ECMA
EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

FORMAL DEFINITION
of the
SYNTAX OF COBOL

September 1970
Free copies of this document are available from ECMA,
European Computer Manufacturers Association,
114 Rue du Rhône — 1204 Geneva (Switzerland)
FORMAL DEFINITION

of the

SYNTAX OF COBOL

September 1970
FORMAL DEFINITION

of the

SYNTAX OF COBOL
## CONTENTS

### PREFACE

PREFACE ix

### INTRODUCTION TO THE NOTATION

INTRODUCTION TO THE NOTATION xiii

### FORMAL DEFINITION OF COBOL SYNTAX

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>G. Syntactic Definitions of General Nature</td>
<td></td>
</tr>
<tr>
<td>EMPTY</td>
<td>1</td>
</tr>
<tr>
<td>COBOL GRAPHICS</td>
<td>2</td>
</tr>
<tr>
<td>COBOL CONTROLS</td>
<td>4</td>
</tr>
<tr>
<td>SOME FREQUENTLY USED SEPARATORS</td>
<td>5</td>
</tr>
<tr>
<td>WORD</td>
<td>6</td>
</tr>
<tr>
<td>PROPER NONNUMERIC LITERAL</td>
<td>7</td>
</tr>
<tr>
<td>PROPER NUMERIC LITERAL</td>
<td>8</td>
</tr>
<tr>
<td>FIGURATIVE CONSTANT</td>
<td>9</td>
</tr>
<tr>
<td>LITERAL</td>
<td>10</td>
</tr>
<tr>
<td>ARITHMETIC OPERATOR</td>
<td>11</td>
</tr>
<tr>
<td>PROPER RELATIONAL OPERATOR</td>
<td>12</td>
</tr>
<tr>
<td>PICTURE CHARACTER STRING</td>
<td>13</td>
</tr>
<tr>
<td>COMMENT STRING</td>
<td>14</td>
</tr>
<tr>
<td>OTHER LANGUAGE STRING</td>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>T. COBOL Text</td>
<td></td>
</tr>
<tr>
<td>SEPARATORS</td>
<td>16</td>
</tr>
<tr>
<td>GENERALIZED CHARACTER STRING</td>
<td>17</td>
</tr>
<tr>
<td>STRUCTURE OF COBOL TEXT</td>
<td>18</td>
</tr>
</tbody>
</table>
N. Names defined by the Implementors
   HARDWARE NAMES  19
   OTHER NAMES  20

C. COBOL Program
   COBOL PROGRAM STRUCTURE  21

I. Identification Division
   IDENTIFICATION DIVISION STRUCTURE  22
   PROGRAM-ID PARAGRAPH  23
   DATE COMPILED PARAGRAPH  24
   OTHER PARAGRAPHS  25
   COMMENT PARAGRAPH BODY  26

E. Environment Division
   ENVIRONMENT DIVISION STRUCTURE  27
   CONFIGURATION SECTION STRUCTURE  28
   INPUT-OUTPUT SECTION STRUCTURE  29
   SOURCE COMPUTER PARAGRAPH  30
   OBJECT COMPUTER PARAGRAPH  31
   SEGMENT LIMIT CLAUSE  32
   SPECIAL-NAMES PARAGRAPH  33
   SPECIAL-NAMES CLAUSE  34
   CURRENCY-SIGN CLAUSE  37
   DECIMAL-POINT CLAUSE  38
   FILE-CONTROL PARAGRAPH  39
   SELECT CLAUSE  41
   ASSIGN CLAUSE  42
   MULTIPLE REEL/UNIT CLAUSE  43
   ALTERNATE AREA CLAUSE  44
   FILE-LIMIT CLAUSE  45
<table>
<thead>
<tr>
<th>Clause/Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCESS MODE CLAUSE</td>
<td>46</td>
</tr>
<tr>
<td>PROCESSING MODE CLAUSE</td>
<td>47</td>
</tr>
<tr>
<td>KEY CLAUSE</td>
<td>48</td>
</tr>
<tr>
<td>I-O-CONTROL PARAGRAPH</td>
<td>49</td>
</tr>
<tr>
<td>RERUN CLAUSE</td>
<td>50</td>
</tr>
<tr>
<td>SAME CLAUSE</td>
<td>51</td>
</tr>
<tr>
<td>MULTIPLE FILE CLAUSE</td>
<td>52</td>
</tr>
<tr>
<td><strong>D. Data Division</strong></td>
<td></td>
</tr>
<tr>
<td>DATA DIVISION STRUCTURE</td>
<td>53</td>
</tr>
<tr>
<td>FILE SECTION</td>
<td>54</td>
</tr>
<tr>
<td>WORKING STORAGE SECTION</td>
<td>55</td>
</tr>
<tr>
<td>REPORT SECTION</td>
<td>56</td>
</tr>
<tr>
<td>FD SKELETON</td>
<td>57</td>
</tr>
<tr>
<td>SD SKELETON</td>
<td>59</td>
</tr>
<tr>
<td>RD SKELETON</td>
<td>61</td>
</tr>
<tr>
<td>FILE AND SORT FILE RECORD</td>
<td>63</td>
</tr>
<tr>
<td>DESCRIPTION SKELETON</td>
<td></td>
</tr>
<tr>
<td>WORKING-STORAGE DATA DESCRIPTION SKELETON</td>
<td>71</td>
</tr>
<tr>
<td>REPORT-GROUP DESCRIPTION SKELETON</td>
<td>75</td>
</tr>
<tr>
<td>BLANK WHEN ZERO CLAUSE</td>
<td>81</td>
</tr>
<tr>
<td>BLOCK CLAUSE</td>
<td>82</td>
</tr>
<tr>
<td>CODE CLAUSE</td>
<td>83</td>
</tr>
<tr>
<td>COLUMN NUMBER CLAUSE</td>
<td>84</td>
</tr>
<tr>
<td>CONTROL CLAUSE</td>
<td>85</td>
</tr>
<tr>
<td>DATA RECORDS CLAUSE</td>
<td>86</td>
</tr>
<tr>
<td>GROUP INDICATE CLAUSE</td>
<td>87</td>
</tr>
<tr>
<td>JUSTIFIED CLAUSE</td>
<td>88</td>
</tr>
<tr>
<td>LABEL RECORDS CLAUSE</td>
<td>89</td>
</tr>
<tr>
<td>LINE NUMBER CLAUSE</td>
<td>90</td>
</tr>
<tr>
<td>NEXT GROUP CLAUSE</td>
<td>91</td>
</tr>
<tr>
<td>OCCURS CLAUSE</td>
<td>92</td>
</tr>
<tr>
<td>Clause</td>
<td>Page</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>PAGE LIMIT CLAUSE</td>
<td>94</td>
</tr>
<tr>
<td>PICTURE CLAUSE</td>
<td>95</td>
</tr>
<tr>
<td>RECORD CONTAINS CLAUSE</td>
<td>100</td>
</tr>
<tr>
<td>REDEFINES CLAUSE</td>
<td>101</td>
</tr>
<tr>
<td>RENAMES CLAUSE</td>
<td>102</td>
</tr>
<tr>
<td>REPORT CLAUSE</td>
<td>103</td>
</tr>
<tr>
<td>RESET CLAUSE</td>
<td>104</td>
</tr>
<tr>
<td>SOURCE CLAUSE</td>
<td>105</td>
</tr>
<tr>
<td>SUM CLAUSE</td>
<td>106</td>
</tr>
<tr>
<td>SYNCHRONIZED CLAUSE</td>
<td>107</td>
</tr>
<tr>
<td>TYPE CLAUSE</td>
<td>108</td>
</tr>
<tr>
<td>USAGE CLAUSE</td>
<td>109</td>
</tr>
<tr>
<td>VALUE CLAUSE</td>
<td>110</td>
</tr>
<tr>
<td>VALUE OF CLAUSE</td>
<td>111</td>
</tr>
<tr>
<td>IDENTIFIERS</td>
<td>112</td>
</tr>
</tbody>
</table>

