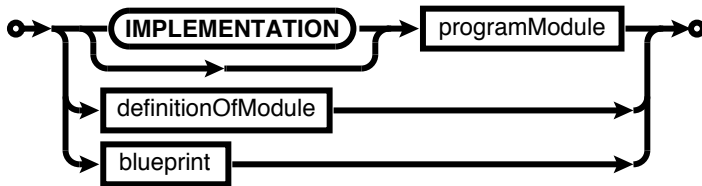


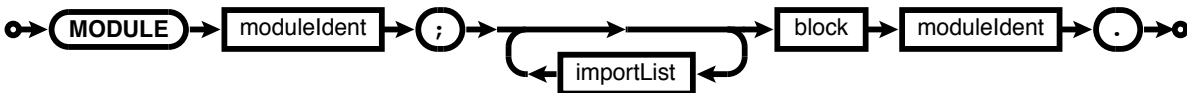
Appendix B: Syntax Diagrams

B.1 Non-Terminal Symbols

#1 Compilation Unit



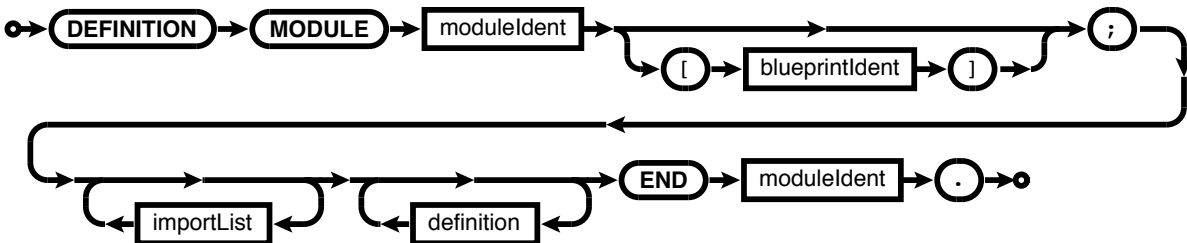
#2 Program Module



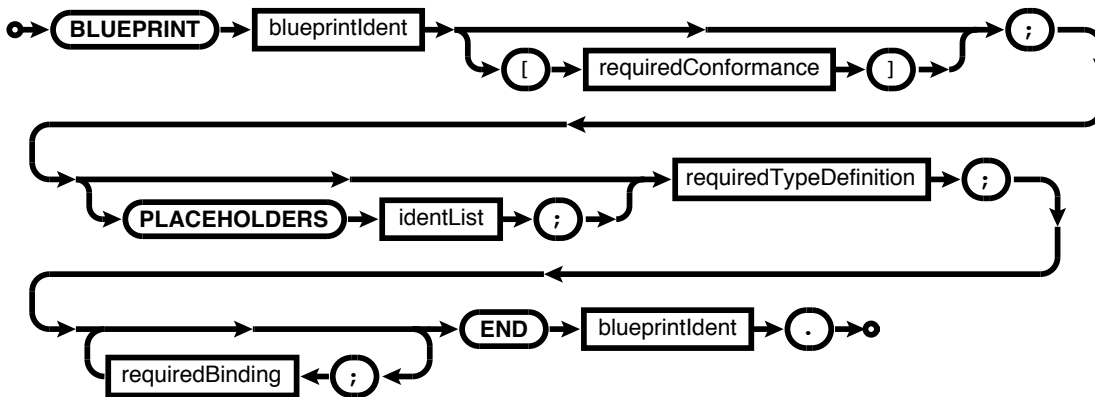
#2.1 Module Identifier



#3 Definition Of Module



#4 Blueprint



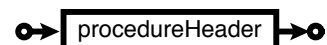
#4.1 Blueprint Identifier



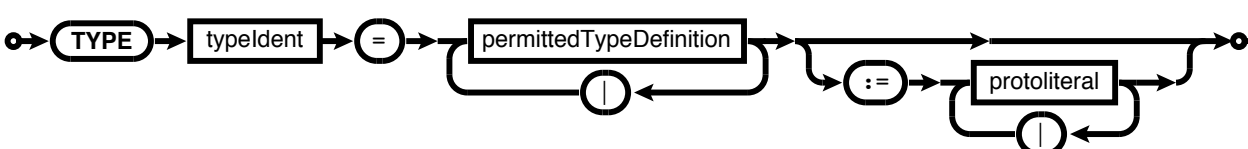
#4.2 Required Conformance

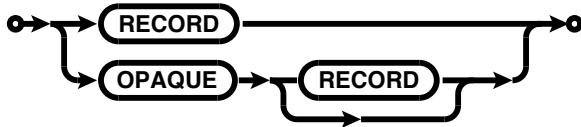
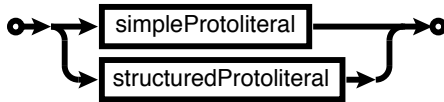
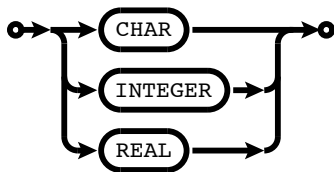
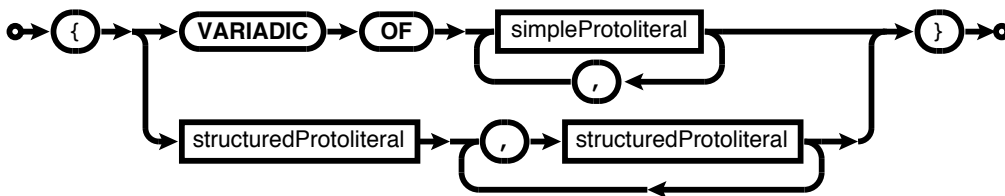
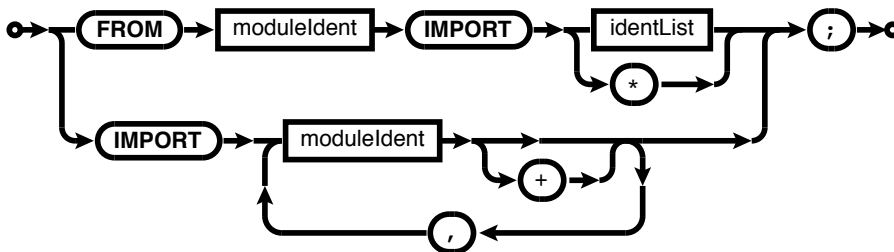
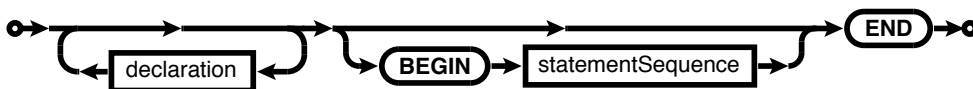
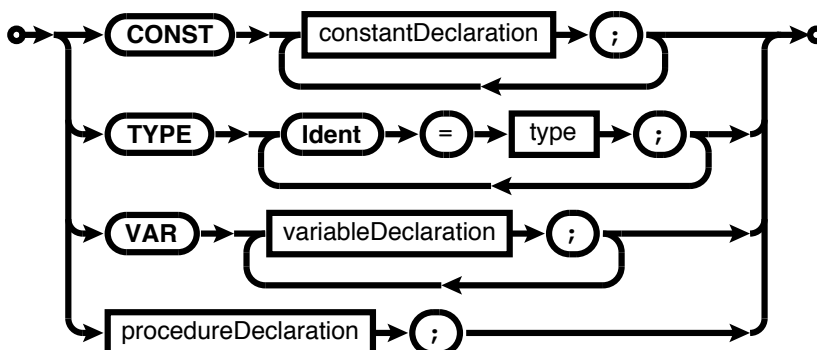


