Section 1
Sorted BNF

```
ACTION_annotation ::=                          "ACTION annotation" section (3.6.3.7)
     synchronous
   | asynchronous

alias ::=                                      "Generic Objects" section (3.4.7)
     ALIAS identifier = identifier ;

all_purpose_item ::=                           "Auxiliary Objects" section (3.4.6)
     annotation
   | annotation_container
   | generic_object
   | template_instantiation
   | cell_instantiation

all_purpose_items ::=                          "Auxiliary Objects" section (3.4.6)
     all_purpose_item { all_purpose_item }

analog measurement ::=        "Models for interpolateable tables and equations" section (3.6.8.1)
     CURRENT
   | ENERGY
   | FREQUENCY
   | POWER
   | TEMPERATURE
   | TIME
   | VOLTAGE

annotation ::=                                  "Auxiliary Objects" section (3.4.6)
     assignment
   | annotation_assignment { all_purpose_items }

annotation_assignment ::=                      "Assignments" section (3.4.1)
     named_assignment_base
   | unnamed_assignment_base

annotation_container ::=                       "Auxiliary Objects" section (3.4.6)
     context_sensitive_keyword { all_purpose_items }

any_character ::=                    "Reserved and Non-reserved Characters" section (3.2.4)
     reserved_character
   | nonreserved_character
   | escape_character
   | whitespace

arithmetic_binary_operator ::=                 "Operators" section (3.4.5)
   + | - | * | / | ** | %
```

```
arithmetic_expression ::=                    "Expressions" section (3.4.2)
     [ arithmetic_unary_operator ] arithmetic_primary
   | arithmetic_expression  arithmetic_binary_operator
     arithmetic_expression
   | arithmetic_function_operator ( arithmetic_expression
     { , arithmetic_expression } )

arithmetic_function_operator ::=             "Operators" section (3.4.5)
     abs
   | exp
   | log
   | min
   | max

arithmetic_model ::=                         "Arithmetic Model" section (3.4.15)
     context_sensitive_keyword [ identifier ]
     { [all_purpose_items] [header] bodies }
   | context_sensitive_keyword [identifier] = primary ;
   | context_sensitive_keyword [identifier] =
     primary  { all_purpose_items }
   | arithmetic_model_template_instantiation

arithmetic_primary ::=                       "Expressions" section (3.4.2)
     number
   | identifier
   | ( arithmetic_expression )

arithmetic_unary_operator ::=                "Operators" section (3.4.5)
     + | -

assignment ::=                               "Assignments" section (3.4.1)
     named_assignment
   | unnamed_assignment

assignments ::=                              "Assignments" section (3.4.1)
     assignment { assignment }

attribute ::=                                "Generic Objects" section (3.4.7)
     ATTRIBUTE { attribute_items }

attribute_item ::=                           "Generic Objects" section (3.4.7)
     identifier [ { unnamed_assignments } ]

attribute_items ::=                          "Generic Objects" section (3.4.7)
     attribute_item { attribute_item }

based_literal ::=                            "Based Literals" section (3.2.9)
     binary_base { _ | binary_digit }
   | octal_base { _ | octal_digit }
   | decimal_base { _ | decimal_digit }
   | hex_base { _ | hex_digit }

behavior ::=                                 "Function" section (3.4.16)
     BEHAVIOR [ identifier ]  { behavior_body }
```

```
behavior_body ::=                                   "Function" section (3.4.16)
     primitives
   | combinational_assignments
   | sequential_assignments
   | primitive_instantiations

binary_arithmetic_operator ::=                      "Arithmetic operators" section (3.5.1)
   + |- |* | / |** |%

binary_base ::=                                     "Based Literals" section (3.2.9)
   'B | 'b

binary_bitwise_operator ::=                         "Boolean operators on words" section (3.5.3)
     & | | | ^ | ~^

binary_boolean_operator ::=                         "Boolean operators on scalars" section (3.5.2)
     && | & | || | | == | ~^ | != | ^

binary_digit ::=                                    "Based Literals" section (3.2.9)
