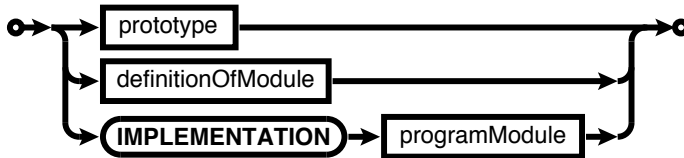


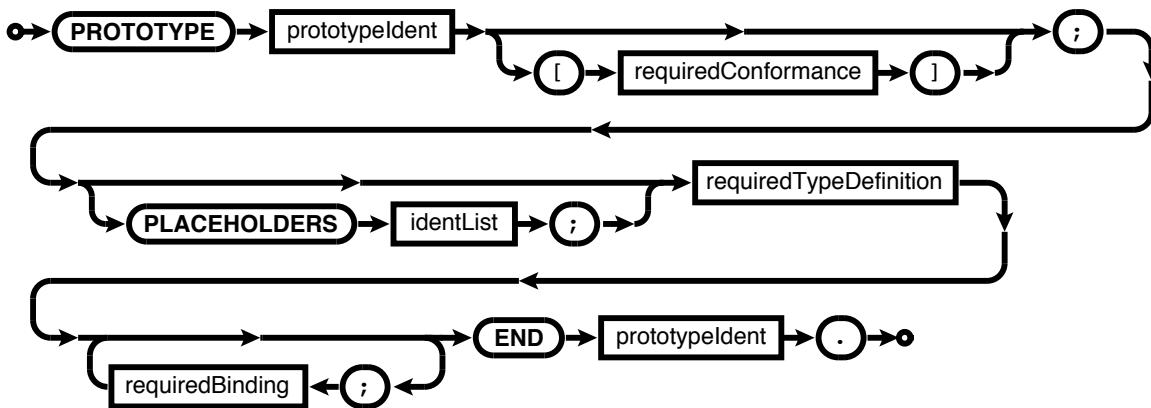
Appendix B: Syntax Diagrams

B.1 Non-Terminal Symbols

#1 Compilation Unit



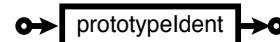
#2 Prototype Definition



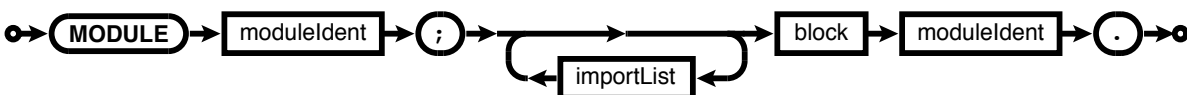
#2.1 Prototype Identifier



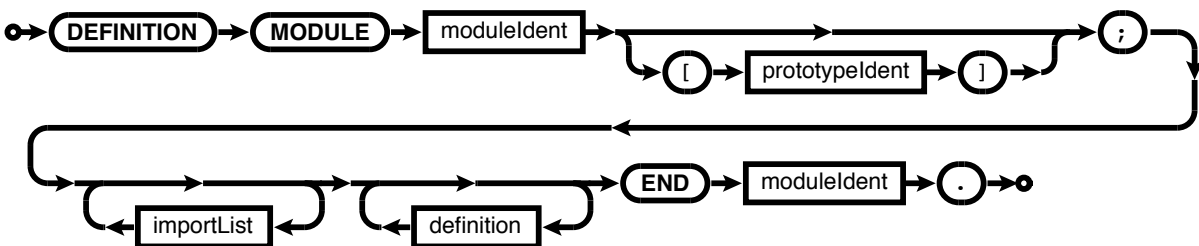
#2.2 Required Conformance



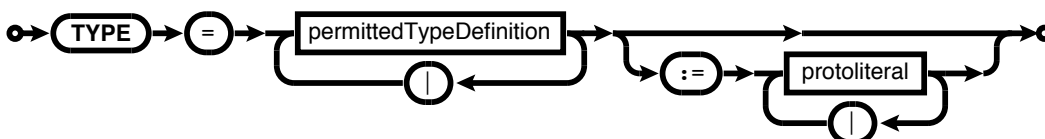
#3 Program Module

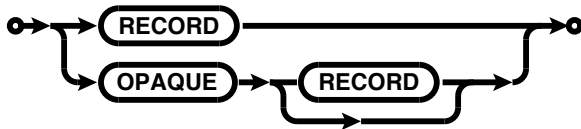
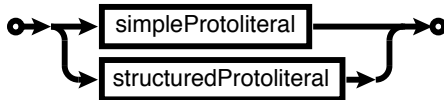
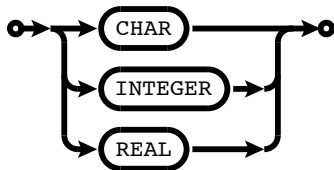
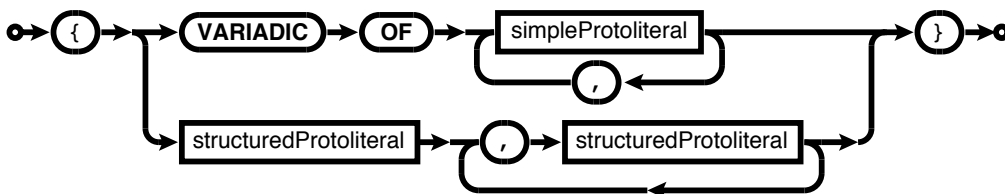
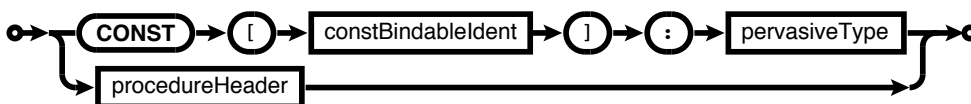
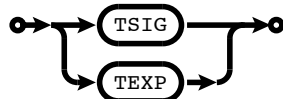
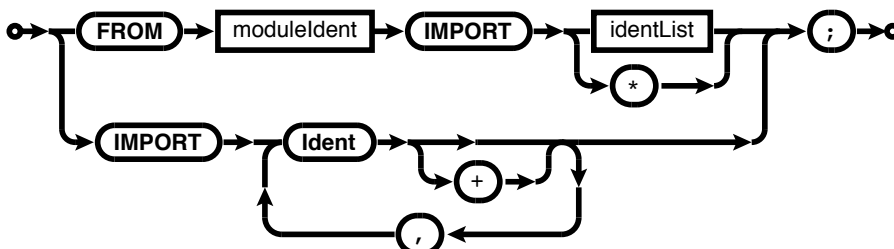
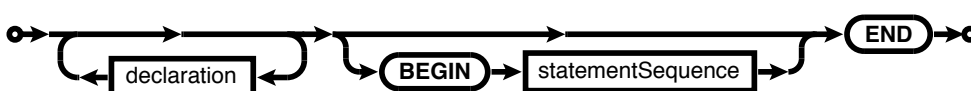