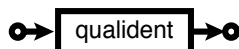
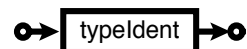
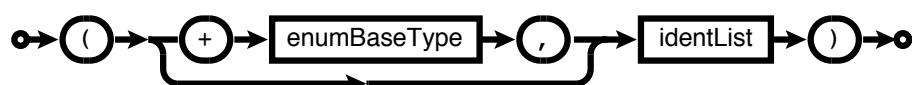
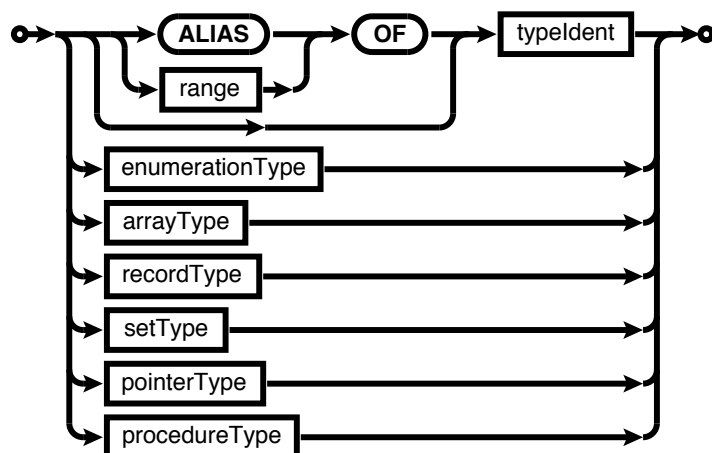
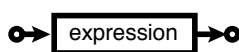
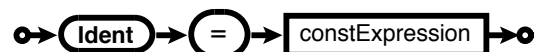
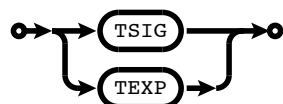
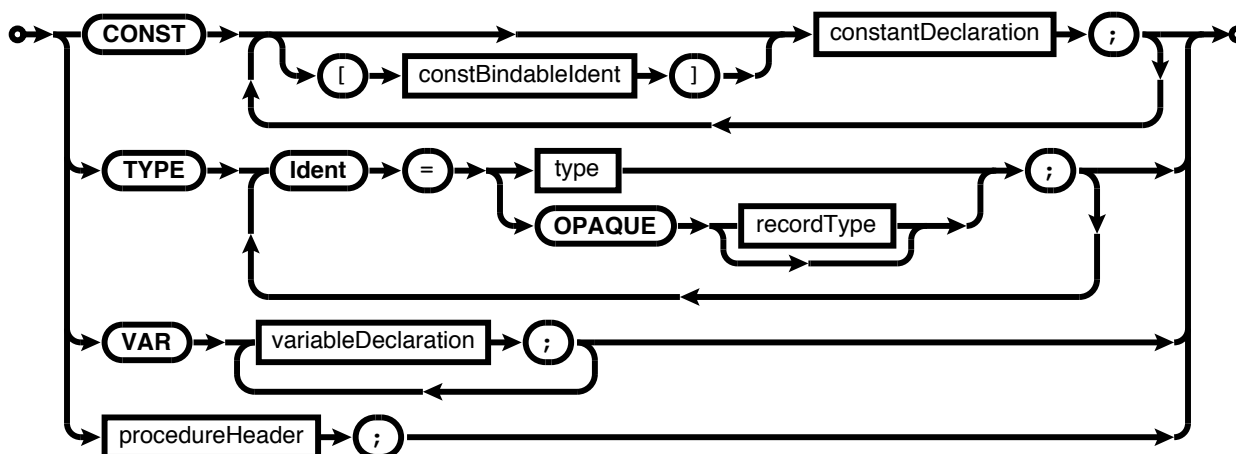
#4.3 Required Binding



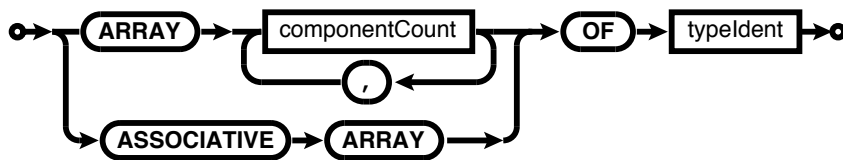
#5 Required Type Definition



#6 Permitted Type Definition**#7 Proto-Literal****#7.1 Simple Proto-Literal****#8 Structured Proto-Literal****#9 Import List****#10 Block****#11 Declaration**