     bit_literal

binary_operator ::=                                 "Boolean operators on words" section (3.5.3)
     << | >> | + | - | * | / | %

binary_reduction_operator ::=                       "Boolean operators on words" section (3.5.3)
     > | < | >= | <=

binary_vector_operator ::=                          "Vector operators" section (3.5.4)
     -> | <-> | &> | <&>

bit_edge_literal ::=                                "Edge Literals" section (3.2.10)
     bit_literal  bit_literal

bit_literal ::=                                     "Boolean Literals" section (3.2.8)
   X | Z | L | H | U | W | ? | 0 | 1 |
   x | z | l | h | u | w

bodies ::=                                          "Arithmetic Model" section (3.4.15)
     body { body }

body ::=                                            "Arithmetic Model" section (3.4.15)
     table
   | equation

boolean_binary_operator ::=                         "Operators" section (3.4.5)
     + | - | * | / | % | && | || | & | | | ^ | ~& | ~| | ~^
   | > | < | >= | <= | == | !=

boolean_expression ::=                              "Expressions" section (3.4.2)
     [ boolean_unary_operator ]  boolean_primary
   | boolean_expression boolean_binary_operator boolean_expression
   | boolean_expression ? boolean_expression : boolean_expression
```

```
boolean_primary ::=                          "Expressions" section (3.4.2)
     logic_literal
   | identifier [ index ]
   | ( boolean_expression )

boolean_unary_operator ::=                   "Operators" section (3.4.5)
     ! | ~ | & | ~& | | | ~| | ^ | ~^

BUFFERTYPE_annotation ::=                    "BUFFERTYPE annotation" section (3.6.5.2)
     input
   | output
   | inout
   | internal

cell ::=                                     "Cell Object" section (3.4.8)
     CELL cell_identifier  [ { cell_items | ; } ]
   | cell_template_instantiation

cell_instantiation ::=                       "Instantiations" section (3.4.3)
     cell_identifier { primaries }
   | cell_identifier { pin_assignments }

cell_instantiations ::=                      "Instantiations" section (3.4.3)
     cell_instantiation  { cell_instantiation }

cell_item ::=                                "Cell Object" section (3.4.8)
     all_purpose_item
   | pin
   | primitive
   | function
   | arithmetic_model
   | vector

cell_items ::=                               "Cell Object" section (3.4.8)
     cell_item {cell_item}

CELL_object_ATTRIBUTE ::=      "ATTRIBUTE within a CELL object" section (3.6.6.2)
     RAM
   | ROM
   | CAM
   | static
   | dynamic
   | asynchronous
   | synchronous

cells ::=                                    "Cell Object" section (3.4.8)
     cell { cell }

CELLTYPE_annotation ::=                      "CELLTYPE annotation" section (3.6.5.1)
      buffer
   | combinational
   | multiplexor
   | flipflop
   | latch
```

```
class::=                                        "Generic Objects" section (3.4.7)
     CLASS identifier  ;
     CLASS identifier  [ { generic_objects } ]

combinational_assignment ::=                    "Assignments" section (3.4.1)
     identifier [index] = boolean_expression ;

combinational_assignments ::=                   "Assignments" section (3.4.1)
     combinational_assignment  { combinational_assignment }

comment ::=                                     "Comments" section (3.2.6)
     single_line_comment
   | block_comment

connect_class_annotation ::=                    "CONNECT_CLASS annotation" section (3.6.3.12)
   CONNECT_CLASS_annotation = string ;

connectivity_data ::=                           "Models for non-interpolateable tables" section (3.6.8.2)
     CONNECTIVITY
   | DRIVER
   | RECEIVER

CONNECT_RULE_annotation ::=                     "CONNECT_RULE annotation" section (3.6.7.4)
     must_short
   | can_short
   | cannot_short

constant ::=                                    "Generic Objects" section (3.4.7)
     CONSTANT identifier = number ;
     CONSTANT identifier = logic_literal ;

context_sensitive_keyword ::=                   "Literals" section (3.4.4)
     nonescaped_identifier

