

The Language Preprocessed HAL/S

BNF-converter

December 16, 2022

This document was automatically generated by the *BNF-Converter*. It was generated together with the lexer, the parser, and the abstract syntax module, which guarantees that the document matches with the implementation of the language (provided no hand-hacking has taken place).

The lexical structure of HAL/S

Literals

BitIdentifierToken = {"b_"} $\langle letter \rangle ((\langle letter \rangle \mid \langle digit \rangle \mid \{ _ \}) * (\langle letter \rangle \mid \langle digit \rangle))?$
BitFunctionIdentifierToken = {"bf_"} $\langle letter \rangle ((\langle letter \rangle \mid \langle digit \rangle \mid \{ _ \}) * (\langle letter \rangle \mid \langle digit \rangle))?$
CharFunctionIdentifierToken = {"cf_"} $\langle letter \rangle ((\langle letter \rangle \mid \langle digit \rangle \mid \{ _ \}) * (\langle letter \rangle \mid \langle digit \rangle))?$
CharIdentifierToken = {"c_"} $\langle letter \rangle ((\langle letter \rangle \mid \langle digit \rangle \mid \{ _ \}) * (\langle letter \rangle \mid \langle digit \rangle))?$
StructIdentifierToken = {"s_"} $\langle letter \rangle ((\langle letter \rangle \mid \langle digit \rangle \mid \{ _ \}) * (\langle letter \rangle \mid \langle digit \rangle))?$
StructFunctionIdentifierToken = {"sf_"} $\langle letter \rangle ((\langle letter \rangle \mid \langle digit \rangle \mid \{ _ \}) * (\langle letter \rangle \mid \langle digit \rangle))?$
LabelToken = {"l_"} $\langle letter \rangle ((\langle letter \rangle \mid \langle digit \rangle \mid \{ _ \}) * (\langle letter \rangle \mid \langle digit \rangle))?$
EventToken = {"e_"} $\langle letter \rangle ((\langle letter \rangle \mid \langle digit \rangle \mid \{ _ \}) * (\langle letter \rangle \mid \langle digit \rangle))?$
ArithFieldToken = {"a_"} $\langle letter \rangle ((\langle letter \rangle \mid \langle digit \rangle \mid \{ _ \}) * (\langle letter \rangle \mid \langle digit \rangle))?$
IdentifierToken = $\langle letter \rangle ((\langle letter \rangle \mid \langle digit \rangle \mid \{ _ \}) * (\langle letter \rangle \mid \langle digit \rangle))?$
StringToken = ["'"] $(\langle letter \rangle \mid \langle digit \rangle \mid \{ _ \} \mid [_ -] \% \$ () [* . / \& \neg = < > \# @ , ; : \{ \} ! ?] * [_])$
TextToken = [""] $(\langle letter \rangle \mid \langle digit \rangle \mid [_ -] \% \$ () [* . / \& \neg = < > \# @ , ; : \{ \} ! ?] * [_])$
LevelToken = ["123456789"] \mid ["12"] $\langle digit \rangle$
NumberToken = $\langle digit \rangle +$
CompoundToken = $(\langle digit \rangle + ([_ .] \langle digit \rangle *)? \mid [_ .] \langle digit \rangle +) ([_ \text{EBH}] [_ -]? \langle digit \rangle +)?$

Reserved words and symbols

The set of reserved words is the set of terminals appearing in the grammar. Those reserved words that consist of non-letter characters are called symbols, and they are treated in a different way from those that are similar to identifiers. The lexer follows rules familiar from languages like Haskell, C, and Java, including longest match and spacing conventions.

The reserved words used in HAL/S are the following:

ABS	ABVAL	ACCESS
AFTER	ALIGNED	AND
ARCCOS	ARCCOSH	ARCSIN
ARCSINH	ARCTAN	ARCTAN2
ARCTANH	ARRAY	ASSIGN
AT	AUTOMATIC	BIN
BIT	BOOLEAN	BY
CALL	CANCEL	CASE
CAT	CEILING	CHAR
CHARACTER	CLOCKTIME	CLOSE
COLUMN	COMPOOL	CONSTANT
COS	COSH	DATE
DEC	DECLARE	DENSE
DEPENDENT	DET	DIV
DO	DOUBLE	ELSE
END	EQUATE	ERRGRP
ERROR	EVENT	EVERY
EXCLUSIVE	EXIT	EXP
EXTERNAL	FALSE	FILE
FLOOR	FOR	FUNCTION
GO	HEX	IF
IGNORE	IN	INDEX
INITIAL	INTEGER	INVERSE
LATCHED	LENGTH	LINE
LJUST	LOCK	LOG
MATRIX	MAX	MIDVAL
MIN	MOD	NAME
NEXTTIME	NONHAL	NOT
NULL	OCT	ODD
OFF	ON	OR
PAGE	PRIO	PRIORITY
PROCEDURE	PROD	PROGRAM
RANDOM	READ	READALL
REENTRANT	REMAINDER	REMOTE
REPEAT	REPLACE	RESET
RETURN	RIGID	RJUST
ROUND	RUNTIME	SCALAR
SCHEDULE	SEND	SET
SHL	SHR	SIGN
SIGNAL	SIGNUM	SIN
SINGLE	SINH	SIZE
SKIP	SQRT	STATIC
STRUCTURE	SUBBIT	SUM
SYSTEM	TAB	TAN
TANH	TASK	TEMPORARY
TERMINATE	THEN	TO
TRACE	TRANSPOSE	TRIM
TRUE	TRUNCATE	UNIT
UNTIL	UPDATE	VECTOR
WAIT	WHILE	WRITE
XOR		

The symbols used in HAL/S are the following:

/	+	-
*	.	**
()	,
#	[]
{	}	\$
@	;	:
	&	
¬	^	~
<	=	>
<=	>=	%

Comments

There are no single-line comments in the grammar.
Multiple-line comments are enclosed with */** and **/*.

The syntactic structure of HAL/S

Non-terminals are enclosed between \langle and \rangle . The symbols $::=$ (production), $|$ (union) and ϵ (empty rule) belong to the BNF notation. All other symbols are terminals.

$$\begin{aligned}
 \langle ARITH-EXP \rangle & ::= \langle TERM \rangle \\
 & \quad | \langle PLUS \rangle \langle TERM \rangle \\
 & \quad | \langle MINUS \rangle \langle TERM \rangle \\
 & \quad | \langle ARITH-EXP \rangle \langle PLUS \rangle \langle TERM \rangle \\
 & \quad | \langle ARITH-EXP \rangle \langle MINUS \rangle \langle TERM \rangle \\
 \langle TERM \rangle & ::= \langle PRODUCT \rangle \\
 & \quad | \langle PRODUCT \rangle / \langle TERM \rangle \\
 \langle PLUS \rangle & ::= + \\
 \langle MINUS \rangle & ::= - \\
 \langle PRODUCT \rangle & ::= \langle FACTOR \rangle \\
 & \quad | \langle FACTOR \rangle * \langle PRODUCT \rangle \\
 & \quad | \langle FACTOR \rangle . \langle PRODUCT \rangle \\
 & \quad | \langle FACTOR \rangle \langle PRODUCT \rangle \\
 \langle FACTOR \rangle & ::= \langle PRIMARY \rangle \\
 & \quad | \langle PRIMARY \rangle \langle EXPONENTIATION \rangle \langle FACTOR \rangle \\
 \langle EXPONENTIATION \rangle & ::= ** \\
 \langle PRIMARY \rangle & ::= \langle ARITH-VAR \rangle \\
 & \quad | \langle PRE-PRIMARY \rangle \\
 & \quad | \langle MODIFIED-ARITH-FUNC \rangle \\
 & \quad | \langle PRE-PRIMARY \rangle \langle QUALIFIER \rangle
 \end{aligned}$$

$\langle \text{ARITH-VAR} \rangle ::= \langle \text{ARITH-ID} \rangle$
 $\quad \quad \quad | \quad \langle \text{ARITH-ID} \rangle \langle \text{SUBSCRIPT} \rangle$
 $\quad \quad \quad | \quad [\langle \text{ARITH-ID} \rangle]$
 $\quad \quad \quad | \quad [\langle \text{ARITH-ID} \rangle] \langle \text{SUBSCRIPT} \rangle$
 $\quad \quad \quad | \quad \{ \langle \text{QUAL-STRUCT} \rangle \}$
 $\quad \quad \quad | \quad \{ \langle \text{QUAL-STRUCT} \rangle \} \langle \text{SUBSCRIPT} \rangle$
 $\quad \quad \quad | \quad \langle \text{QUAL-STRUCT} \rangle . \langle \text{ARITH-ID} \rangle$
 $\quad \quad \quad | \quad \langle \text{QUAL-STRUCT} \rangle . \langle \text{ARITH-ID} \rangle \langle \text{SUBSCRIPT} \rangle$