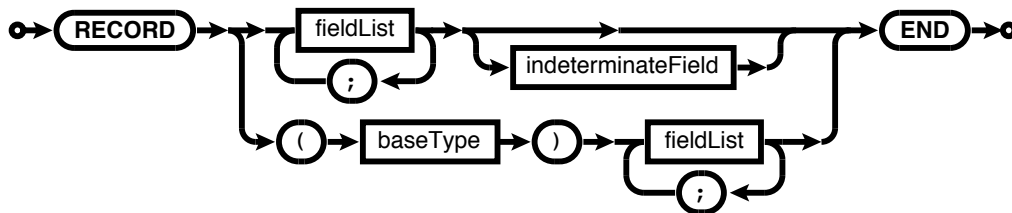
#17 Array Type



#17.1 Component Count



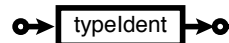
#18 Record Type



#18.1 Field List



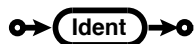
#18.2 Base Type



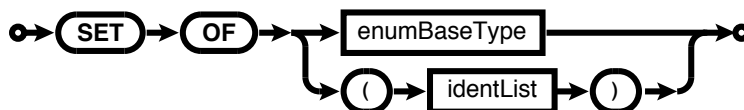
#19 Indeterminate Field



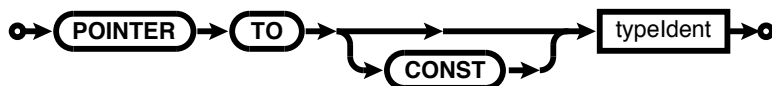
#19.1 Discriminant Field



#20 Set Type



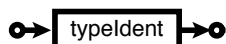
#21 Pointer Type



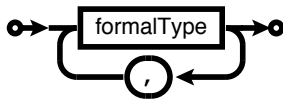
#22 Procedure Type



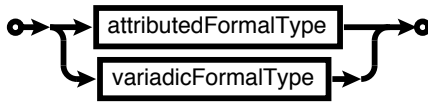
#22.1 Returned Type



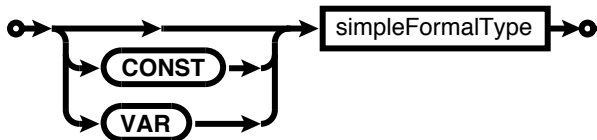
#23 Formal Type List



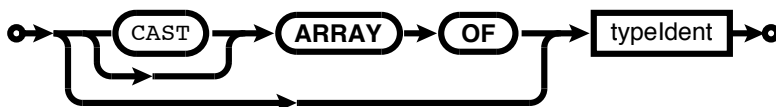
#24 Formal Type



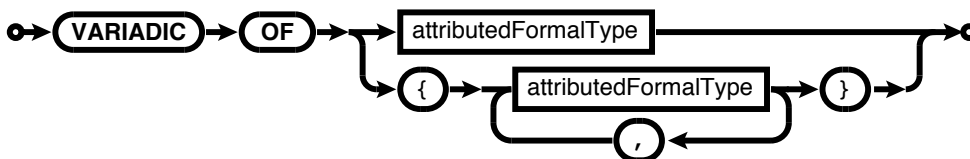
#25 Attributed Formal Type



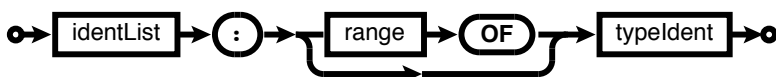
#26 Simple Formal Type



#27 Variadic Formal Type



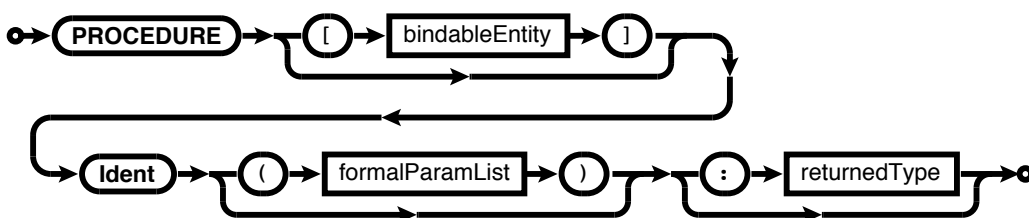
#28 Variable Declaration



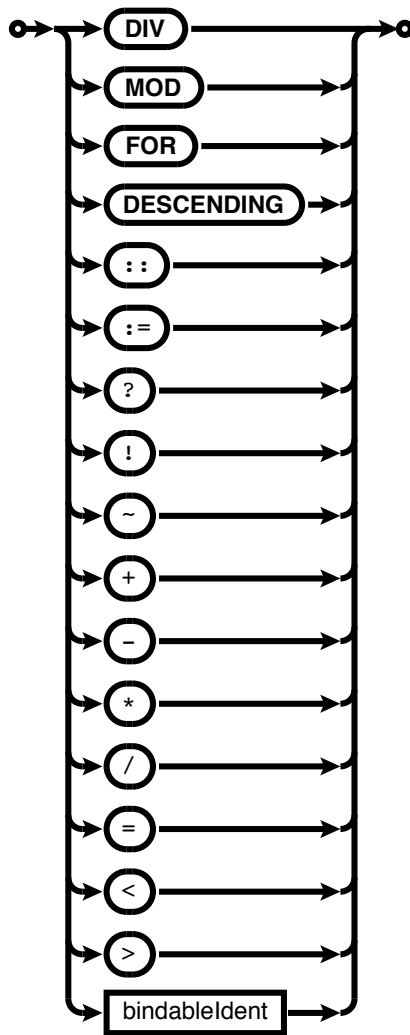
#29 Procedure Declaration



