<AFFIRM>PARSER..11 10-Jul-81 18:09:14

```
(FILECREATED " 4-Jul-81 20:13:48" <AFFIRM>PARSER..11 127843 )

(PRETTYCOMPRINT PARSERCOMS)

(RPAQQ PARSERCOMS ((* Parser PARSER defined by GOP. This parser requires an input generator Handle,
                      the name of the non-terminal symbol which the parser will attempt to match, and
                      a boolean which says whether the input must match entirely
                      (the generator must have terminated)
                      or whether a parse tree handle will suffice. The resulting parse tree may be
                      accessed via the records defined with the same names as the nonterminals at the
                      beginning of the listing.)
        (FNS * PARSERFNS)
        (* Standard Parser Input Interface. This function must be compiled.)
        (* Universal record definitions)
        (RECORDS COMMONTYPE COMMONSUBTYPE all arrayType assertStatement assertion assignmentStatement
                 assumeStatement block bracketExprList caseElementList caseLabel caseStatement
                 compoundStatement concurrentAssignmentStatement constDefinition constant coord
                 declareType declareopt denotePair denoteSpec direction expression expressionSeq
                 fieldList fileType firstOne forStatement formalParameterSection functionDecl
                 goToStatement greaterThanEqual identifier identifierSeq ifExpr ifStatement infixOp
                 interfaceList label labelStatement lastOne lessThanEqual machinePair machineSpec
                 normalInfixOp notEqual number op packed parameterGroup parameterKind parenExpr
                 pointerType prefixExpr prefixOp primary procedureOrFunctionDeclaration
                 procedureStatement program proveStatement qualifier quantifiedExpression quantifier
                 range rangeSpec rangedInterfaceList rangedOp recordSection recordType
                 repeatStatement returnStatement rule ruleSeq scalarType setType simpleStatement
                 simpleType specialPrefixExpr specialPrefixOp statement structuredType subrangeType
                 type typeDefinition typeIdentifier unitKind unpackedStructuredType varDeclaration
                 varDeclarePart variable variableDecl variant variantPart whileStatement
                 withStatement)
        (* Macro used to generate input values of NIL when generator halts.)
        (VARS (DWIMIFYCOMPFLG T))
        (* The filter file must contain the following functions: identifierFilter infixOpFilter
           unsignedInteger prefixOpFilter specialPrefixOpFilter labelFilter simplifyStatement)
        (* If the user redefines the production records, currently represented as TYPERECORDS, he
           must redefine the CopyTopRecord function and compiler macro consistent with it.)
        (* The following LOAD and definition should normally be replaced by a load from a file
           defining UsersNextInput and an input initialization routine)
        (* The user might consider eliminating some of the BLKAPPLYFNS if he can predict the roots
           which might be called externally.)
        (BLOCKS (PARSERBLK PARSER a0062 a0063 a0064 a0065 a0066 a0067 a0068 a0069 a0070 a0071 a0072
                           a0073 a0074 a0075 a0076 a0077 a0078 a0079 a0080 a0081 a0082 a0083 a0084
                           a0085 a0086 a0087 a0088 a0089 a0090 a0091 a0092 a0093 a0094 a0095 a0096
                           a0097 a0098 a0099 a0100 a0101 a0102 a0103 a0104 a0105 a0106 a0107 a0108
                           a0109 a0110 a0111 a0112 a0113 a0114 a0115 a0116 a0117 a0118 a0119 a0120
                           a0121 a0122 a0123 all arrayType assertStatement assertion
                           assignmentStatement assumeStatement block bracketExprList caseElementList
                           caseLabel caseStatement compoundStatement concurrentAssignmentStatement
                           constDefinition constant coord declareType declareopt denotePair
                           denoteSpec direction expression expressionSeq fieldList fileType firstOne
                           forStatement formalParameterSection functionDecl goToStatement
                           greaterThanEqual identifier identifierSeq ifExpr ifStatement infixOp
                           interfaceList label labelStatement lastOne lessThanEqual machinePair
                           machineSpec normalInfixOp notEqual number op packed parameterGroup
                           parameterKind parenExpr pointerType prefixExpr prefixOp primary
                           procedureOrFunctionDeclaration procedureStatement program proveStatement
                           qualifier quantifiedExpression quantifier range rangeSpec
                           rangedInterfaceList rangedOp recordSection recordType repeatStatement
                           returnStatement rule ruleSeq scalarType setType simpleStatement simpleType
                           specialPrefixExpr specialPrefixOp statement structuredType subrangeType
                           type typeDefinition typeIdentifier unitKind unpackedStructuredType
                           varDeclaration varDeclarePart variable variableDecl variant variantPart
                           whileStatement withStatement CopyTopRecord MatchConstant MatchLexeme
                           PARSEPROGRAM PARSE\ASSERTION ParserRATOM ReadAtom SIFromStack? SIMark
                           SINewStack SINext SINextNew SINextSaved SISaveLexeme SIToMark SIUnmark
                           UsersNextInput identifierFilter infixOpFilter labelFilter prefixOpFilter
                           simplifyStatement specialPrefixOpFilter unsignedInteger
                           (ENTRIES PARSER PARSEPROGRAM PARSE\ASSERTION)
                           (SPECVARS CurrentRecord InputFileHandle CurrentLexeme ProductionValue
                                     Terminator SIEntries OutputStream)
                           (BLKAPPLYFNS program all bracketExprList coord denoteSpec denotePair
                                        expression expressionSeq firstOne functionDecl
                                        greaterThanEqual identifier identifierSeq ifExpr infixOp
                                        lastOne lessThanEqual machinePair machineSpec normalInfixOp
                                        notEqual number op parenExpr prefixExpr prefixOp primary
                                        quantifiedExpression quantifier range rangedInterfaceList
                                        rangedOp rangeSpec rule ruleSeq specialPrefixExpr
                                        specialPrefixOp variable variableDecl interfaceList arrayType
                                        assertion assertStatement assignmentStatement assumeStatement
                                        block caseElementList caseLabel caseStatement
```

```
                                    compoundStatement concurrentAssignmentStatement constant
                                    constDefinition declareopt declareType direction fieldList
                                    fileType formalParameterSection forStatement gotoStatement
                                    ifStatement label labelStatement packed parameterGroup
                                    parameterKind pointerType procedureOrFunctionDeclaration
                                    procedureStatement proveStatement qualifier recordSection
                                    recordType repeatStatement returnStatement scalarType setType
                                    simpleStatement simpleType statement structuredType
                                    subrangeType type typeDefinition typeIdentifier unitKind
                                    unpackedStructuredType varDeclaration varDeclarePart variant
                                    variantPart whileStatement withStatement)))
        (FNS * PARSERPLUSINS)
        (P (CLISPDEC (QUOTE FAST)))
        (DECLARE: DOEVAL@COMPILE
                  (PROP MACRO CopyTopRecord SIFromStack? SIMark SINewStack SINext SINextNew
                        SINextSaved SISaveLexeme SIToMark SIUnmark UsersNextInput labelFilter))
        (VARS Delimiters KeyWordList ReserveWordList SpecialPrefixOps UCaseParseAtoms UpperCaseVars)
        (VARS (PARSERPROMPT (QUOTE ~)))
              (USESLOWERCASE T)
              (PARSERTRACE NIL)
              (COLLECTTOKENS T))
        (P (SETQ PASCAL\READ\TABLE (COPYREADTABLE (QUOTE ORIG)))
           (SETBRK (QUOTE (4 5 6 14 16 17 18 19 20 21 27 28 29 34 30 33 38 40 41 42 43 44 45 46 47 58
                             59 60 61 62 64 91 93 94 123 124 125 126))
                   NIL PASCAL\READ\TABLE))))
[DECLARE: DONTEVAL@LOAD DONTCOPY
```

```
(* Parser PARSER defined by GOP. This parser requires an input generator Handle, the name of the
   non-terminal symbol which the parser will attempt to match, and a boolean which says whether the
   input must match entirely (the generator must have terminated)
   or whether a parse tree handle will suffice. The resulting parse tree may be accessed via the
   records defined with the same names as the nonterminals at the beginning of the listing.)

(RPAQQ PARSERFNS (PARSER a0062 a0063 a0064 a0065 a0066 a0067 a0068 a0069 a0070 a0071 a0072 a0073
                         a0074 a0075 a0076 a0077 a0078 a0079 a0080 a0081 a0082 a0083 a0084 a0085
                         a0086 a0087 a0088 a0089 a0090 a0091 a0092 a0093 a0094 a0095 a0096 a0097
                         a0098 a0099 a0100 a0101 a0102 a0103 a0104 a0105 a0106 a0107 a0108 a0109
                         a0110 a0111 a0112 a0113 a0114 a0115 a0116 a0117 a0118 a0119 a0120 a0121
                         a0122 a0123 all arrayType assertStatement assertion assignmentStatement
                         assumeStatement block bracketExprList caseElementList caseLabel
                         caseStatement compoundStatement concurrentAssignmentStatement
                         constDefinition constant coord declareType declareopt denotePair denoteSpec
                         direction expression expressionSeq fieldList fileType firstOne forStatement
                         formalParameterSection functionDecl goToStatement greaterThanEqual
                         identifier identifierSeq ifExpr ifStatement infixOp interfaceList label
                         labelStatement lastOne lessThanEqual machinePair machineSpec normalInfixOp
                         notEqual number op packed parameterGroup parameterKind parenExpr pointerType
                         prefixExpr prefixOp primary procedureOrFunctionDeclaration
                         procedureStatement program proveStatement qualifier quantifiedExpression
                         quantifier range rangeSpec rangedInterfaceList rangedOp recordSection
                         recordType repeatStatement returnStatement rule ruleSeq scalarType setType
                         simpleStatement simpleType specialPrefixExpr specialPrefixOp statement
                         structuredType subrangeType type typeDefinition typeIdentifier unitKind
                         unpackedStructuredType varDeclaration varDeclarePart variable variableDecl
                         variant variantPart whileStatement withStatement))
(DEFINEQ

                                                                                        1

(PARSER
  [LAMBDA (Root InputFileHandle ForceTermination?)
    (PROG (CurrentLexeme SIEntries OutputStream ProductionValue (Terminator (CONS NIL NIL)))
          (SINewStack)
          (SINextNew)
          (RETURN (if (FMEMB Root
                             (QUOTE (program all bracketExprList coord denoteSpec denotePair
                                             expression expressionSeq firstOne functionDecl
                                             greaterThanEqual identifier identifierSeq ifExpr infixOp
                                             lastOne lessThanEqual machinePair machineSpec
                                             normalInfixOp notEqual number op parenExpr prefixExpr
                                             prefixOp primary quantifiedExpression quantifier range
                                             rangedInterfaceList rangedOp rangeSpec rule ruleSeq
                                             specialPrefixExpr specialPrefixOp variable variableDecl
                                             interfaceList arrayType assertion assertStatement
                                             assignmentStatement assumeStatement block
                                             caseElementList caseLabel caseStatement
                                             compoundStatement concurrentAssignmentStatement constant
                                             constDefinition declareopt declareType direction
                                             fieldList fileType formalParameterSection forStatement
                                             goToStatement ifStatement label labelStatement packed
                                             parameterGroup parameterKind pointerType
                                             procedureOrFunctionDeclaration procedureStatement
                                             proveStatement qualifier recordSection recordType
                                             repeatStatement returnStatement scalarType setType
                                             simpleStatement simpleType statement structuredType
                                             subrangeType type typeDefinition typeIdentifier unitKind
                                             unpackedStructuredType varDeclaration varDeclarePart
                                             variant variantPart whileStatement withStatement)))
                      then (PROG ((Value (BLKAPPLY Root NIL)))
                                 (if ~ ForceTermination? or ForceTermination? and CurrentLexeme =
                                                                                        Terminator
                                     then (RETURN Value])

                                                                                        2

(a0062
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (ProductionValue ←(procedureOrFunctionDeclaration))
              then (PROGN (CurrentRecord : procedureOrFunctionDeclaration ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
```

```
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        [if (ProductionValue ←(block))
            then (PROGN (CurrentRecord : block ← ProductionValue)
                        (PROGN (SIUnmark)
                               (RETURN T]
        (SIToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (SIUnmark)
        (RETURN NIL])
```

3

```
(a0063
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (ProductionValue ←(expression))
              then (PROGN (do (CurrentRecord : expression ←(NCONC1 CurrentRecord : expression
                                                                   ProductionValue))
                              (SIMark) repeatwhile (AND (MatchConstant (QUOTE (.))
                                                                       NIL)
                                                        (ProductionValue ←(expression))
                                                        (PROGN (SIUnmark)
                                                               T)))
                          (SIToMark)
                          (SIUnmark)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SIUnmark)
                          (RETURN T)))
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SIUnmark)
          (RETURN NIL])
```

4

```
(a0064
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (ProductionValue ←(number))
              then (PROGN (CurrentRecord : number ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (MatchConstant (QUOTE (-))
                             NIL)
              then (if (ProductionValue ←(number))
                       then (PROGN (CurrentRecord : number# ← ProductionValue)
                                   (PROGN (SIUnmark)
                                          (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(all))
              then (PROGN (CurrentRecord : all ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(lastOne))
              then (PROGN (CurrentRecord : lastOne ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(firstOne))
              then (PROGN (CurrentRecord : firstOne ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SIUnmark)
          (RETURN NIL])
```

5

```
(a0065
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (ProductionValue ←(infixOp))
              then (PROGN (CurrentRecord : infixOp ← ProductionValue)
                          (if (ProductionValue ←(expression))
                              then (PROGN (CurrentRecord : expression ← ProductionValue)
                                          (PROGN (SIUnmark)
                                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SIUnmark)
                          (RETURN T)))
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SIUnmark)
          (RETURN NIL])
```

6

```
(a0066
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (MatchConstant (QUOTE (ELSE))
                             NIL)
              then (if (ProductionValue ←(expression))
                       then (PROGN (CurrentRecord : expression## ← ProductionValue)
                                   (PROGN (SIUnmark)
                                          (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SIUnmark)
                          (RETURN T)))
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SIUnmark)
          (RETURN NIL])
```