$\langle \text{PRE-PRIMARY} \rangle ::= (\langle \text{ARITH-EXP} \rangle)$
 $\quad \quad \quad | \quad \langle \text{NUMBER} \rangle$
 $\quad \quad \quad | \quad \langle \text{COMPOUND-NUMBER} \rangle$
 $\quad \quad \quad | \quad \langle \text{ARITH-FUNC-HEAD} \rangle (\langle \text{CALL-LIST} \rangle)$
 $\quad \quad \quad | \quad \langle \text{LabelToken} \rangle (\langle \text{CALL-LIST} \rangle)$

$\langle \text{NUMBER} \rangle ::= \langle \text{SIMPLE-NUMBER} \rangle$
 $\quad \quad \quad | \quad \langle \text{LEVEL} \rangle$

$\langle \text{LEVEL} \rangle ::= \langle \text{LevelToken} \rangle$

$\langle \text{COMPOUND-NUMBER} \rangle ::= \langle \text{CompoundToken} \rangle$

$\langle \text{SIMPLE-NUMBER} \rangle ::= \langle \text{NumberToken} \rangle$

$\langle \text{MODIFIED-ARITH-FUNC} \rangle ::= \langle \text{NO-ARG-ARITH-FUNC} \rangle$
 $\quad \quad \quad | \quad \langle \text{NO-ARG-ARITH-FUNC} \rangle \langle \text{SUBSCRIPT} \rangle$
 $\quad \quad \quad | \quad \langle \text{QUAL-STRUCT} \rangle . \langle \text{NO-ARG-ARITH-FUNC} \rangle$
 $\quad \quad \quad | \quad \langle \text{QUAL-STRUCT} \rangle . \langle \text{NO-ARG-ARITH-FUNC} \rangle \langle \text{SUBSCRIPT} \rangle$

$\langle \text{ARITH-FUNC-HEAD} \rangle ::= \langle \text{ARITH-FUNC} \rangle$
 $\quad \quad \quad | \quad \langle \text{ARITH-CONV} \rangle \langle \text{SUBSCRIPT} \rangle$

$\langle \text{CALL-LIST} \rangle ::= \langle \text{LIST-EXP} \rangle$
 $\quad \quad \quad | \quad \langle \text{CALL-LIST} \rangle , \langle \text{LIST-EXP} \rangle$

$\langle \text{LIST-EXP} \rangle ::= \langle \text{EXPRESSION} \rangle$
 $\quad \quad \quad | \quad \langle \text{ARITH-EXP} \rangle \# \langle \text{EXPRESSION} \rangle$
 $\quad \quad \quad | \quad \langle \text{QUAL-STRUCT} \rangle$

$\langle \text{EXPRESSION} \rangle ::= \langle \text{ARITH-EXP} \rangle$
 $\quad \quad \quad | \quad \langle \text{BIT-EXP} \rangle$
 $\quad \quad \quad | \quad \langle \text{CHAR-EXP} \rangle$
 $\quad \quad \quad | \quad \langle \text{NAME-EXP} \rangle$
 $\quad \quad \quad | \quad \langle \text{STRUCTURE-EXP} \rangle$

$\langle \text{ARITH-ID} \rangle ::= \langle \text{IDENTIFIER} \rangle$
 $\quad \quad \quad | \quad \langle \text{ArithFieldToken} \rangle$

$\langle \text{IDENTIFIER} \rangle ::= \langle \text{IdentifierToken} \rangle$

$\langle \text{NO-ARG-ARITH-FUNC} \rangle ::= \text{CLOCKTIME}$
 $\quad \quad \quad | \quad \text{DATE}$
 $\quad \quad \quad | \quad \text{ERRGRP}$
 $\quad \quad \quad | \quad \text{PRIO}$
 $\quad \quad \quad | \quad \text{RANDOM}$
 $\quad \quad \quad | \quad \text{RUNTIME}$

$\langle ARITH-FUNC \rangle$::=	NEXTTIME
		ABS
		CEILING
		DIV
		FLOOR
		MIDVAL
		MOD
		ODD
		REMAINDER
		ROUND
		SIGN
		SIGNUM
		TRUNCATE
		ARCCOS
		ARCCOSH
		ARCSIN
		ARCSINH
		ARCTAN2
		ARCTAN
		ARCTANH
		COS
		COSH
		EXP
		LOG
		SIN
		SINH
		SQRT
		TAN
		TANH
		SHL
		SHR
		ABVAL
		DET
		TRACE
		UNIT
		MATRIX
		INDEX
		LENGTH
		INVERSE
		TRANSPOSE
		PROD
		SUM
		SIZE
		MAX
		MIN
		INTEGER
		SCALAR
$\langle SUBSCRIPT \rangle$::=	$\langle SUB-HEAD \rangle$)
		$\langle QUALIFIER \rangle$
		\$ $\langle NUMBER \rangle$
		\$ $\langle ARITH-VAR \rangle$

$$\begin{aligned}
\langle \text{QUALIFIER} \rangle & ::= \$ (@ \langle \text{PREC-SPEC} \rangle) \\
& \quad | \$ (\langle \text{SCALE-HEAD} \rangle \langle \text{ARITH-EXP} \rangle) \\
& \quad | \$ (@ \langle \text{PREC-SPEC} \rangle , \langle \text{SCALE-HEAD} \rangle \langle \text{ARITH-EXP} \rangle) \\
& \quad | \$ (@ \langle \text{RADIX} \rangle) \\
\langle \text{SCALE-HEAD} \rangle & ::= @ \\
& \quad | @@ \\
\langle \text{PREC-SPEC} \rangle & ::= \text{SINGLE} \\
& \quad | \text{DOUBLE} \\
\langle \text{SUB-START} \rangle & ::= \$ (\\
& \quad | \$ (@ \langle \text{PREC-SPEC} \rangle , \\
& \quad | \langle \text{SUB-HEAD} \rangle ; \\
& \quad | \langle \text{SUB-HEAD} \rangle : \\
& \quad | \langle \text{SUB-HEAD} \rangle , \\
\langle \text{SUB-HEAD} \rangle & ::= \langle \text{SUB-START} \rangle \\
& \quad | \langle \text{SUB-START} \rangle \langle \text{SUB} \rangle \\
\langle \text{SUB} \rangle & ::= \langle \text{SUB-EXP} \rangle \\
& \quad | * \\
& \quad | \langle \text{SUB-RUN-HEAD} \rangle \langle \text{SUB-EXP} \rangle \\
& \quad | \langle \text{ARITH-EXP} \rangle \text{ AT } \langle \text{SUB-EXP} \rangle \\
\langle \text{SUB-RUN-HEAD} \rangle & ::= \langle \text{SUB-EXP} \rangle \text{ TO } \\
\langle \text{SUB-EXP} \rangle & ::= \langle \text{ARITH-EXP} \rangle \\
& \quad | \langle \text{POUND-EXPRESSION} \rangle \\
\langle \text{POUND-EXPRESSION} \rangle & ::= \# \\
& \quad | \langle \text{POUND-EXPRESSION} \rangle \langle \text{PLUS} \rangle \langle \text{TERM} \rangle \\
& \quad | \langle \text{POUND-EXPRESSION} \rangle \langle \text{MINUS} \rangle \langle \text{TERM} \rangle \\
\langle \text{ARITH-CONV} \rangle & ::= \text{INTEGER} \\
& \quad | \text{SCALAR} \\
& \quad | \text{VECTOR} \\
& \quad | \text{MATRIX} \\
\langle \text{BIT-EXP} \rangle & ::= \langle \text{BIT-FACTOR} \rangle \\
& \quad | \langle \text{BIT-EXP} \rangle \langle \text{OR} \rangle \langle \text{BIT-FACTOR} \rangle \\
\langle \text{BIT-FACTOR} \rangle & ::= \langle \text{BIT-CAT} \rangle \\
& \quad | \langle \text{BIT-FACTOR} \rangle \langle \text{AND} \rangle \langle \text{BIT-CAT} \rangle \\
\langle \text{BIT-CAT} \rangle & ::= \langle \text{BIT-PRIM} \rangle \\
& \quad | \langle \text{BIT-CAT} \rangle \langle \text{CAT} \rangle \langle \text{BIT-PRIM} \rangle \\
& \quad | \langle \text{NOT} \rangle \langle \text{BIT-PRIM} \rangle \\
& \quad | \langle \text{BIT-CAT} \rangle \langle \text{CAT} \rangle \langle \text{NOT} \rangle \langle \text{BIT-PRIM} \rangle \\
\langle \text{OR} \rangle & ::= \langle \text{CHAR-VERTICAL-BAR} \rangle \\
& \quad | \text{OR} \\
\langle \text{CHAR-VERTICAL-BAR} \rangle & ::= | \\
\langle \text{AND} \rangle & ::= \& \\
& \quad | \text{AND}
\end{aligned}$$