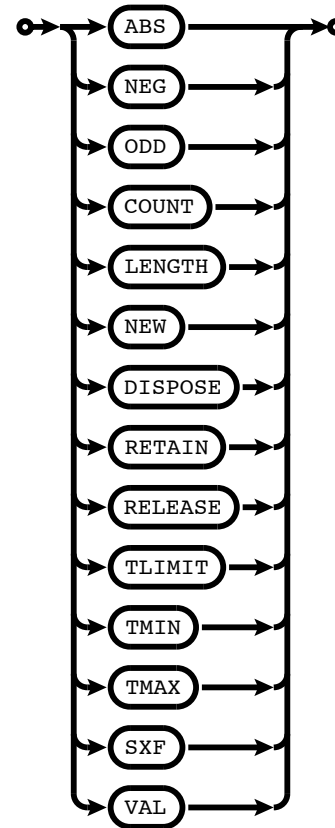
#30 Procedure Header



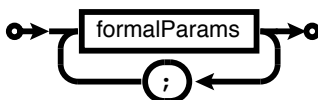
#31 Bindable Entity



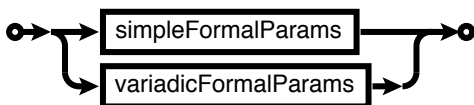
#31.1 Bindable Identifier



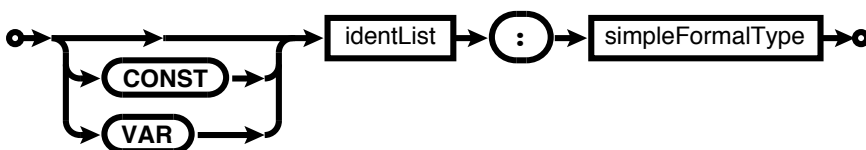
#32 Formal Parameter List



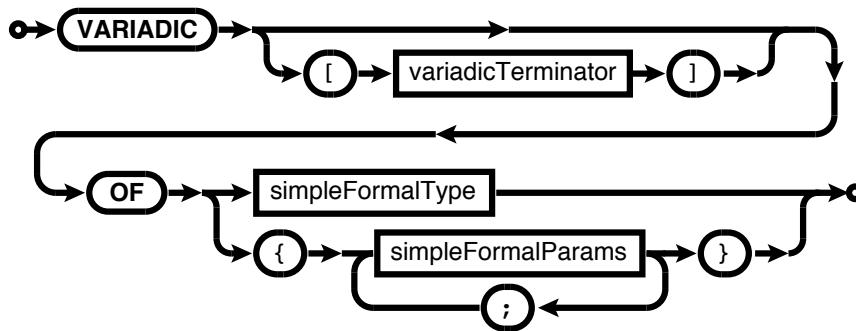
#33 Formal Parameters



#34 Simple Formal Parameters



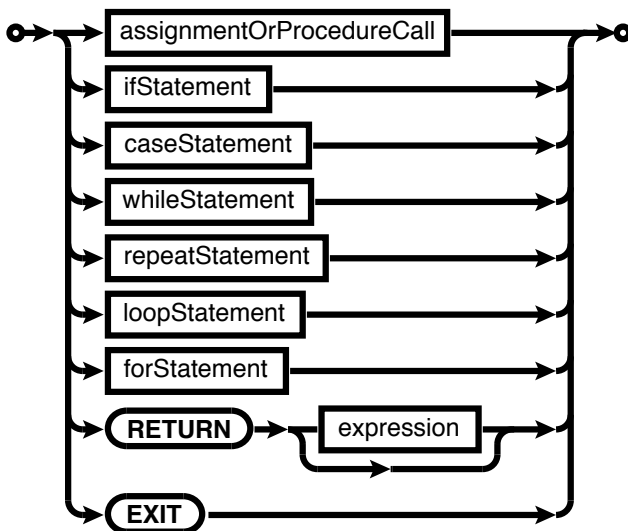
#35 Variadic Formal Parameters



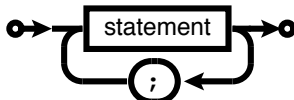
#35.1 Variadic Terminator



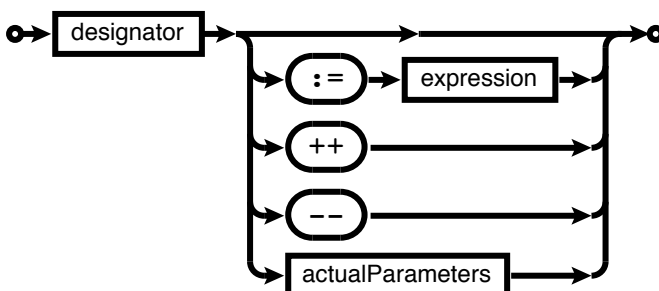
#36 Statement



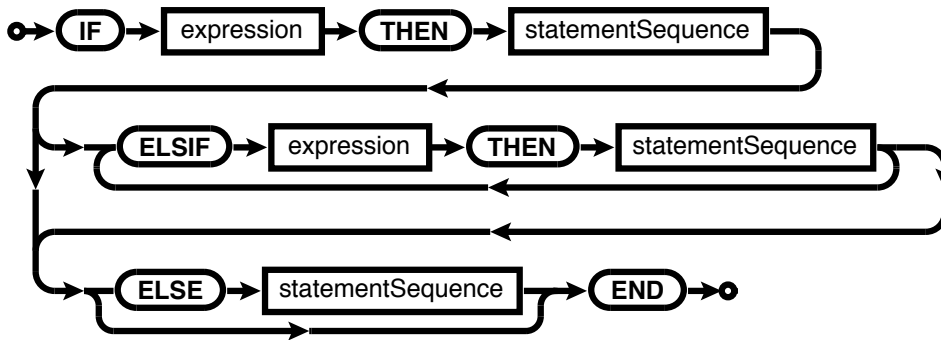
#37 StatementSequence



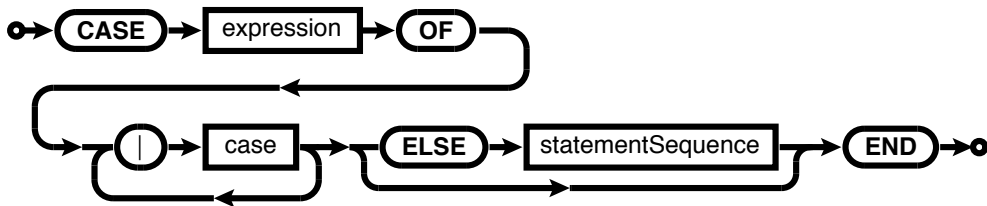
#38 Assignment Or Procedure Call



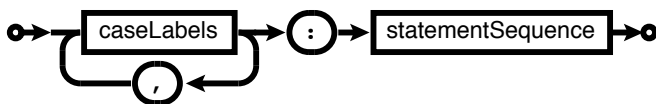
#39 IF Statement



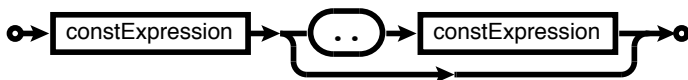
#40 CASE Statement



#41 Case



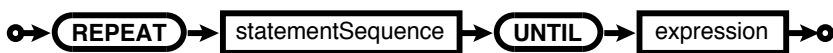
#42 Case Labels



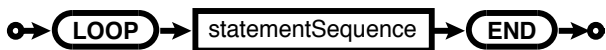
#43 WHILE Statement



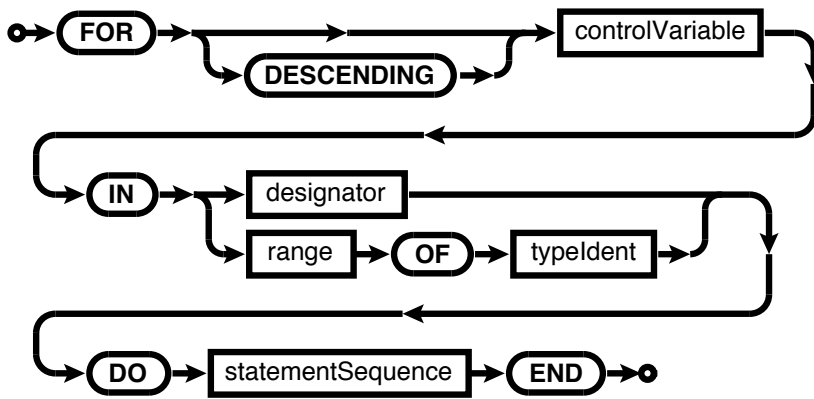