7

```
(a0067
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (ProductionValue ←(notEqual))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(lessThanEqual))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(greaterThanEqual))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(normalInfixOp))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SIUnmark)
          (RETURN NIL])
```

8

```
(a0068
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (BY))
                             NIL)
              then (if (ProductionValue ←(identifierSeq))
                       then (PROGN (CurrentRecord : identifierSeq# ← ProductionValue)
                                   (PROGN (SlUnmark)
                                          (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

9

```
(a0069
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (ProductionValue ←(identifierSeq))
              then (PROGN (CurrentRecord : identifierSeq ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

10

```
(a0070
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (~))
                             NIL)
              then (if (MatchConstant (QUOTE (=))
                                      NIL)
                       then (PROGN (SlUnmark)
                                   (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (MatchConstant (QUOTE (!))
                             NIL)
              then (if (MatchConstant (QUOTE (=))
                                      NIL)
                       then (PROGN (SlUnmark)
                                   (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

11

```
(a0071
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (ProductionValue ←(expression))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
```

```
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        [if (ProductionValue ←(infixOp))
            then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                        (PROGN (SIUnmark)
                               (RETURN T]
        (SIToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        [if (ProductionValue ←(specialPrefixOp))
            then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                        (PROGN (SIUnmark)
                               (RETURN T]
        (SIToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        [if (ProductionValue ←(prefixOp))
            then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                        (PROGN (SIUnmark)
                               (RETURN T]
        (SIToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (SIUnmark)
        (RETURN NIL])
```

12

```
(a0072
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (MatchConstant (QUOTE (IMPORTS))
                             NIL)
              then (if (MatchConstant (QUOTE (%())
                                      NIL)
                       then (if (ProductionValue ←(identifier))
                                then (PROGN (do (CurrentRecord : identifier ←(NCONC1 CurrentRecord :
                                                                                identifier
                                                                              ProductionValue))
                                                (SIMark) repeatwhile (AND (MatchConstant
                                                                            (QUOTE (;))
                                                                            NIL)
                                                                          (ProductionValue ←(
                                                                                      identifier))
                                                                          (PROGN (SIUnmark)
                                                                                 T)))
                                            (SIToMark)
                                            (SIUnmark)
                                            (if (MatchConstant (QUOTE (%)))
                                                               NIL)
                                                then (PROGN (SIUnmark)
                                                            (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SIUnmark)
                          (RETURN T)))
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SIUnmark)
          (RETURN NIL])
```

13

```
(a0073
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (ProductionValue ←(expression))
              then (PROGN (do (CurrentRecord : expression ←(NCONC1 CurrentRecord : expression
                                                                                ProductionValue))
                              (SIMark) repeatwhile (AND (MatchConstant (QUOTE (,))
                                                                       NIL)
                                                        (ProductionValue ←(expression))
                                                        (PROGN (SIUnmark)
                                                               T)))
                          (SIToMark)
                          (SIUnmark)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
```

```
        (if T
            then (PROGN (SIUnmark)
                        (RETURN T)))
        (SIToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (SIUnmark)
        (RETURN NIL])
```

14

```
(a0074
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (ProductionValue ←(prefixExpr))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(variable))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(number))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(specialPrefixExpr))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(parenExpr))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(bracketExprList))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(ifExpr))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(quantifiedExpression))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SIUnmark)
          (RETURN NIL])
```

15

```
(a0075
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (MatchConstant (QUOTE (:))
                             NIL)
              then (if (ProductionValue ←(coord))
                       then (PROGN (CurrentRecord : coord# ← ProductionValue)
                                   (PROGN (SIUnmark)
                                          (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
```

```
        (if T
            then (PROGN (SlUnmark)
                        (RETURN T)))
        (SlToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (SlUnmark)
        (RETURN NIL])
```

16

```
(a0076
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (ProductionValue ←(coord))
              then (PROGN (CurrentRecord : coord ← ProductionValue)
                          (if (a0075)
                              then (PROGN (SlUnmark)
                                          (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if (MatchConstant (QUOTE (@))
                             NIL)
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

17

```
(a0077
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (ProductionValue ←(rangeSpec))
              then (PROGN (CurrentRecord : rangeSpec ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

18

```
(a0078
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (ProductionValue ←(op))
              then (PROGN (CurrentRecord : op ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

19

```
(a0079
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (=))
```

```
                        NIL)
            then (if (MatchConstant (QUOTE (=))
                                    NIL)
                     then (PROGN (SIUnmark)
                                 (RETURN T]
        (SIToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        [if (MatchConstant (QUOTE (<))
                           NIL)
            then (if (MatchConstant (QUOTE (=))
                                    NIL)
                     then (if (MatchConstant (QUOTE (>))
                                             NIL)
                              then (PROGN (SIUnmark)
                                          (RETURN T]
        (SIToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        [if (MatchConstant (QUOTE (<))
                           NIL)
            then (if (MatchConstant (QUOTE (-))
                                    NIL)
                     then (if (MatchConstant (QUOTE (>))
                                             NIL)
                              then (PROGN (SIUnmark)
                                          (RETURN T]
        (SIToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        [if (MatchConstant (QUOTE (:))
                           NIL)
            then (if (MatchConstant (QUOTE (:))
                                    NIL)
                     then (PROGN (SIUnmark)
                                 (RETURN T]
        (SIToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (SIUnmark)
        (RETURN NIL])
```

20

```
(a0080
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (a0079)
              then (if (ProductionValue ←(expression))
                       then (PROGN (CurrentRecord : expression# ← ProductionValue)
                                   (PROGN (SIUnmark)
                                          (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SIUnmark)
                          (RETURN T)))
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SIUnmark)
          (RETURN NIL])
```

21

```
(a0081
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (ProductionValue ←(compoundStatement))
              then (PROGN (CurrentRecord : compoundStatement ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SIUnmark)
                          (RETURN T)))
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SIUnmark)
          (RETURN NIL])
```

22

```
(a0082
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (ProductionValue ←(declareopt))
              then (PROGN (do (CurrentRecord : declareopt ←(NCONC1 CurrentRecord : declareopt
                                                                   ProductionValue))
                              (SlMark) repeatwhile (AND (MatchConstant (QUOTE (;))
                                                                       NIL)
                                                        (ProductionValue ←(declareopt))
                                                        (PROGN (SlUnmark)
                                                               T)))
                          (SlToMark)
                          (SlUnmark)
                          (if (MatchConstant (QUOTE (;))
                                             NIL)
                              then (PROGN (SlUnmark)
                                          (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

23

```
(a0083
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (EXIT POST))
                             NIL)
              then (if (ProductionValue ←(assertion))
                       then (PROGN (CurrentRecord : assertion# ← ProductionValue)
                                   (if (MatchConstant (QUOTE (;))
                                                      NIL)
                                       then (PROGN (SlUnmark)
                                                   (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

24

```
(a0084
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (ENTRY PRE))
                             NIL)
              then (if (ProductionValue ←(assertion))
                       then (PROGN (CurrentRecord : assertion ← ProductionValue)
                                   (if (MatchConstant (QUOTE (;))
                                                      NIL)
                                       then (PROGN (SlUnmark)
                                                   (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

25

```
(a0085
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (ProductionValue ←(statement))
              then (PROGN (CurrentRecord : statement ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SIUnmark)
                          (RETURN T)))
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SIUnmark)
          (RETURN NIL])
```

26

```
(a0086
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (MatchConstant (QUOTE (;))
                             NIL)
              then (if (MatchConstant (QUOTE (ELSE OTHERWISE))
                                      NIL)
                       then (if (ProductionValue ←(statement))
                                then (PROGN (CurrentRecord : statement ← ProductionValue)
                                            (PROGN (SIUnmark)
                                                   (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SIUnmark)
                          (RETURN T)))
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SIUnmark)
          (RETURN NIL])
```

27

```
(a0087
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (MatchConstant (QUOTE (LABEL))
                             NIL)
              then (if (ProductionValue ←(label))
                       then (PROGN (do (CurrentRecord : label ←(NCONC1 CurrentRecord : label
                                                                       ProductionValue))
                                       (SIMark) repeatwhile (AND (MatchConstant (QUOTE (,))
                                                                                NIL)
                                                                 (ProductionValue ←(label))
                                                                 (PROGN (SIUnmark)
                                                                        T)))
                                   (SIToMark)
                                   (SIUnmark)
                                   (PROGN (SIUnmark)
                                          (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (MatchConstant (QUOTE (CONST))
                             NIL)
              then (if (ProductionValue ←(constDefinition))
                       then (PROGN (do (CurrentRecord : constDefinition ←(NCONC1 CurrentRecord :
                                                                                 constDefinition
                                                                                 ProductionValue))
                                       (SIMark) repeatwhile (AND (MatchConstant (QUOTE (;))
                                                                                NIL)
                                                                 (ProductionValue ←(constDefinition))
                                                                 (PROGN (SIUnmark)
                                                                        T)))
                                   (SIToMark)
```

```
                                    (SIUnmark)
                                    (PROGN (SIUnmark)
                                           (RETURN T]
            (SIToMark)
            (CurrentRecord ←(CopyTopRecord RecordCopy))
            [if (MatchConstant (QUOTE (TYPE))
                               NIL)
                then (if (ProductionValue ←(typeDefinition))
                         then (PROGN (do (CurrentRecord : typeDefinition ←(NCONC1 CurrentRecord :
                                                                                  typeDefinition
                                                                                  ProductionValue))
                                         (SIMark) repeatwhile (AND (MatchConstant (QUOTE (:))
                                                                                  NIL)
                                                                   (ProductionValue ←(typeDefinition))
                                                                   (PROGN (SIUnmark)
                                                                          T)))
                                    (SIToMark)
                                    (SIUnmark)
                                    (PROGN (SIUnmark)
                                           (RETURN T]
            (SIToMark)
            (CurrentRecord ←(CopyTopRecord RecordCopy))
            [if (MatchConstant (QUOTE (VAR))
                               NIL)
                then (if (ProductionValue ←(varDeclaration))
                         then (PROGN (do (CurrentRecord : varDeclaration ←(NCONC1 CurrentRecord :
                                                                                  varDeclaration
                                                                                  ProductionValue))
                                         (SIMark) repeatwhile (AND (MatchConstant (QUOTE (:))
                                                                                  NIL)
                                                                   (ProductionValue ←(varDeclaration))
                                                                   (PROGN (SIUnmark)
                                                                          T)))
                                    (SIToMark)
                                    (SIUnmark)
                                    (PROGN (SIUnmark)
                                           (RETURN T]
            (SIToMark)
            (CurrentRecord ←(CopyTopRecord RecordCopy))
            [if (ProductionValue ←(procedureOrFunctionDeclaration))
                then (PROGN (CurrentRecord : procedureOrFunctionDeclaration ← ProductionValue)
                            (PROGN (SIUnmark)
                                   (RETURN T]
            (SIToMark)
            (CurrentRecord ←(CopyTopRecord RecordCopy))
            (SIUnmark)
            (RETURN NIL])
```

28

```
(a0088
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (MatchConstant (QUOTE (:))
                             NIL)
              then (if (ProductionValue ←(variantPart))
                       then (PROGN (CurrentRecord : variantPart ← ProductionValue)
                                   (PROGN (SIUnmark)
                                          (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SIUnmark)
                          (RETURN T)))
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SIUnmark)
          (RETURN NIL])
```

29

```
(a0089
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (ProductionValue ←(recordSection))
              then (PROGN (do (CurrentRecord : recordSection ←(NCONC1 CurrentRecord : recordSection
                                                                      ProductionValue))
```

```
                    (SlMark) repeatwhile (AND (MatchConstant (QUOTE (:))
                                                        NIL)
                                              (ProductionValue ←(recordSection))
                                              (PROGN (SlUnmark)
                                                     T)))
                (SlToMark)
                (SlUnmark)
                (PROGN (SlUnmark)
                       (RETURN T]
        (SlToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (if T
            then (PROGN (SlUnmark)
                        (RETURN T)))
        (SlToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (SlUnmark)
        (RETURN NIL])
```

30

```
(a0090
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (ProductionValue ←(parameterKind))
              then (PROGN (CurrentRecord : parameterKind ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

31

```
(a0091
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (THUS))
                             NIL)
              then (if (ProductionValue ←(assertion))
                       then (PROGN (CurrentRecord : assertion# ← ProductionValue)
                                   (PROGN (SlUnmark)
                                          (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

32

```
(a0092
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (MAINTAIN))
                             NIL)
              then (if (ProductionValue ←(assertion))
                       then (PROGN (CurrentRecord : assertion ← ProductionValue)
                                   (PROGN (SlUnmark)
                                          (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
```

```
        (SlToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (SlUnmark)
        (RETURN NIL])
```

33

```
(a0093
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (ASSERTING))
                             NIL)
              then (if (ProductionValue ←(assertion))
                       then (PROGN (CurrentRecord : assertion ← ProductionValue)
                                   (PROGN (SlUnmark)
                                          (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

34

```
(a0094
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          (if (MatchConstant (QUOTE (GOTO))
                             NIL)
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (MatchConstant (QUOTE (GO))
                             NIL)
              then (if (MatchConstant (QUOTE (TO))
                                      NIL)
                       then (PROGN (SlUnmark)
                                   (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

35

```
(a0095
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (ELSE))
                             NIL)
              then (if (ProductionValue ←(statement))
                       then (PROGN (CurrentRecord : statement# ← ProductionValue)
                                   (PROGN (SlUnmark)
                                          (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

36

```
(a0096
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
```

```
        (SlMark)
        [if (ProductionValue ←(simpleStatement))
            then (PROGN (CurrentRecord : simpleStatement ← ProductionValue)
                        (PROGN (SlUnmark)
                               (RETURN T]
        (SlToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (if T
            then (PROGN (SlUnmark)
                        (RETURN T)))
        (SlToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (SlUnmark)
        (RETURN NIL])
```

37

```
(a0097
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (:))
                             NIL)
              then (if (ProductionValue ←(type))
                       then (PROGN (CurrentRecord : type ← ProductionValue)
                                   (PROGN (SlUnmark)
                                          (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

38