#4 Definition Of Module

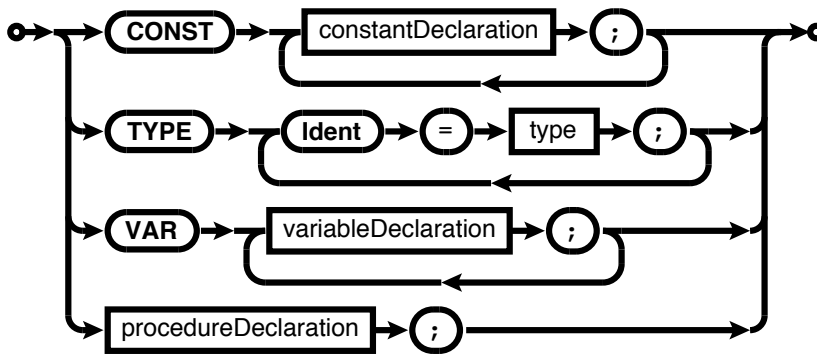


#5 Required Type Definition

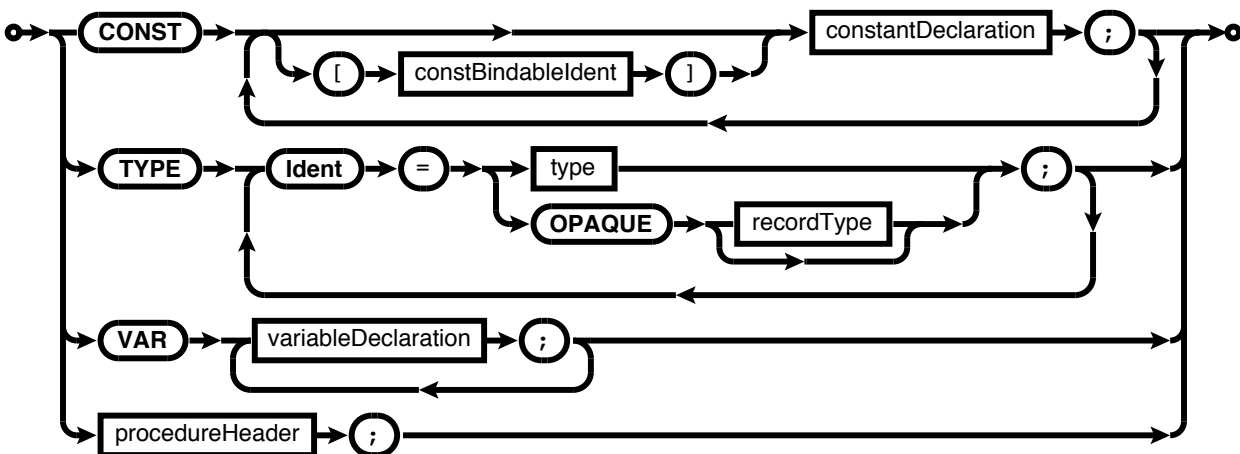


#6 Permitted Type Definition**#7 Proto-Literal****#7.1 Simple Proto-Literal****#8 Structured Proto-Literal****#9 Required Binding****#9.1 CONST Bindable Identifier****#9.2 Pervasive Type****#10 Import List****#11 Block**