$\langle \text{BIT-PRIM} \rangle ::= \langle \text{BIT-VAR} \rangle$
 $\quad \quad \quad | \langle \text{LABEL-VAR} \rangle$
 $\quad \quad \quad | \langle \text{EVENT-VAR} \rangle$
 $\quad \quad \quad | \langle \text{BIT-CONST} \rangle$
 $\quad \quad \quad | (\langle \text{BIT-EXP} \rangle)$
 $\quad \quad \quad | \langle \text{SUBBIT-HEAD} \rangle \langle \text{EXPRESSION} \rangle$
 $\quad \quad \quad | \langle \text{BIT-FUNC-HEAD} \rangle (\langle \text{CALL-LIST} \rangle)$
 $\quad \quad \quad | [\langle \text{BIT-VAR} \rangle]$
 $\quad \quad \quad | [\langle \text{BIT-VAR} \rangle] \langle \text{SUBSCRIPT} \rangle$
 $\quad \quad \quad | \{ \langle \text{BIT-VAR} \rangle \}$
 $\quad \quad \quad | \{ \langle \text{BIT-VAR} \rangle \} \langle \text{SUBSCRIPT} \rangle$
 $\langle \text{CAT} \rangle ::= ||$
 $\quad \quad \quad | \text{CAT}$
 $\langle \text{NOT} \rangle ::= \neg$
 $\quad \quad \quad | \text{NOT}$
 $\quad \quad \quad | \wedge$
 $\quad \quad \quad | \sim$
 $\langle \text{BIT-VAR} \rangle ::= \langle \text{BIT-ID} \rangle$
 $\quad \quad \quad | \langle \text{BIT-ID} \rangle \langle \text{SUBSCRIPT} \rangle$
 $\quad \quad \quad | \langle \text{QUAL-STRUCT} \rangle . \langle \text{BIT-ID} \rangle$
 $\quad \quad \quad | \langle \text{QUAL-STRUCT} \rangle . \langle \text{BIT-ID} \rangle \langle \text{SUBSCRIPT} \rangle$
 $\langle \text{LABEL-VAR} \rangle ::= \langle \text{LABEL} \rangle$
 $\quad \quad \quad | \langle \text{LABEL} \rangle \langle \text{SUBSCRIPT} \rangle$
 $\quad \quad \quad | \langle \text{QUAL-STRUCT} \rangle . \langle \text{LABEL} \rangle$
 $\quad \quad \quad | \langle \text{QUAL-STRUCT} \rangle . \langle \text{LABEL} \rangle \langle \text{SUBSCRIPT} \rangle$
 $\langle \text{EVENT-VAR} \rangle ::= \langle \text{EVENT} \rangle$
 $\quad \quad \quad | \langle \text{EVENT} \rangle \langle \text{SUBSCRIPT} \rangle$
 $\quad \quad \quad | \langle \text{QUAL-STRUCT} \rangle . \langle \text{EVENT} \rangle$
 $\quad \quad \quad | \langle \text{QUAL-STRUCT} \rangle . \langle \text{EVENT} \rangle \langle \text{SUBSCRIPT} \rangle$
 $\langle \text{BIT-CONST-HEAD} \rangle ::= \langle \text{RADIX} \rangle$
 $\quad \quad \quad | \langle \text{RADIX} \rangle (\langle \text{NUMBER} \rangle)$
 $\langle \text{BIT-CONST} \rangle ::= \langle \text{BIT-CONST-HEAD} \rangle \langle \text{CHAR-STRING} \rangle$
 $\quad \quad \quad | \text{TRUE}$
 $\quad \quad \quad | \text{FALSE}$
 $\quad \quad \quad | \text{ON}$
 $\quad \quad \quad | \text{OFF}$
 $\langle \text{RADIX} \rangle ::= \text{HEX}$
 $\quad \quad \quad | \text{OCT}$
 $\quad \quad \quad | \text{BIN}$
 $\quad \quad \quad | \text{DEC}$
 $\langle \text{CHAR-STRING} \rangle ::= \langle \text{StringToken} \rangle$
 $\langle \text{SUBBIT-HEAD} \rangle ::= \langle \text{SUBBIT-KEY} \rangle ($
 $\quad \quad \quad | \langle \text{SUBBIT-KEY} \rangle \langle \text{SUBSCRIPT} \rangle ($
 $\langle \text{SUBBIT-KEY} \rangle ::= \text{SUBBIT}$
 $\langle \text{BIT-FUNC-HEAD} \rangle ::= \langle \text{BIT-FUNC} \rangle$
 $\quad \quad \quad | \text{BIT}$
 $\quad \quad \quad | \text{BIT} \langle \text{SUB-OR-QUALIFIER} \rangle$

$\langle \text{BIT-ID} \rangle ::= \langle \text{BitIdentifierToken} \rangle$

$\langle \text{LABEL} \rangle ::= \langle \text{LabelToken} \rangle$
 $\quad \quad \quad | \quad \langle \text{BitFunctionIdentifierToken} \rangle$
 $\quad \quad \quad | \quad \langle \text{CharFunctionIdentifierToken} \rangle$
 $\quad \quad \quad | \quad \langle \text{StructFunctionIdentifierToken} \rangle$

$\langle \text{BIT-FUNC} \rangle ::= \text{XOR}$
 $\quad \quad \quad | \quad \langle \text{BitFunctionIdentifierToken} \rangle$

$\langle \text{EVENT} \rangle ::= \langle \text{EventToken} \rangle$

$\langle \text{SUB-OR-QUALIFIER} \rangle ::= \langle \text{SUBSCRIPT} \rangle$
 $\quad \quad \quad | \quad \langle \text{BIT-QUALIFIER} \rangle$

$\langle \text{BIT-QUALIFIER} \rangle ::= < \$ (@ \langle \text{RADIX} \rangle)$

$\langle \text{CHAR-EXP} \rangle ::= \langle \text{CHAR-PRIM} \rangle$
 $\quad \quad \quad | \quad \langle \text{CHAR-EXP} \rangle \langle \text{CAT} \rangle \langle \text{CHAR-PRIM} \rangle$
 $\quad \quad \quad | \quad \langle \text{CHAR-EXP} \rangle \langle \text{CAT} \rangle \langle \text{ARITH-EXP} \rangle$
 $\quad \quad \quad | \quad \langle \text{ARITH-EXP} \rangle \langle \text{CAT} \rangle \langle \text{ARITH-EXP} \rangle$
 $\quad \quad \quad | \quad \langle \text{ARITH-EXP} \rangle \langle \text{CAT} \rangle \langle \text{CHAR-PRIM} \rangle$

$\langle \text{CHAR-PRIM} \rangle ::= \langle \text{CHAR-VAR} \rangle$
 $\quad \quad \quad | \quad \langle \text{CHAR-CONST} \rangle$
 $\quad \quad \quad | \quad \langle \text{CHAR-FUNC-HEAD} \rangle (\langle \text{CALL-LIST} \rangle)$
 $\quad \quad \quad | \quad (\langle \text{CHAR-EXP} \rangle)$

$\langle \text{CHAR-FUNC-HEAD} \rangle ::= \langle \text{CHAR-FUNC} \rangle$
 $\quad \quad \quad | \quad \text{CHARACTER} \langle \text{SUB-OR-QUALIFIER} \rangle$

$\langle \text{CHAR-VAR} \rangle ::= \langle \text{CHAR-ID} \rangle$
 $\quad \quad \quad | \quad \langle \text{CHAR-ID} \rangle \langle \text{SUBSCRIPT} \rangle$
 $\quad \quad \quad | \quad \langle \text{QUAL-STRUCT} \rangle . \langle \text{CHAR-ID} \rangle$
 $\quad \quad \quad | \quad \langle \text{QUAL-STRUCT} \rangle . \langle \text{CHAR-ID} \rangle \langle \text{SUBSCRIPT} \rangle$

