
	Factor = • "num_int" {A15}	num_int	48	
	Factor = • "string_literal" {A17}	string_literal	49	
	Factor = • "bool_literal" {A18}	bool_literal	50	
	Factor = • "id" H FactorTail {A16}	id	51	
	Factor = • "(" LogicalExpr ")" {A32}	(	52	
40	LstId = "," "id" B • LstId {A4}	LstId	56	
	LstId = • "," "id" B LstId {A4}	,	21	
	LstId = • ":" TypeSpec {A5}	:	22	
41	TypeSpec = "array" "[" • "num_int" "]" "of" TypeSpec {A25}	num_int	57	
42	Stmt = "id" C AssignmentTarget "ass" • Expr {A7}	Expr	58	
	Expr = • SimpleExpr	SimpleExpr	44	
	Expr = • Expr "||" E SimpleExpr {A9}	Expr	58	
	SimpleExpr = • Term	Term	45	
	SimpleExpr = • SimpleExpr "+" F Term {A11}	SimpleExpr	44	
	SimpleExpr = • SimpleExpr "-" F Term {A30}	SimpleExpr	44	
	Term = • Factor	Factor	46	
	Term = • Term "*" G Factor {A13}	Term	45	
	Term = • Term "/" G Factor {A31}	Term	45	
	Factor = • "num_real" {A14}	num_real	47	
	Factor = • "num_int" {A15}	num_int	48	
	Factor = • "string_literal" {A17}	string_literal	49	
	Factor = • "bool_literal" {A18}	bool_literal	50	
	Factor = • "id" H FactorTail {A16}	id	51	
	Factor = • "(" LogicalExpr ")" {A32}	(	52	
43	AssignmentTarget = "[" Expr • "]" {A28}	]	59	
	Expr = Expr • "||" E SimpleExpr {A9}	||	60	
44	Expr = SimpleExpr •	;, ,,], ||,), >, >=		R19
	SimpleExpr = SimpleExpr • "+" F Term {A11}	+	61	
	SimpleExpr = SimpleExpr • "-" F Term {A30}	-	62	
45	SimpleExpr = Term •	;, ,,], ||, +, -,), >, >=		R21
	Term = Term • "*" G Factor {A13}	*	63	
	Term = Term • "/" G Factor {A31}	/	64	
46	Term = Factor •	;, ,,], ||, +, -, *, /,), >, >=		R24
47	Factor = "num_real" {A14} •	;, ,,], ||, +, -, *, /,), >, >=		R27 {A14}
48	Factor = "num_int" {A15} •	;, ,,], ||, +, -, *, /,), >, >=		R28 {A15}
49	Factor = "string_literal" {A17} •	;, ,,], ||, +, -, *, /,), >, >=		R29 {A17}
50	Factor = "bool_literal" {A18} •	;, ,,], ||, +, -, *, /,), >, >=		R30 {A18}
51	Factor = "id" • H FactorTail {A16}	H	65	
	H = {A41} •	;, ,, [,], ||, +, -, *, /, (,), >, >=		R52 {A41}
52	Factor = "(" • LogicalExpr ")" {A32}	LogicalExpr	66	
	LogicalExpr = • SimpleLogicalExpr	SimpleLogicalExpr	54	
	LogicalExpr = • LogicalExpr RelOp M SimpleLogicalExpr {A39}	LogicalExpr	66	
	SimpleLogicalExpr = • Expr {A40}	Expr	55	
	Expr = • SimpleExpr	SimpleExpr	44	
	Expr = • Expr "||" E SimpleExpr {A9}	Expr	55	

№ сост	Конфигурация	Символ перехода (свертки)	Состояние-преемник	Свертка {действие}
	SimpleExpr = • Term	Term	45	
	SimpleExpr = • SimpleExpr "+" F Term {A11}	SimpleExpr	44	
	SimpleExpr = • SimpleExpr "-" F Term {A30}	SimpleExpr	44	
	Term = • Factor	Factor	46	
	Term = • Term "*" G Factor {A13}	Term	45	
	Term = • Term "/" G Factor {A31}	Term	45	
	Factor = • "num_real" {A14}	num_real	47	
	Factor = • "num_int" {A15}	num_int	48	
	Factor = • "string_literal" {A17}	string_literal	49	