#44 REPEAT Statement



#45 LOOP Statement



#46 FOR Statement



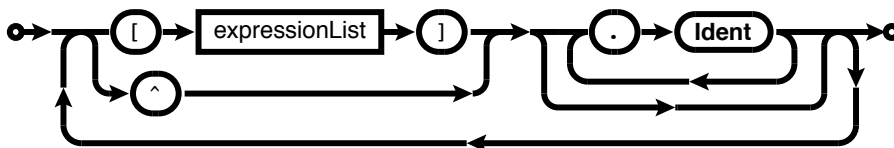
#46.1 Control Variable



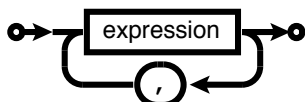
#47 Designator



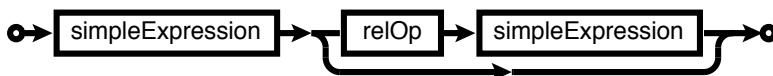
#48 Designator Tail



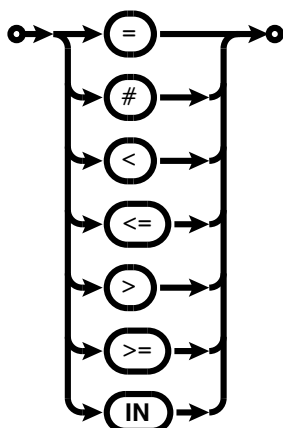
#49 Expression List



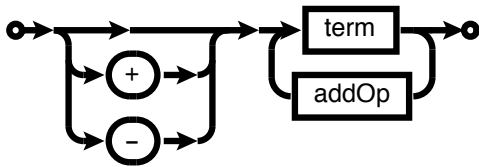
#50 Expression



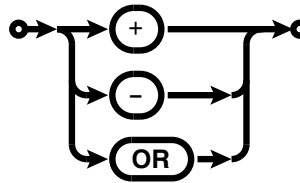
#50.1 Relational Operator



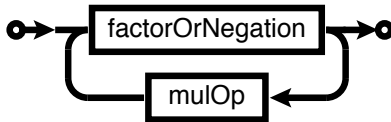
#51 Simple Expression



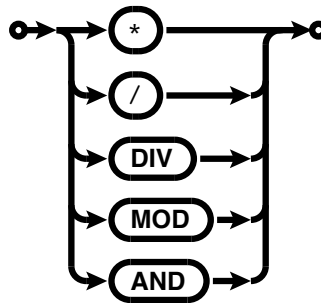
#51.1 Add Operator



#52 Term



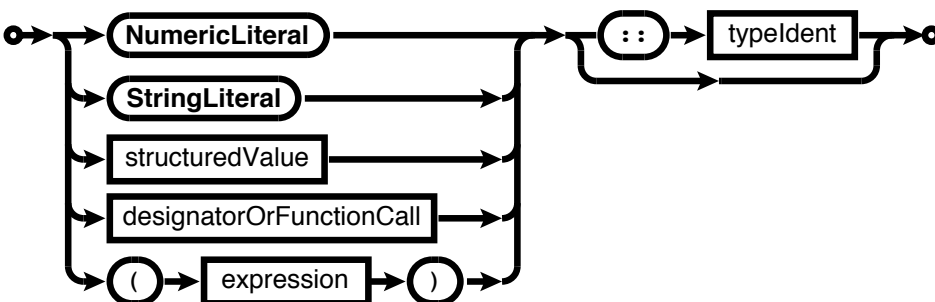
#52.1 Multiply Operator



#53 Factor Or Negation



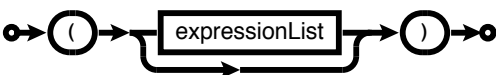
#54 Factor



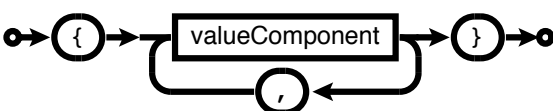
#55 Designator Or Function Call

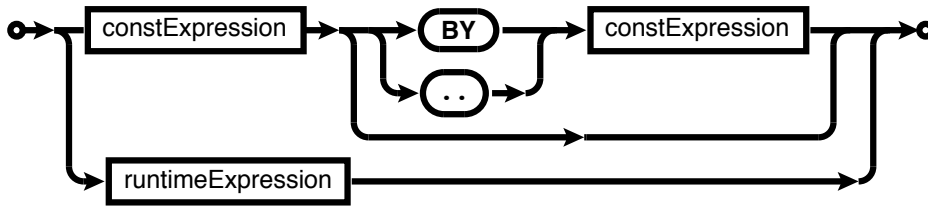
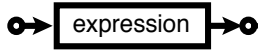
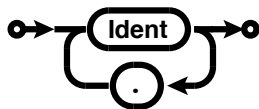
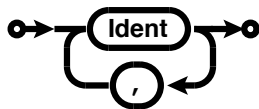