DATATYPE_annotation ::=                         "DATATYPE annotation" section (3.6.3.13)
     signed
   | unsigned

decimal_base ::=                                "Based Literals" section (3.2.9)
   'D | 'd

decimal_digit ::=                               "Based Literals" section (3.2.9)
   0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

default_annotation ::=
   DEFAULT_annotation= number ;                 "DEFAULT annotation" section (3.6.7.1)

delimiter ::=                                   "Delimiters" section (3.2.5)
     reserved_character
   | && | ~& | || | ~| | ~^ | == | != | ** | >= | <=
   | ?! | ?~ | ?- | ?? | -> | <-> | &> | <&> | >> | <<

DIRECTION_annotation ::=                        "DIRECTION annotation" section (3.6.3.5)
     input
```

```
   | output
   | both
   | none

DRIVERTYPE_annotation ::=                    "DRIVERTYPE annotation" section (3.6.5.3)
     predriver
   | slotdriver
   | both

DRIVETYPE_annotation ::=                     "DRIVETYPE annotation" section (3.6.3.4)
     cmos
   | nmos
   | pmos
   | nmos_pass
   | pmos_pass
   | cmos_pass
   | ttl
   | open_drain
   | open_source

edge_literal ::=                    "Edge Literals" section (3.2.10),"Literals" section (3.4.4)
     bit_edge_literal
   | word_edge_literal
   | symbolic_edge_literal

edge_literals::=                             "Literals" section (3.4.4)
     edge_literal { edge_literal }

enable_pin_annotation ::=                    "ENABLE_PIN annotation" section (3.6.3.9)
   ENABLE_PIN_annotation ::= string

equation ::=                                 "Arithmetic Model" section (3.4.15)
     EQUATION { arithmetic_expression }
   | equation_template_instantiation

escape_character ::=                "Reserved and Non-reserved Characters" section (3.2.4)
     \

escaped_identifier ::=                       "Identifiers" section (3.2.12)
   escape_character { nonreserved_character | reserved_character }

extraction_data ::=          "Models for interpolateable tables and equations" section (3.6.8.1)
     CAPACITANCE
   | RESISTANCE

function ::=                                 "Function" section (3.4.16)
     FUNCTION [ identifier ] { [all_purpose_items] [primitives]
     [function_bodies] }
   | function_template_instantiation

function_arithmetic_operator ::=             "Arithmetic operators" section (3.5.1)
     LOG | EXP | ABS | MIN | MAX
```

```
function_bodies ::=                          "Function" section (3.4.16)
     function_body { function_body }

function_body ::=                            "Function" section (3.4.16)
     behavior
   | statetable

functions ::=                                "Function" section (3.4.16)
     function { function }

generic_object ::=                           "Auxiliary Objects" section (3.4.6)
     alias
   | attribute
   | constant
   | class
   | group
   | include
   | property
   | template

generic_objects ::=                          "Auxiliary Objects" section (3.4.6)
     generic_object { generic_object }

group ::=                                    "Generic Objects" section (3.4.7)
     GROUP group_identifier { identifiers }
   | GROUP group_identifier { numbers }
   | GROUP group_identifier { edge_literals }
   | GROUP group_identifier { logic_literals }
   | GROUP group_identifier { indexed_identifiers }
   | GROUP group_identifier { integer : integer }

header ::=                                   "Arithmetic Model" section (3.4.15)
     HEADER { header_items [body] }
   | header_template_instantiation

header_items ::=                             "Arithmetic Model" section (3.4.15)
     header_item { header_item }

header_item ::=                              "Arithmetic Model" section (3.4.15)
     identifier
   | all_purpose_item
   | arithmetic_model

hex_base ::=                                 "Based Literals" section (3.2.9)
   'H | 'h

hex_digit ::=                                "Based Literals" section (3.2.9)
   octal_digit | 8 | 9 | A | B | C | D | E | F | a | b | c | d | e | f

identifier ::=                               "Literals" section (3.4.4)
     nonescaped_identifier
   | escaped_identifier