```
(a0098
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (ALTERS))
                             NIL)
              then (if (ProductionValue ←(identifier))
                       then (PROGN (do (CurrentRecord : identifier## ←(NCONC1 CurrentRecord :
                                                                              identifier##
                                                                              ProductionValue))
                                       (SlMark) repeatwhile (AND (MatchConstant (QUOTE (,))
                                                                                NIL)
                                                                 (ProductionValue ←(identifier))
                                                                 (PROGN (SlUnmark)
                                                                        T)))
                                   (SlToMark)
                                   (SlUnmark)
                                   (PROGN (SlUnmark)
                                          (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

39

```
(a0099
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (IMPORTS))
                             NIL)
              then (if (MatchConstant (QUOTE (%())
                                      NIL)
```

```
                then (if (ProductionValue ←(formalParameterSection))
                         then (PROGN (do (CurrentRecord : formalParameterSection# ←(NCONC1
                                                          CurrentRecord :
                                                          formalParameterSection#
                                                          ProductionValue))
                                         (SlMark) repeatwhile (AND (MatchConstant
                                                                     (QUOTE (:))
                                                                     NIL)
                                                                   (ProductionValue ←(
                                                                    formalParameterSection))
                                                                   (PROGN (SlUnmark)
                                                                          T)))
                                     (SlToMark)
                                     (SlUnmark)
                                     (if (MatchConstant (QUOTE (4)))
                                                        NIL)
                                         then (PROGN (SlUnmark)
                                                     (RETURN T]
        (SlToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (if T
            then (PROGN (SlUnmark)
                        (RETURN T)))
        (SlToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (SlUnmark)
        (RETURN NIL])
```

40

```
(a0100
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (:))
                             NIL)
              then (if (ProductionValue ←(type))
                       then (PROGN (CurrentRecord : type ← ProductionValue)
                                   (PROGN (SlUnmark)
                                          (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

41

```
(a0101
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (ProductionValue ←(identifier))
              then (PROGN (CurrentRecord : identifier# ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

42

```
(a0102
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (%()))
                             NIL)
```

```
            then (if (ProductionValue ←(formalParameterSection))
                     then (PROGN (do (CurrentRecord : formalParameterSection ←(NCONC1 Current:Record
                                                                                      :
                                                                                formalParameterSection
                                                                                  ProductionValue))
                                     (SIMark) repeatwhile (AND (MatchConstant (QUOTE (;))
                                                               NIL)
                                                          (ProductionValue ←(
                                                                  formalParameterSection))
                                                          (PROGN (SIUnmark)
                                                                 T)))
                                 (SIToMark)
                                 (SIUnmark)
                                 (if (MatchConstant (QUOTE (%)))
                                                    NIL)
                                     then (PROGN (SIUnmark)
                                                 (RETURN T]
        (SIToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (if T
            then (PROGN (SIUnmark)
                        (RETURN T)))
        (SIToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (SIUnmark)
        (RETURN NIL])
```

43

```
(a0103
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (MatchConstant (QUOTE (ALTERS))
                             NIL)
              then (if (ProductionValue ←(variable))
                       then (PROGN (do (CurrentRecord : variable ←(NCONC1 CurrentRecord : variable
                                                                          ProductionValue))
                                       (SIMark) repeatwhile (AND (MatchConstant (QUOTE (,))
                                                                 NIL)
                                                            (ProductionValue ←(variable))
                                                            (PROGN (SIUnmark)
                                                                   T)))
                                   (SIToMark)
                                   (SIUnmark)
                                   (PROGN (SIUnmark)
                                          (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SIUnmark)
                          (RETURN T)))
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SIUnmark)
          (RETURN NIL])
```

44

```
(a0104
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (MatchConstant (QUOTE (IMPORTS))
                             NIL)
              then (if (MatchConstant (QUOTE (%()
                                      NIL)
                       then (if (ProductionValue ←(identifier))
                                then (PROGN (do (CurrentRecord : identifier# ←(NCONC1 CurrentRecord :
                                                                                      identifier#
                                                                                ProductionValue))
                                                (SIMark) repeatwhile (AND (MatchConstant
                                                                            (QUOTE (;))
                                                                            NIL)
                                                                          (ProductionValue ←(
                                                                                  identifier))
                                                                          (PROGN (SIUnmark)
                                                                                 T)))
                                            (SIToMark)
```

```
                                        (SlUnmark)
                                        (if (MatchConstant (QUOTE (%)))
                                                    NIL)
                                            then (PROGN (SlUnmark)
                                                        (RETURN T]
        (SlToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (if T
            then (PROGN (SlUnmark)
                        (RETURN T)))
        (SlToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (SlUnmark)
        (RETURN NIL])
```

45

```
(a0105
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (%()))
                             NIL)
              then (if (ProductionValue ←(expression))
                       then (PROGN (do (CurrentRecord : expression ←(NCONC1 CurrentRecord :
                                                                            expression
                                                                            ProductionValue))
                                       (SlMark) repeatwhile (AND (MatchConstant (QUOTE (.))
                                                                                NIL)
                                                                 (ProductionValue ←(expression))
                                                                 (PROGN (SlUnmark)
                                                                        T)))
                                   (SlToMark)
                                   (SlUnmark)
                                   (if (MatchConstant (QUOTE (%)))
                                                      NIL)
                                       then (PROGN (SlUnmark)
                                                   (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

46

```
(a0106
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (%[))
                             NIL)
              then (if (ProductionValue ←(expression))
                       then (PROGN (do (CurrentRecord : expression ←(NCONC1 CurrentRecord :
                                                                            expression
                                                                            ProductionValue))
                                       (SlMark) repeatwhile (AND (MatchConstant (QUOTE (,))
                                                                                NIL)
                                                                 (ProductionValue ←(expression))
                                                                 (PROGN (SlUnmark)
                                                                        T)))
                                   (SlToMark)
                                   (SlUnmark)
                                   (if (MatchConstant (QUOTE (%]))
                                                      NIL)
                                       then (PROGN (SlUnmark)
                                                   (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (MatchConstant (QUOTE (%.))
                             NIL)
              then (if (ProductionValue ←(identifier))
                       then (PROGN (CurrentRecord : identifier ← ProductionValue)
                                   (PROGN (SlUnmark)
                                          (RETURN T]
```

```
        (SlToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (if (MatchConstant (QUOTE (↑))
                           NIL)
            then (PROGN (SlUnmark)
                        (RETURN T)))
        (SlToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (SlUnmark)
        (RETURN NIL])
```

47

```
(a0107
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (THUS))
                             NIL)
              then (if (ProductionValue ←(assertion))
                       then (PROGN (CurrentRecord : assertion ← ProductionValue)
                                   (PROGN (SlUnmark)
                                          (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

48

```
(a0108
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (ASSERTING))
                             NIL)
              then (if (ProductionValue ←(assertion))
                       then (PROGN (CurrentRecord : assertion ← ProductionValue)
                                   (PROGN (SlUnmark)
                                          (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

49

```
(a0109
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (%())
                             NIL)
              then (if (ProductionValue ←(expression))
                       then (PROGN (CurrentRecord : expression ← ProductionValue)
                                   (if (MatchConstant (QUOTE (%)))
                                                      NIL)
                                       then (PROGN (SlUnmark)
                                                   (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

50

```
(a0110
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (ProductionValue ←(compoundStatement))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(ifStatement))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(caseStatement))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(whileStatement))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(repeatStatement))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(forStatement))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(withStatement))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(goToStatement))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(assertStatement))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(returnStatement))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(proveStatement))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(assumeStatement))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(assignmentStatement))
```

```
            then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                        (PROGN (SIUnmark)
                               (RETURN T]
      (SIToMark)
      (CurrentRecord ←(CopyTopRecord RecordCopy))
      [if (ProductionValue ←(concurrentAssignmentStatement))
          then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                      (PROGN (SIUnmark)
                             (RETURN T]
      (SIToMark)
      (CurrentRecord ←(CopyTopRecord RecordCopy))
      [if (ProductionValue ←(procedureStatement))
          then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                      (PROGN (SIUnmark)
                             (RETURN T]
      (SIToMark)
      (CurrentRecord ←(CopyTopRecord RecordCopy))
      (SIUnmark)
      (RETURN NIL])
```

51

```
(a0111
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (ProductionValue ←(scalarType))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(subrangeType))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(typeIdentifier))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SIUnmark)
          (RETURN NIL])
```

52

```
(a0112
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (ProductionValue ←(assignmentStatement))
              then (PROGN (CurrentRecord : assignmentStatement ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(labelStatement))
              then (PROGN (CurrentRecord : labelStatement ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(simpleStatement))
              then (PROGN (CurrentRecord : simpleStatement ← ProductionValue)
                          (PROGN (SIUnmark)
                                 (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SIUnmark)
                          (RETURN T)))
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SIUnmark)
          (RETURN NIL])
```

53

```
(a0113
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (ProductionValue ←(packed))
              then (PROGN (CurrentRecord : packed ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

54

```
(a0114
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          (if (MatchConstant (QUOTE (*))
                             NIL)
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(expression))
              then (PROGN (CurrentRecord : expression# ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

55

```
(a0115
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          (if (MatchConstant (QUOTE (*))
                             NIL)
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(expression))
              then (PROGN (CurrentRecord : expression ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

56

```
(a0116
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (ProductionValue ←(simpleType))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(structuredType))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
```

```
                                (RETURN T]
        (SlToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        [if (ProductionValue ←(pointerType))
            then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                        (PROGN (SlUnmark)
                               (RETURN T]
        (SlToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (SlUnmark)
        (RETURN NIL])
```

57

```
(a0117
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (ProductionValue ←(arrayType))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(recordType))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(setType))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          [if (ProductionValue ←(fileType))
              then (PROGN (CurrentRecord : ALTERNATIVESUBNODE ← ProductionValue)
                          (PROGN (SlUnmark)
                                 (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

58

```
(a0118
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (:))
                             NIL)
              then (if (MatchConstant (QUOTE (*))
                                      NIL)
                       then (if (ProductionValue ←(expression))
                                then (PROGN (CurrentRecord : expression# ← ProductionValue)
                                            (PROGN (SlUnmark)
                                                   (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

59

```
(a0119
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (MatchConstant (QUOTE (@))
                             NIL)
              then (if (ProductionValue ←(expression))
```

```
                    then (PROGN (CurrentRecord : expression ← ProductionValue)
                                (PROGN (SlUnmark)
                                       (RETURN T]
        (SlToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (if T
            then (PROGN (SlUnmark)
                        (RETURN T)))
        (SlToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (SlUnmark)
        (RETURN NIL])
```

60

```
(a0120
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (ProductionValue ←(caseLabel))
              then (PROGN (do (CurrentRecord : caseLabel ←(NCONC1 CurrentRecord : caseLabel
                                                                  ProductionValue))
                              (SlMark) repeatwhile (AND (MatchConstant (QUOTE (.))
                                                                       NIL)
                                                        (ProductionValue ←(caseLabel))
                                                        (PROGN (SlUnmark)
                                                               T)))
                          (SlToMark)
                          (SlUnmark)
                          (if (MatchConstant (QUOTE (:))
                                             NIL)
                              then (if (MatchConstant (QUOTE (%())
                                                      NIL)
                                       then (if (ProductionValue ←(fieldList))
                                                then (PROGN (CurrentRecord : fieldList ←
                                                                             ProductionValue)
                                                            (if (MatchConstant (QUOTE (%)))
                                                                               NIL)
                                                                then (PROGN (SlUnmark)
                                                                            (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

61

```
(a0121
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
          [if (ProductionValue ←(identifier))
              then (PROGN (CurrentRecord : identifier ← ProductionValue)
                          (if (MatchConstant (QUOTE (:))
                                             NIL)
                              then (PROGN (SlUnmark)
                                          (RETURN T]
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SlUnmark)
                          (RETURN T)))
          (SlToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SlUnmark)
          (RETURN NIL])
```

62

```
(a0122
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SlMark)
```

```
        [if (MatchConstant (QUOTE (THUS))
                           NIL)
            then (if (ProductionValue ←(assertion))
                     then (PROGN (CurrentRecord : assertion# ← ProductionValue)
                                 (PROGN (SIUnmark)
                                        (RETURN T]
        (SIToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (if T
            then (PROGN (SIUnmark)
                        (RETURN T)))
        (SIToMark)
        (CurrentRecord ←(CopyTopRecord RecordCopy))
        (SIUnmark)
        (RETURN NIL])
```

63

```
(a0123
  [LAMBDA NIL
    (PROG ((RecordCopy (CopyTopRecord CurrentRecord)))
          (SIMark)
          [if (MatchConstant (QUOTE (MAINTAIN))
                             NIL)
              then (if (ProductionValue ←(assertion))
                       then (PROGN (CurrentRecord : assertion ← ProductionValue)
                                   (PROGN (SIUnmark)
                                          (RETURN T]
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (if T
              then (PROGN (SIUnmark)
                          (RETURN T)))
          (SIToMark)
          (CurrentRecord ←(CopyTopRecord RecordCopy))
          (SIUnmark)
          (RETURN NIL])
```

64

```
(all
  [LAMBDA NIL
    (PROG ((CurrentRecord (create all)))
          (if (MatchConstant (QUOTE (ALL))
                             NIL)
              then (PROGN (CurrentRecord : LEXEME ← ProductionValue)
                          (RETURN CurrentRecord])
```

65

```
(arrayType
  [LAMBDA NIL
    (PROG ((CurrentRecord (create arrayType)))
          (if (MatchConstant (QUOTE (ARRAY))
                             NIL)
              then (if (MatchConstant (QUOTE (%[))
                                      NIL)
                       then (if (ProductionValue ←(simpleType))
                                then (PROGN (do (CurrentRecord : simpleType ←(NCONC1 CurrentRecord :
                                                                                     simpleType
                                                                                     ProductionValue))
                                                (SIMark) repeatwhile (AND (MatchConstant
                                                                            (QUOTE (,))
                                                                            NIL)
                                                                          (ProductionValue ←(
                                                                                     simpleType))
                                                                          (PROGN (SIUnmark)
                                                                                 T)))
                                            (SIToMark)
                                            (SIUnmark)
                                            (if (MatchConstant (QUOTE (%]))
                                                               NIL)
                                                then (if (MatchConstant (QUOTE (OF))
                                                                        NIL)
                                                         then (if (ProductionValue ←(type))
                                                                  then (PROGN (CurrentRecord : type ←
                                                                                     ProductionValue)
                                                                              (RETURN CurrentRecord])
```