$\langle \text{CHAR-CONST} \rangle ::= \langle \text{CHAR-STRING} \rangle$
 $\quad \quad \quad | \quad \text{CHAR} (\langle \text{NUMBER} \rangle) \langle \text{CHAR-STRING} \rangle$

$\langle \text{CHAR-FUNC} \rangle ::= \text{LJUST}$
 $\quad \quad \quad | \quad \text{RJUST}$
 $\quad \quad \quad | \quad \text{TRIM}$
 $\quad \quad \quad | \quad \langle \text{CharFunctionIdentifierToken} \rangle$
 $\quad \quad \quad | \quad \text{CHARACTER}$

$\langle \text{CHAR-ID} \rangle ::= \langle \text{CharIdentifierToken} \rangle$

$\langle \text{NAME-EXP} \rangle ::= \langle \text{NAME-KEY} \rangle (\langle \text{NAME-VAR} \rangle)$
 $\quad \quad \quad | \quad \text{NULL}$
 $\quad \quad \quad | \quad \langle \text{NAME-KEY} \rangle (\text{NULL})$

$\langle \text{NAME-KEY} \rangle ::= \text{NAME}$

$\langle \text{NAME-VAR} \rangle ::= \langle \text{VARIABLE} \rangle$
 $\quad \quad \quad | \quad \langle \text{MODIFIED-ARITH-FUNC} \rangle$
 $\quad \quad \quad | \quad \langle \text{LABEL-VAR} \rangle$

$\langle \text{VARIABLE} \rangle ::= \langle \text{ARITH-VAR} \rangle$
 $\quad \quad \quad | \langle \text{BIT-VAR} \rangle$
 $\quad \quad \quad | \langle \text{SUBBIT-HEAD} \rangle \langle \text{VARIABLE} \rangle$
 $\quad \quad \quad | \langle \text{CHAR-VAR} \rangle$
 $\quad \quad \quad | \langle \text{NAME-KEY} \rangle (\langle \text{NAME-VAR} \rangle)$
 $\quad \quad \quad | \langle \text{EVENT-VAR} \rangle$
 $\quad \quad \quad | \langle \text{STRUCTURE-VAR} \rangle$
 $\langle \text{STRUCTURE-EXP} \rangle ::= \langle \text{STRUCTURE-VAR} \rangle$
 $\quad \quad \quad | \langle \text{STRUCT-FUNC-HEAD} \rangle (\langle \text{CALL-LIST} \rangle)$
 $\quad \quad \quad | \langle \text{STRUC-INLINE-DEF} \rangle \langle \text{CLOSING} \rangle ;$
 $\quad \quad \quad | \langle \text{STRUC-INLINE-DEF} \rangle \langle \text{BLOCK-BODY} \rangle \langle \text{CLOSING} \rangle ;$
 $\langle \text{STRUCT-FUNC-HEAD} \rangle ::= \langle \text{STRUCT-FUNC} \rangle$
 $\langle \text{STRUCTURE-VAR} \rangle ::= \langle \text{QUAL-STRUCT} \rangle \langle \text{SUBSCRIPT} \rangle$
 $\langle \text{STRUCT-FUNC} \rangle ::= \langle \text{StructFunctionIdentifierToken} \rangle$
 $\langle \text{QUAL-STRUCT} \rangle ::= \langle \text{STRUCTURE-ID} \rangle$
 $\quad \quad \quad | \langle \text{QUAL-STRUCT} \rangle . \langle \text{STRUCTURE-ID} \rangle$
 $\langle \text{STRUCTURE-ID} \rangle ::= \langle \text{StructIdentifierToken} \rangle$
 $\langle \text{ASSIGNMENT} \rangle ::= \langle \text{VARIABLE} \rangle \langle \text{EQUALS} \rangle \langle \text{EXPRESSION} \rangle$
 $\quad \quad \quad | \langle \text{VARIABLE} \rangle , \langle \text{ASSIGNMENT} \rangle$
 $\quad \quad \quad | \langle \text{QUAL-STRUCT} \rangle \langle \text{EQUALS} \rangle \langle \text{EXPRESSION} \rangle$
 $\quad \quad \quad | \langle \text{QUAL-STRUCT} \rangle , \langle \text{ASSIGNMENT} \rangle$
 $\langle \text{EQUALS} \rangle ::= =$
 $\langle \text{IF-STATEMENT} \rangle ::= \langle \text{IF-CLAUSE} \rangle \langle \text{STATEMENT} \rangle$
 $\quad \quad \quad | \langle \text{TRUE-PART} \rangle \langle \text{STATEMENT} \rangle$
 $\langle \text{IF-CLAUSE} \rangle ::= \langle \text{IF} \rangle \langle \text{RELATIONAL-EXP} \rangle \text{ THEN}$
 $\quad \quad \quad | \langle \text{IF} \rangle \langle \text{BIT-EXP} \rangle \text{ THEN}$
 $\langle \text{TRUE-PART} \rangle ::= \langle \text{IF-CLAUSE} \rangle \langle \text{BASIC-STATEMENT} \rangle \text{ ELSE}$
 $\langle \text{IF} \rangle ::= \text{IF}$
 $\langle \text{RELATIONAL-EXP} \rangle ::= \langle \text{RELATIONAL-FACTOR} \rangle$
 $\quad \quad \quad | \langle \text{RELATIONAL-EXP} \rangle \langle \text{OR} \rangle \langle \text{RELATIONAL-FACTOR} \rangle$
 $\langle \text{RELATIONAL-FACTOR} \rangle ::= \langle \text{REL-PRIM} \rangle$
 $\quad \quad \quad | \langle \text{RELATIONAL-FACTOR} \rangle \langle \text{AND} \rangle \langle \text{REL-PRIM} \rangle$
 $\langle \text{REL-PRIM} \rangle ::= (\langle \text{RELATIONAL-EXP} \rangle)$
 $\quad \quad \quad | \langle \text{NOT} \rangle (\langle \text{RELATIONAL-EXP} \rangle)$
 $\quad \quad \quad | \langle \text{COMPARISON} \rangle$
 $\langle \text{COMPARISON} \rangle ::= \langle \text{ARITH-EXP} \rangle \langle \text{RELATIONAL-OP} \rangle \langle \text{ARITH-EXP} \rangle$
 $\quad \quad \quad | \langle \text{CHAR-EXP} \rangle \langle \text{RELATIONAL-OP} \rangle \langle \text{CHAR-EXP} \rangle$
 $\quad \quad \quad | \langle \text{BIT-CAT} \rangle \langle \text{RELATIONAL-OP} \rangle \langle \text{BIT-CAT} \rangle$
 $\quad \quad \quad | \langle \text{STRUCTURE-EXP} \rangle \langle \text{RELATIONAL-OP} \rangle \langle \text{STRUCTURE-EXP} \rangle$
 $\quad \quad \quad | \langle \text{NAME-EXP} \rangle \langle \text{RELATIONAL-OP} \rangle \langle \text{NAME-EXP} \rangle$

$\langle \text{RELATIONAL-OP} \rangle ::= \langle \text{EQUALS} \rangle$
 $\quad \quad \quad \langle \text{NOT} \rangle \langle \text{EQUALS} \rangle$
 $\quad \quad \quad <$
 $\quad \quad \quad >$
 $\quad \quad \quad \leq$
 $\quad \quad \quad \geq$
 $\quad \quad \quad \langle \text{NOT} \rangle <$
 $\quad \quad \quad \langle \text{NOT} \rangle >$

$\langle \text{STATEMENT} \rangle ::= \langle \text{BASIC-STATEMENT} \rangle$
 $\quad \quad \quad \langle \text{OTHER-STATEMENT} \rangle$
 $\quad \quad \quad \langle \text{INLINE-DEFINITION} \rangle$

