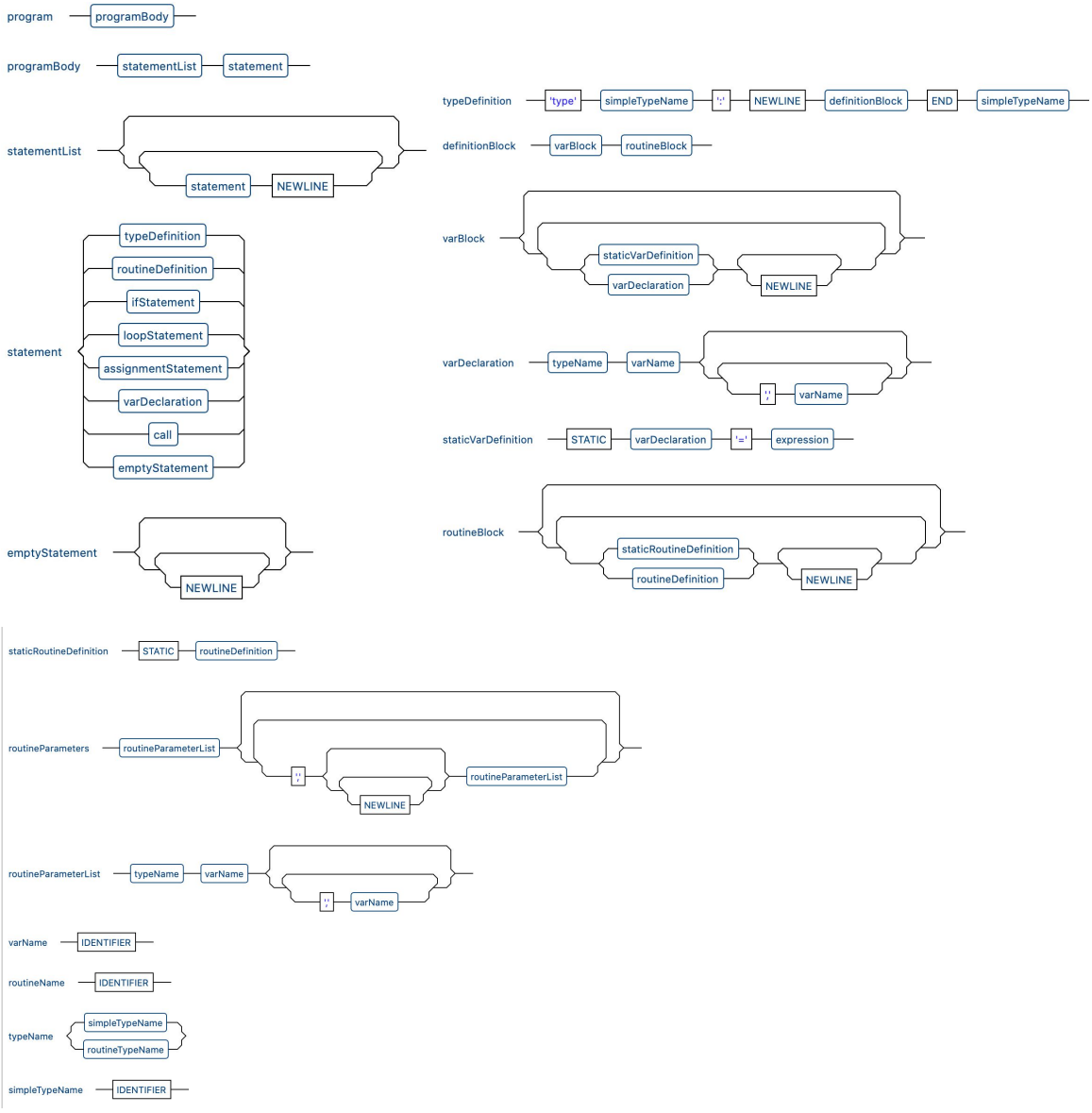
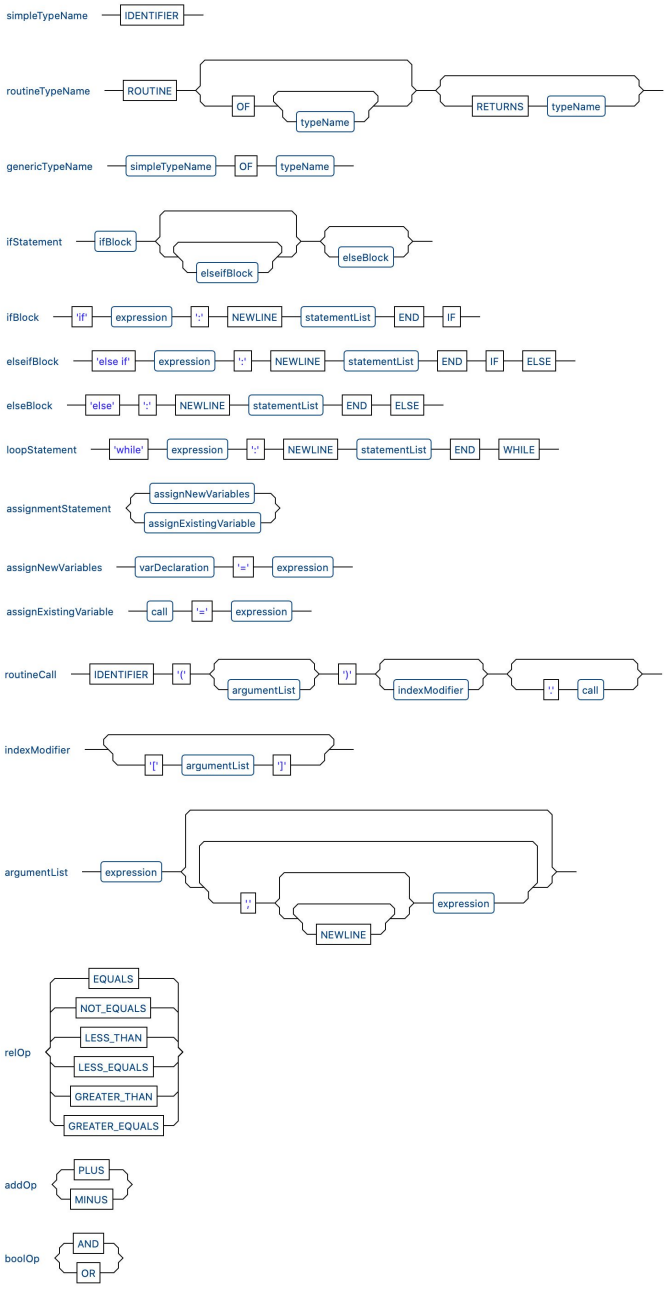
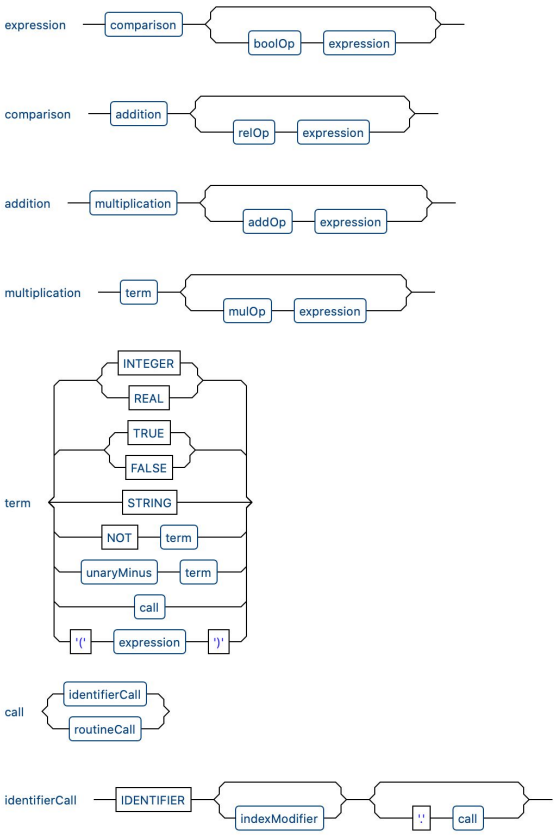