	Factor = • "bool_literal" {A18}	bool_literal	50	
	Factor = • "id" H FactorTail {A16}	id	51	
	Factor = • "(" LogicalExpr ")" {A32}	(	52	
53	Stmt = "repeat" LstStmt "until" LogicalExpr {A26} •	;		R16 {A26}
	LogicalExpr = LogicalExpr • RelOp M SimpleLogicalExpr {A39}	RelOp	67	
	RelOp = • ">" {A19}	>	68	
	RelOp = • ">=" {A20}	>=	69	
54	LogicalExpr = SimpleLogicalExpr •	;,), >, >=		R40
55	SimpleLogicalExpr = Expr {A40} •	;,), >, >=		R42 {A40}
	Expr = Expr • "||" E SimpleExpr {A9}	||	60	
56	LstId = "," "id" B LstId {A4} •	;		R6 {A4}
57	TypeSpec = "array" "[" "num_int" • "]" "of" TypeSpec {A25}	]	70	
58	Stmt = "id" C AssignmentTarget "ass" Expr {A7} •	;		R15 {A7}
	Expr = Expr • "||" E SimpleExpr {A9}	||	60	
59	AssignmentTarget = "[" Expr "]" {A28} •	ass		R18 {A28}
60	Expr = Expr "||" • E SimpleExpr {A9}	E	71	
	E = {A8} •	id, num_int, num_real, string_literal, bool_literal, (		R49 {A8}
61	SimpleExpr = SimpleExpr "+" • F Term {A11}	F	72	
	F = {A10} •	id, num_int, num_real, string_literal, bool_literal, (		R50 {A10}
62	SimpleExpr = SimpleExpr "-" • F Term {A30}	F	73	
	F = {A10} •	id, num_int, num_real, string_literal, bool_literal, (		R50 {A10}
63	Term = Term "*" • G Factor {A13}	G	74	
	G = {A12} •	id, num_int, num_real, string_literal, bool_literal, (		R51 {A12}
64	Term = Term "/" • G Factor {A31}	G	75	
	G = {A12} •	id, num_int, num_real, string_literal, bool_literal, (		R51 {A12}
65	Factor = "id" H • FactorTail {A16}	FactorTail	76	
	FactorTail = • I	I	77	
	FactorTail = • "[" Expr "]" {A28}	[	78	
	FactorTail = • "(" Args ")" {A34}	(	79	
	I = {A33} •	;, ,,], ||, +, -, *, /,), >, >=		R53 {A33}
66	Factor = "(" LogicalExpr • ")" {A32}	)	80	
	LogicalExpr = LogicalExpr • RelOp M SimpleLogicalExpr {A39}	RelOp	67	
	RelOp = • ">" {A19}	>	68	
	RelOp = • ">=" {A20}	>=	69	
67	LogicalExpr = LogicalExpr RelOp • M SimpleLogicalExpr {A39}	M	81	
	M = {A43} •	id, num_int, num_real, string_literal, bool_literal, (		R57 {A43}

№ сост	Конфигурация	Символ перехода (свертки)	Состояние-преемник	Свертка {действие}
68	RelOp = ">" {A19} •	id, num_int, num_real, string_literal, bool_literal, (		R43 {A19}
69	RelOp = ">=" {A20} •	id, num_int, num_real, string_literal, bool_literal, (		R44 {A20}
70	TypeSpec = "array" "[" "num_int" "]" • "of" TypeSpec {A25}	of	82	
71	Expr = Expr "||" E • SimpleExpr {A9}	SimpleExpr	83	
	SimpleExpr = • Term	Term	45	
	SimpleExpr = • SimpleExpr "+" F Term {A11}	SimpleExpr	83	
	SimpleExpr = • SimpleExpr "-" F Term {A30}	SimpleExpr	83	
	Term = • Factor	Factor	46	
	Term = • Term "*" G Factor {A13}	Term	45	
	Term = • Term "/" G Factor {A31}	Term	45	
	Factor = • "num_real" {A14}	num_real	47	
	Factor = • "num_int" {A15}	num_int	48	
	Factor = • "string_literal" {A17}	string_literal	49	