**P. Procedure Division**

<table>
<thead>
<tr>
<th>Step</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCEDURE DIVISION STRUCTURE</td>
<td>119</td>
</tr>
<tr>
<td>DECLARATIVE PORTION</td>
<td>120</td>
</tr>
<tr>
<td>NON-DECLARATIVE PORTION</td>
<td>121</td>
</tr>
<tr>
<td>SECTIONS</td>
<td>122</td>
</tr>
<tr>
<td>SECTION NAME</td>
<td>123</td>
</tr>
<tr>
<td>SECTION BODY</td>
<td>124</td>
</tr>
<tr>
<td>PARAGRAPH</td>
<td>125</td>
</tr>
<tr>
<td>PARAGRAPH NAME</td>
<td>126</td>
</tr>
<tr>
<td>PROCEDURE NAME</td>
<td>127</td>
</tr>
<tr>
<td>PARAGRAPH BODY</td>
<td>128</td>
</tr>
<tr>
<td>SENTENCES</td>
<td>129</td>
</tr>
<tr>
<td>IMPERATIVE SENTENCES</td>
<td>130</td>
</tr>
<tr>
<td>CONDITIONAL SENTENCE</td>
<td>131</td>
</tr>
<tr>
<td>COMPILER DIRECTING SENTENCES</td>
<td>132</td>
</tr>
<tr>
<td>DECLARATIVE SENTENCE</td>
<td>133</td>
</tr>
<tr>
<td>Statement</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------</td>
</tr>
<tr>
<td>SORT STATEMENT</td>
<td>167</td>
</tr>
<tr>
<td>STOP STATEMENT</td>
<td>168</td>
</tr>
<tr>
<td>SUBTRACT STATEMENT</td>
<td>169</td>
</tr>
<tr>
<td>TERMINATE STATEMENT</td>
<td>170</td>
</tr>
<tr>
<td>USE STATEMENT</td>
<td>171</td>
</tr>
<tr>
<td>WRITE STATEMENT</td>
<td>172</td>
</tr>
</tbody>
</table>

L. COBOL Library

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRUCTURE OF LIBRARY CALLS</td>
</tr>
<tr>
<td>LIBRARY NAME</td>
</tr>
</tbody>
</table>

R. Reserved Words

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDEX OF THE ECMA TC6 SYNTAX</td>
</tr>
<tr>
<td>DEFINITION OF COBOL</td>
</tr>
</tbody>
</table>

EXPLANATORY NOTES

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix: A METALANGUAGE FOR THE DESCRIPTION OF PROGRAMMING LANGUAGES</td>
</tr>
</tbody>
</table>
PREFACE
This formal definition of the syntax of COBOL was prepared by the ECMA Technical Committee on COBOL (TC6).

The work was initially undertaken at the request of the CODASYL COBOL Publication Subcommittee. It resulted in the publication in 1967 of a Preliminary Edition based on COBOL Edition 65. This new edition is based on the ISO Draft Recommendation 1989 on COBOL.

The document comprises four distinct parts and an appendix. The first part briefly describes the notation used, the second part is the formal definition of the COBOL syntax, the third part is an index showing where each meta-variable is defined and where it is used, the fourth part contains explanatory notes for those definitions marked with an asterisk, and the appendix is a complete and rigorous description of the metalanguage. The second part is divided into three sections: syntactic definitions of general nature, level 1 syntax defining the COBOL text and level 2 syntax defining the COBOL program. The level 1 syntax describes the basic structure of the COBOL language. It defines a set of strings, called COBOL texts, in terms of generalized words (including COBOL words, literals, arithmetic and relational operators, etc.) and word separators. The level 2 syntax describes the detailed structure of the COBOL language. It defines a set of strings, called COBOL programs, in terms of specific sequences of generalized words and word separators. Although a COBOL text and a COBOL program have each been defined as a string of characters, an attempt has been made to show the relationship between such a string and the Reference Format.

The metalanguage used is an extension of the metalanguage used in the ALGOL 60 Report, known as the Backus normal form. It is introduced in the first part: "Introduction to the notation used" and described in detail in the appendix under the title "Formalism for syntactical definition". Most extensions have been introduced to reduce the number and complexity of production rules constituting the formal definition of the COBOL syntax. For example certain extensions greatly simplify the description of the nested structure of records. Whenever these extensions are used, the usual Backus notation, based on Chomsky context-free grammars (type 2), could have been used. However, the convention adopted to show relationship between declaration of data-names and the subsequent use of those data-names is different in that this relationship could not be expressed in Backus notation. This is a well known context-dependent aspect of programming languages. English text has been used where needed to adequately supplement the meta-language.
It has been difficult to decide whether some COBOL rules should be included in the syntax and somewhat arbitrary decisions had to be made. The level of detail expressed in the production rules is also somewhat arbitrary. It is often founded on an attempt to facilitate the use of this formal definition by the human reader, in conjunction with the existing descriptions of COBOL. For the same reason, the names of metavariables have been chosen to reflect their meaning, and the names defined in the draft ISO Recommendation on COBOL have been used wherever feasible.

The application of the production rules given in level 2 syntax will generate all valid COBOL programs. However, invalid programs will also be generated. For example the following is not reflected:

- uniqueness of names
- relationship between qualifiers and the corresponding data hierarchy.
- relationships between subscripts or indices and the corresponding table declarations
- some relationships between clauses and/or statements
- possible indentation of data description entries.

With the exceptions mentioned above, this formal definition is believed to be in agreement with the ISO Recommendation on COBOL.

However, the modular structure of the ISO Recommendation is not reflected; the syntax shown applies to the combination of the upper levels of all modules.
INTRODUCTION
TO THE NOTATION
Introduction to the NOTATION

1. General

In the following it is assumed that the reader is familiar with the standard COBOL specification. This informal explanation is intended to further the understanding of the notation used in this Formal Syntactic Definition by way of examples where the new symbols are described in the order of appearance. It is followed by a summary of all the symbols concerned (*).

Observe that the standard COBOL specification already takes advantage of the existence of a kind of formal syntactic definition as shown by the Formats. The present Formal Syntactic Definition is intended to be merely more rigorous, leaving less or (hopefully) no place for ambiguous interpretation of the syntax.

2. Informal Approach through Examples

2.1 As a first example consider entry D124 (page 52):

D124 <block-clause> ::= 
  BLOCK # [CONTAINS #] 
  [<positive-integer> # TO #] 
  <positive-integer> 
  {[# CHARACTERS] ! # RECORDS}

and the corresponding format of the COBOL specification (ref. 2 SEQ, chapter 7, 2.3.2):

BLOCK CONTAINS [integer-1 TO] integer-2 {CHARACTERS RECORDS}

Now, compare these two descriptions element by element:

<table>
<thead>
<tr>
<th>Formal Syntactic Definition</th>
<th>COBOL Specification Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOCK # [CONTAINS #]</td>
<td>BLOCK CONTAINS [integer-1 TO]</td>
</tr>
<tr>
<td>[&lt;positive-integer&gt; TO #]</td>
<td>integer-2 {CHARACTERS}</td>
</tr>
<tr>
<td>&lt;positive-integer&gt;</td>
<td>RECORDS</td>
</tr>
<tr>
<td>{[# CHARACTERS] ! # RECORDS}</td>
<td></td>
</tr>
</tbody>
</table>

It will be seen that the formal notation has been designed to be as closely related as possible to the COBOL specification formats. Thus

a) the appearance of upper-case letters means, in both descriptions, the actual occurrence of these letters in program text;

b) in both descriptions, square brackets [ ] mean that the contents may or may not appear in the program text at the user's option;

Note, however, some differences:

c) underlined upper-case words of the COBOL specification are not underlined in the formal notation, whereas

* A more formal and complete description of the notation is to be found in the Appendix.
noise words, shown with non-underlined upper-case words in the COBOL formats, are described in the formal notation by upper-case words enclosed in square brackets (comprising only the noise word and possibly the symbol #, which is explained below),

\[ \text{e.g. [CONTAINS #]} \text{ instead of CONTAINS [CHARACTERS] instead of CHARACTERS} \]

the symbol ! (‘or’ represented by exclamation mark) has been introduced to express that an alternative has to be selected from two or more possibilities; thus

\[ \text{[# CHARACTERS] ! # RECORDS} \]

means that either # CHARACTERS or # RECORDS exclusively, may appear in the program text. If the former, it may be omitted as shown by the surrounding square brackets.