#56 Actual Parameters



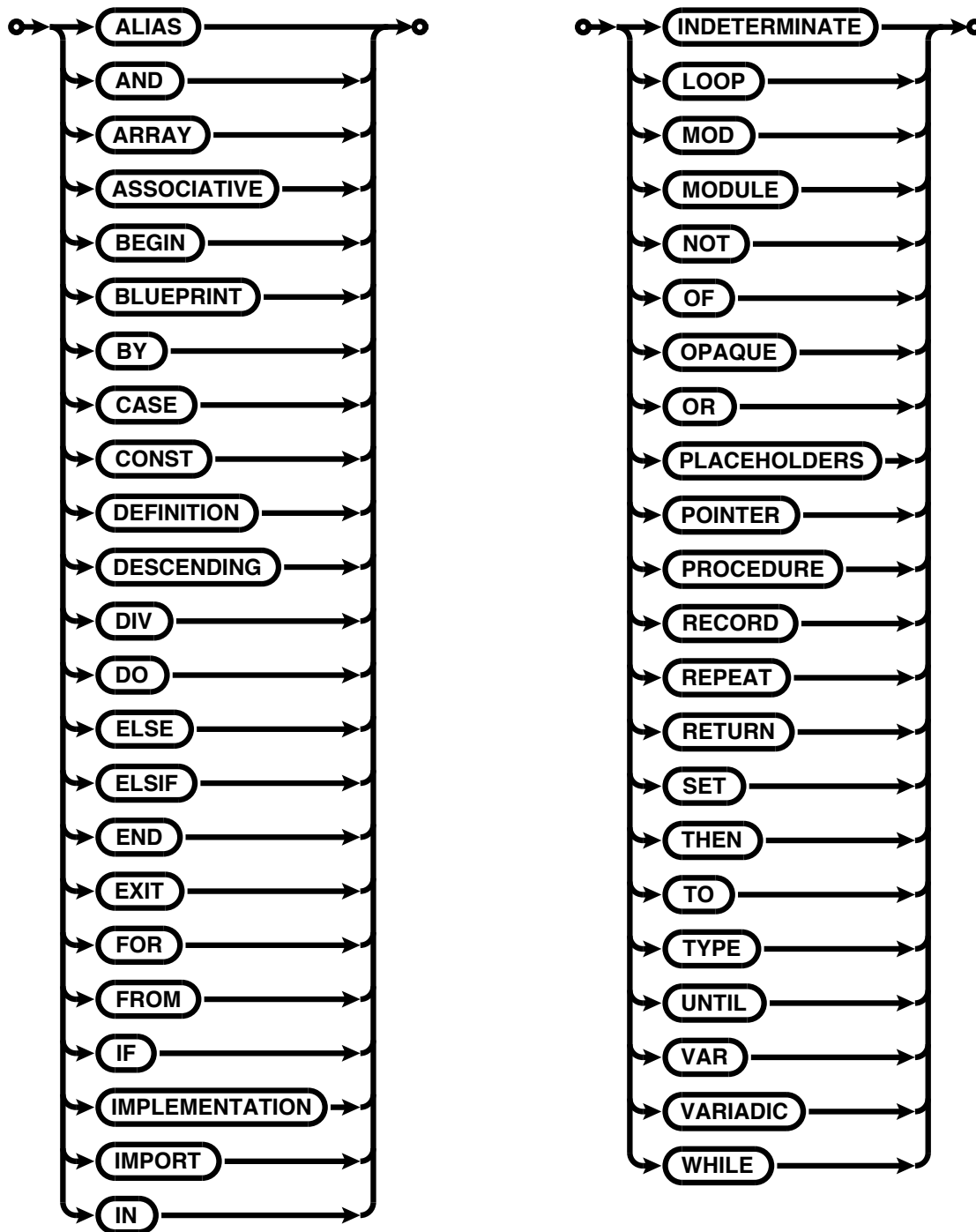
#57 Structured Value



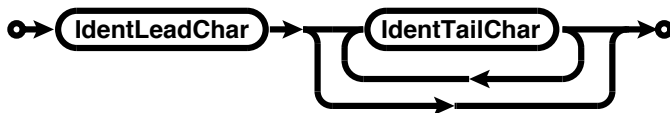
#58 Value Component**#58.1 Runtime Expression****#59 Qualified Identifier****#60 Identifier List**

B.2 Terminal Symbols

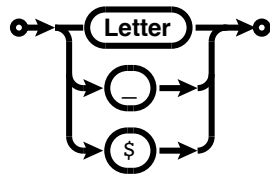
#1 Reserved Words



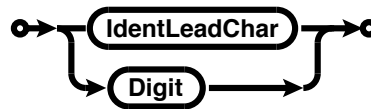
#2 Identifier



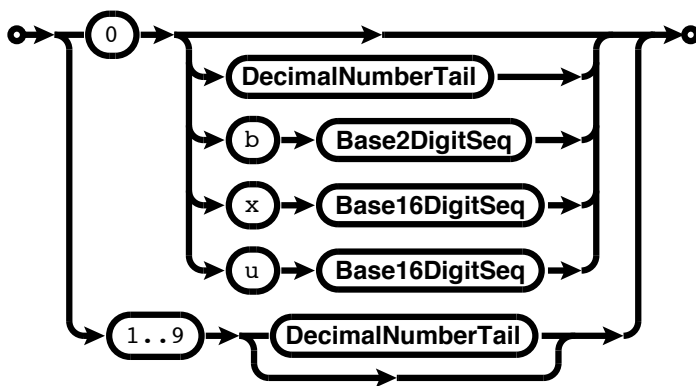
#2.1 Identifier Leading Character



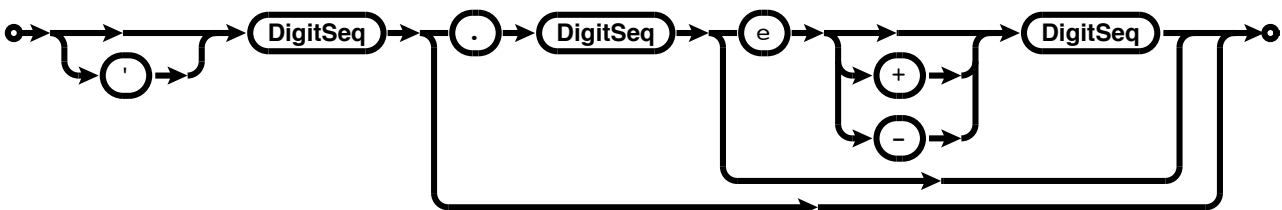
#2.2 Identifier Tail Character



#3 Numeric Literal



#3.1 Decimal Number Tail



#3.2 Digit Sequence



#3.2b Digit Group



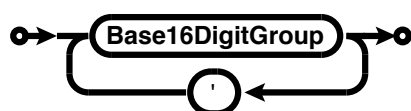
#3.3 Base-2 Digit Sequence



#3.3b Base-2 Digit Group

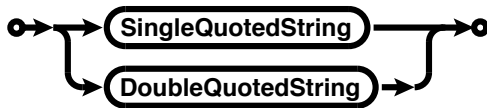
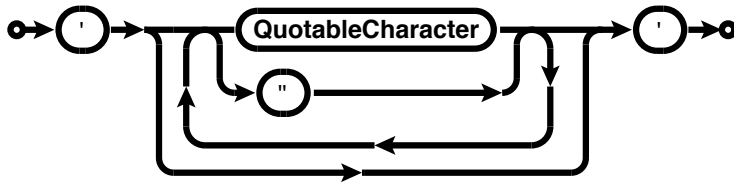
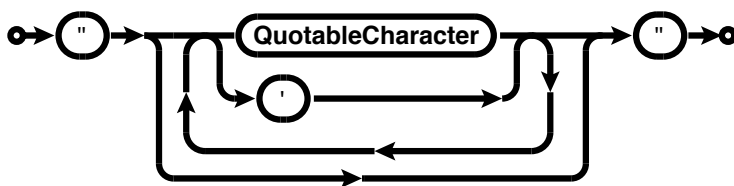
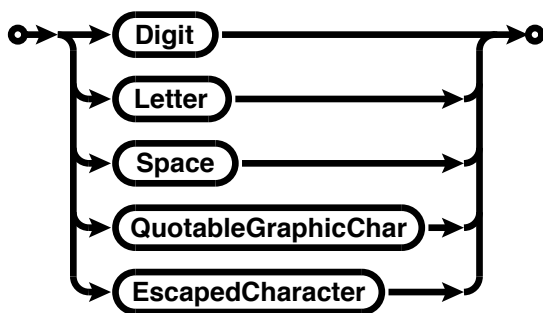
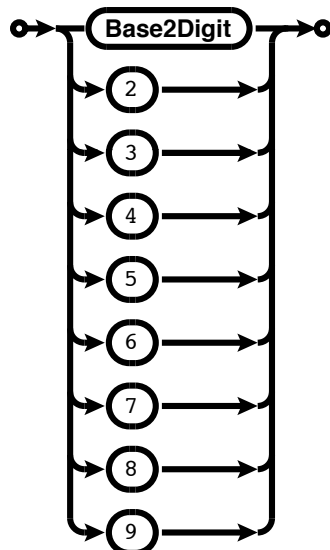
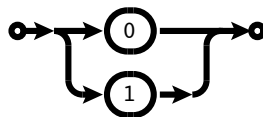
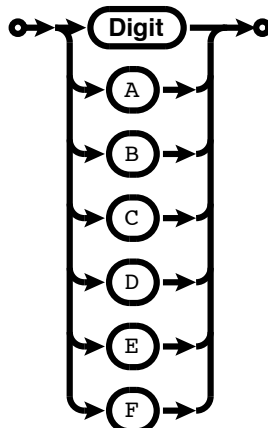
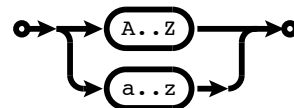