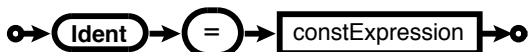
#12 Declaration



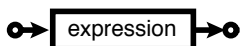
#13 Definition



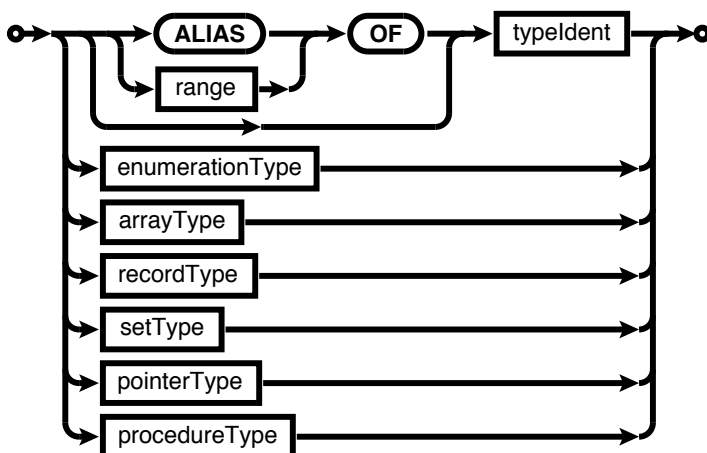
#14 Constant Declaration



#14.1 Constant Expression



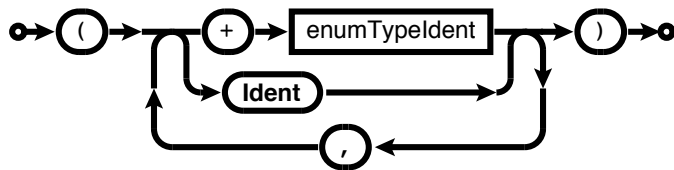
#15 Type



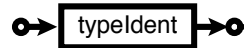
#16 Range



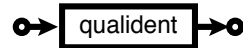
#17 Enumeration Type



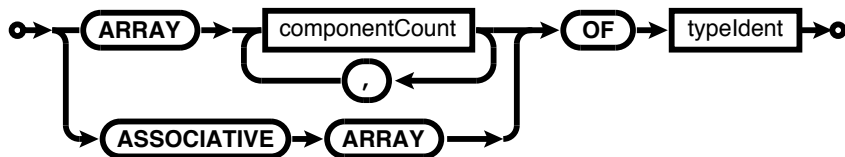
#17.1 Enumeration Type Identifier



#17.2 Type Identifier



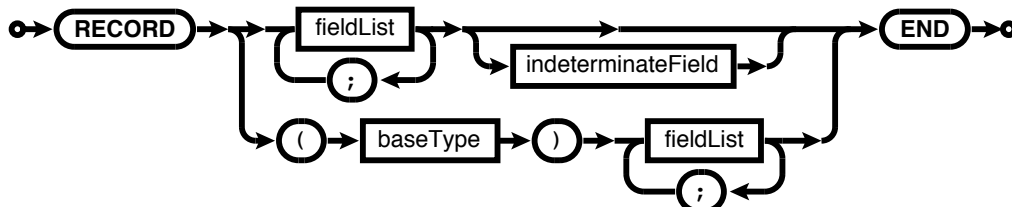
#18 Array Type



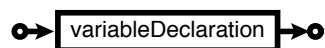
#18.1 Component Count



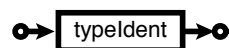
#19 Record Type



#19.1 Field List



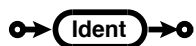
#19.2 Base Type



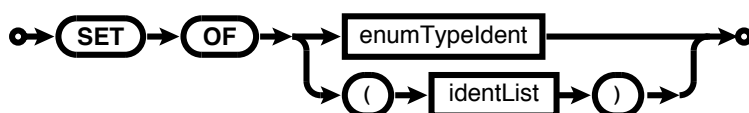
#20 Indeterminate Field



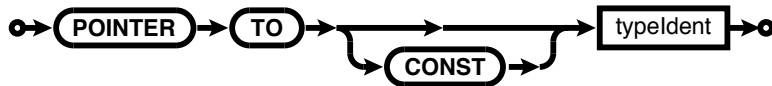
#20.1 Discriminant Field



#21 Set Type



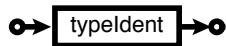
#22 Pointer Type



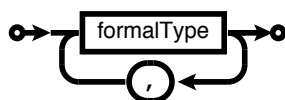
#23 Procedure Type



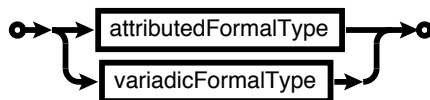
#23.1 Returned Type



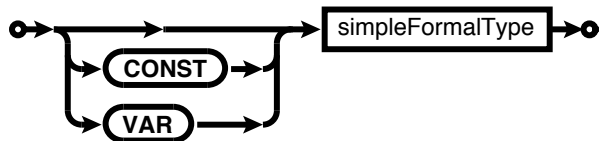
#24 Formal Type List



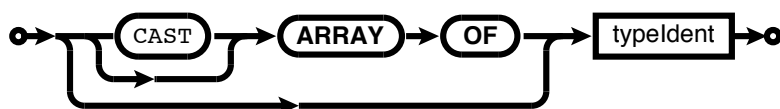
#25 Formal Type



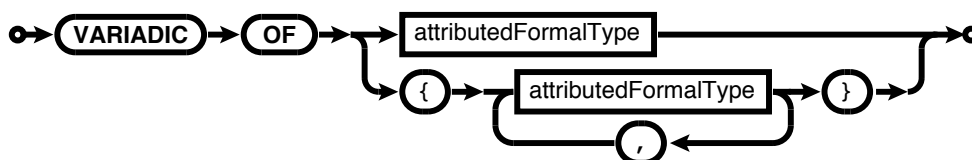
#26 Attributed Formal Type



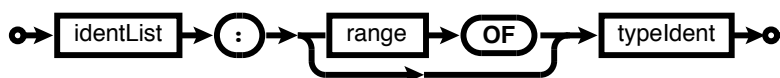
#27 Simple Formal Type



#28 Variadic Formal Type

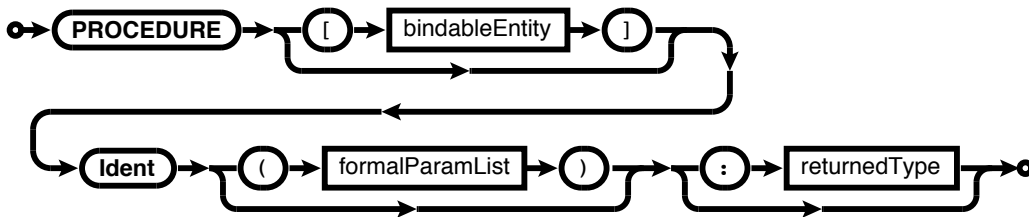
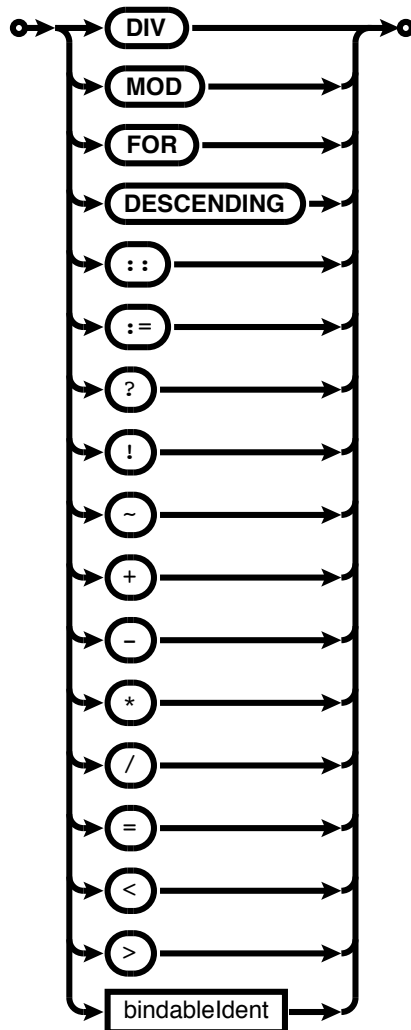
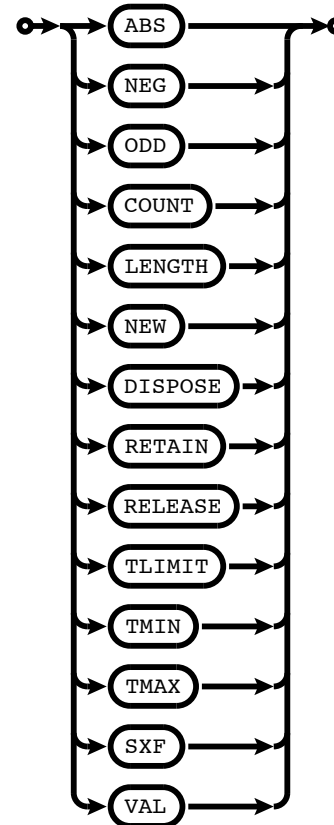
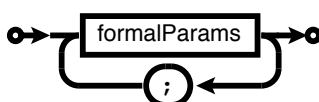
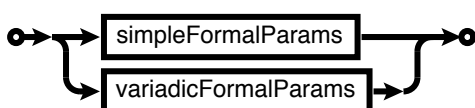