	Factor = • "bool_literal" {A18}	bool_literal	50	
	Factor = • "id" H FactorTail {A16}	id	51	
	Factor = • "(" LogicalExpr ")" {A32}	(	52	
72	SimpleExpr = SimpleExpr "+" F • Term {A11}	Term	84	
	Term = • Factor	Factor	46	
	Term = • Term "*" G Factor {A13}	Term	84	
	Term = • Term "/" G Factor {A31}	Term	84	
	Factor = • "num_real" {A14}	num_real	47	
	Factor = • "num_int" {A15}	num_int	48	
	Factor = • "string_literal" {A17}	string_literal	49	
	Factor = • "bool_literal" {A18}	bool_literal	50	
	Factor = • "id" H FactorTail {A16}	id	51	
	Factor = • "(" LogicalExpr ")" {A32}	(	52	
73	SimpleExpr = SimpleExpr "-" F • Term {A30}	Term	85	
	Term = • Factor	Factor	46	
	Term = • Term "*" G Factor {A13}	Term	85	
	Term = • Term "/" G Factor {A31}	Term	85	
	Factor = • "num_real" {A14}	num_real	47	
	Factor = • "num_int" {A15}	num_int	48	
	Factor = • "string_literal" {A17}	string_literal	49	
	Factor = • "bool_literal" {A18}	bool_literal	50	
	Factor = • "id" H FactorTail {A16}	id	51	
	Factor = • "(" LogicalExpr ")" {A32}	(	52	
74	Term = Term "*" G • Factor {A13}	Factor	86	
	Factor = • "num_real" {A14}	num_real	47	
	Factor = • "num_int" {A15}	num_int	48	
	Factor = • "string_literal" {A17}	string_literal	49	
	Factor = • "bool_literal" {A18}	bool_literal	50	
	Factor = • "id" H FactorTail {A16}	id	51	
	Factor = • "(" LogicalExpr ")" {A32}	(	52	
75	Term = Term "/" G • Factor {A31}	Factor	87	
	Factor = • "num_real" {A14}	num_real	47	
	Factor = • "num_int" {A15}	num_int	48	
	Factor = • "string_literal" {A17}	string_literal	49	
	Factor = • "bool_literal" {A18}	bool_literal	50	

№ сост	Конфигурация	Символ перехода (свертки)	Состояние-преемник	Свертка {действие}
	Factor = • "id" H FactorTail {A16}	id	51	
	Factor = • "(" LogicalExpr ")" {A32}	(	52	
76	Factor = "id" H FactorTail {A16} •	;, ,,], ||, +, -, *, /,), >, >=		R31 {A16}
77	FactorTail = I •	;, ,,], ||, +, -, *, /,), >, >=		R33
78	FactorTail = "[" • Expr "]" {A28}	Expr	88	
	Expr = • SimpleExpr	SimpleExpr	44	
	Expr = • Expr "||" E SimpleExpr {A9}	Expr	88	
	SimpleExpr = • Term	Term	45	
	SimpleExpr = • SimpleExpr "+" F Term {A11}	SimpleExpr	44	
	SimpleExpr = • SimpleExpr "-" F Term {A30}	SimpleExpr	44	
	Term = • Factor	Factor	46	
	Term = • Term "*" G Factor {A13}	Term	45	
	Term = • Term "/" G Factor {A31}	Term	45	
	Factor = • "num_real" {A14}	num_real	47	
	Factor = • "num_int" {A15}	num_int	48	
	Factor = • "string_literal" {A17}	string_literal	49	
	Factor = • "bool_literal" {A18}	bool_literal	50	
	Factor = • "id" H FactorTail {A16}	id	51	
	Factor = • "(" LogicalExpr ")" {A32}	(	52	
79	FactorTail = "(" • Args ")" {A34}	Args	89	
	Args = • Expr J ArgsTail {A35}	Expr	90	
	Args = • K	K	91	
	Expr = • SimpleExpr	SimpleExpr	44	