#3.4 Base-16 Digit Sequence

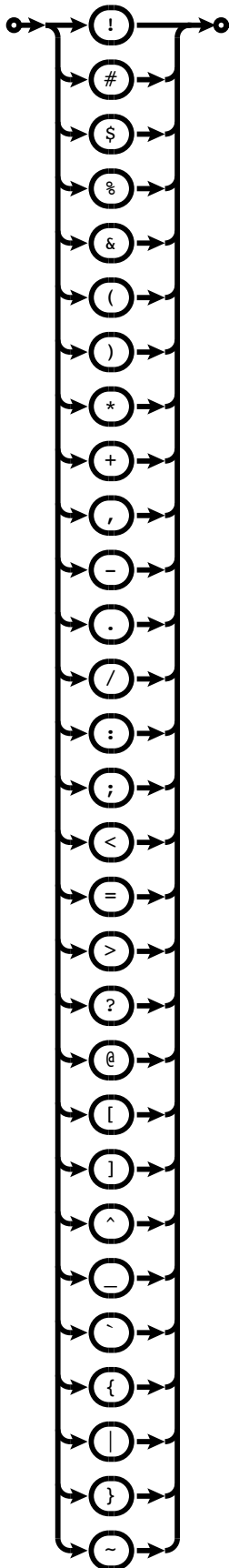
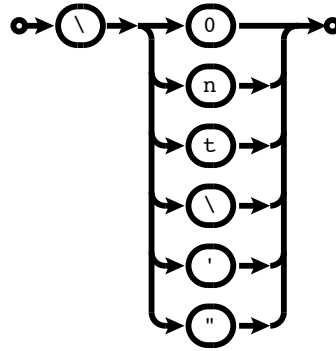


#3.4b Base-16 Digit Group



#4 String Literal**#4.1 Single Quoted String****#4.2 Double Quoted String****#4.3 Quotable Character****#4.4 Digit****#4.5 Base-2 Digit****#4.6 Base-16 Digit****#4.7 Letter****#4.8 Space**

CONST Space = CHR(32)