   | placeholder_identifier
```

```
identifiers ::=                                  "Literals" section (3.4.4)
   identifier { identifier }

include ::=                                      "Generic Objects" section (3.4.7)
     INCLUDE quoted_string ;

index ::=                                        "Literals" section (3.4.4)
     [ index_primary ]
   | [ index_primary : index_primary ]

indexed_identifier ::=                           "Literals" section (3.4.4)
     identifier index

indexed_identifiers ::=                          "Literals" section (3.4.4)
     indexed_identifier { indexed_identifier }

indexed_primary ::=                              "Literals" section (3.4.4)
     unsigned
   | identifier

information_annotation_container ::=      "Information container" section (3.6.1.5)
     VERSION
   | TITLE
   | PRODUCT
   | AUTHOR
   | DATETIME

integer_digit ::=                                "Number Literals" section (3.2.7)
   0  |  1  |  2  |  3  |  4  |  5  |  6  |  7  |  8  |  9

integer ::=                                      "Number Literals" section (3.2.7)
   [ sign ] unsigned

label_annotation ::=                             "LABEL annotation" section (3.6.4.1)
   LABEL_annotation= string ;

layout_data ::=              "Models for interpolateable tables and equations" section (3.6.8.1)
     AREA
   | DISTANCE
   | HEIGHT
   | LENGTH
   | WIDTH

libraries ::=                                    "Library Object" section (3.4.9)
     library { library }

library ::=                                      "Library Object" section (3.4.9)
     LIBRARY library_identifier { library_items [sublibraries] }
   | library_template_instantiation

library_item ::=                                 "Library Object" section (3.4.9)
     all_purpose_item
   | arithmetic_model
   | cell
```

```
   | primitive
   | wire

library_items ::=                          "Library Object" section (3.4.9)
     library_item { library_item }

library_specific_object ::=                "Auxiliary Objects" section (3.4.6)
     annotation
   | annotation_container
   | cell
   | function
   | library
   | pin
   | primitive
   | sublibrary
   | vector
   | wire

logic_literal ::=                          "Literals" section (3.4.4)
     bit_literal
   | based_literal

logic_literals::=                          "Literals" section (3.4.4)
    logic_literal { logic_literal }

logic_value ::=                            "Literals" section (3.4.4)
     logic_literal
   | edge_literal
   | ( [!] logic_variable )

logic_values ::=                           "Literals" section (3.4.4)
     logic_value {logic_value}

logic_variable ::=                         "Literals" section (3.4.4)
     pin_identifier [index]

logic_variables ::=                        "Literals" section (3.4.4)
     logic_variable {logic_variable}

MEASUREMENT_annotation ::=                 "MEASUREMENT annotation" section (3.6.7.3)
     transient
   | static
   | average
   | rms
   | peak

named_assignment ::=                       "Assignments" section (3.4.1)
     named_assignment_base  ;

named_assignment_base ::=                  "Assignments" section (3.4.1)
     context_sensitive_keyword identifier = number
   | context_sensitive_keyword identifier = string
```

```
named_assignments ::=                          "Assignments" section (3.4.1)
     named_assignment { named_assignment }

nonescaped_identifier ::=                      "Identifiers" section (3.2.12)
   nonreserved_character { nonreserved_character }

non_negative_number ::=                        "Number Literals" section (3.2.7)
     unsigned [ .  unsigned ]
   | unsigned [ . unsigned ] E [ sign ] unsigned

nonreserved_character ::=          "Reserved and Non-reserved Characters" section (3.2.4)
     letter | digit | _ | $

number ::=                                     "Number Literals" section (3.2.7)
     [ sign ] non_negative_number

numbers ::=                                    "Literals" section (3.4.4)
     number { number }

object ::=                                     "Auxiliary Objects" section (3.4.6)
     generic_object