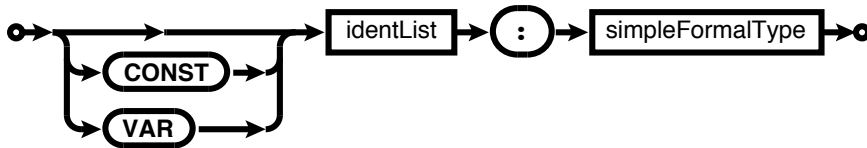
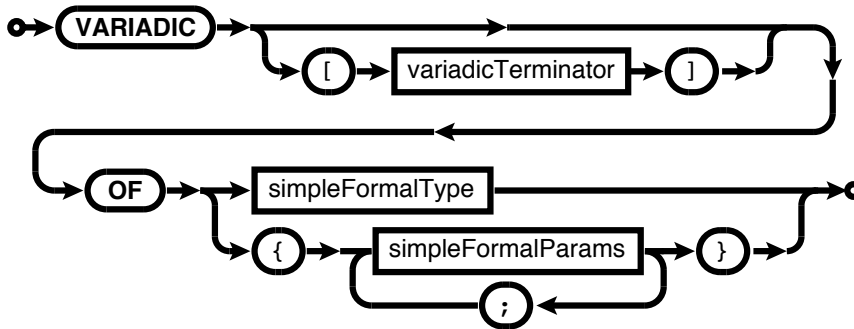
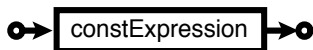
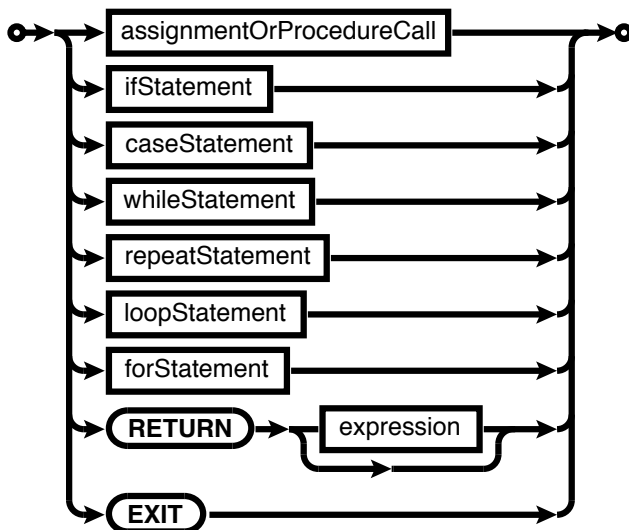
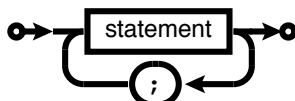
#29 Variable Declaration

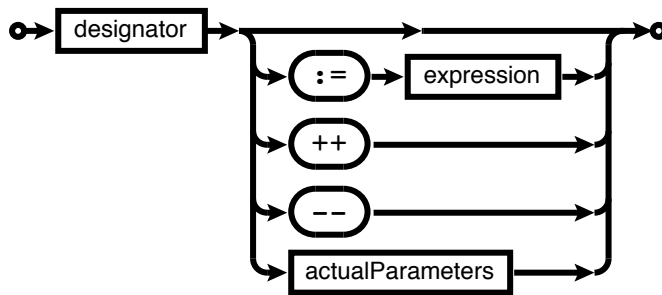
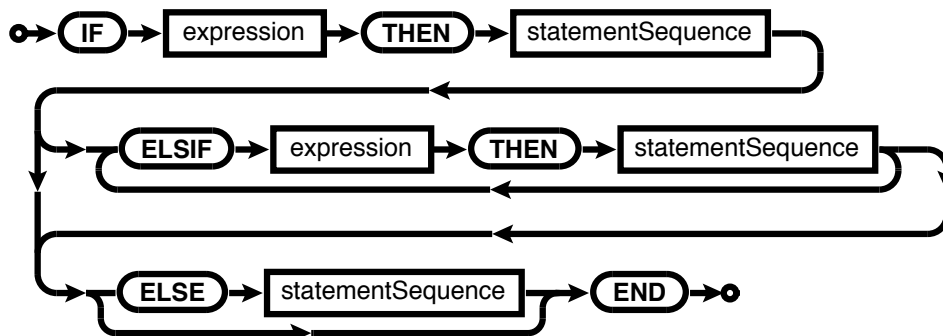
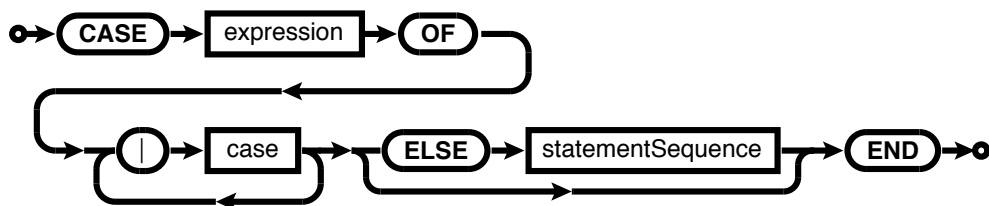
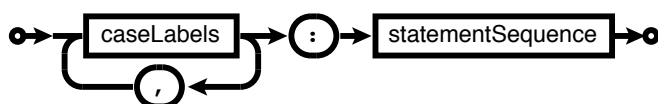
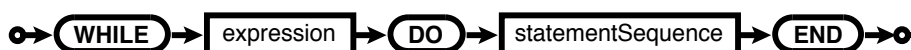
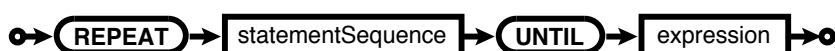
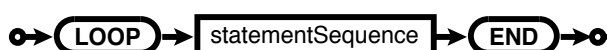