It will be noticed that the right hand part of D124 shows the exact punctuation allowed or required in the COBOL text:

\[ \text{# means the actual occurrence of one or more spaces. Note also the introduction of the symbols} \]

\[ <> <> \]

These are various types of delimiters.

Syntactic units representing parts of the program text to be filled in at the user’s option from a given set of possible words, strings of words or other entities are represented by lower-case words (or hyphenated words) enclosed in Backus brackets < >. In the above example

\[ <\text{positive-integer}> \]

is such an instance. In fact such a lower-case word enclosed in Backus brackets can be considered as a variable, that is, a notation to be replaced by a variable content; "variables" of this kind have been given the name meta-variables.

As will be seen a meta-variable can be used to represent not only an element (like positive-integer), but any specified portion of a COBOL program. For instance, the whole compound

\[ \text{BLOCK # [CONTAINS #]} \\
\text{[<positive-integer> # TO #]} \\
\text{<positive-integer>} \\
\text{<[# CHARACTERS] ! # RECORDS >} \]

is represented in the example by the single meta-variable

\[ <\text{block-clause}> \]

This is possible because at entry D124 <block-clause> has been precisely defined as being equivalent to the above compound.

This equivalence is specified by the symbol ::= separating <block-clause> from its definition which is given by the compound.
The symbol ::= means "is defined by" and the whole entry is called a meta-definition.
Notice in addition that the alternatives

```
[ # CHARACTERS ] ! # RECORDS
```

appear enclosed in the special braces `< ->` as follows:

```
`{[ # CHARACTERS ] ! # RECORDS }`
```

The aim of the special braces `< ->` is to delimit a specific portion in the formal notation. In this particular case the function of the braces is to determine the scope of the alternatives defined by the `!` operator. Later, other uses of delimitation by braces will be described (in particular in connection with the ellipsis).

2.2 As a second example, consider the D 16 entry (page 57):

```
D 16 <fd-styles> ::= 
    `{<;><block-clause]>`  
    `{<;><record-contains-clause>]`  
    `{<;><label-records-clause>`  
    `{<;><value-of-clause>]`  
    `{<;><data-records-clause>`  
    `{<;><report-clause>`
```

Here is a whole structure named (i.e. defining) `<fd-styles>`, of which the above mentioned `<block-clause>` is just one constituent part, and moreover an optional one, since it is enclosed in square brackets.

This second example introduces some new symbols.

j) There is the symbol `<;>` representing the occurrence of one or more spaces optionally preceded by a semicolon.

k) Then there are the permutation brackets `†` and the permutation separators `‡`

meaning that all syntactic units contained between the permutation brackets and delimited by the permutation separators may appear in any order in the program text, at the user's option.

It is now possible to interpret the meta-definition D 16. Remember that the `!` designates (and separates) two possible alternatives. Thus, D 16 means that

```
<fd-styles>
```

represents a compound of clauses optionally preceded by a semicolon, where the compound comprises one mandatory clause, namely

```
<label-records-clause>
```

three optional clauses, namely

```
<block-clause>
<record-contains-clause>
<value-of-clause>
```
and either the \texttt{<data-records-clause>} which is optional or the \texttt{<report-clause>} and, further, that these clauses may appear in any order by virtue of the permutation brackets.

In the same way as \texttt{<block-clause>} was defined in the first example as D124, the other meta-variables

\begin{itemize}
\item \texttt{<record-contains-clause>}
\item \texttt{<label-records-clause>}
\item \texttt{<value-of-clause>}
\item \texttt{<data-records-clause>}
\item \texttt{<report-clause>}
\end{itemize}

are defined elsewhere in the book, that is at their proper entries.

In D16 only the meta-variable \texttt{<fd-clauses>} is defined.

2.3 Going one more step backwards consider now the entry D15 (page 57):

D15 \texttt{<ms-file-description> ::=}

\begin{itemize}
\item \texttt{<-FD # »}
\item \texttt{<<sequential-ms-file-name-declaration>!}
\item \texttt{<random-ms-file-name-declaration>»}
\item \texttt{<<;><copy-clause>!}
\item \texttt{<fd-clauses>>.>}
\end{itemize}

In this third example new symbols are introduced, namely \texttt{<<.>, < and »}.

l) The symbol \texttt{<<.>} represents the actual occurrence of a period followed by one or more spaces.

m) The symbol \texttt{<-} (horizontal arrow) means that the next character must be in area A of a new line of the reference format.

n) The symbol \texttt{»} (vertical arrow) means that the next character must be in area B of either the same line or the following one of the reference format.

Thus, this third example reads

\texttt{<ms-file-description>} is defined by the letter \texttt{F} in area A of a new line, followed by the letter \texttt{D}, followed by one or more spaces, then, beginning in area B,

\begin{itemize}
\item either a \texttt{<sequential-file-name-declaration>},
\item or a \texttt{<random-file-name-declaration>},
\end{itemize}

then

\begin{itemize}
\item either one or more spaces optionally preceded by a semicolon, and followed by a \texttt{<copy-clause>}
\item or the \texttt{<fd-clauses>}
\end{itemize}

in each case, followed by a period and one or more spaces.

2.4 Looking further backwards it will be seen that \texttt{<ms-file-description>} appears in the definition of \texttt{<file-specification>}:
D 5 <file-specification> ::= 
<non-ms-file-description> 
[<non-ms-record-description>]...!
<ms-file-description> 
[<ms-record-description>]...

This meta-definition demonstrates the usage of the last symbol to be described in this introduction, namely the repetition symbol, called ellipsis, and represented by three dots ...

The ellipsis specifies that the immediately preceding syntactic unit may be repeated any number of times at the user's option, exactly as stated in the familiar COBOL specification. The delimitation of the "immediately preceding syntactic unit" may be found by searching for the immediately preceding closing brace, bracket or Backus bracket and finding the logically matching opening brace, bracket or Backus bracket. Between these two lies the syntactic unit concerned.

Examining D 5, it will easily be seen that <file-specification> is defined as one of two alternatives. For instance, the second alternative (after the ! symbol) consists of

the appearance of <ms-file-description> followed by no, one, or more occurrences of <ms-record-description>

Notice also that since the latter is a meta-variable these successive occurrences will in general represent successive different representations of <ms-record-description>.

2.5 Following the track still further backwards, it will be seen that <file-specification> appears in the definition of <file-section-body> (D4), which, in turn, appears in <file-section> (D3), and so on through <data-division-body> (D2), back, eventually, to <data-division> (D1).

So, the Data Division is defined by means of a chaining of successive meta-definitions, forming a completely determined tree and showing the exact layout of the Data Division of every syntactically correct COBOL program.

Having acquainted himself with the notation by working through the above examples, the reader will then find no difficulty going through the following summary thereby recapitulating and refining the knowledge gained.

3. Summary of the NOTATION used in the Formal Syntactic Definition

3.1 The Meta-Definition

A meta-definition is a syntax rule expressed in a formal notation. Each meta-definition defines a new syntactical entity as a specific arrangement of COBOL characters and other syntactical entities. The name of the syntactical entity to be defined appears on the left-hand side of the definition symbol which is followed by the definition.
3.2 The Metalanguage

3.2.1 Definition Symbol

The following symbol, \( \equiv \), is used as definition symbol.