66

```
(assertStatement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create assertStatement)))
          (if (MatchConstant (QUOTE (ASSERT))
                             NIL)
              then (if (ProductionValue ←(assertion))
                       then (PROGN (CurrentRecord : assertion ← ProductionValue)
                                   (RETURN CurrentRecord])
```

67

```
(assertion
  [LAMBDA NIL
    (PROG ((CurrentRecord (create assertion)))
          (if (ProductionValue ←(expression))
              then (PROGN (CurrentRecord : expression ← ProductionValue)
                          (RETURN CurrentRecord])
```

68

```
(assignmentStatement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create assignmentStatement)))
          (if (ProductionValue ←(variable))
              then (PROGN (CurrentRecord : variable ← ProductionValue)
                          (if (MatchConstant (QUOTE (:))
                                             NIL)
                              then (if (MatchConstant (QUOTE (=))
                                                      NIL)
                                       then (if (ProductionValue ←(expression))
                                                then (PROGN (CurrentRecord : expression ←
                                                                             ProductionValue)
                                                            (RETURN CurrentRecord])
```

69

```
(assumeStatement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create assumeStatement)))
          (if (MatchConstant (QUOTE (ASSUME))
                             NIL)
              then (if (ProductionValue ←(assertion))
                       then (PROGN (CurrentRecord : assertion ← ProductionValue)
                                   (RETURN CurrentRecord])
```

70

```
(block
  [LAMBDA NIL
    (PROG ((CurrentRecord (create block)))
          (if (a0084)
              then (if (a0083)
                       then (if (a0082)
                                then (if (a0081)
                                         then (RETURN CurrentRecord])
```

71

```
(bracketExprList
  [LAMBDA NIL
    (PROG ((CurrentRecord (create bracketExprList)))
          (if (MatchConstant (QUOTE (%[))
                             NIL)
              then (if (a0063)
                       then (if (MatchConstant (QUOTE (%]))
                                               NIL)
                                then (RETURN CurrentRecord])
```

72

```
(caseElementList
```

```
[LAMBDA NIL
  (PROG ((CurrentRecord (create caseElementList)))
        (if (ProductionValue ←(caseLabel))
            then (PROGN (do (CurrentRecord : caseLabel ←(NCONC1 CurrentRecord : caseLabel
                                                                ProductionValue))
                            (SIMark) repeatwhile (AND (MatchConstant (QUOTE (.))
                                                                     NIL)
                                                      (ProductionValue ←(caseLabel))
                                                      (PROGN (SIUnmark)
                                                             T)))
                        (SIToMark)
                        (SIUnmark)
                        (if (MatchConstant (QUOTE (:))
                                           NIL)
                            then (if (a0085)
                                     then (RETURN CurrentRecord])
```

73

```
(caseLabel
  [LAMBDA NIL
    (PROG ((CurrentRecord (create caseLabel)))
          (if (ProductionValue ←(constant))
              then (PROGN (CurrentRecord : constant ← ProductionValue)
                          (RETURN CurrentRecord])
```

74

```
(caseStatement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create caseStatement)))
          (if (MatchConstant (QUOTE (CASE))
                             NIL)
              then (if (ProductionValue ←(expression))
                       then (PROGN (CurrentRecord : expression ← ProductionValue)
                                   (if (MatchConstant (QUOTE (OF))
                                                      NIL)
                                       then
                                        (if (ProductionValue ←(caseElementList))
                                            then (PROGN (do (CurrentRecord : caseElementList ←(NCONC1
                                                                                               CurrentRecord :
                                                                                               caseElementList
                                                                                               ProductionValue))
                                                            (SIMark)
                                                            repeatwhile (AND (MatchConstant
                                                                               (QUOTE (;))
                                                                               NIL)
                                                                             (ProductionValue
                                                                               ←(caseElementList))
                                                                             (PROGN (SIUnmark)
                                                                                    T)))
                                                        (SIToMark)
                                                        (SIUnmark)
                                                        (if (a0086)
                                                            then
                                                             (if (MatchConstant (QUOTE (;))
                                                                                T)
                                                                 then (if (MatchConstant
                                                                            (QUOTE (END))
                                                                            NIL)
                                                                          then (RETURN CurrentRecord])
```

75

```
(compoundStatement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create compoundStatement)))
          (if (MatchConstant (QUOTE (BEGIN))
                             NIL)
              then (if (ProductionValue ←(statement))
                       then (PROGN (do (CurrentRecord : statement ←(NCONC1 CurrentRecord : statement
                                                                   ProductionValue))
                                       (SIMark) repeatwhile (AND (MatchConstant (QUOTE (;))
                                                                                NIL)
                                                                 (ProductionValue ←(statement))
                                                                 (PROGN (SIUnmark)
                                                                        T)))
                                   (SIToMark)
```

```
                    (SlUnmark)
                    (if (MatchConstant (QUOTE (END))
                                       NIL)
                        then (RETURN CurrentRecord])
```

76

```
(concurrentAssignmentStatement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create concurrentAssignmentStatement)))
          (if (ProductionValue ←(variable))
              then (PROGN (do (CurrentRecord : variable ←(NCONC1 CurrentRecord : variable
                                                                 ProductionValue))
                              (SlMark) repeatwhile (AND (MatchConstant (QUOTE (,))
                                                                       NIL)
                                                        (ProductionValue ←(variable))
                                                        (PROGN (SlUnmark)
                                                               T)))
                          (SlToMark)
                          (SlUnmark)
                          (if (MatchConstant (QUOTE (:))
                                             NIL)
                              then (if (MatchConstant (QUOTE (=))
                                                      NIL)
                                       then (if (ProductionValue ←(expression))
                                                then (PROGN (do (CurrentRecord : expression ←(NCONC1
                                                                                              CurrentRecord :
                                                                                              expression
                                                                                              ProductionValue))
                                                                (SlMark)
                                                                repeatwhile
                                                                (AND (MatchConstant (QUOTE (,))
                                                                                    NIL)
                                                                     (ProductionValue ←(expression))
                                                                     (PROGN (SlUnmark)
                                                                            T)))
                                                            (SlToMark)
                                                            (SlUnmark)
                                                            (RETURN CurrentRecord])
```

77

```
(constDefinition
  [LAMBDA NIL
    (PROG ((CurrentRecord (create constDefinition)))
          (if (ProductionValue ←(identifier))
              then (PROGN (CurrentRecord : identifier ← ProductionValue)
                          (if (MatchConstant (QUOTE (=))
                                             NIL)
                              then (if (ProductionValue ←(expression))
                                       then (PROGN (CurrentRecord : expression ← ProductionValue)
                                                   (RETURN CurrentRecord])
```

78

```
(constant
  [LAMBDA NIL
    (PROG ((CurrentRecord (create constant)))
          (if (MatchLexeme)
              then (RETURN CurrentRecord])
```

79

```
(coord
  [LAMBDA NIL
    (PROG ((CurrentRecord (create coord)))
          (if (a0064)
              then (RETURN CurrentRecord])
```

80

```
(declareType
  [LAMBDA NIL
    (PROG ((CurrentRecord (create declareType)))
          (if (a0087)
              then (RETURN CurrentRecord])
```

81

```
(declareopt
  [LAMBDA NIL
    (PROG ((CurrentRecord (create declareopt)))
          (if (MatchConstant (QUOTE (XPUBLIC PUBLIC))
                             T)
              then (if (ProductionValue ←(declareType))
                       then (PROGN (CurrentRecord : declareType ← ProductionValue)
                                   (RETURN CurrentRecord])
```

82

```
(denotePair
  [LAMBDA NIL
    (PROG ((CurrentRecord (create denotePair)))
          (if (ProductionValue ←(expression))
              then (PROGN (CurrentRecord : expression ← ProductionValue)
                          (if (MatchConstant (QUOTE (BY))
                                             NIL)
                              then (if (ProductionValue ←(identifier))
                                       then (PROGN (CurrentRecord : identifier ← ProductionValue)
                                                   (RETURN CurrentRecord])
```

83

```
(denoteSpec
  [LAMBDA NIL
    (PROG ((CurrentRecord (create denoteSpec)))
          (if (ProductionValue ←(denotePair))
              then (PROGN (do (CurrentRecord : denotePair ←(NCONC1 CurrentRecord : denotePair
                                                                   ProductionValue))
                              (SlMark) repeatwhile (AND (MatchConstant (QUOTE (,))
                                                                       NIL)
                                                        (ProductionValue ←(denotePair))
                                                        (PROGN (SlUnmark)
                                                               T)))
                          (SlToMark)
                          (SlUnmark)
                          (RETURN CurrentRecord])
```

84

```
(direction
  [LAMBDA NIL
    (PROG ((CurrentRecord (create direction)))
          (if (MatchConstant (QUOTE (TO DOWNTO))
                             NIL)
              then (PROGN (CurrentRecord : LEXEME ← ProductionValue)
                          (RETURN CurrentRecord])
```

85

```
(expression
  [LAMBDA NIL
    (PROG ((CurrentRecord (create expression)))
          (if (ProductionValue ←(primary))
              then (PROGN (CurrentRecord : primary ← ProductionValue)
                          (if (a0065)
                              then (RETURN CurrentRecord])
```

86

```
(expressionSeq
  [LAMBDA NIL
    (PROG ((CurrentRecord (create expressionSeq)))
          (if (ProductionValue ←(expression))
              then (PROGN (do (CurrentRecord : expression ←(NCONC1 CurrentRecord : expression
                                                                   ProductionValue))
                              (SlMark) repeatwhile (AND (MatchConstant (QUOTE (,))
                                                                       NIL)
                                                        (ProductionValue ←(expression))
                                                        (PROGN (SlUnmark)
                                                               T)))
```

```
                    (SlToMark)
                    (SlUnmark)
                    (RETURN CurrentRecord])
```

87

```
(fieldList
  [LAMBDA NIL
    (PROG ((CurrentRecord (create fieldList)))
          (if (a0089)
              then (if (a0088)
                       then (if (MatchConstant (QUOTE (:))
                                               T)
                                then (RETURN CurrentRecord])
```

88

```
(fileType
  [LAMBDA NIL
    (PROG ((CurrentRecord (create fileType)))
          (if (MatchConstant (QUOTE (FILE))
                             NIL)
              then (if (MatchConstant (QUOTE (OF))
                                      NIL)
                       then (if (ProductionValue ←(type))
                                then (PROGN (CurrentRecord : type ← ProductionValue)
                                            (RETURN CurrentRecord])
```

89

```
(firstOne
  [LAMBDA NIL
    (PROG ((CurrentRecord (create firstOne)))
          (if (MatchConstant (QUOTE (FIRST))
                             NIL)
              then (PROGN (CurrentRecord : LEXEME ← ProductionValue)
                          (RETURN CurrentRecord])
```

90

```
(forStatement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create forStatement)))
          (if (a0092)
              then
               (if (MatchConstant (QUOTE (FOR))
                                  NIL)
                   then
                    (if (ProductionValue ←(identifier))
                        then
                         (PROGN
                           (CurrentRecord : identifier ← ProductionValue)
                           (if (MatchConstant (QUOTE (:))
                                              NIL)
                               then
                                (if (MatchConstant (QUOTE (=))
                                                   NIL)
                                    then
                                     (if (ProductionValue ←(expression))
                                         then
                                          (PROGN
                                            (CurrentRecord : expression ← ProductionValue)
                                            (if (ProductionValue ←(direction))
                                                then
                                                 (PROGN
                                                   (CurrentRecord : direction ← ProductionValue)
                                                   (if (ProductionValue ←(expression))
                                                       then
                                                        (PROGN
                                                          (CurrentRecord : expression# ←
                                                                           ProductionValue)
                                                          (if (MatchConstant (QUOTE (DO))
                                                                             NIL)
                                                              then (if (ProductionValue ←(statement))
                                                                       then
                                                                        (PROGN (CurrentRecord :
                                                                                       statement ←
```

```
                                                    Productic  .lue)
                                                (if (a0091)
                                                    then (RI  .N
                                                          Current  ord])
```

91

```
(formalParameterSection
  [LAMBDA NIL
    (PROG ((CurrentRecord (create formalParameterSection)))
          (if (a0090)
              then (if (ProductionValue ←(parameterGroup))
                       then (PROGN (CurrentRecord : parameterGroup ← ProductionValue)
                                   (RETURN CurrentRecord])
```

92

```
(functionDecl
  [LAMBDA NIL
    (PROG ((CurrentRecord (create functionDecl)))
          (if (ProductionValue ←(expression))
              then (PROGN (do (CurrentRecord : expression ←(NCONC1 CurrentRecord : expression
                                                                   ProductionValue))
                              (SIMark) repeatwhile (AND (MatchConstant (QUOTE (.))
                                                                       NIL)
                                                        (ProductionValue ←(expression))
                                                        (PROGN (SIUnmark)
                                                               T)))
                          (SIToMark)
                          (SIUnmark)
                          (if (MatchConstant (QUOTE (:))
                                             NIL)
                              then (if (ProductionValue ←(expression))
                                       then (PROGN (CurrentRecord : expression# ← ProductionValue)
                                                   (RETURN CurrentRecord])
```

93

```
(goToStatement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create goToStatement)))
          (if (a0094)
              then (if (ProductionValue ←(label))
                       then (PROGN (CurrentRecord : label ← ProductionValue)
                                   (if (a0093)
                                       then (RETURN CurrentRecord])
```

94

```
(greaterThanEqual
  [LAMBDA NIL
    (PROG ((CurrentRecord (create greaterThanEqual)))
          (if (MatchConstant (QUOTE (>))
                             NIL)
              then (if (MatchConstant (QUOTE (=))
                                      NIL)
                       then (RETURN CurrentRecord])
```

95

```
(identifier
  [LAMBDA NIL
    (PROG ((CurrentRecord (create identifier)))
          (if (MatchLexeme)
              then (PROGN (CurrentRecord ←(identifierFilter CurrentRecord))
                          (RETURN CurrentRecord])
```