	Expr = • Expr "||" E SimpleExpr {A9}	Expr	90	
	K = {A36} •	)		R55 {A36}
	SimpleExpr = • Term	Term	45	
	SimpleExpr = • SimpleExpr "+" F Term {A11}	SimpleExpr	44	
	SimpleExpr = • SimpleExpr "-" F Term {A30}	SimpleExpr	44	
	Term = • Factor	Factor	46	
	Term = • Term "*" G Factor {A13}	Term	45	
	Term = • Term "/" G Factor {A31}	Term	45	
	Factor = • "num_real" {A14}	num_real	47	
	Factor = • "num_int" {A15}	num_int	48	
	Factor = • "string_literal" {A17}	string_literal	49	
	Factor = • "bool_literal" {A18}	bool_literal	50	
	Factor = • "id" H FactorTail {A16}	id	51	
	Factor = • "(" LogicalExpr ")" {A32}	(	52	
80	Factor = "(" LogicalExpr ")" {A32} •	;, ,,], ||, +, -, *, /,), >, >=		R32 {A32}
81	LogicalExpr = LogicalExpr RelOp M • SimpleLogicalExpr {A39}	SimpleLogicalExpr	92	
	SimpleLogicalExpr = • Expr {A40}	Expr	55	
	Expr = • SimpleExpr	SimpleExpr	44	
	Expr = • Expr "||" E SimpleExpr {A9}	Expr	55	
	SimpleExpr = • Term	Term	45	
	SimpleExpr = • SimpleExpr "+" F Term {A11}	SimpleExpr	44	
	SimpleExpr = • SimpleExpr "-" F Term {A30}	SimpleExpr	44	
	Term = • Factor	Factor	46	
	Term = • Term "*" G Factor {A13}	Term	45	
	Term = • Term "/" G Factor {A31}	Term	45	
	Factor = • "num_real" {A14}	num_real	47	

№ сост	Конфигурация	Символ перехода (свертки)	Состояние-преемник	Свертка {действие}
	Factor = • "num_int" {A15}	num_int	48	
	Factor = • "string_literal" {A17}	string_literal	49	
	Factor = • "bool_literal" {A18}	bool_literal	50	
	Factor = • "id" H FactorTail {A16}	id	51	
	Factor = • "(" LogicalExpr ")" {A32}	(	52	
82	TypeSpec = "array" "[" "num_int" "]" "of" • TypeSpec {A25}	TypeSpec	93	
	TypeSpec = • "integer" {A21}	integer	30	
	TypeSpec = • "real" {A22}	real	31	
	TypeSpec = • "string" {A23}	string	32	
	TypeSpec = • "bool" {A24}	bool	33	
	TypeSpec = • "array" "[" "num_int" "]" "of" TypeSpec {A25}	array	34	
83	Expr = Expr "||" E SimpleExpr {A9} •	;, ,,], ||,), >, >=		R20 {A9}
	SimpleExpr = SimpleExpr • "+" F Term {A11}	+	61	
	SimpleExpr = SimpleExpr • "-" F Term {A30}	-	62	
84	SimpleExpr = SimpleExpr "+" F Term {A11} •	;, ,,], ||, +, -,), >, >=		R22 {A11}
	Term = Term • "*" G Factor {A13}	*	63	
	Term = Term • "/" G Factor {A31}	/	64	
85	SimpleExpr = SimpleExpr "-" F Term {A30} •	;, ,,], ||, +, -,), >, >=		R23 {A30}
	Term = Term • "*" G Factor {A13}	*	63	
	Term = Term • "/" G Factor {A31}	/	64	
86	Term = Term "*" G Factor {A13} •	;, ,,], ||, +, -, *, /,), >, >=		R25 {A13}
87	Term = Term "/" G Factor {A31} •	;, ,,], ||, +, -, *, /,), >, >=		R26 {A31}
88	FactorTail = "[" Expr • "]" {A28}	]	94	
	Expr = Expr • "||" E SimpleExpr {A9}	||	60	
89	FactorTail = "(" Args • ")" {A34}	)	95	
90	Args = Expr • J ArgsTail {A35}	J	96	
	Expr = Expr • "||" E SimpleExpr {A9}	||	60	
	J = {A42} •	,,)		R54 {A42}