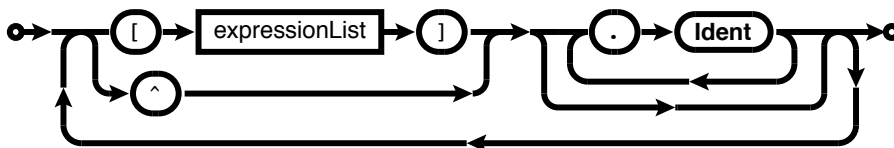
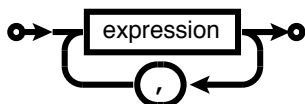
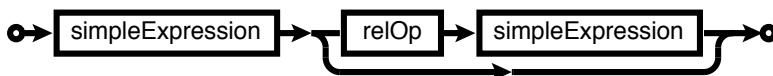
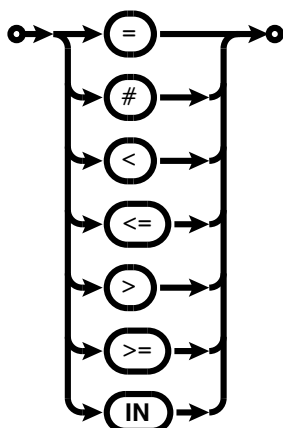
#30 Procedure Declaration

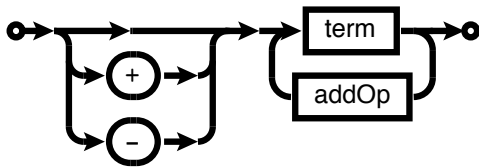
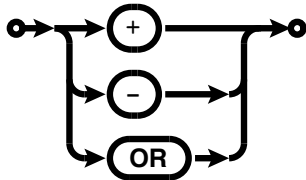
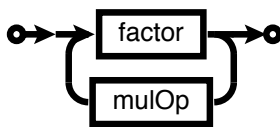
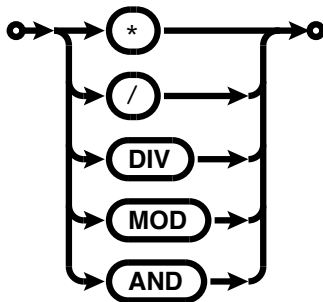
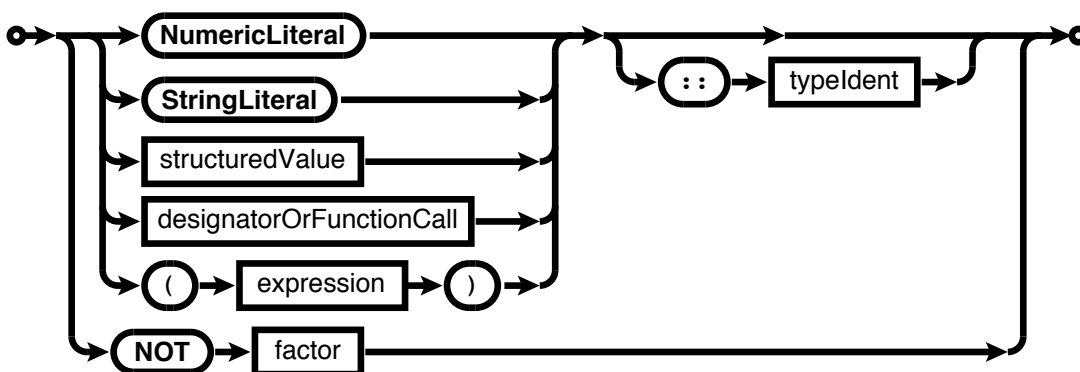
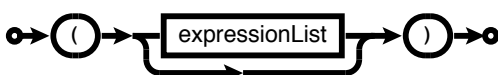


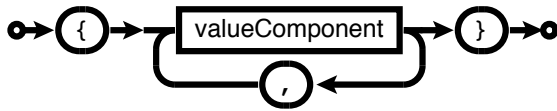
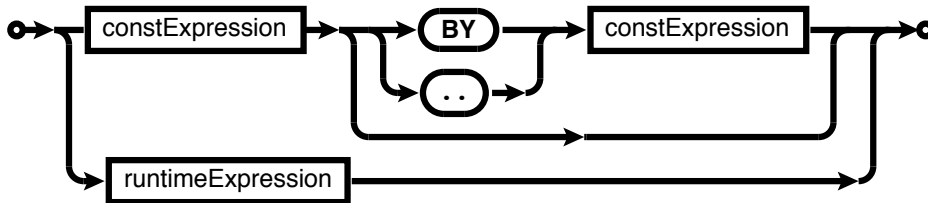
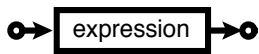
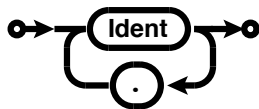
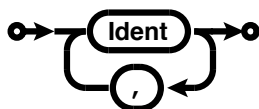
#31 Procedure Header**#32 Bindable Entity****#32.1 Bindable Identifier****#33 Formal Parameter List****#34 Formal Parameters**

#35 Simple Formal Parameters**#36 Variadic Formal Parameters****#36.1 Variadic Terminator****#37 Statement****#38 StatementSequence**

#39 Assignment Or Procedure Call**#40 IF Statement****#41 CASE Statement****#42 Case****#43 Case Labels****#44 WHILE Statement****#45 REPEAT Statement****#46 LOOP Statement**