96

```
(identifierSeq
  [LAMBDA NIL
    (PROG ((CurrentRecord (create identifierSeq)))
          (if (ProductionValue ←(identifier))
              then (PROGN (do (CurrentRecord : identifier ←(NCONC1 CurrentRecord : identifier
```

```
                                                    ProductionValue))
                        (SlMark) repeatwhile (AND (MatchConstant (QUOTE (.))
                                                  NIL)
                                                 (ProductionValue ←(identifier))
                                                 (PROGN (SlUnmark)
                                                        T)))
                    (SlToMark)
                    (SlUnmark)
                    (RETURN CurrentRecord])
```

97

```
(ifExpr
  [LAMBDA NIL
    (PROG ((CurrentRecord (create ifExpr)))
          (if (MatchConstant (QUOTE (IF))
                             NIL)
              then (if (ProductionValue ←(expression))
                       then (PROGN (CurrentRecord : expression ← ProductionValue)
                                   (if (MatchConstant (QUOTE (THEN))
                                                      NIL)
                                       then (if (ProductionValue ←(expression))
                                                then (PROGN (CurrentRecord : expression# ←
                                                                             ProductionValue)
                                                            (if (a0066)
                                                                then (RETURN CurrentRecord])
```

98

```
(ifStatement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create ifStatement)))
          (if (MatchConstant (QUOTE (IF))
                             NIL)
              then (if (ProductionValue ←(expression))
                       then (PROGN (CurrentRecord : expression ← ProductionValue)
                                   (if (MatchConstant (QUOTE (THEN))
                                                      NIL)
                                       then (if (ProductionValue ←(statement))
                                                then (PROGN (CurrentRecord : statement ←
                                                                             ProductionValue)
                                                            (if (a0095)
                                                                then (RETURN CurrentRecord])
```

99

```
(infixOp
  [LAMBDA NIL
    (PROG ((CurrentRecord (create infixOp)))
          (if (a0067)
              then (RETURN CurrentRecord])
```

100

```
(interfaceList
  [LAMBDA NIL
    (PROG ((CurrentRecord (create interfaceList)))
          (if (ProductionValue ←(op))
              then (PROGN (do (CurrentRecord : op ←(NCONC1 CurrentRecord : op ProductionValue))
                              (SlMark) repeatwhile (AND (MatchConstant (QUOTE (.))
                                                                       NIL)
                                                        (ProductionValue ←(op))
                                                        (PROGN (SlUnmark)
                                                               T)))
                          (SlToMark)
                          (SlUnmark)
                          (RETURN CurrentRecord])
```

101

```
(label
  [LAMBDA NIL
    (PROG ((CurrentRecord (create label)))
          (if (MatchLexeme)
              then (PROGN (CurrentRecord ←(labelFilter CurrentRecord))
                          (RETURN CurrentRecord])
```

102

```
(labelStatement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create labelStatement)))
          (if (ProductionValue ←(label))
              then (PROGN (CurrentRecord : label ← ProductionValue)
                          (if (MatchConstant (QUOTE (:))
                                             NIL)
                              then (if (a0096)
                                       then (RETURN CurrentRecord])
```

103

```
(lastOne
  [LAMBDA NIL
    (PROG ((CurrentRecord (create lastOne)))
          (if (MatchConstant (QUOTE (LAST))
                             NIL)
              then (PROGN (CurrentRecord : LEXEME ← ProductionValue)
                          (RETURN CurrentRecord])
```

104

```
(lessThanEqual
  [LAMBDA NIL
    (PROG ((CurrentRecord (create lessThanEqual)))
          (if (MatchConstant (QUOTE (<))
                             NIL)
              then (if (MatchConstant (QUOTE (=))
                                      NIL)
                       then (RETURN CurrentRecord])
```

105

```
(machinePair
  [LAMBDA NIL
    (PROG ((CurrentRecord (create machinePair)))
          (if (a0069)
              then (if (a0068)
                       then (RETURN CurrentRecord])
```

106

```
(machineSpec
  [LAMBDA NIL
    (PROG ((CurrentRecord (create machineSpec)))
          (if (ProductionValue ←(machinePair))
              then (PROGN (do (CurrentRecord : machinePair ←(NCONC1 CurrentRecord : machinePair
                                                                    ProductionValue))
                              (SIMark) repeatwhile (AND (MatchConstant (QUOTE (,))
                                                                       NIL)
                                                        (ProductionValue ←(machinePair))
                                                        (PROGN (SIUnmark)
                                                               T)))
                          (SIToMark)
                          (SIUnmark)
                          (RETURN CurrentRecord])
```

107

```
(normalInfixOp
  [LAMBDA NIL
    (PROG ((CurrentRecord (create normalInfixOp)))
          (if (MatchLexeme)
              then (PROGN (CurrentRecord ←(infixOpFilter CurrentRecord))
                          (RETURN CurrentRecord])
```

108

```
(notEqual
  [LAMBDA NIL
    (PROG ((CurrentRecord (create notEqual)))
```

```
        (if (a0070)
            then (RETURN CurrentRecord])
```

109

```
(number
  [LAMBDA NIL
    (PROG ((CurrentRecord (create number)))
          (if (MatchLexeme)
              then (PROGN (CurrentRecord ←(unsignedInteger CurrentRecord))
                          (RETURN CurrentRecord])
```

110

```
(op
  [LAMBDA NIL
    (PROG ((CurrentRecord (create op)))
          (if (a0071)
              then (RETURN CurrentRecord])
```

111

```
(packed
  [LAMBDA NIL
    (PROG ((CurrentRecord (create packed)))
          (if (MatchConstant (QUOTE (PACKED))
                             NIL)
              then (PROGN (CurrentRecord : LEXEME ← ProductionValue)
                          (RETURN CurrentRecord])
```

112

```
(parameterGroup
  [LAMBDA NIL
    (PROG ((CurrentRecord (create parameterGroup)))
          (if (ProductionValue ←(identifier))
              then (PROGN (do (CurrentRecord : identifier ←(NCONC1 CurrentRecord : identifier
                                                                   ProductionValue))
                              (SlMark) repeatwhile (AND (MatchConstant (QUOTE (,))
                                                                       NIL)
                                                        (ProductionValue ←(identifier))
                                                        (PROGN (SlUnmark)
                                                               T)))
                          (SlToMark)
                          (SlUnmark)
                          (if (a0097)
                              then (RETURN CurrentRecord])
```

113

```
(parameterKind
  [LAMBDA NIL
    (PROG ((CurrentRecord (create parameterKind)))
          (if (MatchConstant (QUOTE (VAR FUNCTION PROCEDURE))
                             NIL)
              then (PROGN (CurrentRecord : LEXEME ← ProductionValue)
                          (RETURN CurrentRecord])
```

114

```
(parenExpr
  [LAMBDA NIL
    (PROG ((CurrentRecord (create parenExpr)))
          (if (MatchConstant (QUOTE (%()))
                             NIL)
              then (if (ProductionValue ←(expression))
                       then (PROGN (CurrentRecord : expression ← ProductionValue)
                                   (if (MatchConstant (QUOTE (%)))
                                                      NIL)
                                       then (RETURN CurrentRecord])
```

115

```
(pointerType
```

```
[LAMBDA NIL
  (PROG ((CurrentRecord (create pointerType)))
        (if (MatchConstant (QUOTE (↑))
                           NIL)
            then (if (ProductionValue ←(identifier))
                     then (PROGN (CurrentRecord : identifier ← ProductionValue)
                                 (RETURN CurrentRecord])
```

116

```
(prefixExpr
  [LAMBDA NIL
    (PROG ((CurrentRecord (create prefixExpr)))
          (if (ProductionValue ←(prefixOp))
              then (PROGN (CurrentRecord : prefixOp ← ProductionValue)
                          (if (MatchConstant (QUOTE (%())
                                             NIL)
                              then (if (a0073)
                                       then (if (MatchConstant (QUOTE (%)))
                                                NIL)
                                                then (if (a0072)
                                                         then (RETURN CurrentRecord])
```

117

```
(prefixOp
  [LAMBDA NIL
    (PROG ((CurrentRecord (create prefixOp)))
          (if (MatchLexeme)
              then (PROGN (CurrentRecord ←(prefixOpFilter CurrentRecord))
                          (RETURN CurrentRecord])
```

118

```
(primary
  [LAMBDA NIL
    (PROG ((CurrentRecord (create primary)))
          (if (a0074)
              then (RETURN CurrentRecord])
```

119

```
(procedureOrFunctionDeclaration
  [LAMBDA NIL
    (PROG ((CurrentRecord (create procedureOrFunctionDeclaration)))
          (if (MatchConstant (QUOTE (INLINE))
                             T)
              then
               (if (ProductionValue ←(unitKind))
                   then
                    (PROGN
                      (CurrentRecord : unitKind ← ProductionValue)
                      (if (ProductionValue ←(identifier))
                          then
                           (PROGN
                             (CurrentRecord : identifier ← ProductionValue)
                             (if (a0102)
                                 then
                                  (if (MatchConstant (QUOTE (RETURNS))
                                                     T)
                                      then
                                       (if (a0101)
                                           then
                                            (if (a0100)
                                                then (if (a0099)
                                                         then
                                                          (if (a0098)
                                                              then (if (MatchConstant
                                                                         (QUOTE (;))
                                                                       NIL)
                                                                       then
                                                                        (if (ProductionValue
                                                                              ←(block))
                                                                            then (PROGN (CurrentRecord
                                                                                          : block ←
                                                                                          ProductionValue)
                                                                                        (RETURN
```

```
                                                                CurrentRecord])
```

120

```
(procedureStatement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create procedureStatement)))
          (if (ProductionValue ←(identifier))
              then (PROGN (CurrentRecord : identifier ← ProductionValue)
                          (if (a0105)
                              then (if (a0104)
                                       then (if (a0103)
                                                then (RETURN CurrentRecord])
```

121

```
(program
  [LAMBDA NIL
    (PROG ((CurrentRecord (create program)))
          (if (a0062)
              then (if (MatchConstant (QUOTE (;))
                                      T)
                       then (if (MatchConstant (QUOTE (%.))
                                               T)
                                then (RETURN CurrentRecord])
```

122

```
(proveStatement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create proveStatement)))
          (if (MatchConstant (QUOTE (PROVE))
                             NIL)
              then (if (ProductionValue ←(assertion))
                       then (PROGN (CurrentRecord : assertion ← ProductionValue)
                                   (RETURN CurrentRecord])
```

123

```
(qualifier
  [LAMBDA NIL
    (PROG ((CurrentRecord (create qualifier)))
          (if (a0106)
              then (RETURN CurrentRecord])
```

124

```
(quantifiedExpression
  [LAMBDA NIL
    (PROG ((CurrentRecord (create quantifiedExpression)))
          (if (ProductionValue ←(quantifier))
              then (PROGN (CurrentRecord : quantifier ← ProductionValue)
                          (if (ProductionValue ←(identifier))
                              then (PROGN (do (CurrentRecord : identifier ←(NCONC1 CurrentRecord :
                                                                                   identifier
                                                                                   ProductionValue))
                                              (SIMark) repeatwhile (AND (MatchConstant
                                                                          (QUOTE (,))
                                                                          NIL)
                                                                        (ProductionValue ←(identifier)
                                                                                         )
                                                                        (PROGN (SIUnmark)
                                                                               T)))
                                          (SIToMark)
                                          (SIUnmark)
                                          (if (MatchConstant (QUOTE (%())
                                                             NIL)
                                              then (if (ProductionValue ←(expression))
                                                       then (PROGN (CurrentRecord : expression ←
                                                                                    ProductionValue)
                                                                   (if (MatchConstant
                                                                         (QUOTE (%)))
                                                                       NIL)
                                                                   then (RETURN CurrentRecord])
```

125

```
(quantifier
  [LAMBDA NIL
    (PROG ((CurrentRecord (create quantifier)))
          (if (MatchConstant (QUOTE (ALL FORALL SOME EXISTS))
                             NIL)
              then (PROGN (CurrentRecord : LEXEME ← ProductionValue)
                          (RETURN CurrentRecord])
```

126

```
(range
  [LAMBDA NIL
    (PROG ((CurrentRecord (create range)))
          (if (a0076)
              then (RETURN CurrentRecord])
```

127

```
(rangeSpec
  [LAMBDA NIL
    (PROG ((CurrentRecord (create rangeSpec)))
          (if (MatchConstant (QUOTE (%|))
                             NIL)
              then (if (ProductionValue ←(range))
                       then (PROGN (do (CurrentRecord : range ←(NCONC1 CurrentRecord : range
                                                                       ProductionValue))
                                       (SIMark) repeatwhile (AND (MatchConstant (QUOTE (.))
                                                                                NIL)
                                                                 (ProductionValue ←(range))
                                                                 (PROGN (SIUnmark)
                                                                        T)))
                                   (SIToMark)
                                   (SIUnmark)
                                   (if (MatchConstant (QUOTE (%|))
                                                      NIL)
                                       then (RETURN CurrentRecord])
```

128

```
(rangedInterfaceList
  [LAMBDA NIL
    (PROG ((CurrentRecord (create rangedInterfaceList)))
          (if (ProductionValue ←(rangedOp))
              then (PROGN (do (CurrentRecord : rangedOp ←(NCONC1 CurrentRecord : rangedOp
                                                                 ProductionValue))
                              (SIMark) repeatwhile (AND (MatchConstant (QUOTE (,))
                                                                       NIL)
                                                        (ProductionValue ←(rangedOp))
                                                        (PROGN (SIUnmark)
                                                               T)))
                          (SIToMark)
                          (SIUnmark)
                          (RETURN CurrentRecord])
```

129

```
(rangedOp
  [LAMBDA NIL
    (PROG ((CurrentRecord (create rangedOp)))
          (if (a0078)
              then (if (a0077)
                       then (RETURN CurrentRecord])
```