   | library_specific_object
   | arithmetic_model
   | header

object_keyword ::=          "Keywords for referencing objects used as annotation" section (3.6.2)
     CELL
   | PRIMITIVE
   | PIN
   | CLASS

objects ::=                                    "Auxiliary Objects" section (3.4.6)
     object { object }

octal_base ::=                                 "Based Literals" section (3.2.9)
   'O | 'o

octal_digit ::=                                "Based Literals" section (3.2.9)
   binary_digit  | 2 | 3 | 4 | 5 | 6 | 7

OFF_STATE_annotation ::=                       "OFF_STATE annotation" section (3.6.3.16)
      inverted
    | non_inverted

ORIENTATION_annotation ::=                     "ORIENTATION annotation" section (3.6.3.11)
     left
   | right
   | top
   | bottom

pin ::=                                        "Pin Object" section (3.4.10)
     PIN [ index ] pin_identifier [ index ] [{ pin_items | ;  }]
   | pin_template_instantiation
```

```
pin_assignment ::=                          "Assignments" section (3.4.1)
     identifier [index] = identifier [index] ;
   | identifier [index] = logic_literal ;
   | logic_literal  = identifier [index] ;

pin_assignments ::=                         "Assignments" section (3.4.1)
     pin_assignment { pin_assignment }

pin_item ::=                                "Pin Object" section (3.4.10)
     all_purpose_item

pin_items ::=                               "Pin Object" section (3.4.10)pin_item {
   pin_item }

PIN_object_ATTRIBUTE ::=      "ATTRIBUTE within a PIN object" section (3.6.6.1)
     SCHMITT
   | TRISTATE
   | XTAL
   | PAD

pin_related_data ::=          "Models for interpolateable tables and equations" section (3.6.8.1)
     THRESHOLD
   | DRIVE_STRENGTH
   | SWITCHING_BITS
   | FANOUT
   | FANIN
   | CONNECTIONS

pins ::=                                    "Pin Object" section (3.4.10)
     pin { pin }

PINTYPE_annotation ::=                      "PINTYPE annotation" section (3.6.3.2)
     digital
   | analog
   | supply

placeholder_identifier ::=                  "Identifiers" section (3.2.12)
   < nonescaped_identifier >

POLARITY_annotation_attribute::=       "ATTRIBUTE within a PIN object" section (3.6.6.1)
     TIE
   | READ
   | WRITE

POLARITY_input_annotation ::=               "POLARITY annotation" section (3.6.3.8)
     high
   | low
   | rising_edge
   | falling_edge
   | double_edge

POLARITY_output_annotation ::=              "POLARITY annotation" section (3.6.3.8)
     inverted
   | non_inverted
```

```
   |,both
   | none
```

```
predefined_derating_case ::=    "Models for non-interpolateable tables and equations" section
   (3.6.8.3)
     bccom
   | bcind
   | bcmil
   | wccom
   | wcind
   | wcmil

predefined_process_name ::="Models for non-interpolateable tables and equations" section (3.6.8.3)
     snsp
   | snwp
   | wnsp
   | wnwp

primaries ::=                                "Literals" section (3.4.4)
     primary { primary }

primary ::=                                  "Literals" section (3.4.4)
     number
   | identifier

primitive ::=                                "Primitive Object" section (3.4.11)
     PRIMITIVE primitive_identifier { primitive_items }
   | primitive_template_instantiation

primitive_instantiation ::=                  "Instantiations" section (3.4.3)
     primitive_identifier [ identifier ] { primaries }
   | primitive_identifier [ identifier ]
     { combinational_assignments }
   | primitive_identifier [ identifier ] { pin_assignments }

primitive_instantiations ::=                 "Instantiations" section (3.4.3)
     primitive_instantiation  { primitive_instantiation }

primitive_item ::=                           "Primitive Object" section (3.4.11)
     all_purpose_item
   | pin
   | function

primitive_items ::=                          "Primitive Object" section (3.4.11)
     primitive_item { primitive_item }

primitives ::=                               "Primitive Object" section (3.4.11)
     primitive { primitive }

process_data ::=           "Models for non-interpolateable tables and equations" section (3.6.8.3)
     DERATE_CASE
   | PROCESS
```