$\langle \text{BASIC-STATEMENT} \rangle ::= \langle \text{ASSIGNMENT} \rangle ;$
 $\quad \quad \quad \langle \text{LABEL-DEFINITION} \rangle \langle \text{BASIC-STATEMENT} \rangle$
 $\quad \quad \quad \text{EXIT} ;$
 $\quad \quad \quad \text{EXIT} \langle \text{LABEL} \rangle ;$
 $\quad \quad \quad \text{REPEAT} ;$
 $\quad \quad \quad \text{REPEAT} \langle \text{LABEL} \rangle ;$
 $\quad \quad \quad \text{GO TO} \langle \text{LABEL} \rangle ;$
 $\quad \quad \quad ;$
 $\quad \quad \quad \langle \text{CALL-KEY} \rangle ;$
 $\quad \quad \quad \langle \text{CALL-KEY} \rangle (\langle \text{CALL-LIST} \rangle) ;$
 $\quad \quad \quad \langle \text{CALL-KEY} \rangle \langle \text{ASSIGN} \rangle (\langle \text{CALL-ASSIGN-LIST} \rangle) ;$
 $\quad \quad \quad \langle \text{CALL-KEY} \rangle (\langle \text{CALL-LIST} \rangle) \langle \text{ASSIGN} \rangle (\langle \text{CALL-ASSIGN-LIST} \rangle) ;$
 $\quad \quad \quad \text{RETURN} ;$
 $\quad \quad \quad \text{RETURN} \langle \text{EXPRESSION} \rangle ;$
 $\quad \quad \quad \langle \text{DO-GROUP-HEAD} \rangle \langle \text{ENDING} \rangle ;$
 $\quad \quad \quad \langle \text{READ-KEY} \rangle ;$
 $\quad \quad \quad \langle \text{READ-PHRASE} \rangle ;$
 $\quad \quad \quad \langle \text{WRITE-KEY} \rangle ;$
 $\quad \quad \quad \langle \text{WRITE-PHRASE} \rangle ;$
 $\quad \quad \quad \langle \text{FILE-EXP} \rangle \langle \text{EQUALS} \rangle \langle \text{EXPRESSION} \rangle ;$
 $\quad \quad \quad \langle \text{VARIABLE} \rangle \langle \text{EQUALS} \rangle \langle \text{FILE-EXP} \rangle ;$
 $\quad \quad \quad \langle \text{FILE-EXP} \rangle \langle \text{EQUALS} \rangle \langle \text{QUAL-STRUCT} \rangle ;$
 $\quad \quad \quad \langle \text{WAIT-KEY} \rangle \text{FOR DEPENDENT} ;$
 $\quad \quad \quad \langle \text{WAIT-KEY} \rangle \langle \text{ARITH-EXP} \rangle ;$
 $\quad \quad \quad \langle \text{WAIT-KEY} \rangle \text{UNTIL} \langle \text{ARITH-EXP} \rangle ;$
 $\quad \quad \quad \langle \text{WAIT-KEY} \rangle \text{FOR} \langle \text{BIT-EXP} \rangle ;$
 $\quad \quad \quad \langle \text{TERMINATOR} \rangle ;$
 $\quad \quad \quad \langle \text{TERMINATOR} \rangle \langle \text{TERMINATE-LIST} \rangle ;$
 $\quad \quad \quad \text{UPDATE PRIORITY TO} \langle \text{ARITH-EXP} \rangle ;$
 $\quad \quad \quad \text{UPDATE PRIORITY} \langle \text{LABEL-VAR} \rangle \text{TO} \langle \text{ARITH-EXP} \rangle ;$
 $\quad \quad \quad \langle \text{SCHEDULE-PHRASE} \rangle ;$
 $\quad \quad \quad \langle \text{SCHEDULE-PHRASE} \rangle \langle \text{SCHEDULE-CONTROL} \rangle ;$
 $\quad \quad \quad \langle \text{SIGNAL-CLAUSE} \rangle ;$
 $\quad \quad \quad \text{SEND ERROR} \langle \text{SUBSCRIPT} \rangle ;$
 $\quad \quad \quad \text{SEND ERROR} ;$
 $\quad \quad \quad \langle \text{ON-CLAUSE} \rangle ;$
 $\quad \quad \quad \langle \text{ON-CLAUSE} \rangle \text{AND} \langle \text{SIGNAL-CLAUSE} \rangle ;$
 $\quad \quad \quad \text{OFF ERROR} \langle \text{SUBSCRIPT} \rangle ;$
 $\quad \quad \quad \text{OFF ERROR} ;$
 $\quad \quad \quad \langle \text{PERCENT-MACRO-NAME} \rangle ;$
 $\quad \quad \quad \langle \text{PERCENT-MACRO-HEAD} \rangle \langle \text{PERCENT-MACRO-ARG} \rangle) ;$

$\langle \text{OTHER-STATEMENT} \rangle ::= \langle \text{IF-STATEMENT} \rangle$
 $\quad \quad \quad | \quad \langle \text{ON-PHRASE} \rangle \langle \text{STATEMENT} \rangle$
 $\quad \quad \quad | \quad \langle \text{LABEL-DEFINITION} \rangle \langle \text{OTHER-STATEMENT} \rangle$

$\langle \text{ANY-STATEMENT} \rangle ::= \langle \text{STATEMENT} \rangle$
 $\quad \quad \quad | \quad \langle \text{BLOCK-DEFINITION} \rangle$

$\langle \text{ON-PHRASE} \rangle ::= \text{ON ERROR } \langle \text{SUBSCRIPT} \rangle$
 $\quad \quad \quad | \quad \text{ON ERROR}$

$\langle \text{ON-CLAUSE} \rangle ::= \text{ON ERROR } \langle \text{SUBSCRIPT} \rangle \text{ SYSTEM}$
 $\quad \quad \quad | \quad \text{ON ERROR } \langle \text{SUBSCRIPT} \rangle \text{ IGNORE}$
 $\quad \quad \quad | \quad \text{ON ERROR SYSTEM}$
 $\quad \quad \quad | \quad \text{ON ERROR IGNORE}$

$\langle \text{LABEL-DEFINITION} \rangle ::= \langle \text{LABEL} \rangle :$

$\langle \text{CALL-KEY} \rangle ::= \text{CALL } \langle \text{LABEL-VAR} \rangle$

$\langle \text{ASSIGN} \rangle ::= \text{ASSIGN}$

$\langle \text{CALL-ASSIGN-LIST} \rangle ::= \langle \text{VARIABLE} \rangle$
 $\quad \quad \quad | \quad \langle \text{CALL-ASSIGN-LIST} \rangle , \langle \text{VARIABLE} \rangle$
 $\quad \quad \quad | \quad \langle \text{QUAL-STRUCT} \rangle$
 $\quad \quad \quad | \quad \langle \text{CALL-ASSIGN-LIST} \rangle , \langle \text{QUAL-STRUCT} \rangle$

$\langle \text{DO-GROUP-HEAD} \rangle ::= \text{DO ;}$
 $\quad \quad \quad | \quad \text{DO } \langle \text{FOR-LIST} \rangle ;$
 $\quad \quad \quad | \quad \text{DO } \langle \text{FOR-LIST} \rangle \langle \text{WHILE-CLAUSE} \rangle ;$
 $\quad \quad \quad | \quad \text{DO } \langle \text{WHILE-CLAUSE} \rangle ;$
 $\quad \quad \quad | \quad \text{DO CASE } \langle \text{ARITH-EXP} \rangle ;$
 $\quad \quad \quad | \quad \langle \text{CASE-ELSE} \rangle \langle \text{STATEMENT} \rangle$
 $\quad \quad \quad | \quad \langle \text{DO-GROUP-HEAD} \rangle \langle \text{ANY-STATEMENT} \rangle$
 $\quad \quad \quad | \quad \langle \text{DO-GROUP-HEAD} \rangle \langle \text{TEMPORARY-STMT} \rangle$

$\langle \text{ENDING} \rangle ::= \text{END}$
 $\quad \quad \quad | \quad \text{END } \langle \text{LABEL} \rangle$
 $\quad \quad \quad | \quad \langle \text{LABEL-DEFINITION} \rangle \langle \text{ENDING} \rangle$

$\langle \text{READ-KEY} \rangle ::= \text{READ (} \langle \text{NUMBER} \rangle \text{)}$
 $\quad \quad \quad | \quad \text{READALL (} \langle \text{NUMBER} \rangle \text{)}$

$\langle \text{WRITE-KEY} \rangle ::= \text{WRITE (} \langle \text{NUMBER} \rangle \text{)}$

$\langle \text{READ-PHRASE} \rangle ::= \langle \text{READ-KEY} \rangle \langle \text{READ-ARG} \rangle$
 $\quad \quad \quad | \quad \langle \text{READ-PHRASE} \rangle , \langle \text{READ-ARG} \rangle$