130

```
(recordSection
  [LAMBDA NIL
    (PROG ((CurrentRecord (create recordSection)))
          (if (ProductionValue ←(identifier))
              then (PROGN (do (CurrentRecord : identifier ←(NCONC1 CurrentRecord : identifier
                                                                   ProductionValue))
                              (SIMark) repeatwhile (AND (MatchConstant (QUOTE (,))
                                                                       NIL)
```

```
                                                    (ProductionValue ←(identifier))
                                                    (PROGN (SIUnmark)
                                                           T)))
                                (SIToMark)
                                (SIUnmark)
                                (if (MatchConstant (QUOTE (:))
                                                   NIL)
                                    then (if (ProductionValue ←(type))
                                             then (PROGN (CurrentRecord : type ← ProductionValue)
                                                         (RETURN CurrentRecord])
```

131

```
(recordType
  [LAMBDA NIL
    (PROG ((CurrentRecord (create recordType)))
          (if (MatchConstant (QUOTE (RECORD))
                             NIL)
              then (if (ProductionValue ←(fieldList))
                       then (PROGN (CurrentRecord : fieldList ← ProductionValue)
                                   (if (MatchConstant (QUOTE (END))
                                                      NIL)
                                       then (RETURN CurrentRecord])
```

132

```
(repeatStatement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create repeatStatement)))
          (if (MatchConstant (QUOTE (REPEAT))
                             NIL)
              then (if (ProductionValue ←(statement))
                       then (PROGN (do (CurrentRecord : statement ←(NCONC1 CurrentRecord : statement
                                                                           ProductionValue))
                                       (SIMark) repeatwhile (AND (MatchConstant (QUOTE (:))
                                                                                NIL)
                                                                 (ProductionValue ←(statement))
                                                                 (PROGN (SIUnmark)
                                                                        T)))
                                   (SIToMark)
                                   (SIUnmark)
                                   (if (MatchConstant (QUOTE (UNTIL))
                                                      NIL)
                                       then (if (ProductionValue ←(expression))
                                                then (PROGN (CurrentRecord : expression ←
                                                                           ProductionValue)
                                                            (if (a0107)
                                                                then (RETURN CurrentRecord])
```

133

```
(returnStatement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create returnStatement)))
          (if (MatchConstant (QUOTE (RETURN))
                             NIL)
              then (if (a0109)
                       then (if (a0108)
                                then (RETURN CurrentRecord])
```

134

```
(rule
  [LAMBDA NIL
    (PROG ((CurrentRecord (create rule)))
          (if (ProductionValue ←(expression))
              then (PROGN (CurrentRecord : expression ← ProductionValue)
                          (if (a0080)
                              then (RETURN CurrentRecord])
```

135

```
(ruleSeq
  [LAMBDA NIL
    (PROG ((CurrentRecord (create ruleSeq)))
          (if (ProductionValue ←(rule))
```

```
            then (PROGN (do (CurrentRecord : rule ←(NCONC1 CurrentRecord : rule ProductionValue))
                            (SIMark) repeatwhile (AND (MatchConstant (QUOTE (.))
                                                              NIL)
                                                      (ProductionValue ←(rule))
                                                      (PROGN (SIUnmark)
                                                             T)))
                        (SIToMark)
                        (SIUnmark)
                        (RETURN CurrentRecord])
```

136

```
(scalarType
  [LAMBDA NIL
    (PROG ((CurrentRecord (create scalarType)))
          (if (MatchConstant (QUOTE (%())
                             NIL)
              then (if (ProductionValue ←(identifier))
                       then (PROGN (do (CurrentRecord : identifier ←(NCONC1 CurrentRecord :
                                                                             identifier
                                                                             ProductionValue))
                                       (SIMark) repeatwhile (AND (MatchConstant (QUOTE (.))
                                                                                NIL)
                                                                 (ProductionValue ←(identifier))
                                                                 (PROGN (SIUnmark)
                                                                        T)))
                                   (SIToMark)
                                   (SIUnmark)
                                   (if (MatchConstant (QUOTE (%)))
                                                      NIL)
                                       then (RETURN CurrentRecord])
```

137

```
(setType
  [LAMBDA NIL
    (PROG ((CurrentRecord (create setType)))
          (if (MatchConstant (QUOTE (SET))
                             NIL)
              then (if (MatchConstant (QUOTE (OF))
                                      NIL)
                       then (if (ProductionValue ←(simpleType))
                                then (PROGN (CurrentRecord : simpleType ← ProductionValue)
                                            (RETURN CurrentRecord])
```

138

```
(simpleStatement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create simpleStatement)))
          (if (a0110)
              then (RETURN CurrentRecord])
```

139

```
(simpleType
  [LAMBDA NIL
    (PROG ((CurrentRecord (create simpleType)))
          (if (a0111)
              then (RETURN CurrentRecord])
```

140

```
(specialPrefixExpr
  [LAMBDA NIL
    (PROG ((CurrentRecord (create specialPrefixExpr)))
          (if (ProductionValue ←(specialPrefixOp))
              then (PROGN (CurrentRecord : specialPrefixOp ← ProductionValue)
                          (if (ProductionValue ←(primary))
                              then (PROGN (CurrentRecord : primary ← ProductionValue)
                                          (RETURN CurrentRecord])
```

141

```
(specialPrefixOp
```

```
  [LAMBDA NIL
    (PROG ((CurrentRecord (create specialPrefixOp)))
          (if (MatchLexeme)
              then (PROGN (CurrentRecord ←(specialPrefixOpFilter CurrentRecord))
                          (RETURN CurrentRecord])
```

142

```
(statement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create statement)))
          (if (a0112)
              then (PROGN (CurrentRecord ←(simplifyStatement CurrentRecord))
                          (RETURN CurrentRecord])
```

143

```
(structuredType
  [LAMBDA NIL
    (PROG ((CurrentRecord (create structuredType)))
          (if (a0113)
              then (if (ProductionValue ←(unpackedStructuredType))
                       then (PROGN (CurrentRecord : unpackedStructuredType ← ProductionValue)
                                   (RETURN CurrentRecord])
```

144

```
(subrangeType
  [LAMBDA NIL
    (PROG ((CurrentRecord (create subrangeType)))
          (if (a0115)
              then (if (MatchConstant (QUOTE (%.))
                                      NIL)
                       then (if (MatchConstant (QUOTE (%.))
                                               NIL)
                                then (if (a0114)
                                         then (RETURN CurrentRecord])
```

145

```
(type
  [LAMBDA NIL
    (PROG ((CurrentRecord (create type)))
          (if (a0116)
              then (RETURN CurrentRecord])
```

146

```
(typeDefinition
  [LAMBDA NIL
    (PROG ((CurrentRecord (create typeDefinition)))
          (if (ProductionValue ←(identifier))
              then (PROGN (CurrentRecord : identifier ← ProductionValue)
                          (if (MatchConstant (QUOTE (=))
                                             NIL)
                              then (if (ProductionValue ←(type))
                                       then (PROGN (CurrentRecord : type ← ProductionValue)
                                                   (RETURN CurrentRecord])
```

147

```
(typeIdentifier
  [LAMBDA NIL
    (PROG ((CurrentRecord (create typeIdentifier)))
          (if (ProductionValue ←(identifier))
              then (PROGN (CurrentRecord : identifier ← ProductionValue)
                          (RETURN CurrentRecord])
```

148

```
(unitKind
  [LAMBDA NIL
    (PROG ((CurrentRecord (create unitKind)))
          (if (MatchConstant (QUOTE (PROCEDURE FUNCTION PROGRAM))
```

```
                    NIL)
            then (PROGN (CurrentRecord : LEXEME ← ProductionValue)
                        (RETURN CurrentRecord])
```

149

```
(unpackedStructuredType
  [LAMBDA NIL
    (PROG ((CurrentRecord (create unpackedStructuredType)))
          (if (a0117)
              then (RETURN CurrentRecord])
```

150

```
(varDeclaration
  [LAMBDA NIL
    (PROG ((CurrentRecord (create varDeclaration)))
          (if (ProductionValue ←(varDeclarePart))
              then (PROGN (do (CurrentRecord : varDeclarePart ←(NCONC1 CurrentRecord : varDeclarePart
                                                                       ProductionValue))
                              (SlMark) repeatwhile (AND (MatchConstant (QUOTE (,))
                                                                       NIL)
                                                        (ProductionValue ←(varDeclarePart))
                                                        (PROGN (SlUnmark)
                                                               T)))
                          (SlToMark)
                          (SlUnmark)
                          (if (MatchConstant (QUOTE (:))
                                             NIL)
                              then (if (ProductionValue ←(type))
                                       then (PROGN (CurrentRecord : type ← ProductionValue)
                                                   (RETURN CurrentRecord])
```

151

```
(varDeclarePart
  [LAMBDA NIL
    (PROG ((CurrentRecord (create varDeclarePart)))
          (if (ProductionValue ←(identifier))
              then (PROGN (CurrentRecord : identifier ← ProductionValue)
                          (if (a0119)
                              then (if (a0118)
                                       then (RETURN CurrentRecord])
```

152

```
(variable
  [LAMBDA NIL
    (PROG ((CurrentRecord (create variable)))
          (if (ProductionValue ←(identifier))
              then (PROGN (CurrentRecord : identifier ← ProductionValue)
                          (RETURN CurrentRecord])
```

153

```
(variableDecl
  [LAMBDA NIL
    (PROG ((CurrentRecord (create variableDecl)))
          (if (ProductionValue ←(identifier))
              then (PROGN (do (CurrentRecord : identifier ←(NCONC1 CurrentRecord : identifier
                                                                   ProductionValue))
                              (SlMark) repeatwhile (AND (MatchConstant (QUOTE (,))
                                                                       NIL)
                                                        (ProductionValue ←(identifier))
                                                        (PROGN (SlUnmark)
                                                               T)))
                          (SlToMark)
                          (SlUnmark)
                          (if (MatchConstant (QUOTE (:))
                                             NIL)
                              then (if (ProductionValue ←(expression))
                                       then (PROGN (CurrentRecord : expression ← ProductionValue)
                                                   (RETURN CurrentRecord])
```

154

```
(variant
  [LAMBDA NIL
    (PROG ((CurrentRecord (create variant)))
          (if (a0120)
              then (RETURN CurrentRecord])
```

155

```
(variantPart
  [LAMBDA NIL
    (PROG ((CurrentRecord (create variantPart)))
          (if (MatchConstant (QUOTE (CASE))
                             NIL)
              then (if (a0121)
                       then (if (ProductionValue ←(typeIdentifier))
                                then
                                 (PROGN (CurrentRecord : typeIdentifier ← ProductionValue)
                                        (if (MatchConstant (QUOTE (OF))
                                                           NIL)
                                            then (if (ProductionValue ←(variant))
                                                     then
                                                      (PROGN (do (CurrentRecord : variant ←(NCONC1
                                                                                            CurrentRecord :
                                                                                            variant
                                                                                            ProductionValue));
                                                                 (SlMark)
                                                                 repeatwhile
                                                                  (AND (MatchConstant (QUOTE (;))
                                                                                      NIL)
                                                                       (ProductionValue ←(variant))
                                                                       (PROGN (SlUnmark)
                                                                              T)))
                                                             (SlToMark)
                                                             (SlUnmark)
                                                             (RETURN CurrentRecord])
```

156

```
(whileStatement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create whileStatement)))
          (if (a0123)
              then (if (MatchConstant (QUOTE (WHILE))
                                      NIL)
                       then (if (ProductionValue ←(expression))
                                then (PROGN (CurrentRecord : expression ← ProductionValue)
                                            (if (MatchConstant (QUOTE (DO))
                                                               NIL)
                                                then (if (ProductionValue ←(statement))
                                                         then (PROGN (CurrentRecord : statement ←
                                                                                      ProductionValue)
                                                                     (if (a0122)
                                                                         then (RETURN CurrentRecord])
```

157

```
(withStatement
  [LAMBDA NIL
    (PROG ((CurrentRecord (create withStatement)))
          (if (MatchConstant (QUOTE (WITH))
                             NIL)
              then (if (ProductionValue ←(variable))
                       then (PROGN (do (CurrentRecord : variable ←(NCONC1 CurrentRecord : variable
                                                                   ProductionValue))
                                       (SlMark) repeatwhile (AND (MatchConstant (QUOTE (,))
                                                                                NIL)
                                                                 (ProductionValue ←(variable))
                                                                 (PROGN (SlUnmark)
                                                                        T)))
                                   (SlToMark)
                                   (SlUnmark)
                                   (if (MatchConstant (QUOTE (DO))
                                                      NIL)
                                       then (if (ProductionValue ←(statement))
                                                then (PROGN (CurrentRecord : statement ←
                                                                             ProductionValue)
                                                            (RETURN CurrentRecord])
```

)
[DECLARE DONTEVAL@LOAD DONTCOPY

(* Standard Parser Input Interface. This function must be compiled.)]