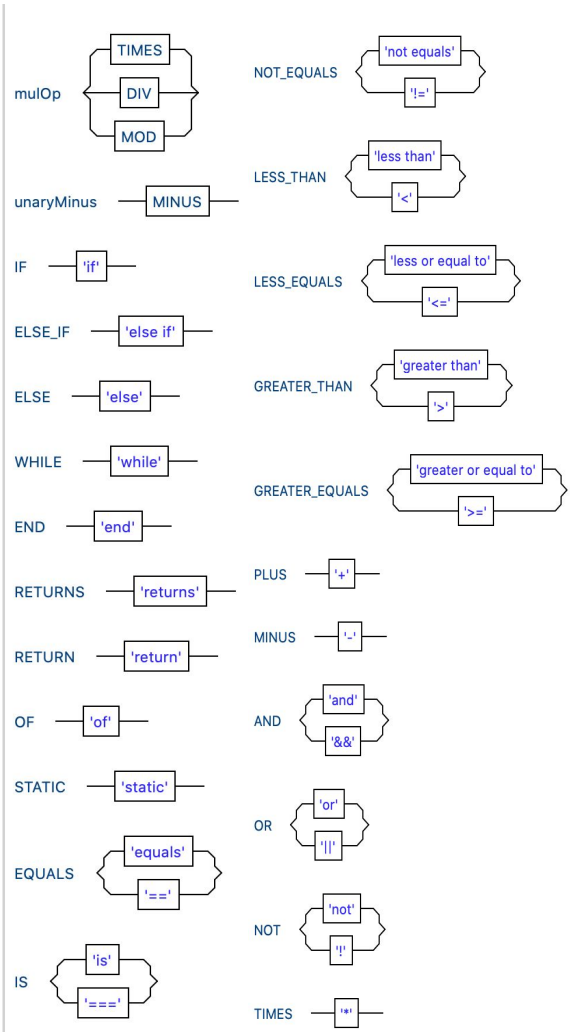
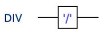