$\langle \text{WRITE-PHRASE} \rangle ::= \langle \text{WRITE-KEY} \rangle \langle \text{WRITE-ARG} \rangle$
 $\quad \quad \quad | \quad \langle \text{WRITE-PHRASE} \rangle , \langle \text{WRITE-ARG} \rangle$

$\langle \text{READ-ARG} \rangle ::= \langle \text{VARIABLE} \rangle$
 $\quad \quad \quad | \quad \langle \text{IO-CONTROL} \rangle$

$\langle \text{WRITE-ARG} \rangle ::= \langle \text{EXPRESSION} \rangle$
 $\quad \quad \quad | \quad \langle \text{IO-CONTROL} \rangle$
 $\quad \quad \quad | \quad \langle \text{StructIdentifierToken} \rangle$

$\langle \text{FILE-EXP} \rangle ::= \langle \text{FILE-HEAD} \rangle , \langle \text{ARITH-EXP} \rangle$

$\langle \text{FILE-HEAD} \rangle ::= \text{FILE (} \langle \text{NUMBER} \rangle$
 $\langle \text{IO-CONTROL} \rangle ::= \text{SKIP (} \langle \text{ARITH-EXP} \rangle)$
 $\quad \quad \quad | \quad \text{TAB (} \langle \text{ARITH-EXP} \rangle)$
 $\quad \quad \quad | \quad \text{COLUMN (} \langle \text{ARITH-EXP} \rangle)$
 $\quad \quad \quad | \quad \text{LINE (} \langle \text{ARITH-EXP} \rangle)$
 $\quad \quad \quad | \quad \text{PAGE (} \langle \text{ARITH-EXP} \rangle)$
 $\langle \text{WAIT-KEY} \rangle ::= \text{WAIT}$
 $\langle \text{TERMINATOR} \rangle ::= \text{TERMINATE}$
 $\quad \quad \quad | \quad \text{CANCEL}$
 $\langle \text{TERMINATE-LIST} \rangle ::= \langle \text{LABEL-VAR} \rangle$
 $\quad \quad \quad | \quad \langle \text{TERMINATE-LIST} \rangle , \langle \text{LABEL-VAR} \rangle$
 $\langle \text{SCHEDULE-HEAD} \rangle ::= \text{SCHEDULE } \langle \text{LABEL-VAR} \rangle$
 $\quad \quad \quad | \quad \langle \text{SCHEDULE-HEAD} \rangle \text{ AT } \langle \text{ARITH-EXP} \rangle$
 $\quad \quad \quad | \quad \langle \text{SCHEDULE-HEAD} \rangle \text{ IN } \langle \text{ARITH-EXP} \rangle$
 $\quad \quad \quad | \quad \langle \text{SCHEDULE-HEAD} \rangle \text{ ON } \langle \text{BIT-EXP} \rangle$
 $\langle \text{SCHEDULE-PHRASE} \rangle ::= \langle \text{SCHEDULE-HEAD} \rangle$
 $\quad \quad \quad | \quad \langle \text{SCHEDULE-HEAD} \rangle \text{ PRIORITY (} \langle \text{ARITH-EXP} \rangle)$
 $\quad \quad \quad | \quad \langle \text{SCHEDULE-PHRASE} \rangle \text{ DEPENDENT}$
 $\langle \text{SCHEDULE-CONTROL} \rangle ::= \langle \text{STOPPING} \rangle$
 $\quad \quad \quad | \quad \langle \text{TIMING} \rangle$
 $\quad \quad \quad | \quad \langle \text{TIMING} \rangle \langle \text{STOPPING} \rangle$
 $\langle \text{TIMING} \rangle ::= \langle \text{REPEAT} \rangle \text{ EVERY } \langle \text{ARITH-EXP} \rangle$
 $\quad \quad \quad | \quad \langle \text{REPEAT} \rangle \text{ AFTER } \langle \text{ARITH-EXP} \rangle$
 $\quad \quad \quad | \quad \langle \text{REPEAT} \rangle$
 $\langle \text{REPEAT} \rangle ::= , \text{ REPEAT}$
 $\langle \text{STOPPING} \rangle ::= \langle \text{WHILE-KEY} \rangle \langle \text{ARITH-EXP} \rangle$
 $\quad \quad \quad | \quad \langle \text{WHILE-KEY} \rangle \langle \text{BIT-EXP} \rangle$
 $\langle \text{SIGNAL-CLAUSE} \rangle ::= \text{SET } \langle \text{EVENT-VAR} \rangle$
 $\quad \quad \quad | \quad \text{RESET } \langle \text{EVENT-VAR} \rangle$
 $\quad \quad \quad | \quad \text{SIGNAL } \langle \text{EVENT-VAR} \rangle$
 $\langle \text{PERCENT-MACRO-NAME} \rangle ::= \% \langle \text{IDENTIFIER} \rangle$
 $\langle \text{PERCENT-MACRO-HEAD} \rangle ::= \langle \text{PERCENT-MACRO-NAME} \rangle ($
 $\quad \quad \quad | \quad \langle \text{PERCENT-MACRO-HEAD} \rangle \langle \text{PERCENT-MACRO-ARG} \rangle ,$
 $\langle \text{PERCENT-MACRO-ARG} \rangle ::= \langle \text{NAME-VAR} \rangle$
 $\quad \quad \quad | \quad \langle \text{CONSTANT} \rangle$
 $\langle \text{CASE-ELSE} \rangle ::= \text{DO CASE } \langle \text{ARITH-EXP} \rangle ; \text{ ELSE}$
 $\langle \text{WHILE-KEY} \rangle ::= \text{WHILE}$
 $\quad \quad \quad | \quad \text{UNTIL}$
 $\langle \text{WHILE-CLAUSE} \rangle ::= \langle \text{WHILE-KEY} \rangle \langle \text{BIT-EXP} \rangle$
 $\quad \quad \quad | \quad \langle \text{WHILE-KEY} \rangle \langle \text{RELATIONAL-EXP} \rangle$

$\langle \text{FOR-LIST} \rangle ::= \langle \text{FOR-KEY} \rangle \langle \text{ARITH-EXP} \rangle \langle \text{ITERATION-CONTROL} \rangle$
 $\quad \quad \quad | \quad \quad \langle \text{FOR-KEY} \rangle \langle \text{ITERATION-BODY} \rangle$

$\langle \text{ITERATION-BODY} \rangle ::= \langle \text{ARITH-EXP} \rangle$
 $\quad \quad \quad | \quad \quad \langle \text{ITERATION-BODY} \rangle , \langle \text{ARITH-EXP} \rangle$

$\langle \text{ITERATION-CONTROL} \rangle ::= \text{TO } \langle \text{ARITH-EXP} \rangle$
 $\quad \quad \quad | \quad \quad \text{TO } \langle \text{ARITH-EXP} \rangle \text{ BY } \langle \text{ARITH-EXP} \rangle$

$\langle \text{FOR-KEY} \rangle ::= \text{FOR } \langle \text{ARITH-VAR} \rangle \langle \text{EQUALS} \rangle$
 $\quad \quad \quad | \quad \quad \text{FOR TEMPORARY } \langle \text{IDENTIFIER} \rangle =$

$\langle \text{TEMPORARY-STMT} \rangle ::= \text{TEMPORARY } \langle \text{DECLARE-BODY} \rangle ;$

$\langle \text{DECLARE-BODY} \rangle ::= \langle \text{DECLARATION-LIST} \rangle$
 $\quad \quad \quad | \quad \quad \langle \text{ATTRIBUTES} \rangle , \langle \text{DECLARATION-LIST} \rangle$

$\langle \text{DECLARATION-LIST} \rangle ::= \langle \text{DECLARATION} \rangle$
 $\quad \quad \quad | \quad \quad \langle \text{DCL-LIST-COMMA} \rangle \langle \text{DECLARATION} \rangle$

$\langle \text{CONSTANT} \rangle ::= \langle \text{NUMBER} \rangle$
 $\quad \quad \quad | \quad \quad \langle \text{COMPOUND-NUMBER} \rangle$
 $\quad \quad \quad | \quad \quad \langle \text{BIT-CONST} \rangle$
 $\quad \quad \quad | \quad \quad \langle \text{CHAR-CONST} \rangle$