[DECLARE: DONTLOAD DONTCOPY

```
(* Universal record definitions) ]

[DECLARE: EVAL@COMPILE

(RECORD COMMONTYPE (SYNTACTICTYPE))

(ACCESSFNS COMMONSUBTYPE (SYNTACTICSUBTYPE (DATUM : ALTERNATIVESUBNODE : SYNTACTICTYPE)
                                           (DATUM : ALTERNATIVESUBNODE : SYNTACTICTYPE ← NEWVALUE)))

(TYPERECORD all (LEXEME))

(TYPERECORD arrayType (NIL NIL type simpleType))

(TYPERECORD assertStatement (NIL NIL assertion))

(TYPERECORD assertion (expression))

(TYPERECORD assignmentStatement (expression NIL NIL NIL variable))

(TYPERECORD assumeStatement (NIL NIL assertion))

(TYPERECORD block (declareopt compoundStatement assertion NIL NIL assertion#))

(TYPERECORD bracketExprList (expression))

(TYPERECORD caseElementList (NIL NIL NIL statement caseLabel))

(TYPERECORD caseLabel (constant))

(TYPERECORD caseStatement (expression caseElementList NIL statement))

(TYPERECORD compoundStatement (NIL NIL NIL statement))

(TYPERECORD concurrentAssignmentStatement (expression NIL NIL NIL variable))

(TYPERECORD constDefinition (expression identifier))

(TYPERECORD constant (LEXEME))

(TYPERECORD coord (number number# all lastOne firstOne))

(TYPERECORD declareType (constDefinition typeDefinition varDeclaration label
                                         procedureOrFunctionDeclaration))

(TYPERECORD declareopt (declareType))

(TYPERECORD denotePair (expression identifier))

(TYPERECORD denoteSpec (denotePair))

(TYPERECORD direction (LEXEME))

(TYPERECORD expression (expression primary infixOp))

(TYPERECORD expressionSeq (expression))

(TYPERECORD fieldList (recordSection variantPart))

(TYPERECORD fileType (NIL NIL type))

(TYPERECORD firstOne (LEXEME))

(TYPERECORD forStatement (expression identifier assertion statement expression# assertion# direction))

(TYPERECORD formalParameterSection (parameterKind parameterGroup))

(TYPERECORD functionDecl (expression NIL NIL NIL expression#))

(TYPERECORD goToStatement (NIL NIL assertion label))

(TYPERECORD greaterThanEqual (LEXEME))

(TYPERECORD identifier (LEXEME))

(TYPERECORD identifierSeq (NIL identifier))

(TYPERECORD ifExpr (expression expression## NIL NIL expression#))

(TYPERECORD ifStatement (expression statement# NIL statement))

(TYPERECORD infixOp (ALTERNATIVESUBNODE))
```

(TYPERECORD ***interfaceList*** (op))

(TYPERECORD ***label*** (LEXEME))

(TYPERECORD ***labelStatement*** (NIL NIL NIL label simpleStatement))

(TYPERECORD ***lastOne*** (LEXEME))

(TYPERECORD ***lessThanEqual*** (LEXEME))

(TYPERECORD ***machinePair*** (identifierSeq identifierSeq#))

(TYPERECORD ***machineSpec*** (machinePair))

(TYPERECORD ***normalInfixOp*** (LEXEME))

(TYPERECORD ***notEqual*** (LEXEME))

(TYPERECORD ***number*** (LEXEME))

(TYPERECORD ***op*** (ALTERNATIVESUBNODE))

(TYPERECORD ***packed*** (LEXEME))

(TYPERECORD ***parameterGroup*** (NIL identifier type))

(TYPERECORD ***parameterKind*** (LEXEME))

(TYPERECORD ***parenExpr*** (expression))

(TYPERECORD ***pointerType*** (NIL identifier))

(TYPERECORD ***prefixExpr*** (expression identifier prefixOp))

(TYPERECORD ***prefixOp*** (LEXEME))

(TYPERECORD ***primary*** (ALTERNATIVESUBNODE))

(TYPERECORD ***procedureOrFunctionDeclaration*** (unitKind identifier type formalParameterSection
formalParameterSection# block identifier#
identifier##))

(TYPERECORD ***procedureStatement*** (expression identifier NIL NIL variable NIL identifier#))

(TYPERECORD ***program*** (NIL NIL NIL NIL procedureOrFunctionDeclaration block))

(TYPERECORD ***proveStatement*** (NIL NIL assertion))

(TYPERECORD ***qualifier*** (expression identifier))

(TYPERECORD ***quantifiedExpression*** (expression identifier quantifier))

(TYPERECORD ***quantifier*** (LEXEME))

(TYPERECORD ***range*** (coord coord#))

(TYPERECORD ***rangeSpec*** (range))

(TYPERECORD ***rangedInterfaceList*** (rangedOp))

(TYPERECORD ***rangedOp*** (op rangeSpec))

(TYPERECORD ***recordSection*** (NIL identifier type))

(TYPERECORD ***recordType*** (NIL NIL NIL NIL NIL fieldList))

(TYPERECORD ***repeatStatement*** (expression NIL assertion statement))

(TYPERECORD ***returnStatement*** (expression NIL assertion))

(TYPERECORD ***rule*** (expression NIL NIL NIL expression#))

(TYPERECORD ***ruleSeq*** (rule))

(TYPERECORD ***scalarType*** (NIL identifier))

(TYPERECORD ***setType*** (NIL NIL NIL simpleType))

(TYPERECORD ***simpleStatement*** (ALTERNATIVESUBNODE))

```
(TYPERECORD simpleType (ALTERNATIVESUBNODE))
(TYPERECORD specialPrefixExpr (specialPrefixOp primary))
(TYPERECORD specialPrefixOp (LEXEME))
(TYPERECORD statement (assignmentStatement labelStatement NIL NIL simpleStatement))
(TYPERECORD structuredType (packed unpackedStructuredType))
(TYPERECORD subrangeType (expression NIL NIL NIL expression#))
(TYPERECORD type (ALTERNATIVESUBNODE))
(TYPERECORD typeDefinition (NIL identifier type))
(TYPERECORD typeIdentifier (NIL identifier))
(TYPERECORD unitKind (LEXEME))
(TYPERECORD unpackedStructuredType (ALTERNATIVESUBNODE))
(TYPERECORD varDeclaration (varDeclarePart NIL type))
(TYPERECORD varDeclarePart (expression identifier NIL NIL expression#))
(TYPERECORD variable (NIL identifier))
(TYPERECORD variableDecl (expression identifier))
(TYPERECORD variant (NIL NIL NIL NIL caseLabel fieldList))
(TYPERECORD variantPart (typeIdentifier identifier variant))
(TYPERECORD whileStatement (expression NIL assertion statement NIL assertion#))
(TYPERECORD withStatement (NIL NIL NIL statement variable))
]
[DECLARE: DONTEVAL@LOAD DONTCOPY
```

(* Macro used to generate input values of NIL when generator halts.)]

```
(RPAQ DWIMIFYCOMPFLG T)
[DECLARE: DONTEVAL@LOAD DONTCOPY
```

(* The filter file must contain the following functions: identifierFilter infixOpFilter
unsignedInteger prefixOpFilter specialPrefixOpFilter labelFilter simplifyStatement)

[DECLARE: DONTEVAL@LOAD DONTCOPY

(* If the user redefines the production records, currently represented as TYPERECORDS, he must redefine the CopyTopRecord function and compiler macro consistent with it.) |

|DECLARE: DONTEVAL@LOAD DONTCOPY

(* The following LOAD and definition should normally be replaced by a load from a file defining UsersNextInput and an input initialization routine)]

[DECLARE: DONTEVALLOAD DONTCOPY

```
(* The user might consider eliminating some of the BLKAPPLYFNS if he can predict the roots which
   might be called externally.)

[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
(BLOCK: PARSERBLK PARSER a0062 a0063 a0064 a0065 a0066 a0067 a0068 a0069 a0070 a0071 a0072 a0073
        a0074 a0075 a0076 a0077 a0078 a0079 a0080 a0081 a0082 a0083 a0084 a0085 a0086 a0087 a0088
        a0089 a0090 a0091 a0092 a0093 a0094 a0095 a0096 a0097 a0098 a0099 a0100 a0101 a0102 a0103
        a0104 a0105 a0106 a0107 a0108 a0109 a0110 a0111 a0112 a0113 a0114 a0115 a0116 a0117 a0118
        a0119 a0120 a0121 a0122 a0123 all arrayType assertStatement assertion assignmentStatement
        assumeStatement block bracketExprList caseElementList caseLabel caseStatement
        compoundStatement concurrentAssignmentStatement constDefinition constant coord declareType
        declareopt denotePair denoteSpec direction expression expressionSeq fieldList fileType
        firstOne forStatement formalParameterSection functionDecl goToStatement greaterThanEqual
        identifier identifierSeq ifExpr ifStatement infixOp interfaceList label labelStatement
        lastOne lessThanEqual machinePair machineSpec normalInfixOp notEqual number op packed
        parameterGroup parameterKind parenExpr pointerType prefixExpr prefixOp primary
        procedureOrFunctionDeclaration procedureStatement program proveStatement qualifier
        quantifiedExpression quantifier range rangeSpec rangedInterfaceList rangedOp recordSection
        recordType repeatStatement returnStatement rule ruleSeq scalarType setType simpleStatement
        simpleType specialPrefixExpr specialPrefixOp statement structuredType subrangeType type
        typeDefinition typeIdentifier unitKind unpackedStructuredType varDeclaration varDeclarePart
        variable variableDecl variant variantPart whileStatement withStatement CopyTopRecord
        MatchConstant MatchLexeme PARSEPROGRAM PARSE\ASSERTION ParserRATOM ReadAtom SIFromStack?
        SIMark SINewStack SINext SINextNew SINextSaved SISaveLexeme SIToMark SIUnmark UsersNextInput
        identifierFilter infixOpFilter labelFilter prefixOpFilter simplifyStatement
        specialPrefixOpFilter unsignedInteger (ENTRIES PARSER PARSEPROGRAM PARSE\ASSERTION)
        (SPECVARS CurrentRecord InputFileHandle CurrentLexeme ProductionValue Terminator SIEntries
                  OutputStream)
        (BLKAPPLYFNS program all bracketExprList coord denoteSpec denotePair expression expressionSeq
                     firstOne functionDecl greaterThanEqual identifier identifierSeq ifExpr infixOp
                     lastOne lessThanEqual machinePair machineSpec normalInfixOp notEqual number op
                     parenExpr prefixExpr prefixOp primary quantifiedExpression quantifier range
                     rangedInterfaceList rangedOp rangeSpec rule ruleSeq specialPrefixExpr
                     specialPrefixOp variable variableDecl interfaceList arrayType assertion
                     assertStatement assignmentStatement assumeStatement block caseElementList
                     caseLabel caseStatement compoundStatement concurrentAssignmentStatement constant
                     constDefinition declareopt declareType direction fieldList fileType
                     formalParameterSection forStatement goToStatement ifStatement label
                     labelStatement packed parameterGroup parameterKind pointerType
                     procedureOrFunctionDeclaration procedureStatement proveStatement qualifier
                     recordSection recordType repeatStatement returnStatement scalarType setType
                     simpleStatement simpleType statement structuredType subrangeType type
                     typeDefinition typeIdentifier unitKind unpackedStructuredType varDeclaration
                     varDeclarePart variant variantPart whileStatement withStatement))
]

(RPAQQ PARSERPLUSFNS (CopyTopRecord MatchConstant MatchLexeme PARSEPROGRAM PARSE\ASSERTION
                                    ParserRATOM ReadAtom SIFromStack? SIMark SINewStack SINext
                                    SINextNew SINextSaved SISaveLexeme SIToMark SIUnmark
                                    UsersNextInput identifierFilter infixOpFilter labelFilter
                                    prefixOpFilter simplifyStatement specialPrefixOpFilter
                                    unsignedInteger))
(DEFINEQ

                                                                                        158

(CopyTopRecord
  [LAMBDA (x)
    < ! x>])

                                                                                        159

(MatchConstant
  [LAMBDA (constants emptyOK)
    (PROGN (if CurrentLexeme MEMB constants
               then ProductionValue←CurrentLexeme
                    (SINext)
                    ProductionValue
           else emptyOK])

                                                                                        160

(MatchLexeme
  [LAMBDA NIL
    (if CurrentLexeme~=Terminator
        then CurrentRecord:LEXEME←CurrentLexeme
             (SINext)
             T])
```

161

```
(PARSEPROGRAM
  [LAMBDA (FileName)
    (PROG [Value (ERRORTYPELST ('((16 (PROGN BREAKCHK←NIL
                                             PRINTMSG←NIL
                                             (RETFROM ERRORPOS 'STOP 1]
          (if FileName=NIL
              then FileName←T)
          (PARSEDTOKENS← <NIL>)
          (if Value←(PARSER 'program <FileName 'STOP > T)=NIL
              then (if FileName=T
                       then (CLEARBUF T)))
          (RETURN (reduceParseTree Value])
```

162

```
(PARSE\ASSERTION
  [LAMBDA NIL
    (PROG (Value)
          (PARSEDTOKENS← <NIL>)
          (if Value←(PARSER 'expression <T '; > NIL)=NIL
              then (CLEARBUF T))
          (RETURN (reduceExpression Value])
```

163

```
(ParserRATOM
  [LAMBDA (filename FileNameStopAtom ReadTable)          (* R.Bates "19-OCT-78 10:06")
    (PROG (temp)
          (RETURN (if filename=NIL
                      then FileNameStopAtom:2
                    elseif (LISTP filename)
                      then temp←FileNameStopAtom:1:1
                           FileNameStopAtom:1←FileNameStopAtom:1::1
                           temp
                    else (RATOM filename ReadTable])
```

164

```
(ReadAtom
  [LAMBDA (FileNameStopAtom)                             (* R.Erickson "4-Jul-81 18:33")
                                                         (* R.Bates "19-SEP-78 01:07")
    (PROG (atomgot upatom filename)
          (filename←FileNameStopAtom:1)
      TOP (if PARSERPROMPT and filename=T
              then (PROMPTCHAR PARSERPROMPT))
          (atomgot←(ParserRATOM filename FileNameStopAtom PASCAL\READ\TABLE))
          (if atomgot='%"
              then (bind (atomstring) first atomstring←"" eachtime atomgot←(ParserRATOM filename
                                                                                FileNameStopAtom
                                                                              PASCAL\READ\TABLE)
                         while atomgot~='%" do atomstring←(CONCAT atomstring " " atomgot)
                         finally atomgot←(MKATOM atomstring))
            elseif atomgot='{
              then (while atomgot~='} do atomgot←(ParserRATOM filename FileNameStopAtom
                                                                  PASCAL\READ\TABLE))
                   (GO TOP)
            elseif upatom←(UCaseToList atomgot UCaseParseAtoms)
              then atomgot←upatom)
          (if COLLECTTOKENS
              then (TCONC PARSEDTOKENS atomgot))
          (if PARSERTRACE
              then (PRIN1 atomgot)
                   (if ~(MEMB atomgot '(: = < > %( %)))
                       then (PRIN1 " "))
                   (if (MEMB atomgot '(; AND &))
                       then (TERPRI)))
          (RETURN (if (MEMB atomgot FileNameStopAtom::1)
                      then FileNameStopAtom
                    else atomgot])
```

165

```
(SIFromStack?
```

```
  [LAMBDA NIL OutputStream::1])
```

166

```
(SIMark
  [LAMBDA NIL SIEntries← <OutputStream ! SIEntries>])
```

167

```
(SINewStack
  [LAMBDA NIL
    (PROGN SIEntries←NIL
           OutputStream← <NIL>])
```

168

```
(SINext
  [LAMBDA NIL
    (if (SIFromStack?)
        then (SINextSaved)
      else (SINextNew])
```