program
programBody
programBody
statementList
statement
statementList
statement
NEWLINE
statement
typeDefinition
routineDefinition
ifStatement
loopStatement
assignmentStatement
varDeclaration
call
emptyStatement
emptyStatement
NEWLINE
typeDefinition
'type'
simpleTypeName
':'
NEWLINE
definitionBlock
END
simpleTypeName
definitionBlock
varBlock
routineBlock
varBlock
staticVarDefinition
varDeclaration
NEWLINE
varDeclaration
typeName
varName
','
varName
staticVarDefinition
STATIC
varDeclaration
'='
expression
routineBlock
staticRoutineDefinition
routineDefinition
NEWLINE
staticRoutineDefinition
STATIC
routineDefinition
routineParameters
routineParameterList
','
NEWLINE
routineParameterList
routineParameterList
typeName
varName
','
varName
varName
IDENTIFIER
routineName
IDENTIFIER
typeName
simpleTypeName
routineTypeName
simpleTypeName
IDENTIFIER

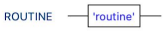

simpleTypeName
IDENTIFIER
routineTypeName
ROUTINE
OF
typeName
RETURNS
typeName
genericTypeName
simpleTypeName
OF
typeName
ifStatement
ifBlock
elseifBlock
elseBlock
ifBlock
'if'
expression
':'
NEWLINE
statementList
END
IF
elseifBlock
'else if'
expression
':'
NEWLINE
statementList
END
IF
ELSE
elseBlock
'else'
':'
NEWLINE
statementList
END
ELSE
loopStatement
'while'
expression
':'
NEWLINE
statementList
END
WHILE
assignmentStatement
assignNewVariables
assignExistingVariable
assignNewVariables
varDeclaration
'='
expression
assignExistingVariable
call
'='
expression
routineCall
IDENTIFIER
'('
argumentList
')'
indexModifier
','
call
indexModifier
'['
argumentList
']'
argumentList
expression
','
NEWLINE
expression
relOp
EQUALS
NOT_EQUALS
LESS_THAN
LESS_EQUALS
GREATER_THAN
GREATER_EQUALS
addOp
PLUS
MINUS
boolOp
AND
OR