91	Args = K •	)		R37
92	LogicalExpr = LogicalExpr RelOp M SimpleLogicalExpr {A39} •	;,), >, >=		R41 {A39}
93	TypeSpec = "array" "[" "num_int" "]" "of" TypeSpec {A25} •	;		R12 {A25}
94	FactorTail = "[" Expr "]" {A28} •	;, ,,], ||, +, -, *, /,), >, >=		R34 {A28}
95	FactorTail = "(" Args ")" {A34} •	;, ,,], ||, +, -, *, /,), >, >=		R35 {A34}
96	Args = Expr J • ArgsTail {A35}	ArgsTail	97	
	ArgsTail = • L	L	98	
	ArgsTail = • "," Expr J ArgsTail {A38}	,	99	
	L = {A37} •	)		R56 {A37}
97	Args = Expr J ArgsTail {A35} •	)		R36 {A35}
98	ArgsTail = L •	)		R38
99	ArgsTail = "," • Expr J ArgsTail {A38}	Expr	100	
	Expr = • SimpleExpr	SimpleExpr	44	
	Expr = • Expr "||" E SimpleExpr {A9}	Expr	100	
	SimpleExpr = • Term	Term	45	
	SimpleExpr = • SimpleExpr "+" F Term {A11}	SimpleExpr	44	
	SimpleExpr = • SimpleExpr "-" F Term {A30}	SimpleExpr	44	
	Term = • Factor	Factor	46	
	Term = • Term "*" G Factor {A13}	Term	45	
	Term = • Term "/" G Factor {A31}	Term	45	

№ сост	Конфигурация	Символ перехода (свертки)	Состояние-преемник	Свертка {действие}
	Factor = • "num_real" {A14}	num_real	47	
	Factor = • "num_int" {A15}	num_int	48	
	Factor = • "string_literal" {A17}	string_literal	49	
	Factor = • "bool_literal" {A18}	bool_literal	50	
	Factor = • "id" H FactorTail {A16}	id	51	
	Factor = • "(" LogicalExpr ")" {A32}	(	52	
100	ArgsTail = "," Expr • J ArgsTail {A38}	J	101	
	Expr = Expr • "||" E SimpleExpr {A9}	||	60	
	J = {A42} •	,,)		R54 {A42}
101	ArgsTail = "," Expr J • ArgsTail {A38}	ArgsTail	102	
	ArgsTail = • L	L	98	
	ArgsTail = • "," Expr J ArgsTail {A38}	,	99	
	L = {A37} •	)		R56 {A37}
102	ArgsTail = "," Expr J ArgsTail {A38} •	)		R39 {A38}

SLR(1)-грамматика

Множества Follow нетерминалов

Follow(S)={}; Follow(A)={;}; Follow(Block)={eof}; Follow(LstVar)={id,start}; Follow(LstStmt)={id,stop,repeat,until};
Follow(DeclVar)={;}; Follow(B)={,,:}; Follow(LstId)={;}; Follow(TypeSpec)={;}; Follow(Stmt)={;}; Follow(C)={[,ass};
Follow(AssignmentTarget)={ass}; Follow(Expr)={;,,,],||,),>,>=}; Follow(LogicalExpr)={;,),>,>=}; Follow(D)={ass};
Follow(SimpleExpr)={;,,,],||,+,-,),>,>=}; Follow(E)={id,num_int,num_real,string_literal,bool_literal,(};
Follow(Term)={;,,,],||,+,-,*,/,),>,>=}; Follow(F)={id,num_int,num_real,string_literal,bool_literal,(};
Follow(Factor)={;,,,],||,+,-,*,/,),>,>=}; Follow(G)={id,num_int,num_real,string_literal,bool_literal,(};
Follow(H)={;,,,[,],||,+,-,*,/,(,),>,>=}; Follow(FactorTail)={;,,,],||,+,-,*,/,),>,>=}; Follow(I)={;,,,],||,+,-,*,/,),>,>=};
Follow(Args)={)}; Follow(J)={,,)}; Follow(ArgsTail)={)}; Follow(K)={)}; Follow(L)={)};
Follow(SimpleLogicalExpr)={;,),>,>=}; Follow(RelOp)={id,num_int,num_real,string_literal,bool_literal,(};
Follow(M)={id,num_int,num_real,string_literal,bool_literal,(};