3.2.2 Meta-Variables (Syntactical Entities)

Lower-case words and other symbols enclosed in Backus brackets represent parts of the COBOL text, whose permissible structures are defined outside the containing meta-definition.

3.2.3 Meta-Constants

Upper-case words, numbers and special characters not enclosed in Backus brackets represent the actual occurrence of these upper-case words, numbers and special characters in the COBOL text.

3.2.4 Meta-Operators and Meta-Delimiters

3.2.4a Alternatives

The OR Symbol, \( \lor \), indicates and separates alternatives.

3.2.4b Braces

Braces, \( [ \; ] \), enclosing a portion of a meta-definition, are used for two different functions:

i) to indicate that a selection of one of the options listed between the braces and separated by the OR symbol, \( \lor \), must be made;

ii) to delimit a portion of the meta-definition, for instance in connection with repetition (see 3.2.4c below).

3.2.4c Repetition

An ellipsis, \( \ldots \), indicates possible repetition of the preceding element. The preceding element may be a COBOL character, a meta-variable or a group of such elements enclosed in brackets or braces.

3.2.4d Permutation

When the order of elements is immaterial these elements are separated by the permutation-separator, \( ; \), and grouped within permutation-brackets \( [ \; ] \). Unless otherwise specified the sequence of elements shown is compulsory.

3.3 Other Conventions

The following symbols are used:

\( \& \) to represent the COBOL character "space"

\( \# \) to represent one or more \( \& \)

\( \leftarrow \) to show that the following element must start in area A of a new line of the reference format

\( \uparrow \) to show that the following element must start in area B of the reference format.
FORMAL DEFINITION
OF COBOL SYNTAX
G. SYNTACTIC DEFINITIONS OF GENERAL NATURE

EMPTY

G1  \langle empty \rangle ::=
COBOL GRAPHICS

G2  \(<\text{non-zero-digit}\> ::=  
    1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 

G3  \(<\text{zero-digit}\> ::=  
    0 

G4  \(<\text{digit}\> ::=  
    \(<\text{zero-digit}\> | \(<\text{non-zero-digit}\> 

G5  \(<\text{letter}\> ::=  
    \text{A} | \text{B} | \text{C} | \text{D} | \text{E} | \text{F} | \text{G} | 
    \text{H} | \text{I} | \text{J} | \text{K} | \text{L} | \text{M} | \text{N} | 
    \text{O} | \text{P} | \text{Q} | \text{R} | \text{S} | \text{T} | \text{U} | 
    \text{V} | \text{W} | \text{X} | \text{Y} | \text{Z} 

* G6  \(<\text{space}\> ::= 
      \text{\hspace{1em}} 

G7  \(<\text{arithmetic-expression-character}\> ::=  
    + | - | * | / 

* G8  \(<\text{relation-character}\> ::=  
    > | \text{<} | \text{=} 

* G9  \(<\text{currency-sign}\> ::=  
      \text{\$}
G10  \texttt{\langle terminating-character \rangle ::=}
    \cdot \mid , \mid ;

G11  \texttt{\langle quotation-mark \rangle ::=} "

G12  \texttt{\langle parenthesis \rangle ::=}
    ( \mid )

G13  \texttt{\langle proper-punctuation-character \rangle ::=}
    \langle terminating-character \rangle \mid
    \langle quotation-mark \rangle \mid
    \langle parenthesis \rangle

G14  \texttt{\langle special-character \rangle ::=}
    \langle arithmetic-expression-character \rangle \mid
    \langle relation-character \rangle \mid
    \langle currency-sign \rangle \mid
    \langle proper-punctuation-character \rangle

G15  \texttt{\langle cobol-character \rangle ::=}
    \langle digit \rangle \mid
    \langle letter \rangle \mid
    \langle space \rangle \mid
    \langle special-character \rangle

G16  \texttt{\langle computer-character \rangle ::=}
    \langle cobol-character \rangle \mid
    \langle additional-data-character \rangle
CDBCL CONTROLS

* G17  \langle strophe-mark \rangle :=

* G18  \langle skip-into-area-b \rangle ::=

\uparrow
SOME FREQUENTLY USED SEPARATORS

G19 \( \langle \text{spaces} \rangle ::= \langle \text{space} \rangle \ldots \)  
   For readability purposes, the abbreviation \# will 
   be used for \( \langle \text{spaces} \rangle \).

G20 \( \langle . \rangle ::= . \# \)

G21 \( \langle , \rangle ::= [ , ] \# \)

G22 \( \langle ; \rangle ::= [ ; ] \# \)
WORD

G23  \texttt{<word-element>} ::=  
     \texttt{<digit>} ! \texttt{<letter>} ! -

G24  \texttt{<word-terminator>} ::=  
     \texttt{<digit>} ! \texttt{<letter>}

G25  \texttt{<word>} ::=  
     \texttt{<word-terminator>}
     \texttt{[[<word-element>...]]  \texttt{<word-terminator>]]}
     \texttt{The maximum number of characters is 30.}

G26  \texttt{<alpha-word>} ::=  
     \texttt{<word> containing <letter>}

G27  \texttt{<non-reserved-word>} ::=  
     \texttt{<word> diff <reserved-word>}

G28  \texttt{<non-reserved-alpha-word>} ::=  
     \texttt{<alpha-word> diff <reserved-word>
PROPER NONNUMERIC LITERAL

G29  \langle literal-string\rangle ::= 
    \langle computer-character\rangle ... 
    not containing " 
    The maximum number of characters is 120.

G30  \langle proper-nonnumeric-literal\rangle ::= 
    " \langle literal-string\rangle "
PROPER NUMERIC LITERAL

G31  \langle integer\rangle ::=
     \langle digit\rangle ...
     The maximum number of digits is 18.

G32  \langle decimal-fraction\rangle ::=
     \langle decimal-point\rangle \langle integer\rangle

G33  \langle unsigned-proper-numeric-literal\rangle ::=
     \langle integer\rangle !
     \langle decimal-fraction\rangle !
     \langle integer\rangle \langle decimal-fraction\rangle
     The maximum number of digits is 18.

G34  \langle sign\rangle ::= 
     + 1  -

G35  \langle proper-numeric-literal\rangle ::= 
     [\langle sign\rangle] \langle unsigned-proper-numeric-literal\rangle

G36  \langle positive-integer\rangle ::= 
     \langle integer\rangle
     containing \langle non-zero-digit\rangle
FIGURATIVE CONSTANT

G37  \(<simple-figurative-constant> ::= \)
    \(ZERO \mid ZEROS \mid ZEROES \mid \)
    \(SPACE \mid SPACES \mid \)
    \(HIGH-VALUE \mid HIGH-VALUES \mid \)
    \(LOW-VALUE \mid LOW-VALUES \mid \)
    \QUOTE \mid QUOTES\)

G38  \(<compound-figurative-constant> ::= \)
    \ALL \ # \ <simple-figurative-constant> \ | \)
    \ALL \ # \ <proper-nonnumeric-literal>\)

G39  \(<figurative-constant> ::= \)
    \(<simple-figurative-constant> \ | \)
    \(<compound-figurative-constant>\)

G40  \(<zero-figurative-constant> ::= \)
    \(ZERO \mid ZEROS \mid ZEROES\)
LITERAL

G41 \( \langle \text{nonnumeric.literal} \rangle ::= \langle \text{proper.nonnumeric.literal} \rangle \ |
\langle \text{figurative.constant} \rangle \)

G42 \( \langle \text{numeric.literal} \rangle ::= \langle \text{proper.numeric.literal} \rangle \ |
\langle \text{zero.figurative.constant} \rangle \)

G43 \( \langle \text{literal} \rangle ::= \langle \text{nonnumeric.literal} \rangle \ |
\langle \text{numeric.literal} \rangle \)
ARITHMETIC OPERATOR