```
property ::=                                    "Generic Objects" section (3.4.7)
     PROPERTY [ identifier ] { unnamed_assignments }

PULL_annotation ::=                             "PULL annotation" section (3.6.3.10)
     up
   | down
   | both
   | none

quoted_string ::=                               "Quoted Strings" section (3.2.11)
     " { any_character } "

reserved_character ::=                          "Reserved and Non-reserved Characters" section (3.2.4)
     & | | | ^ | ~ | + | - | * | / | % | ? | ! | = | < | > | :
   | ( | ) | [ | ] | { | } | @ | ; | , | . | ” | ’

scan_position_annotation ::=                    "SCAN_POSITION annotation" section (3.6.3.14)
   SCAN_POSITION_annotation= unsigned ;

SCAN_TYPE_annotation ::=                        "SCAN_TYPE annotation" section (3.6.5.5)
     muxscan
   | clocked
   | lssd
   | control_0
   | control_1

SCAN_USAGE_annotation ::=                       "SCAN_USAGE annotation" section (3.6.5.6)
     input
   | output
   | hold

SCOPE_annotation ::=                            "SCOPE annotation" section (3.6.3.6)
     behavior
   | measure
   | both
   | none

sequential_assignment ::=                       "Assignments" section (3.4.1)
     @ ( vector_boolean_expression ) { combinational_assignments }
   { : ( vector_boolean_expression ) { combinational_assignments } }

sequential_assignments ::=                      "Assignments" section (3.4.1)
   sequential_assignment  { sequential_assignment }

sign ::=                                        "Number Literals" section (3.2.7)
     +  |  -

SIGNALTYPE_annotation ::=                       "SIGNALTYPE annotation" section (3.6.3.3)
     data
   | scan_data
   | control
   | select
   | enable
```

```
   | out_enable
   | scan_enable
   | scan_out_enable
   | clear
   | set
   | write
   | read
   | scan_clock
   | master_clock
   | slave_clock

source_text ::=                          "Auxiliary Objects" section (3.4.6)
     ALF_REVISION version_string library

statetable ::=                           "Function" section (3.4.16)
     STATETABLE [ identifier ]  {  statetable_body }

statetable_body ::=                      "Function" section (3.4.16)
     logic_variables : logic_variables ;
     logic_values : logic_values ;
     { logic_values : logic_values ; }

string ::=                               "Literals" section (3.4.4)
     quoted_string
   | identifier

STUCK_annotation ::=                     "STUCK annotation" section (3.6.3.15)
      stuck_at_0
    | stuck_at_1
    | both
    | none

sublibraries ::=                         "Sublibrary Object" section (3.4.12)
     sublibrary { sublibrary }

sublibrary ::=                           "Sublibrary Object" section (3.4.12)
     SUBLIBRARY library_identifier { library_items }
   | sublibrary_template_instantiation

symbolic_edge_literal ::=                "Edge Literals" section (3.2.10)
     ?? | ?~ | ?! | ?-

table ::=                                "Arithmetic Model" section (3.4.15)
     TABLE { primaries }
   | table_template_instantiation

template ::=                             "Generic Objects" section (3.4.7)
     TEMPLATE template_identifier { objects }

template_instantiation ::=               "Instantiations" section (3.4.3)
     template_identifier { primaries }
   | template_identifier { unnamed_assignments }
```

```
template_instantiations ::=                    "Instantiations" section (3.4.3)
     template_instantiation  { template_instantiation }

ternary_operator ::=                           "Boolean operators on scalars" section (3.5.2)
      ? :

timing measurement ::=          "Models for interpolateable tables and equations" section (3.6.8.1)
     DELAY
   | HOLD
   | JITTER
   | NOCHANGE
   | PERIOD
   | PULSEWIDTH
   | RECOVERY
   | REMOVAL
   | SETUP
   | SKEW
   | SLEWRATE

unary_arithmetic_operator ::=                  "Arithmetic operators" section (3.5.1)
   + |-

unary_bitwise_operator ::=                     "Boolean operators on words" section (3.5.3)
     ~

unary_boolean_operator ::=                     "Boolean operators on scalars" section (3.5.2)
     !