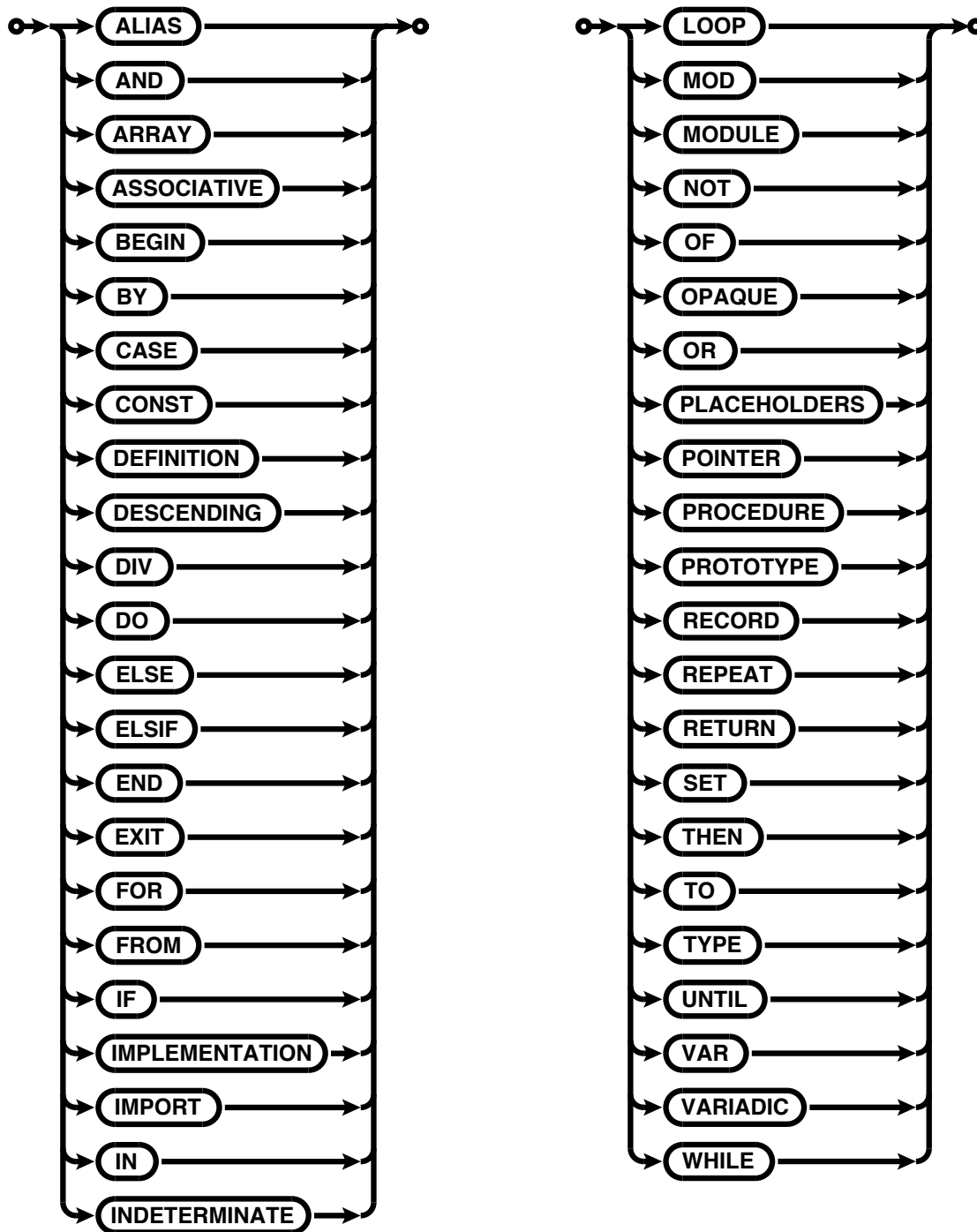
#47 FOR Statement**#47.1 Control Variable****#48 Designator****#49 Designator Tail****#50 Expression List****#51 Expression****#51.1 Relational Operator**

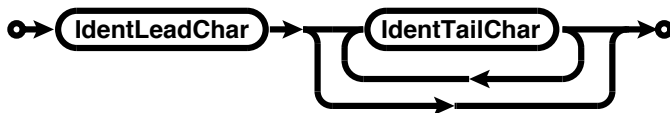
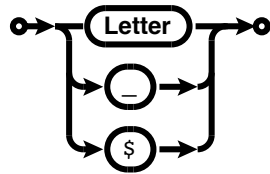
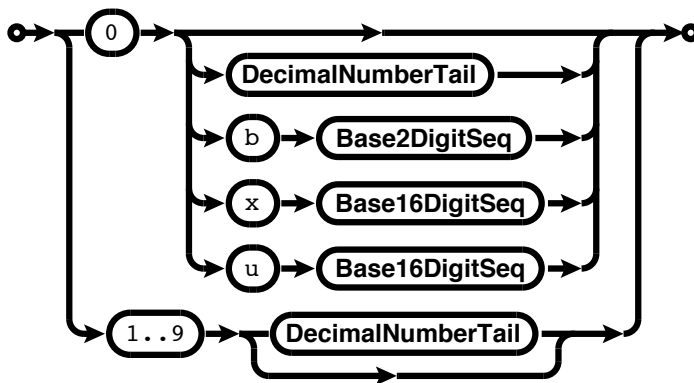
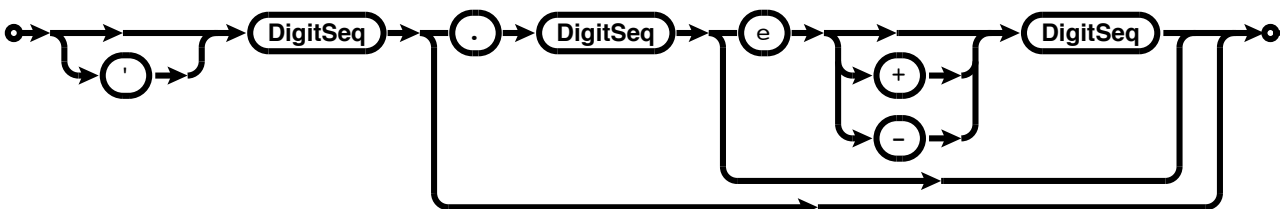
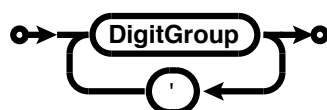
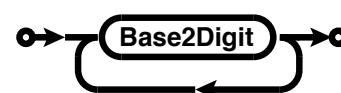
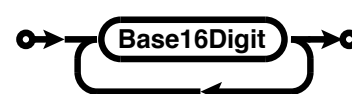
#52 Simple Expression**#52.1 Add Operator****#53 Term****#53.1 Multiply Operator****#54 Factor****#55 Designator Or Function Call****#56 Actual Parameters**