G44 \(<\text{arithmetic-operator}\> ::= + | - | \ast | \div | \^{\ast\ast}\)
PROPER RELATIONAL OPERATOR

* 045  \langle\text{proper-relational-operator}\rangle ::= > \mid < \mid =
046  ⟨picture-character-string⟩ ::= ⟨picture⟩
COMMENT STRING

047  <comment-string> ::=  
      <computer-character>... 
      not containing <.>
OTHER LANGUAGE STRING

G48 \langle other-language-string \rangle ::= \langle computer-character \rangle \ldots
T. COBOL TEXT

SEPARATORS

T1  \( \langle \text{separator} \rangle ::= \# ! \langle . \rangle ! \langle , \rangle ! \langle ; \rangle \)

T2  \( \langle \text{other-language-string-terminator} \rangle ::= \langle \text{empty} \rangle \)

T3  \( \langle \text{generalized-separator} \rangle ::= \langle \text{separator} \rangle ! \langle \text{other-language-string-terminator} \rangle \)
GENERALIZED CHARACTER-STRING

T4  \langle \text{generalized-character-string-type-one} \rangle ::=
    \langle \text{word} \rangle |
    \langle \text{proper-nonnumeric-literal} \rangle |
    \langle \text{proper-numeric-literal} \rangle |
    \langle \text{arithmetic-operator} \rangle |
    \langle \text{proper-relational-operator} \rangle

T5  \langle \text{generalized-character-string-type-two} \rangle ::=
    \langle \text{picture-character-string} \rangle |
    \langle \text{comment-string} \rangle |
    \langle \text{other-language-string} \rangle

T6  \langle \text{generalized-character-string} \rangle ::=
    \langle \text{generalized-character-string-type-one} \rangle |
    \langle \text{generalized-character-string-type-two} \rangle
STRUCTURE OF COBOL TEXT

T7 \( \langle \text{structure} \rangle ::= \langle \text{generalized-character-string} \rangle \mid \langle \text{structure} \rangle \langle \text{generalized-separator} \rangle \ [\langle \text{skip-into-area-b} \rangle] \langle \text{structure} \rangle \mid (\langle \text{structure} \rangle) \)

T8 \( \langle \text{strophe} \rangle ::= \langle \text{strophe-mark} \rangle \langle \text{structure} \rangle \langle \text{separator} \rangle \)

T9 \( \langle \text{cobol-text} \rangle ::= \langle \text{strophe} \rangle \ldots \)
N. NAMES DEFINED BY THE IMPLEMENTOR

HARDWARE NAMES

N1  \texttt{<computer-name> ::= <word> specified by the implementor}

N2  \texttt{<implementor-name-for-type-of-\textit{io-unit}> ::= <word> specified by the implementor}

N3  \texttt{<implementor-name-for-individual-\textit{io-unit}> ::= <word> specified by the implementor}

N4  \texttt{<implementor-name-for-rerun-medium> ::= <word> specified by the implementor}

N5  \texttt{<implementor-name-for-individual-switch> ::= <word> specified by the implementor}
OTHER NAMES

N6  \langle additional-data-character \rangle ::= 
    This proper-meta-variable is specified by the implementor.

N7  \langle implementor-name-for-paper-advance \rangle ::= 
    \langle word \rangle 
    specified by the implementor

N8  \langle implementor-name-for-code-for-report-groups \rangle ::= 
    \langle word \rangle 
    specified by the implementor

N9  \langle other-language-name \rangle ::= 
    \langle word \rangle 
    specified by the implementor
C. COBOL PROGRAM

COBOL PROGRAM STRUCTURE

C1 <cobol-program> ::= <identification-division> <environment-division> <data-division> <procedure-division>
I. IDENTIFICATION DIVISION

IDENTIFICATION DIVISION STRUCTURE

I1  <identification-division> ::= 
    ←IDENTIFICATION # DIVISION <.>
    <identification-division-body>

I2  <identification-division-body> ::= 
    <program-id-paragraph>
    [<author-paragraph>]
    [<installation-paragraph>]
    [<date-written-paragraph>]
    [<date-compiled-paragraph>]
    [<security-paragraph>]
    [<remarks-paragraph>]
PROGRAM-ID PARAGRAPH

I3  \[\langle\text{program-id-paragraph}\rangle ::= \langle\text{PROGRAM-ID} \langle.\rangle \langle\text{program-id-paragraph-body}\rangle\]\n
I4  \[\langle\text{program-id-paragraph-body}\rangle ::= \langle\text{program-id-entry}\rangle\]\n
I5  \[\langle\text{program-id-entry}\rangle ::= \langle\text{program-name-declaration} \langle.\rangle\]\n
I6  \[\langle\text{program-name-declaration}\rangle ::= \langle\text{non-reserved-word}\rangle\]
DATE-COMPILED PARAGRAPH

I7    ⟨date-compiled-paragraph⟩ ::=  
          ←DATE-COMPILED ⟨.⟩  
          [⟨comment-paragraph_body⟩]
OTHER PARAGRAPHS

I8  ⟨author-paragraph⟩ ::=  
    ←AUTHOR ⟨.⟩  
    [⟨comment-paragraph-body⟩]  

I9  ⟨installation-paragraph⟩ ::=  
    ←INSTALLATION ⟨.⟩  
    [⟨comment-paragraph-body⟩]  

I10 ⟨date-written-paragraph⟩ ::=  
    ←DATE-WRITTEN ⟨.⟩  
    [⟨comment-paragraph-body⟩]  

I11 ⟨security-paragraph⟩ ::=  
    ←SECURITY ⟨.⟩  
    [⟨comment-paragraph-body⟩]  

I12 ⟨remarks-paragraph⟩ ::=  
    ←REMARKS ⟨.⟩  
    [⟨comment-paragraph-body⟩]
COMMENT PARAGRAPH BODY

I13 \[ \text{comment-paragraph-body} ::= \]
    \[ \text{comment-entry} \ldots \]

I14 \[ \text{comment-entry} ::= \]
    \[ \text{comment-string} \langle . \rangle \]
E. ENVIRONMENT DIVISION

ENVIRONMENT DIVISION STRUCTURE

E1  <environment-division> ::= 
    ENVIRONMENT DIVISION <.>
   <environment-division-body>

E2  <environment-division-body> ::= 
    <configuration-section>
    [<input-output-section>]
CONFIGURATION SECTION STRUCTURE

E3 \( \langle configuration-section \rangle \ ::= \)
\( \langle configuration-section-body \rangle \)

E4 \( \langle configuration-section-body \rangle \ ::= \)
\( \langle source-computer-paragraph \rangle \)
\( \langle object-computer-paragraph \rangle \)
\( \langle special-names-paragraph \rangle \)
INPUT-OUTPUT SECTION STRUCTURE

E5 \( \langle \text{input-output-section} \rangle ::= \)
\( \langle \text{INPUT-OUTPUT} \ # \ \text{SECTION} \langle . \rangle \) 
\( \langle \text{input-output-section-body} \rangle \)

E6 \( \langle \text{input-output-section-body} \rangle ::= \)
\( \langle \text{file-control-paragraph} \rangle \) 
\( [\langle \text{i-o-control-paragraph} \rangle] \)
E7 \(<\text{source-computer-paragraph}\> ::= \\
\#\text{SOURCE-COMPUTER} <,> \\
\{<\text{copy-entry}> ! \\
<\text{source-computer-paragraph-body}>\}\)

E8 \(<\text{source-computer-paragraph-body}\> ::= \\
<\text{source-computer-entry}>\)

E9 \(<\text{source-computer-entry}\> ::= \\
<\text{computer-name}> <.>\)
OBJECT-COMPUTER PARAGRAPH

E10  <object-computer-paragraph> ::=  
     <OBJECT-COMPUTER <.>
     {<copy-entry> |  
      <object-computer-paragraph-body>::}  