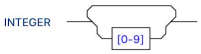

expression
comparison
boolOp
expression
comparison
addition
relOp
expression
addition
multiplication
addOp
expression
multiplication
term
mulOp
expression
term
INTEGER
REAL
TRUE
FALSE
STRING
NOT
term
unaryMinus
term
call
'('
expression
')'
call
identifierCall
routineCall
identifierCall
IDENTIFIER
indexModifier
','
call

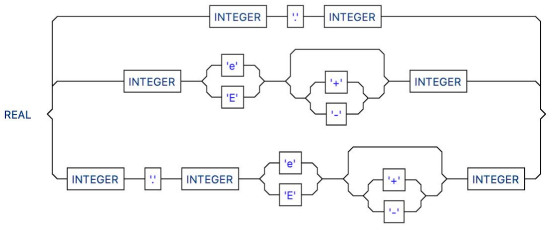

mulOp
TIMES
DIV
MOD
unaryMinus
MINUS
IF
'if'
ELSE_IF
'else if'
ELSE
'else'
WHILE
'while'
END
'end'
RETURNS
'returns'
RETURN
'return'
OF
'of'
STATIC
'static'
EQUALS
'equals'
'=='
IS
'is'
'==='
NOT_EQUALS
'not equals'
'!='
LESS_THAN
'less than'
'<'
LESS_EQUALS
'less or equal to'
'<='
GREATER_THAN
'greater than'
'>'
GREATER_EQUALS
'greater or equal to'
'>='
PLUS
'+'
MINUS
'-'
AND
'and'
'&&'
OR
'or'
'||'
NOT
'not'
'!'
TIMES
'*'

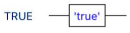

DIV
'/'

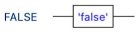

MOD
'mod'
'%'

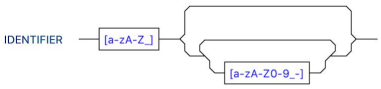

ROUTINE
'routine'


INTEGER
[0-9]


REAL
INTEGER
'.'
INTEGER
INTEGER
'e'
'E'
'+'
'-'
INTEGER
INTEGER
'.'
INTEGER
'e'
'E'
'+'
'-'
INTEGER

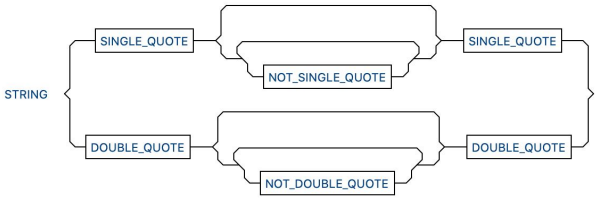

TRUE
'true'


FALSE
'false'


IDENTIFIER
[a-zA-Z_]
[a-zA-Z0-9_-]

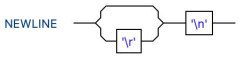

SINGLE_QUOTE
'''

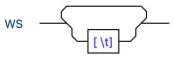

DOUBLE_QUOTE
'"'

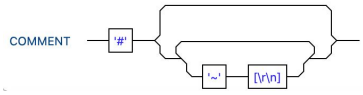

STRING
SINGLE_QUOTE
NOT_SINGLE_QUOTE
SINGLE_QUOTE
DOUBLE_QUOTE
NOT_DOUBLE_QUOTE
DOUBLE_QUOTE


NOT_SINGLE_QUOTE
'~'
'


NOT_DOUBLE_QUOTE
'~'
"


NEWLINE
'\r'
'\n'


WS
[\t]


COMMENT
'#'
'~'
[\r\n]