unary_reduction_operator ::=                   "Boolean operators on words" section (3.5.3)
     & | ~& | | | ~| | ^ | ~^

unary_vector_bit_operator ::=                  "Vector operators" section (3.5.4)
     01 | 10 | 00 | 11 | 0? | 1? | ?0 | ?1 | ??

unary_vector_bit_or_word_operator ::=          "Vector operators" section (3.5.4)
     ?- | ?? | ?! | ?~

 UNIT_annotation ::=                           "UNIT annotation" section (3.6.7.2)
     f* | F* | p* | P* | n* | N* | u* | U* | m* | M* | 1*
   | k* | K* | meg* | MEG* | g* | G*

unnamed_annotation_container ::=               "Annotation containers" section (3.6.1)
     SCAN
   | FROM
   | TO
   | LIMIT
   | VIOLATION
   | INFORMATION

unnamed_assignment ::=                         "Assignments" section (3.4.1)
     unnamed_assignment_base  ;
```

```
unnamed_assignment_base ::=                    "Assignments" section (3.4.1)
     context_sensitive_keyword = number
   | context_sensitive_keyword = string

unnamed_assignments ::=                        "Assignments" section (3.4.1)
     unnamed_assignment { unnamed_assignment }

unsigned ::=                                   "Number Literals" section (3.2.7)
   integer_digit  { _ | integer_digit }

wire ::=                                       "Wire Object" section (3.4.14)
     WIRE wire_identifier { wire_items }
   | wire_template_instantiation

wire_item ::=                                  "Wire Object" section (3.4.14)
     all_purpose_item
   | arithmetic_model

wire_items ::=                                 "Wire Object" section (3.4.14)
     wire_item { wire_item }

wires ::=                                      "Wire Object" section (3.4.14)
     wire { wire }

word_edge_literal ::=                          "Edge Literals" section (3.2.10)
     based_literal  based_literal

vector ::=                                     "Vector Object" section (3.4.13)
     VECTOR ( vector_boolean_expression )  { vector_items }
   | vector_template_instantiation

vector_binary_operator ::=                     "Operators" section (3.4.5)
      -> | <-> | &> | <&>

vector_boolean_expression ::=                  "Expressions" section (3.4.2)
     vector_expression
   | boolean_expression

vector_expression ::=                          "Expressions" section (3.4.2)
     ( vector_expression )
   | vector_unary_operator boolean_expression
   | vector_expression  vector_binary_operator vector_expression
   | vector_expression  boolean_binary_operator vector_expression
   | vector_expression  &&  boolean_expression
   | boolean_expression  &&  vector_expression
   | vector_expression  &  boolean_expression
   | boolean_expression  &  vector_expression
   | boolean_expression ? vector_expression : vector_expression

vector_item ::=                                "Vector Object" section (3.4.13)
     all_purpose_item
   | arithmetic_model
```

```
vector_items ::=                          "Vector Object" section (3.4.13)
     vector_item { vector_item }

vectors ::=                               "Vector Object" section (3.4.13)
     vector { vector }

vector_unary_operator ::=                 "Operators" section (3.4.5)
     edge_literal

VIEW_annotation ::=                       "VIEW annotation" section (3.6.3.1)
     functional
   | physical
   | both
   | none

violation_annotation_container ::=        "Violation container" section (3.6.1.4)
     MESSAGE_TYPE
   | MESSAGE
```