E11  <object-computer-paragraph-body> ::=  
     <object-computer-entry>  

E12  <object-computer-entry> ::=  
     <computer-name> [<,> <memory-size-clause>]
     [<,> <segment-limit-clause>] <.>  

E13  <memory-size-clause> ::=  
     MEMORY # [SIZE #]  
     <integer> # {WORDS | CHARACTERS | MODULES}
SEGMENT-LIMIT CLAUSE

E14  \langle\text{segment-limit-clause}\rangle ::=  
     \text{SEGMENT-LIMIT} \# \text{IS} \#  
     \langle\text{priority-number-limit}\rangle

E15  \langle\text{priority-number-limit}\rangle ::=  
     [0]...  (\langle\text{empty}\rangle 1 1 2 1 3 1 4)  \langle\text{digit}\rangle  
     \text{diff} 40)...
SPECIAL-NAME PARAGRAPH

E16  \(<\text{special-names-paragraph}>::=\n     \langle\text{copy-entry}\rangle\!\\langle\text{special-names-paragraph-body}\rangle\>

E17  \(<\text{special-names-paragraph-body}>::=\n     \langle\text{special-names-entry}\rangle\>

E18  \(<\text{special-names-entry}>::=\n     \langle\text{special-names-clauses}\rangle\!
     \langle\langle,\rangle\langle\text{currency-sign-clause}\rangle\!
     \langle\langle,\rangle\langle\text{decimal-point-clause}\rangle\rangle\!
     \langle\langle,\rangle\langle\text{currency-sign-clause}\rangle\!
     \langle\langle,\rangle\langle\text{decimal-point-clause}\rangle\!
     \langle\langle,\rangle\langle\text{decimal-point-clause}\rangle\rangle\!

E19  \(<\text{special-names-clauses}>::=\n     \langle\text{special-names-clause}\rangle\!
     \langle\langle,\rangle\langle\text{special-names-clause}\rangle\rangle\!

SPECIAL- NAMES CLAUSE

E20  \( \langle \text{special-names-clause} \rangle ::= \)
\( \langle \text{non-switch-special-names-clause} \rangle \)
\( \langle \text{switch-special-names-clause} \rangle \)

E21  \( \langle \text{non-switch-special-names-clause} \rangle ::= \)
\( \langle \text{implementor-name-for-individual-io-unit} \rangle \)
\( \langle \text{mnemonic-name-declaration-for-individual-io-unit} \rangle \)
\( \langle \text{implementor-name-for-type-of-io-unit} \rangle \)
\( \langle \text{mnemonic-name-declaration-for-type-of-io-unit} \rangle \)
\( \langle \text{implementor-name-for-paper-advance} \rangle \)
\( \langle \text{mnemonic-name-declaration-for-paper-advance} \rangle \)
\( \langle \text{implementor-name-for-code-for-report-groups} \rangle \)
\( \langle \text{mnemonic-name-declaration-for-code-for-report-groups} \rangle \)

E22  \( \langle \text{switch-special-names-clause} \rangle ::= \)
\( \langle \text{implementor-name-for-individual-switch} \rangle \)
\( \langle \text{mnemonic-name-declaration-for-individual-switch} \rangle \)
\( \langle \langle \text{switch-status-name-assignment} \rangle \rangle \)
\( \langle \text{implementor-name-for-individual-switch} \rangle \)
\( \langle \text{switch-status-name-assignment} \rangle \)
E23  \<mnemonic-name-declaration-for-individual-io-unit> ::= \<non-reserved-word>

E24  \<mnemonic-name-declaration-for-type-of-io-unit> ::= \<non-reserved-word>

E25  \<mnemonic-name-declaration-for-paper-advance> ::= \<non-reserved-word>

E26  \<mnemonic-name-declaration-for-code-for-report-groups> ::= \<non-reserved-word>

E27  \<mnemonic-name-declaration-for-individual-switch> ::= \<non-reserved-word>

E28  \<mnemonic-name-for-individual-io-unit> ::= \<non-reserved-word> which appears as a \<mnemonic-name-declaration-for-individual-io-unit>

E29  \<mnemonic-name-for-type-of-io-unit> ::= \<non-reserved-word> which appears as a \<mnemonic-name-declaration-for-type-of-io-unit>

E30  \<mnemonic-name-for-paper-advance> ::= \<non-reserved-word> which appears as a \<mnemonic-name-declaration-for-paper-advance>

E31  \<mnemonic-name-for-code-for-report-groups> ::= \<non-reserved-word> which appears as a \<mnemonic-name-declaration-for-code-for-report-groups>
E32  \( \text{switch-status-name-assignment} ::= \)
\( \text{on-status} [\langle,\rangle \text{off-status}] ! \)
\( \text{off-status} [\langle,\rangle \text{on-status}] \)

E33  \( \text{on-status} ::= \)
\( \text{ON} \# [\text{STATUS} \#] \text{IS} \# \)
\( \text{switch-status-name-declaration} \)

E34  \( \text{off-status} ::= \)
\( \text{OFF} \# [\text{STATUS} \#] \text{IS} \# \)
\( \text{switch-status-name-declaration} \)

E35  \( \text{switch-status-name-declaration} ::= \)
\( \text{non-reserved-alpha-word} \)

E36  \( \text{switch-status-name} ::= \)
\( \text{non-reserved-alpha-word} \) which appears as a
\( \text{switch-status-name-declaration} \)
CURRENCY-SIGN CLAUSE

E37  <currency-sign-clause> ::=  
    CURRENCY # [SIGN #] IS # 
    " <currency-sign-declaration> "

E38  <currency-sign-declaration> ::=  
    <possible-character-for-currency-sign>

E39  <possible-character-for-currency-sign> ::=  
    <computer-character> diff 
    { <digit> | 
      A | B | C | D | P | R | S | V | X | Z | <space> !
      + | - | * | . | , | ; | ! | ( | ) }

E40  <currency-symbol> ::=  
    $ <possible-character-for-currency-sign>  
    which appears as a <currency-sign-declaration> .

This language element is dependent on the individual COBOL source program.  
It equals $,  
if no <currency-sign-declaration> is present  
in the <special-names-paragraph> .  
It equals  
<possible-character-for-currency-sign>  
which appears as a  
<currency-sign-declaration> ,  
if a <currency-sign-declaration> is present  
in the <special-names-paragraph> .

E41  <cs> ::=  
    <currency-symbol>
DECIMAL-POINT CLAUSE

E42  <decimal-point-clause> ::=  
    DECIMAL-POINT # IS # COMMA

E43  <decimal-point> ::=  
    . ! ,

This language element is dependent on the individual COBOL source program.
It equals . (period),
if no <decimal-point-clause> is present
in the <special-names-paragraph> .
It equals , (comma),
if a <decimal-point-clause> is present
in the <special-names-paragraph> .