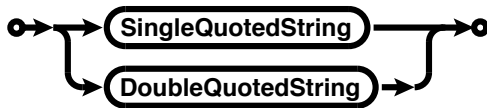
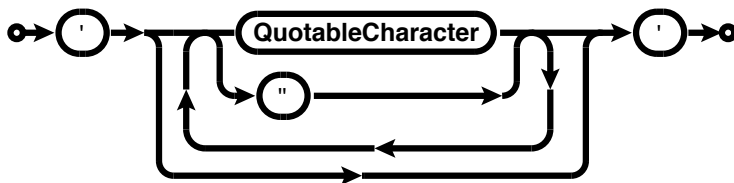
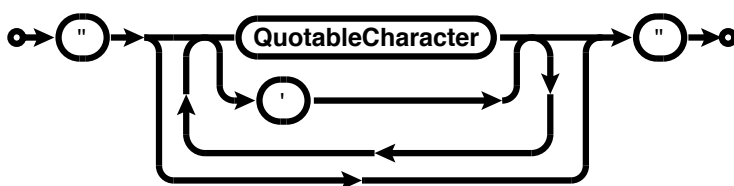
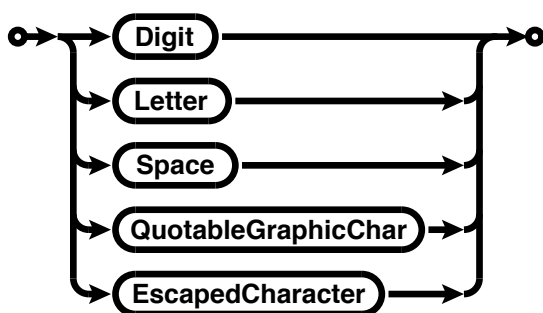
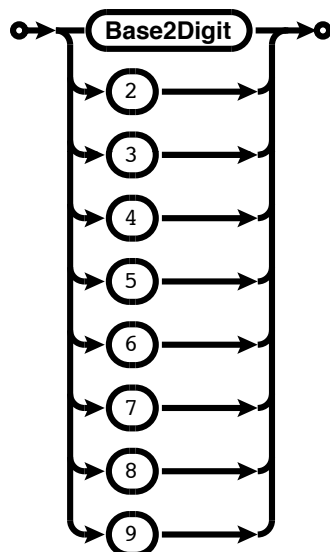
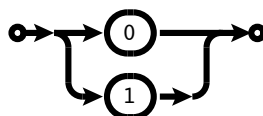
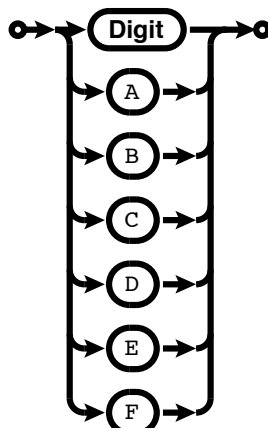
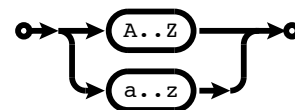
#57 Structured Value**#58 Value Component****#58.1 Runtime Expression****#59 Qualified Identifier****#60 Identifier List**

B.2 Terminal Symbols

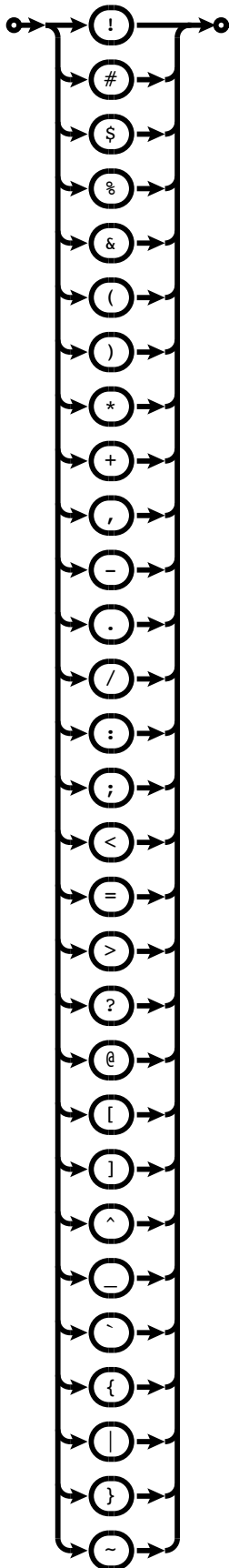
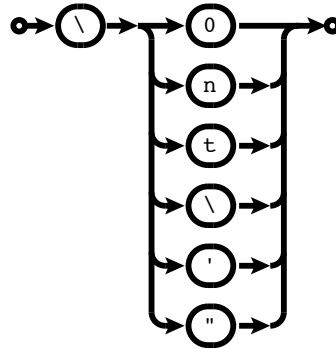
#1 Reserved Words