$\langle \text{ATTRIBUTES} \rangle ::= \langle \text{ARRAY-SPEC} \rangle \langle \text{TYPE-AND-MINOR-ATTR} \rangle$
 $\quad \quad \quad | \quad \quad \langle \text{ARRAY-SPEC} \rangle$
 $\quad \quad \quad | \quad \quad \langle \text{TYPE-AND-MINOR-ATTR} \rangle$

$\langle \text{ARRAY-SPEC} \rangle ::= \langle \text{ARRAY-HEAD} \rangle \langle \text{LITERAL-EXP-OR-STAR} \rangle)$
 $\quad \quad \quad | \quad \quad \text{FUNCTION}$
 $\quad \quad \quad | \quad \quad \text{PROCEDURE}$
 $\quad \quad \quad | \quad \quad \text{PROGRAM}$
 $\quad \quad \quad | \quad \quad \text{TASK}$

$\langle \text{ARRAY-HEAD} \rangle ::= \text{ARRAY (}$
 $\quad \quad \quad | \quad \quad \langle \text{ARRAY-HEAD} \rangle \langle \text{LITERAL-EXP-OR-STAR} \rangle ,$

$\langle \text{TYPE-AND-MINOR-ATTR} \rangle ::= \langle \text{TYPE-SPEC} \rangle$
 $\quad \quad \quad | \quad \quad \langle \text{TYPE-SPEC} \rangle \langle \text{MINOR-ATTR-LIST} \rangle$
 $\quad \quad \quad | \quad \quad \langle \text{MINOR-ATTR-LIST} \rangle$

$\langle \text{MINOR-ATTR-LIST} \rangle ::= \langle \text{MINOR-ATTRIBUTE} \rangle$
 $\quad \quad \quad | \quad \quad \langle \text{MINOR-ATTR-LIST} \rangle \langle \text{MINOR-ATTRIBUTE} \rangle$

$\langle \text{MINOR-ATTRIBUTE} \rangle ::= \text{STATIC}$
 $\quad \quad \quad | \quad \quad \text{AUTOMATIC}$
 $\quad \quad \quad | \quad \quad \text{DENSE}$
 $\quad \quad \quad | \quad \quad \text{ALIGNED}$
 $\quad \quad \quad | \quad \quad \text{ACCESS}$
 $\quad \quad \quad | \quad \quad \text{LOCK (} \langle \text{LITERAL-EXP-OR-STAR} \rangle)$
 $\quad \quad \quad | \quad \quad \text{REMOTE}$
 $\quad \quad \quad | \quad \quad \text{RIGID}$
 $\quad \quad \quad | \quad \quad \langle \text{INIT-OR-CONST-HEAD} \rangle \langle \text{REPEATED-CONSTANT} \rangle)$
 $\quad \quad \quad | \quad \quad \langle \text{INIT-OR-CONST-HEAD} \rangle *)$
 $\quad \quad \quad | \quad \quad \text{LATCHED}$
 $\quad \quad \quad | \quad \quad \text{NONHAL (} \langle \text{LEVEL} \rangle)$

$\langle \text{INIT-OR-CONST-HEAD} \rangle ::= \text{INITIAL (}$
 $\quad | \quad \text{CONSTANT (}$
 $\quad | \quad \langle \text{INIT-OR-CONST-HEAD} \rangle \langle \text{REPEATED-CONSTANT} \rangle ,$
 $\langle \text{REPEATED-CONSTANT} \rangle ::= \langle \text{EXPRESSION} \rangle$
 $\quad | \quad \langle \text{REPEAT-HEAD} \rangle \langle \text{VARIABLE} \rangle$
 $\quad | \quad \langle \text{REPEAT-HEAD} \rangle \langle \text{CONSTANT} \rangle$
 $\quad | \quad \langle \text{NESTED-REPEAT-HEAD} \rangle \langle \text{REPEATED-CONSTANT} \rangle)$
 $\quad | \quad \langle \text{REPEAT-HEAD} \rangle$
 $\langle \text{REPEAT-HEAD} \rangle ::= \langle \text{ARITH-EXP} \rangle \# \langle \text{SIMPLE-NUMBER} \rangle$
 $\langle \text{NESTED-REPEAT-HEAD} \rangle ::= \langle \text{REPEAT-HEAD} \rangle ($
 $\quad | \quad \langle \text{NESTED-REPEAT-HEAD} \rangle \langle \text{REPEATED-CONSTANT} \rangle ,$
 $\langle \text{DECLARATION} \rangle ::= \langle \text{NAME-ID} \rangle$
 $\quad | \quad \langle \text{NAME-ID} \rangle \langle \text{ATTRIBUTES} \rangle$
 $\quad | \quad \langle \text{LabelToken} \rangle \text{PROCEDURE } \langle \text{MINOR-ATTR-LIST} \rangle$
 $\quad | \quad \langle \text{LabelToken} \rangle \text{PROCEDURE}$
 $\quad | \quad \langle \text{EventToken} \rangle \text{EVENT}$
 $\quad | \quad \langle \text{EventToken} \rangle \text{EVENT } \langle \text{MINOR-ATTR-LIST} \rangle$
 $\quad | \quad \langle \text{EventToken} \rangle$
 $\quad | \quad \langle \text{EventToken} \rangle \langle \text{MINOR-ATTR-LIST} \rangle$
 $\langle \text{NAME-ID} \rangle ::= \langle \text{IDENTIFIER} \rangle$
 $\quad | \quad \langle \text{IDENTIFIER} \rangle \text{NAME}$
 $\quad | \quad \langle \text{BIT-ID} \rangle$
 $\quad | \quad \langle \text{CHAR-ID} \rangle$
 $\quad | \quad \langle \text{BitFunctionIdentifierToken} \rangle$
 $\quad | \quad \langle \text{CharFunctionIdentifierToken} \rangle$
 $\quad | \quad \langle \text{StructIdentifierToken} \rangle$
 $\quad | \quad \langle \text{StructFunctionIdentifierToken} \rangle$
 $\langle \text{DCL-LIST-COMMA} \rangle ::= \langle \text{DECLARATION-LIST} \rangle ,$
 $\langle \text{LITERAL-EXP-OR-STAR} \rangle ::= \langle \text{ARITH-EXP} \rangle$
 $\quad | \quad *$
 $\langle \text{TYPE-SPEC} \rangle ::= \langle \text{STRUCT-SPEC} \rangle$
 $\quad | \quad \langle \text{BIT-SPEC} \rangle$
 $\quad | \quad \langle \text{CHAR-SPEC} \rangle$
 $\quad | \quad \langle \text{ARITH-SPEC} \rangle$
 $\quad | \quad \text{EVENT}$
 $\langle \text{BIT-SPEC} \rangle ::= \text{BOOLEAN}$
 $\quad | \quad \text{BIT (} \langle \text{LITERAL-EXP-OR-STAR} \rangle \text{)}$
 $\langle \text{CHAR-SPEC} \rangle ::= \text{CHARACTER (} \langle \text{LITERAL-EXP-OR-STAR} \rangle \text{)}$
 $\langle \text{STRUCT-SPEC} \rangle ::= \langle \text{STRUCT-TEMPLATE} \rangle \langle \text{STRUCT-SPEC-BODY} \rangle$
 $\langle \text{STRUCT-SPEC-BODY} \rangle ::= - \text{STRUCTURE}$
 $\quad | \quad \langle \text{STRUCT-SPEC-HEAD} \rangle \langle \text{LITERAL-EXP-OR-STAR} \rangle)$
 $\langle \text{STRUCT-TEMPLATE} \rangle ::= \langle \text{STRUCTURE-ID} \rangle$

```

<STRUCT-SPEC-HEAD> ::= - STRUCTURE (

<ARITH-SPEC> ::= <PREC-SPEC>
                | <SQ-DQ-NAME>
                | <SQ-DQ-NAME> <PREC-SPEC>
<SQ-DQ-NAME> ::= <DOUBLY-QUAL-NAME-HEAD> <LITERAL-EXP-OR-STAR> )
                | <ARITH-CONV>

<DOUBLY-QUAL-NAME-HEAD> ::= VECTOR (
                                | MATRIX ( <LITERAL-EXP-OR-STAR> ,

<COMPILATION> ::= <ANY-STATEMENT>
                | <COMPILATION> <ANY-STATEMENT>
                | <DECLARE-STATEMENT>
                | <COMPILATION> <DECLARE-STATEMENT>
                | <STRUCTURE-STMT>
                | <COMPILATION> <STRUCTURE-STMT>
                | <REPLACE-STMT> ;
                | <COMPILATION> <REPLACE-STMT> ;

<BLOCK-DEFINITION> ::= <BLOCK-STMT> <CLOSING> ;
                    | <BLOCK-STMT> <BLOCK-BODY> <CLOSING> ;

<BLOCK-STMT> ::= <BLOCK-STMT-TOP> ;

<BLOCK-STMT-TOP> ::= <BLOCK-STMT-TOP> ACCESS
                    | <BLOCK-STMT-TOP> RIGID
                    | <BLOCK-STMT-HEAD>
                    | <BLOCK-STMT-HEAD> EXCLUSIVE
                    | <BLOCK-STMT-HEAD> REENTRANT

<BLOCK-STMT-HEAD> ::= <LABEL-EXTERNAL> PROGRAM
                    | <LABEL-EXTERNAL> COMPOOL
                    | <LABEL-DEFINITION> TASK
                    | <LABEL-DEFINITION> UPDATE
                    | UPDATE
                    | <FUNCTION-NAME>
                    | <FUNCTION-NAME> <FUNC-STMT-BODY>
                    | <PROCEDURE-NAME>
                    | <PROCEDURE-NAME> <PROC-STMT-BODY>

<LABEL-EXTERNAL> ::= <LABEL-DEFINITION>
                    | <LABEL-DEFINITION> EXTERNAL

<CLOSING> ::= CLOSE
            | CLOSE <LABEL>
            | <LABEL-DEFINITION> <CLOSING>

<BLOCK-BODY> ::= <DECLARE-GROUP>
                | <ANY-STATEMENT>
                | <BLOCK-BODY> <ANY-STATEMENT>

<FUNCTION-NAME> ::= <LABEL-EXTERNAL> FUNCTION

<PROCEDURE-NAME> ::= <LABEL-EXTERNAL> PROCEDURE