E44  <digit-separator> ::=  
    , ! ,

This language element is dependent on the individual COBOL source program.
It equals , (comma),
if no <decimal-point-clause> is present
in the <special-names-paragraph> .
It equals . (period),
if a <decimal-point-clause> is present
in the <special-names-paragraph> .
FILE-CONTROL PARAGRAPH

E45  \textit{<file-control-paragraph>} ::= 
     \langle\textit{FILE-CONTROL} \langle.\rangle 
     \langle\textit{copy-entry} \mid 
     \textit{<file-control-paragraph-body}>\rangle

E46  \textit{<file-control-paragraph-body>} ::= 
     \langle\textit{file-control-entry}\ldots

E47  \textit{<file-control-entry>} ::= 
     \langle\textit{file-control-entry-for-non-ms-file} \mid 
     \textit{<file-control-entry-for-sequential-ms-file} \mid 
     \textit{<file-control-entry-for-random-ms-file} \mid 
     \textit{<file-control-entry-for-sort-file}\rangle
E48  \( \text{<file-control-entry-for-non-ms-file> ::=} \)
    \( \text{<select-clause-for-non-ms-file>} \)
    \# \{<assign-clause> ! <sort-output-assign-clause>\}
    \[<\text{alternate-area-clause}>\]
    \(<.>\)

E49  \( \text{<file-control-entry-for-sequential-ms-file> ::=} \)
    \( \text{<select-clause-for-sequential-ms-file>} \)
    \# \{<assign-clause> ! <sort-output-assign-clause>\}
    \[<\text{alternate-area-clause}>\]
    \[<\text{file-limit-clause}>\]
    \(<.>\text{<access-mode-sequential-clause},\>
    \[<\text{processing-mode-sequential-clause}>\]
    \[<\text{actual-key-clause}>\]
    \(<.>\)

E50  \( \text{<file-control-entry-for-random-ms-file> ::=} \)
    \( \text{<select-clause-for-random-ms-file>} \)
    \# <assign-clause>
    \[<\text{file-limit-clause}>\]
    \(<.>\text{<access-mode-random-clause},\>
    \(<.>\text{<processing-mode-sequential-clause},\>
    \(<.>\text{<actual-key-clause}>\)
    \(<.>\)

E51  \( \text{<file-control-entry-for-sort-file> ::=} \)
    \( \text{<select-clause-for-sort-file>} \)
    \# <assign-clause> <.>
SELECT CLAUSE

E52 \(<\text{select-clause-for-non-ms-file}\> ::= \<\text{SELECT}\> [\<\text{optional-phrase}\>] \<\text{non-ms-file-name}\>\)

E53 \(<\text{select-clause-for-sequential-ms-file}\> ::= \<\text{SELECT}\> [\<\text{optional-phrase}\>] \<\text{sequential-ms-file-name}\>\)

E54 \(<\text{select-clause-for-random-ms-file}\> ::= \<\text{SELECT}\> \<\text{random-ms-file-name}\>\)

E55 \(<\text{select-clause-for-sort-file}\> ::= \<\text{SELECT}\> \<\text{sort-file-name}\>\)

E56 \(<\text{optional-phrase}\> ::= \<\text{OPTIONAL}\>\)
ASSIGN CLAUSE

E57  \(<assign-clause> ::= \n\<assign-type-clause> | \n\<assign-individual-units-clause> \n\)

E58  \(<assign-type-clause> ::= \nASSIGN [# TO] \n[# <integer>] \n# <implementor-name-for-type-of-io-unit> \n\)

E59  \(<assign-individual-units-clause> ::= \nASSIGN [# TO] \n# <implementor-name-for-individual-io-unit> \n[<,> <implementor-name-for-individual-io-unit>]... \n\)

E60  \(<sort-output-assign-clause> ::= \nASSIGN [# TO] \n# <implementor-name-for-individual-io-unit> \n[<,> <implementor-name-for-individual-io-unit>]... \n# OR \n# <implementor-name-for-individual-io-unit> \n[<,> <implementor-name-for-individual-io-unit>]... \n\)
MULTIPLE REEL/UNIT CLAUSE

E61  \(<\text{multiple-reel-clause}\> ::= [\text{FOR} \#] \text{MULTIPLE} \# \text{REEL}\)

E62  \(<\text{multiple-unit-clause}\> ::= [\text{FOR} \#] \text{MULTIPLE} \# \text{UNIT}\)
ALTERNATE AREA CLAUSE

E63 \(<alternate\text{-}area\text{-}clause> ::=\)
\hline
RESERVE \# \{\<integer\> | NO\}
\hline
[# ALTERNATE]
\hline
\{[# AREA] | [# AREAS]\}\)
FILE-LIMIT CLAUSE

E64  <file-limit-clause> ::=  
    {FILE-LIMIT # IS ! FILE-LIMITS # ARE}  
    # <file-limit> # {THROUGH | THRU} # <file-limit>  
    [<,> <file-limit> # {THROUGH | THRU} # <file-limit>]...

E65  <file-limit> ::=  
    <data-name-identifier> ! <literal>
ACCESS MODE CLAUSE

E66  \(<\text{access-mode-sequential-clause}> ::=\n    \text{ACCESS \# [MODE \#]} \text{ IS \# SEQUENTIAL}\)

E67  \(<\text{access-mode-random-clause}> ::=\n    \text{ACCESS \# [MODE \#]} \text{ IS \# RANDOM}\)
PROCESSING MODE CLAUSE

E68  \langle\text{processing-mode-sequential-clause}\rangle ::= \text{PROCESSING}\ #\ [\text{MODE}\ #]\ IS\ #\ \text{SEQUENTIAL}
KEY CLAUSE

E69 \(<\text{actual-key-clause}> ::= \)
\hspace{1em} \text{ACTUAL} \# [\text{KEY} \#] \hspace{0.5em} \text{IS} \# \\
\hspace{1em} \text{<data-name-identifier>} \)
I-O-CONTROL PARAGRAPh

E70 \( <\text{i-o-control-paragraph}> ::= \)
\( \langle \text{-I-O-CONTROL} \langle . \rangle \)
\( \langle \text{copy-entry} \rangle ! \)
\( \langle \text{i-o-control-paragraph-body} \rangle \)

E71 \( <\text{i-o-control-paragraph-body}> ::= \)
\( \langle \text{i-o-control-entry} \rangle \)

E72 \( <\text{i-o-control-entry}> ::= \)
\( \langle \text{rerun-clauses} \rangle \)
\( [\langle ; \rangle <\text{same-clauses}>] \)
\( [\langle ; \rangle <\text{multiple-file-clauses}>] <.> ! \)
\( <\text{same-clauses} \rangle \)
\( [\langle ; \rangle <\text{multiple-file-clauses}>] <.> ! \)
\( <\text{multiple-file-clauses} > <.> \)

E73 \( <\text{rerun-clauses}> ::= \)
\( <\text{rerun-clause} \rangle \)
\( [\langle ; \rangle <\text{rerun-clause}>] ... \)

E74 \( <\text{same-clauses}> ::= \)
\( <\text{same-clause} \rangle \)
\( [\langle ; \rangle <\text{same-clause}>] ... \)

E75 \( <\text{multiple-file-clauses}> ::= \)
\( <\text{multiple-file-clause} \rangle \)
\( [\langle ; \rangle <\text{multiple-file-clause}>] ... \)
RERUN CLAUSE

E76 \(<\text{rerun-clause}> ::= \)
   \(\text{RERUN} \# [\langle\text{rerun-medium}\rangle \#] \langle\text{rerun-condition-1}\rangle !\)
   \(\text{RERUN} \# \langle\text{rerun-medium}\rangle \# \langle\text{rerun-condition-2}\rangle\)

E77 \(<\text{rerun-medium}> ::= \)
   \(\text{ON} \# \langle\text{non-sort-file-name}\rangle !\)
   \(\text{ON} \# \langle\text{implementor-name-for-rerun-medium}\rangle\)

E78 \(<\text{rerun-condition-1}> ::= \)
   \(\langle\text{end-of-reel-rerun-condition}\rangle\)

E79 \(<\text{end-of-reel-rerun-condition}> ::= \)
   \([\text{EVERY} \#] [\text{END} \# [\text{OF} \#]] \langle\text{REEL} ! \text{UNIT}\rangle \#\)
   \([\text{OF} \#] \langle\text{non-sort-file-name}\rangle\)

E80 \(<\text{rerun-condition-2}> ::= \)
   \(\langle\text{integer-records-rerun-condition}\rangle !\)
   \(\langle\text{clock-units-rerun-condition}\rangle !\)
   \(\langle\text{switch-rerun-condition}\rangle\)

E81 \(<\text{integer-records-rerun-condition}> ::= \)
   \([\text{EVERY} \#] \langle\text{positive-integer}\rangle \# \text{RECORDS} \#\)
   \([\text{OF} \#] \langle\text{non-sort-file-name}\rangle\)

E82 \(<\text{clock-units-rerun-condition}> ::= \)
   \([\text{EVERY} \#] \langle\text{positive-integer}\rangle \# \text{CLOCK-UNITS}\)

E83 \(<\text{switch-rerun-condition}> ::= \)
   \([\text{EVERY} \#] \langle\text{switch-status-name}\rangle\)
SAME CLAUSE

E84  (same-clause) ::=  
      (same-record-area-clause) !
      (same-block-area-clause) !
      (same-sort-area-clause)

E85  (same-record-area-clause) ::=  
      SAME # RECORD # [AREA #] [FOR #]
      <file-name> {<,> <file-name>}...