169

```
(SINextNew
  [LAMBDA NIL
    (PROGN (if CurrentLexeme~=Terminator
               then CurrentLexeme←(UsersNextInput InputFileHandle))
           (if CurrentLexeme=InputFileHandle
               then CurrentLexeme←Terminator)
           (SISaveLexeme])
```

170

```
(SINextSaved
  [LAMBDA NIL
    (PROGN OutputStream←OutputStream::1
           CurrentLexeme←OutputStream:1])
```

171

```
(SISaveLexeme
  [LAMBDA NIL
    (PROGN OutputStream::1← <CurrentLexeme> OutputStream←OutputStream::1])
```

172

```
(SIToMark
  [LAMBDA NIL
    (PROGN OutputStream←SIEntries:1
           CurrentLexeme←OutputStream:1])
```

173

```
(SIUnmark
  [LAMBDA NIL SIEntries←SIEntries::1])
```

174

```
(UsersNextInput
  [LAMBDA (FileNameStopAtom)
    (ReadAtom FileNameStopAtom])
```

175

```
(identifierFilter
  [LAMBDA (x)                                          (* R.Bates "17-Sep-79 13:51")
    x←x:2
    (if (NUMBERP x) or (MEMB x ReserveWordList) or (MEMB x SpecialPrefixOps) or (MEMB x Delimiters)
        or (MEMB x '(< = > :))
      then NIL
```

```
    else x])
```

176

```
(infixOpFilter
  [LAMBDA (x)
    (if (MEMB x:LEXEME KeyWordList) or (MEMB x:LEXEME Delimiters)
        then NIL
      else x:LEXEME])
```

177

```
(labelFilter
  [LAMBDA (X)
    (OR (unsignedInteger X)
        (identifierFilter X])
```

178

```
(prefixOpFilter
  [LAMBDA (x)                                                    (* R.Bates "17-Sep-79 13:46")
    (if (MEMB x:2 KeyWordList) or (MEMB x:2 Delimiters) or (MEMB x:LEXEME '(< = : >))
        then NIL
      else x:2])
```

179

```
(simplifyStatement
  [LAMBDA (X)
    (PROG (value)
          (if value←X:assignmentStatement
              then (RETURN value)
            elseif value←X:labelStatement
              then value:simpleStatement←value:simpleStatement:ALTERNATIVESUBNODE
                   (RETURN value)
            elseif value←X:simpleStatement
              then (RETURN value:ALTERNATIVESUBNODE)
            else (RETURN X])
```

180

```
(specialPrefixOpFilter
  [LAMBDA (x)
    (if (MEMB x:LEXEME SpecialPrefixOps)
        then x:LEXEME
      else NIL])
```

181

```
(unsignedInteger
  [LAMBDA (X)
    (NUMBERP X:LEXEME])
)
(CLISPDEC (QUOTE FAST))
(DECLARE: DOEVAL@COMPILE

(PUTPROPS CopyTopRecord MACRO ((x)
                               (APPEND x)))

(PUTPROPS SIFromStack? MACRO (NIL (CDR OutputStream)))

(PUTPROPS SIMark MACRO (NIL (SETQ SIEntries (CONS OutputStream SIEntries))))

(PUTPROPS SINewStack MACRO [NIL (PROGN (SETQ SIEntries NIL)
                                       (SETQ OutputStream (LIST NIL])

(PUTPROPS SINext MACRO [NIL (COND
                             ((SIFromStack?)
                               (SINextSaved))
                             (T (SINextNew])

(PUTPROPS SINextNew MACRO (NIL (PROGN [COND
                                       ((NEQ CurrentLexeme Terminator)
                                         (SETQ CurrentLexeme (UsersNextInput InputFileHandle]
                                      (COND
```

```
                                        ((EQ CurrentLexeme InputFileHandle)
                                          (SETQ CurrentLexeme Terminator)))
                                      (SISaveLexeme))))

(PUTPROPS SINextSaved MACRO [NIL (PROGN (SETQ OutputStream (CDR OutputStream))
                                        (SETQ CurrentLexeme (CAR OutputStream])

(PUTPROPS SISaveLexeme MACRO [NIL (PROGN (RPLACD OutputStream (LIST CurrentLexeme))
                                         (SETQ OutputStream (CDR OutputStream])

(PUTPROPS SIToMark MACRO [NIL (PROGN (SETQ OutputStream (CAR SIEntries))
                                     (SETQ CurrentLexeme (CAR OutputStream])

(PUTPROPS SIUnmark MACRO (NIL (SETQ SIEntries (CDR SIEntries))))

(PUTPROPS UsersNextInput MACRO ((FileNameStopAtom)
                                (ReadAtom FileNameStopAtom)))

(PUTPROPS labelFilter MACRO [LAMBDA (X)
                            (OR (unsignedInteger X)
                                (identifierFilter X])
)

(RPAQQ Delimiters (%( %) , %. : ; @ %[ %] ↑ %|))

(RPAQQ KeyWordList (ALL ALTERS ARRAY ASSERT ASSERTING ASSUME BEGIN BY CASE CONST DO DOWNTO ELSE END
                        ENTRY EXISTS EXIT FILE FOR FORALL FUNCTION GO GOTO IF IMPORTS INLINE LABEL
                        MAINTAIN OF OTHERWISE PACKED POST PRE PROCEDURE PROGRAM PROVE PUBLIC RECORD
                        REPEAT RETURN RETURNS SET SOME THEN THUS TO TYPE UNTIL VAR WHILE WITH XPUBLIC)
)

(RPAQQ ReserveWordList (= ALL ALTERS AND ARRAY ASSERT ASSERTING ASSUME BEGIN BY CASE CONST DIFFERENCE
                          DIV DO DOWNTO ELSE END ENTRY EQ EQV EXISTS EXIT EXPT FILE FIRST FOR FORALL
                          FUNCTION GE GO GOTO GT IF IMP IMPORTS INLINE LABEL LAST LE LT MAINTAIN MAX
                          MIN MOD NE NOT OF OR OTHERWISE PACKED PLUS POST PRE PROCEDURE PROGRAM PROVE
                          PUBLIC RECORD REPEAT RETURN RETURNS SET SOME THEN THUS TIMES TO TYPE UNTIL
                          VAR WHILE WITH XPUBLIC))

(RPAQQ SpecialPrefixOps (! + - ~ NOT))

(RPAQQ UCaseParseAtoms (TRUE FALSE = ALL ALTERS AND ARRAY ASSERT ASSERTING ASSUME BEGIN BY CASE CONST
                             DIFFERENCE DIV DO DOWNTO ELSE END ENTRY EQ EQV EXISTS EXIT EXPT FILE
                             FIRST FOR FORALL FUNCTION GE GO GOTO GT IF IMP IMPORTS INLINE LABEL LAST
                             LE LT MAINTAIN MAX MIN MOD NE NOT OF OR OTHERWISE PACKED PLUS POST PRE
                             PROCEDURE PROGRAM PROVE PUBLIC RECORD REPEAT RETURN RETURNS SET SOME
                             THEN THUS TIMES TO TYPE UNTIL VAR WHILE WITH XPUBLIC))

(RPAQQ UpperCaseVars (TRUE FALSE))

(RPAQQ PARSERPROMPT ~>)

(RPAQ USESLOWERCASE T)

(RPAQ PARSERTRACE NIL)

(RPAQ COLLECTTOKENS T)
(SETQ PASCAL\READ\TABLE (COPYREADTABLE (QUOTE ORIG)))
(SETBRK (QUOTE (4 5 6 14 16 17 18 19 20 21 27 28 29 34 30 33 38 40 41 42 43 44 45 46 47 58 59 60 61
                  62 64 91 93 94 123 124 125 126))
        NIL PASCAL\READ\TABLE)
(DECLARE: DONTCOPY
  (FILEMAP (NIL (9289 108353 (PARSER 9301 . 11217) (a0062 11221 . 11923) (a0063 11927 . 12795) (a0064
12799 . 14276) (a0065 14280 . 14981) (a0066 14985 . 15646) (a0067 15650 . 16876) (a0068 16880 . 17544)
 (a0069 17548 . 18134) (a0070 18138 . 18860) (a0071 18864 . 20080) (a0072 20084 . 21249) (a0073 21253
. 22121) (a0074 22125 . 24408) (a0075 24412 . 25059) (a0076 25063 . 25717) (a0077 25721 . 26299) (
a0078 26303 . 26867) (a0079 26871 . 28301) (a0080 28305 . 28929) (a0081 28933 . 29527) (a0082 29531 .
30475) (a0083 30479 . 31222) (a0084 31226 . 31968) (a0085 31972 . 32550) (a0086 32554 . 33294) (a0087
33298 . 36318) (a0088 36322 . 36980) (a0089 36984 . 37867) (a0090 37871 . 38457) (a0091 38461 . 39119)
 (a0092 39123 . 39764) (a0093 39788 . 40450) (a0094 40454 . 41107) (a0095 41111 . 41769) (a0096 41773
. 42363) (a0097 42367 . 43011) (a0098 43015 . 43989) (a0099 43993 . 45204) (a0100 45208 . 45852) (
a0101 45856 . 46437) (a0102 46441 . 47561) (a0103 47565 . 48506) (a0104 48510 . 49678) (a0105 49682 .
50725) (a0106 50729 . 52148) (a0107 52152 . 52809) (a0108 52813 . 53475) (a0109 53479 . 54217) (a0110
54221 . 58424) (a0111 58428 . 59386) (a0112 59390 . 60518) (a0113 60522 . 61094) (a0114 61098 . 61722)
 (a0115 61726 . 62349) (a0116 62353 . 63310) (a0117 63314 . 64524) (a0118 64528 . 65258) (a0119 65262
. 65918) (a0120 65922 . 67164) (a0121 67168 . 67824) (a0122 67828 . 68486) (a0123 68490 . 69151) (all
69155 . 69387) (arrayType 69391 . 70414) (assertStatement 70418 . 70744) (assertion 70748 . 70985) (
assignmentStatement 70989 . 71534) (assumeStatement 71538 . 71864) (block 71868 . 72122) (
bracketExprList 72126 . 72436) (caseElementList 72440 . 73087) (caseLabel 73091 . 73324) (
caseStatement 73328 . 74483) (compoundStatement 74487 . 75183) (concurrentAssignmentStatement 75187 .
76378) (constDefinition 76382 . 76834) (constant 76838 . 76995) (coord 76999 . 77144) (declareType
77148 . 77305) (declareopt 77309 . 77635) (denotePair 77639 . 78082) (denoteSpec 78086 . 78613) (
```

```
(FILECREATED "28-Sep-81 14:37:26" <AFFIRM>DWIMPARSER..8 1278

     changes to:  DWIMPARSER

     previous date: "24-Feb-81 16:05:38" <AFFIRM>DWIMPARSER..6)


(PRETTYCOMPRINT DWIMPARSERCOMS)

(RPAQQ DWIMPARSERCOMS ((FNS * DWIMPARSERFNS)))

(RPAQQ DWIMPARSERFNS (DWIMPARSER))
(DEFINEQ

(DWIMPARSER
  [LAMBDA NIL                                        (* R.Erickson "28-Sep-81 14:34")

          (* * After GOP has been run, this dwimifies the parser and adds information from parserplus)


    (LOAD 'PARSER)
    NOSPELLFLG←T
    CLISPIFYPRETTYFLG←NIL
    (DWIMIFYFNS PARSERFNS)
    (DWIMIFYFNS PARSERPLUSFNS)
    (EDITE PARSERCOMS '(F (BLKAPPLYFNS --)
                          (DELETE (3 THRU -1))
                          (I N ParserRoots)
                          (BO -1)))                  (* The only productions which may be called upon
                                                     externally are the root. and those in ParserRoots.)

    (EDITE (GETD 'PARSER)
           '(F RETURN F QUOTE 2 (DELETE (2 THRU -1))
               (I N ParserRoots)
               (BO -1)))

          (* The function PARSER only tries to work if given a production which is BLKAPPLYable. We do an EDITE to avoid any
          adivce on EDITF.)


    (MAKEFILE 'PARSER.DWIM '(NEW NOCLISP])
)
(DECLARE: DONTCOPY
  (FILEMAP (NIL (303 1254 (DWIMPARSER 315 . 1251)))))
STOP
```

```
(FILECREATED "25-Sep-81 19:56:21" <AFFIRM>MINFS..2 1082

     changes to:  MINFSCOMS

     previous date: "29-Jun-81 16:40:32" <AFFIRM>MINFS..1)


(PRETTYCOMPRINT MINFSCOMS)

(RPAQQ MINFSCOMS [(VARS (SAVEDBFLG (QUOTE YES))
                        (LOADDBFLG (QUOTE YES)))
                  (P (MINFS 1000 (QUOTE ARRAYP))
                     (MINFS 512 (QUOTE SWPARRAYP))
                     (MINFS 512 (QUOTE STACKP))
                     (MINFS 10000 (QUOTE LISTP))
                     (MINFS 512 (QUOTE VCELLP))
                     (MINFS 1000 (QUOTE LITATOM))
                     (MINFS 512 (QUOTE FLOATP))
                     (MINFS 3000 (QUOTE FIXP))
                     (MINFS 512 (QUOTE STRINGP))
                     (MINFS 512 (QUOTE ATOM.CHARS))
                     (MINFS 512 (QUOTE STRING.CHARS])

(RPAQQ SAVEDBFLG YES)

(RPAQQ LOADDBFLG YES)
(MINFS 1000 (QUOTE ARRAYP))
(MINFS 512 (QUOTE SWPARRAYP))
(MINFS 512 (QUOTE STACKP))
(MINFS 10000 (QUOTE LISTP))
(MINFS 512 (QUOTE VCELLP))
(MINFS 1000 (QUOTE LITATOM))
(MINFS 512 (QUOTE FLOATP))
(MINFS 3000 (QUOTE FIXP))
(MINFS 512 (QUOTE STRINGP))
(MINFS 512 (QUOTE ATOM.CHARS))
(MINFS 512 (QUOTE STRING.CHARS))
(DECLARE: DONTCOPY
  (FILEMAP (NIL)))
STOP
```