```

$\langle \text{FUNC-STMT-BODY} \rangle ::= \langle \text{PARAMETER-LIST} \rangle$
 $\quad \quad \quad | \quad \langle \text{TYPE-SPEC} \rangle$
 $\quad \quad \quad | \quad \langle \text{PARAMETER-LIST} \rangle \langle \text{TYPE-SPEC} \rangle$
 $\langle \text{PROC-STMT-BODY} \rangle ::= \langle \text{PARAMETER-LIST} \rangle$
 $\quad \quad \quad | \quad \langle \text{ASSIGN-LIST} \rangle$
 $\quad \quad \quad | \quad \langle \text{PARAMETER-LIST} \rangle \langle \text{ASSIGN-LIST} \rangle$
 $\langle \text{DECLARE-GROUP} \rangle ::= \langle \text{DECLARE-ELEMENT} \rangle$
 $\quad \quad \quad | \quad \langle \text{DECLARE-GROUP} \rangle \langle \text{DECLARE-ELEMENT} \rangle$
 $\langle \text{DECLARE-ELEMENT} \rangle ::= \langle \text{DECLARE-STATEMENT} \rangle$
 $\quad \quad \quad | \quad \langle \text{REPLACE-STMT} \rangle ;$
 $\quad \quad \quad | \quad \langle \text{STRUCTURE-STMT} \rangle$
 $\quad \quad \quad | \quad \text{EQUATE EXTERNAL } \langle \text{IDENTIFIER} \rangle \text{ TO } \langle \text{VARIABLE} \rangle ;$
 $\langle \text{PARAMETER} \rangle ::= \langle \text{IdentifierToken} \rangle$
 $\quad \quad \quad | \quad \langle \text{BitIdentifierToken} \rangle$
 $\quad \quad \quad | \quad \langle \text{CharIdentifierToken} \rangle$
 $\quad \quad \quad | \quad \langle \text{StructIdentifierToken} \rangle$
 $\quad \quad \quad | \quad \langle \text{EventToken} \rangle$
 $\quad \quad \quad | \quad \langle \text{LabelToken} \rangle$
 $\langle \text{PARAMETER-LIST} \rangle ::= \langle \text{PARAMETER-HEAD} \rangle \langle \text{PARAMETER} \rangle)$
 $\langle \text{PARAMETER-HEAD} \rangle ::= ($
 $\quad \quad \quad | \quad \langle \text{PARAMETER-HEAD} \rangle \langle \text{PARAMETER} \rangle ,$
 $\langle \text{DECLARE-STATEMENT} \rangle ::= \text{DECLARE } \langle \text{DECLARE-BODY} \rangle ;$
 $\langle \text{ASSIGN-LIST} \rangle ::= \langle \text{ASSIGN} \rangle \langle \text{PARAMETER-LIST} \rangle$
 $\langle \text{TEXT} \rangle ::= \langle \text{TextToken} \rangle$
 $\langle \text{REPLACE-STMT} \rangle ::= \text{REPLACE } \langle \text{REPLACE-HEAD} \rangle \text{ BY } \langle \text{TEXT} \rangle$
 $\langle \text{REPLACE-HEAD} \rangle ::= \langle \text{IDENTIFIER} \rangle$
 $\quad \quad \quad | \quad \langle \text{IDENTIFIER} \rangle (\langle \text{ARG-LIST} \rangle)$
 $\langle \text{ARG-LIST} \rangle ::= \langle \text{IDENTIFIER} \rangle$
 $\quad \quad \quad | \quad \langle \text{ARG-LIST} \rangle , \langle \text{IDENTIFIER} \rangle$
 $\langle \text{STRUCTURE-STMT} \rangle ::= \text{STRUCTURE } \langle \text{STRUCT-STMT-HEAD} \rangle \langle \text{STRUCT-STMT-TAIL} \rangle$
 $\langle \text{STRUCT-STMT-HEAD} \rangle ::= \langle \text{STRUCTURE-ID} \rangle : \langle \text{LEVEL} \rangle$
 $\quad \quad \quad | \quad \langle \text{STRUCTURE-ID} \rangle \langle \text{MINOR-ATTR-LIST} \rangle : \langle \text{LEVEL} \rangle$
 $\quad \quad \quad | \quad \langle \text{STRUCT-STMT-HEAD} \rangle \langle \text{DECLARATION} \rangle , \langle \text{LEVEL} \rangle$
 $\langle \text{STRUCT-STMT-TAIL} \rangle ::= \langle \text{DECLARATION} \rangle ;$
 $\langle \text{INLINE-DEFINITION} \rangle ::= \langle \text{ARITH-INLINE} \rangle$
 $\quad \quad \quad | \quad \langle \text{BIT-INLINE} \rangle$
 $\quad \quad \quad | \quad \langle \text{CHAR-INLINE} \rangle$
 $\quad \quad \quad | \quad \langle \text{STRUCTURE-EXP} \rangle$
 $\langle \text{ARITH-INLINE} \rangle ::= \langle \text{ARITH-INLINE-DEF} \rangle \langle \text{CLOSING} \rangle ;$
 $\quad \quad \quad | \quad \langle \text{ARITH-INLINE-DEF} \rangle \langle \text{BLOCK-BODY} \rangle \langle \text{CLOSING} \rangle ;$

$$\begin{aligned}
\langle \text{ARITH-INLINE-DEF} \rangle & ::= \text{FUNCTION } \langle \text{ARITH-SPEC} \rangle ; \\
& \quad | \quad \text{FUNCTION} ; \\
\langle \text{BIT-INLINE} \rangle & ::= \langle \text{BIT-INLINE-DEF} \rangle \langle \text{CLOSING} \rangle ; \\
& \quad | \quad \langle \text{BIT-INLINE-DEF} \rangle \langle \text{BLOCK-BODY} \rangle \langle \text{CLOSING} \rangle ; \\
\langle \text{BIT-INLINE-DEF} \rangle & ::= \text{FUNCTION } \langle \text{BIT-SPEC} \rangle ; \\
\langle \text{CHAR-INLINE} \rangle & ::= \langle \text{CHAR-INLINE-DEF} \rangle \langle \text{CLOSING} \rangle ; \\
& \quad | \quad \langle \text{CHAR-INLINE-DEF} \rangle \langle \text{BLOCK-BODY} \rangle \langle \text{CLOSING} \rangle ; \\
\langle \text{CHAR-INLINE-DEF} \rangle & ::= \text{FUNCTION } \langle \text{CHAR-SPEC} \rangle ; \\
\langle \text{STRUC-INLINE-DEF} \rangle & ::= \text{FUNCTION } \langle \text{STRUCT-SPEC} \rangle ;
\end{aligned}$$