	EPSILON_VALUE	IDENTIFIER	TYPE_DEFINITION	CONST	END	ASSIGNATION	END_OF_INSTRUCTION	BOOLEAN_TYPE	REAL_TYPE	INTEGER_TYPE	WHILE	FOR	TERNARY_ELSE	COMMA	TERNARY_IF	INTEGER	REAL	BOOLEAN	NEGATION	BITWISE_NOT	MINUS	PLUS	LEFT_PARENTHESIS	RIGHT_PARENTHESIS	LAZY_OR	LAZY_AND	GREATER_THAN	LESS_THAN	GREATER_OR_EQUALS_THAN	LESS_OR_EQUALS_THAN	EQUALITY	INEQUALITY	BITWISE_OR	BITWISE_XOR	ARITHMETIC_SHIFT_LEFT	ARITHMETIC_SHIFT_RIGHT	TIMES	DIVIDE	REMAINDER	BITWISE_AND	INVERSE_DIVIDE	POWER	IF	ELSE_IF	ELSE	LET	BOOLEAN_TYPE	REAL_TYPE	INTEGER_TYPE	READ_REAL	READ_INTEGER	INTEGER_CAST	REAL_CAST	BOOLEAN_CAST	PRINTLN	FUNCTION	RETURN
<Program>	1	1		1			1				1	1																															1			1				1	1	1	1	1	1	1	
<Instruction>		2		3							5	5																															5			4				6	6	6	6	6	6	7	
<InstructionList>	9	8		8	9		9				8	8																															8	9	9	8				8	8	8	8	8	8	8	9
<InstructionListTail>	11				11		10																																					11	11												11
<IdentifierInstruction>		12																																																							
<IdentifierInstructionTail>			14			13								13									15																																			
<AssignationTail>						16								17																																											
<ConstDefinition>				18																																																					
<Block>																																													19												
<Loop>											21	22																													20															
<ForTail>							23						24																																											
<Type>								25	26	27																																				25	26	27								
<Expression>		28														28	28	28	28	28	28	28	28																											28	28	28	28	28	28		
<TernaryIfExpression>	30				30		30						30	30	29									30																				30	30													30
<TernaryElseExpression>													31																																											
<AtomicExpression>		32														33	34	35																															36	36	36	36	36	36		
<AtomicIdentifierExpression>		37																																																							
<AtomicIdentifierExpressionTail>	39				39		39						39	39	39						39	39	38	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39	39		39	39													39
<UnaryExpression>		41														41	41	41	40	41	41	41	41																											41	41	41	41	41	41		
<UnaryBitwiseNotExpression>		43														43	43	43		42	43	43	43																											43	43	43	43	43	43		
<UnaryMinusPlusExpression>		46														46	46	46			44	45	46																											46	46	46	46	46	46		
<UnaryAtomicExpression>		47														47	47	47					48																											47	47	47	47	47	47		
<BinaryExpression>		49														49	49	49	49	49	49	49	49																											49	49	49	49	49	49		
<BinaryExpression'>	51				51		51						51	51	51									51	50																			51	51													51
<BinaryLazyOrExpression>		52														52	52	52	52	52	52	52	52																											52	52	52	52	52	52		
<BinaryLazyOrExpression'>	54				54		54						54	54	54									54	54	53																		54	54													54
<BinaryLazyAndExpression>		55														55	55	55	55	55	55	55	55																											55	55	55	55	55	55		
<BinaryLazyAndExpression'>	62				62		62						62	62	62									62	62	62	56	57	58	59	60	61												62	62													62
<BinaryNumericExpression>		63														63	63	63	63	63	63	63	63																											63	63	63	63	63	63		
<BinaryNumericExpression'>	68				68		68						68	68	68						65	64		68	68	68	68	68	68	68	68	68	66	67										68	68													68
<BinaryTermExpression>		69														69	69	69	69	69	69	69	69																											69	69	69	69	69	69		
<BinaryTermExpression'>	72				72		72						72	72	72						72	72		72	72	72	72	72	72	72	72	72	72	72	70	71								72	72													72
<BinaryShiftedExpression>		73														73	73	73	73	73	73	73	73																											73	73	73	73	73	73		
<BinaryShiftedExpression'>	79				79		79						79	79	79						79	79		79	79	79	79	79	79	79	79	79	79	79	79	74	75	76	77	78			79	79													79
<BinaryFactorExpression>		80														80	80	80	80	80	80	80	80																											80	80	80	80	80	80		
<BinaryFactorExpression'>	82				82		82						82	82	82						82	82		82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	81			82	82													82
<If>																																										83															
<IfEnd>					86																																						84	85													
<BuiltInFunctionCall>																																																	87	88	89	90	91	92		
<FunctionCallTail>																							93																																		
<Parameter>	95	94														94	94	94	94	94	94	94	94	95																										94	94	94	94	94	94		
<ParameterTail>	97													96										97																																
<FunctionDefinition>																																																							98	
<FunctionDefinitionEnd>					100																																																			99
<Argument>	102	101																						102																																
<ArgumentTail>	104													103										104																																