#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)
```

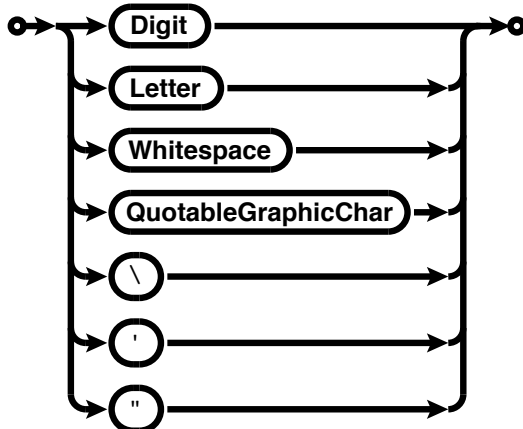
#2 Single-line Comment



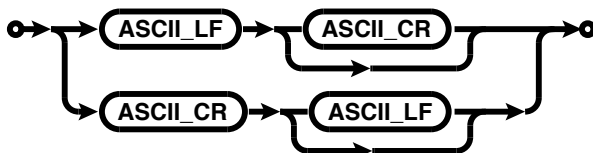
#3 Multi-line Comment



#3.1 Comment Character



#4 End Of Line Marker



#4.1 ASCII Line Feed

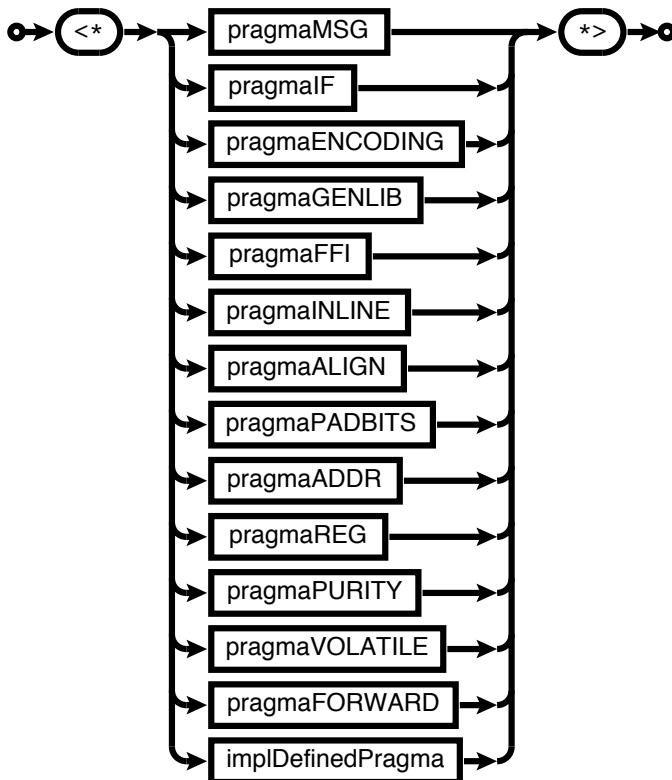
```
CONST ASCII_LF = CHR(10)
```

#4.2 ASCII Carriage Return

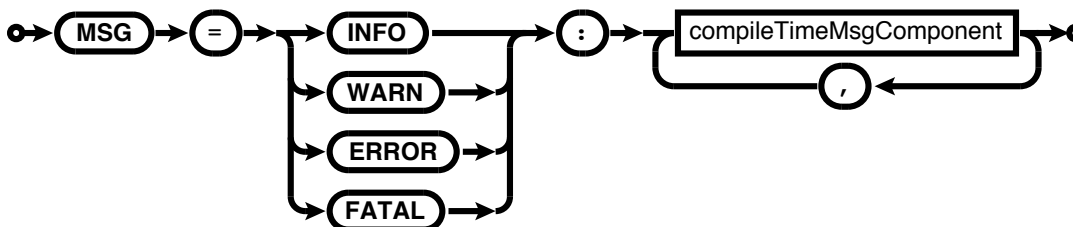
```
CONST ASCII_CR = CHR(13)
```


B.4 Pragma Grammar

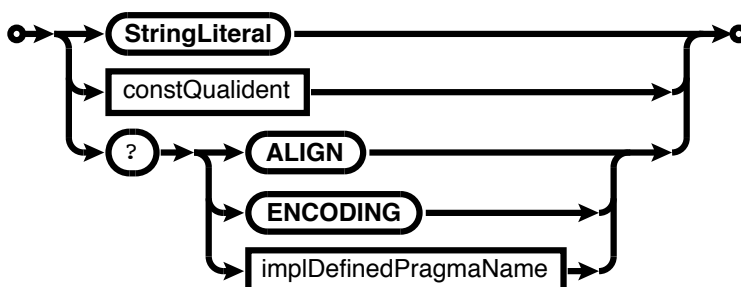
#1 Pragma



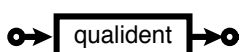
#2 Body Of Compile Time Message Pragma



#3 Compile Time Message Component

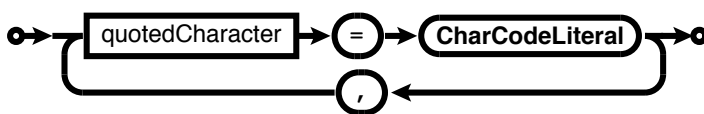
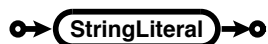
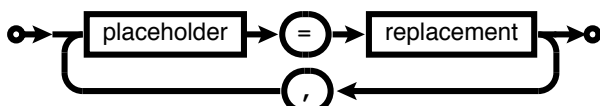
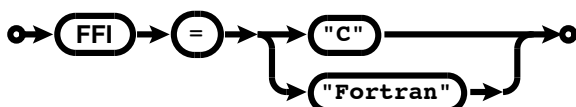
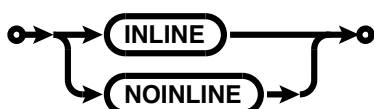


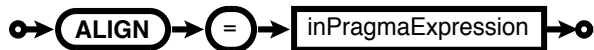
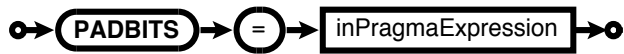
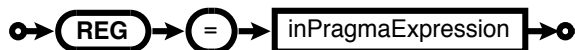
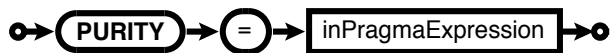
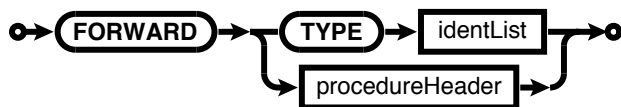
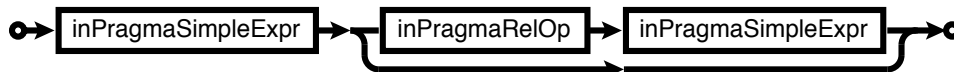
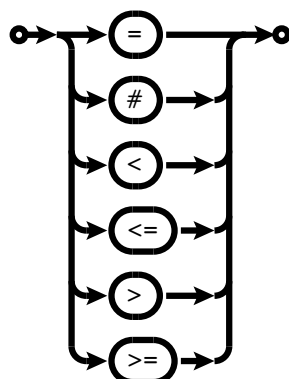
#3.1 Constant Qualified Identifier

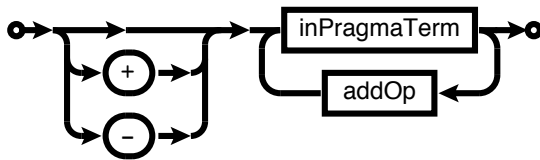
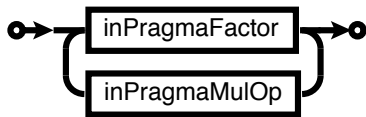
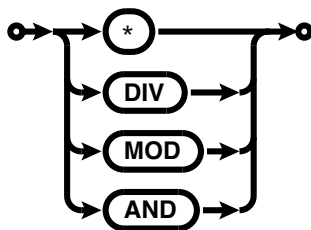
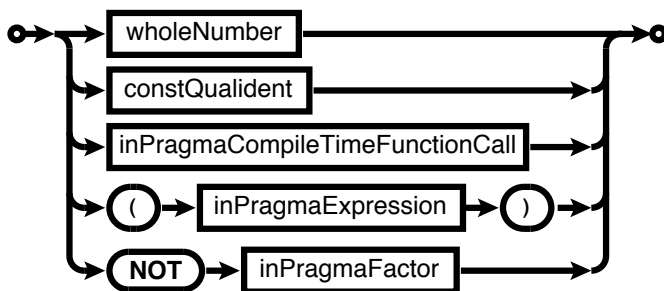


#3.2 Implementation Defined Pragma Name



#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 Volatile 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****#22.1 Whole Number****#23 In-Pragma Compile-Time Function Call**