#4.9 Quotable Graphic Character**#4.10 Escaped Character**

B.3 Ignore Symbols

#1 Whitespace



#1.1 ASCII Tabulator

```
CONST ASCII_TAB = CHR(8)
```

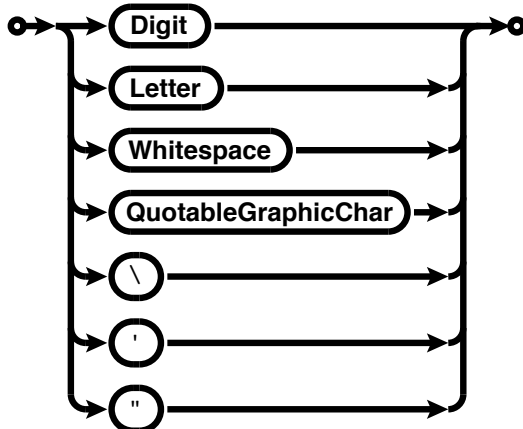
#2 Single-line Comment



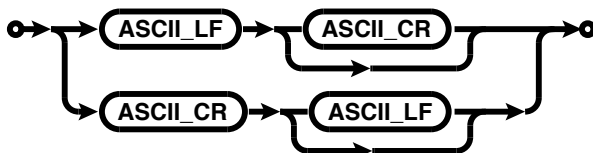
#3 Multi-line Comment



#3.1 Comment Character



#4 End Of Line Marker



#4.1 ASCII Line Feed

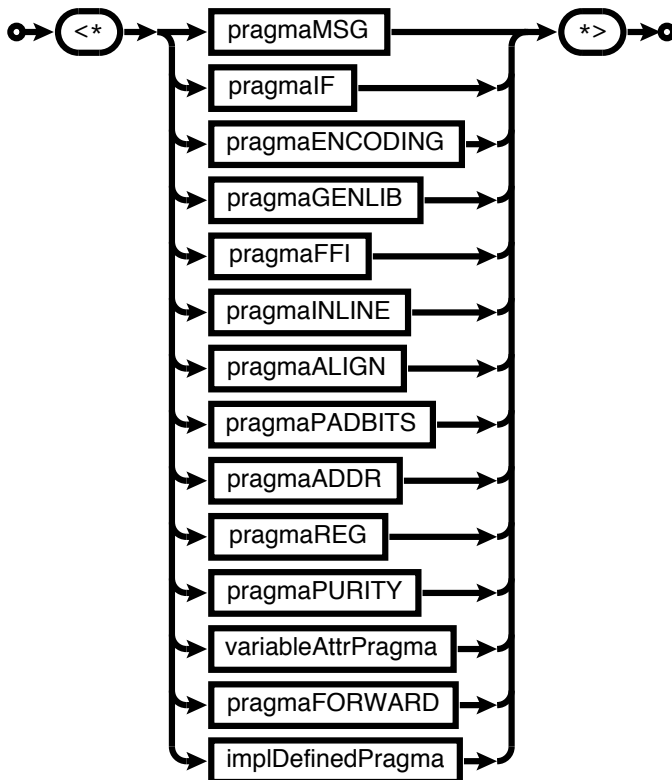
```
CONST ASCII_LF = CHR(10)
```

#4.2 ASCII Carriage Return

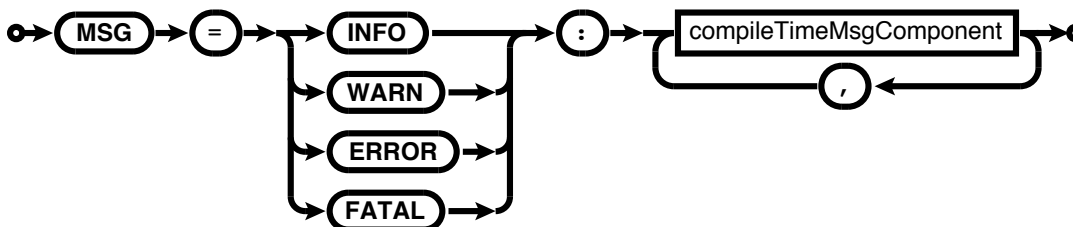
```
CONST ASCII_CR = CHR(13)
```


B.4 Pragma Grammar

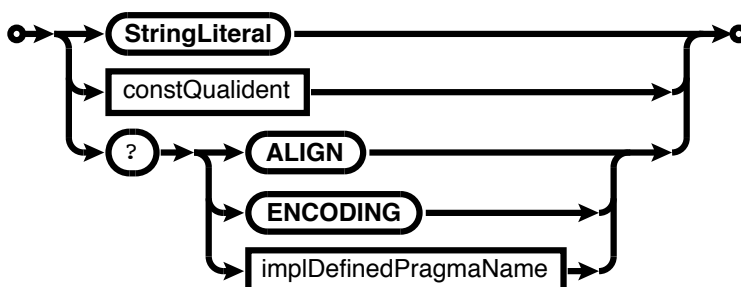
#1 Pragma



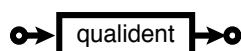
#2 Body Of Compile Time Message Pragma



#3 Compile Time Message Component

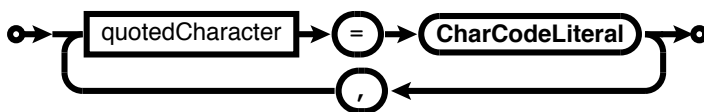
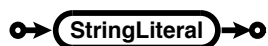
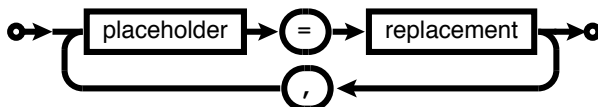
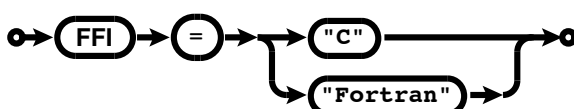
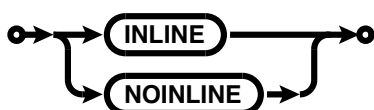
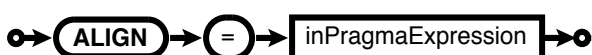


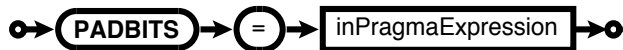
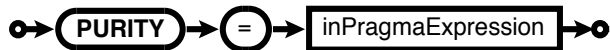
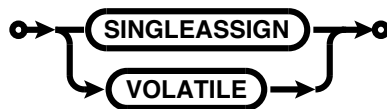
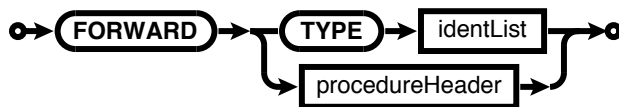
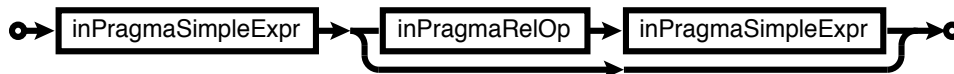
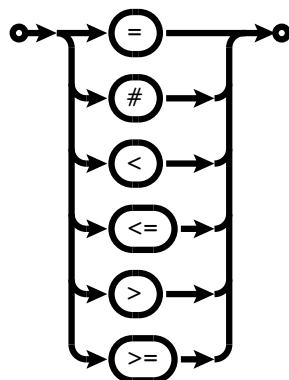
#3.1 Constant Qualified Identifier

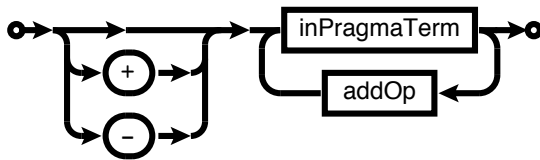
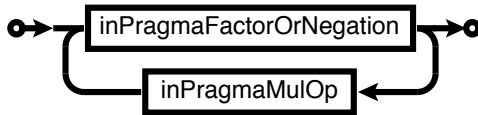
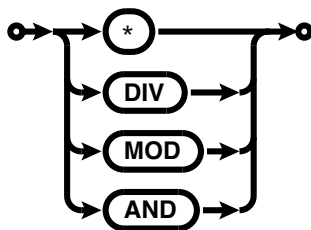
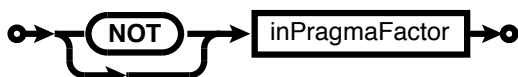
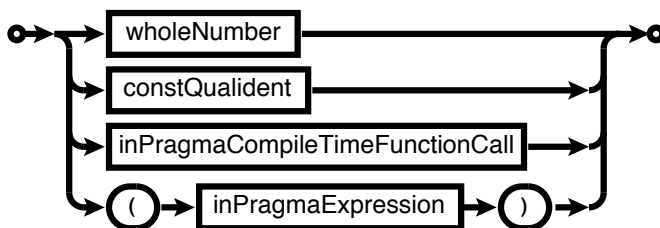


#3.2 Implementation Defined Pragma Name



#4 Body Of Conditional Compilation Pragma**#5 Body Of Character Encoding Pragma****#6 Code Point Sample List****#6.1 Quoted Character****#6.2 Character Code Literal****#7 Body Of Library Template Expansion Pragma****#8 Template Parameter List****#8.1 Placeholder****#8.2 Replacement****#9 Body Of Foreign Function Interface Pragma****#10 Body Of Procedure Inlining Pragma****#11 Body Of Memory Alignment Pragma**

#12 Body Of Bit Padding Pragma**#13 Body Of Memory Mapping Pragma****#14 Body Of Register Mapping Pragma****#15 Body Of Purity Attribute Pragma****#16 Body Of Variable Attribute Pragma****#17 Body Of Forward Declaration Pragma****#18 Body Of Implementation Defined Pragma****#19 In-Pragma Expression****#19.1 In-Pragma Relational Operator**

#20 In-Pragma Simple Expression**#21 In-Pragma Term****#21.1 In-Pragma Multiply Operator****#22 In-Pragma Factor Or Negation****#23 In-Pragma Factor****#23.1 Whole Number****#24 In-Pragma Compile-Time Function Call**