	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_XOR	BITWISE_OR	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	1	
<InstructionList>	9	2		3	9		8				5	5																														1	5	9	9	4			6	6	6	6	6	6	7	9	
<IdentifierInstruction>		10																																																						
<IdentifierInstructionTail>			12			11								11									13																																	
<AssignationTail>						14								15																																										
<ConstDefinition>				16																																																			
<Block>																																																17								
<Loop>											19	20																															18													
<ForTail>							21						22																																											
<Type>								23	24	25																																								23	24	25						
<Expression>		26														26	26	26	26	26	26	26	26																													26	26	26	26	26	26			
<TernaryIfExpression>	28			28		28							28	28	27									28																																		
<TernaryElseExpression>													29																																												
<AtomicExpression>		30														31	32	33																																	34	34	34	34	34	34			
<AtomicIdentifierExpression>		35																																																						
<AtomicIdentifierExpressionTail>	37			37		37							37	37	37						37	37	36	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37																	
<UnaryExpression>		39														39	39	39	38	39	39	39	39																													39	39	39	39	39	39			
<UnaryBitwiseNotExpression>		41														41	41	41		40	41	41	41																													41	41	41	41	41	41			
<UnaryMinusPlusExpression>		44														44	44	44			42	43	44																													44	44	44	44	44	44			
<UnaryAtomicExpression>		45														45	45	45					46																													45	45	45	45	45	45			
<BinaryExpression>		47														47	47	47	47	47	47	47	47																													47	47	47	47	47	47			
<BinaryExpression'>	49			49		49							49	49	49									49	48																																		
<BinaryLazyOrExpression>		50														50	50	50	50	50	50	50	50																													50	50	50	50	50	50			
<BinaryLazyOrExpression'>	52			52		52							52	52	52									52	52	51																																	
<BinaryLazyAndExpression>		53														53	53	53	53	53	53	53	53																													53	53	53	53	53	53			
<BinaryLazyAndExpression'>	60			60		60							60	60	60									60	60	60	54	55	56	57	58	59																									
<BinaryNumericExpression>		61														61	61	61	61	61	61	61	61																													61	61	61	61	61	61			
<BinaryNumericExpression'>	66			66		66							66	66	66						63	62		66	66	66	66	66	66	66	66	66	64	65																							
<BinaryTermExpression>		67														67	67	67	67	67	67	67	67																													67	67	67	67	67	67			
<BinaryTermExpression'>	70			70		70							70	70	70						70	70		70	70	70	70	70	70	70	70	70	70	70	68	69																					
<BinaryShiftedExpression>		71														71	71	71	71	71	71	71	71																													71	71	71	71	71	71			
<BinaryShiftedExpression'>	77			77		77							77	77	77						77	77		77	77	77	77	77	77	77	77	77	77	77	77	77	72	73	74	75	76																	
<BinaryFactorExpression>		78														78	78	78	78	78	78	78	78																													78	78	78	78	78	78			
<BinaryFactorExpression'>	80			80		80							80	80	80						80	80		80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	80	79																	
<If>																																																									
<IfEnd>				84																																																					
<BuiltInFunctionCall>																																																								
<FunctionCallTail>																							91																																		
<Parameter>	93	92														92	92	92	92	92	92	92	92	93																													92	92	92	92	92	92			
<ParameterTail>	95													94										95																																		
<FunctionDefinition>																																																								96	
<FunctionDefinitionEnd>					98																																																				97
<Argument>	100	99																						100																																	
<ArgumentTail>	102													101										102																																	