E86  (same-block-area-clause) ::=  
      SAME # [AREA #] [FOR #]
      <non-sort-file-name> {<,> <non-sort-file-name>}...

E87  (same-sort-area-clause) ::=  
      SAME # SORT # [AREA #] [FOR #]
      <file-name> {<,> <file-name>}...
MULTIPLE FILE CLAUSE

E88  \( \langle \text{multiple-file-clause} \rangle ::= \)
    MULTIPLE # FILE # [TAPE #] [CONTAINS #]
    \( \langle \text{non-sort-file-name} \rangle \) [# POSITION # \( \langle \text{integer} \rangle \)]
    [\( \langle, \rangle \) \( \langle \text{non-sort-file-name} \rangle \) [# POSITION #
    \( \langle \text{integer} \rangle \rangle \) ]...
D. DATA DIVISION

DATA DIVISION STRUCTURE

\[ \text{D1} \quad \text{<data-division> ::=} \]
\[ \quad \text{<DATA # DIVISION <.>}} \]
\[ \quad \text{<data-division-body>} \]

\[ \text{D2} \quad \text{<data-division-body> ::=} \]
\[ \quad \text{<file-section> \}
\[ \quad \text{[<working-storage-section>] \}
\[ \quad \text{[<report-section>]} \}
\[ \quad \text{[<working-storage-section>]} \]
FILE SECTION

D3 $\langle\text{file-section}\rangle ::=$
   $\langle\text{-FILE } \# \text{ SECTION} \rangle$
   $\langle\text{file-section-body}\rangle$

D4 $\langle\text{file-section-body}\rangle ::=$
   "$\{\langle\text{file-specification}\rangle \mid$
   $\langle\text{sort-file-specification}\rangle\}\}$

D5 $\langle\text{file-specification}\rangle ::=$
   $\langle\text{non-ms-file-description}\rangle$
   $[\langle\text{non-ms-record-description}\rangle] \mid$
   $\langle\text{ms-file-description}\rangle$
   $[\langle\text{ms-record-description}\rangle] \mid$
   $\langle\text{sort-file-specification}\rangle$

D6 $\langle\text{sort-file-specification}\rangle ::=$
   $\langle\text{sort-file-description}\rangle$
   $\langle\text{sort-record-description}\rangle\}$
WORKING-STORAGE SECTION

D7 <working-storage-section> ::=  
<WORKING-STORAGE # SECTION .>  
<working-storage-section-body>

D8 <working-storage-section-body> ::=  
[<77-descriptions>]  
<ws-record-descriptions> !  
<77-descriptions>

D9 <77-descriptions> ::=  
{"<77-description>  
[<redefining-77-description>]...}...

D10 <ws-record-descriptions> ::=  
{"<ws-record-description>  
[<redefining-ws-record-description>]...}...
REPORT SECTION

D11 (report-section) ::= 
  !REPORT # SECTION <.*> 
  (report-section-body)

D12 (report-section-body) ::= 
  (report-specification)...

D13 (report-specification) ::= 
  (report-description) 
  (report-group-description)...

FD SKELETON

D14  \textless\textit{non-m}s-file-description\textgreater \ :=
\begin{itemize}
  \item \textlt{FD} \#\textup{\textdagger}
  \item \textlt{non-m}s-file-name-declaration\textgreater
  \item \textlt{copy-clause}\textgreater !
\end{itemize}
\textlt{fd-clauses}\textgreater \textlt{.}\textgreater

D15  \textlt{m}s-file-description\textgreater \ :=
\begin{itemize}
  \item \textlt{FD} \#\textup{\textdagger}
  \item \textlt{sequential-m}s-file-name-declaration\textgreater !
  \item \textlt{random-m}s-file-name-declaration\textgreater
  \item \textlt{copy-clause}\textgreater !
\end{itemize}
\textlt{fd-clauses}\textgreater \textlt{.}\textgreater

D16  \textlt{fd-clauses}\textgreater \ :=
\begin{itemize}
  \item \[\textlt{block-clause}\textgreater \textup{\textdagger}
  \item \[\textlt{record-contains-clause}\textgreater \textup{\textdagger}
  \item \[\textlt{label-records-clause}\textgreater \textup{\textdagger}
  \item \[\textlt{value-of-clause}\textgreater \textup{\textdagger}
  \item \[\textlt{data-records-clause}\textgreater !
  \item \[\textlt{report-clause}\textgreater \textup{\textdagger}
\end{itemize}
D17 ⟨non-ms-file-name-declaration⟩ ::= 
    ⟨non-reserved-alpha-word⟩

D18 ⟨non-ms-file-name⟩ ::= 
    ⟨non-reserved-alpha-word⟩
    which appears as a 
    ⟨non-ms-file-name-declaration⟩

D19 ⟨sequential-ms-file-name-declaration⟩ ::= 
    ⟨non-reserved-alpha-word⟩

D20 ⟨sequential-ms-file-name⟩ ::= 
    ⟨non-reserved-alpha-word⟩
    which appears as a 
    ⟨sequential-ms-file-name-declaration⟩

D21 ⟨random-ms-file-name-declaration⟩ ::= 
    ⟨non-reserved-alpha-word⟩

D22 ⟨random-ms-file-name⟩ ::= 
    ⟨non-reserved-alpha-word⟩
    which appears as a 
    ⟨random-ms-file-name-declaration⟩

D23 ⟨sequential-file-name⟩ ::= 
    ⟨non-ms-file-name⟩ ! 
    ⟨sequential-ms-file-name⟩

D24 ⟨ms-file-name⟩ ::= 
    ⟨sequential-ms-file-name⟩ ! 
    ⟨random-ms-file-name⟩

D25 ⟨non-sort-file-name⟩ ::= 
    ⟨non-ms-file-name⟩ ! 
    ⟨ms-file-name⟩

D26 ⟨file-name⟩ ::= 
    ⟨non-sort-file-name⟩ ! 
    ⟨sort-file-name⟩
SD SKELETON

D26  <sort-file-description> ::=  
    <-SD #  ↦
    <sort-file-name-declaration>  
    {<;> <copy-clause> !}  
    <sd-clauses> } .>

D28  <sd-clauses> ::=  
    {[<;> <record-contains-clause>]  ﾆ  
    <;> <data-records-clause>}

D29  \texttt{<sort-file-name-declaration> ::=}
      \texttt{<non-reserved-alpha-word>}

D30  \texttt{<sort-file-name> ::=}
      \texttt{<non-reserved-alpha-word>}
      \texttt{which appears as a}
      \texttt{<sort-file-name-declaration>}

RD SKELETON

D31 \[
\text{<report-description> ::=}
\text{\hphantom{=}}
\text{\textasciitilde RD \# \uparrow}
\text{\hphantom{=}}
\text{\textasciitilde report-name-declaration}
\text{\hphantom{=}}
\{\langle ; \rangle \text{<copy-clause>} \}
\text{\hphantom{=}}
\text{\textasciitilde rd-clauses\} <.>}
\]

D32 \[
\text{<rd-clauses> ::=}
\text{\hphantom{=}}
\{[\langle ; \rangle \text{<code-clause>} \}
\text{\hphantom{=}}
\{[\langle ; \rangle \text{<control-clause>} \}
\text{\hphantom{=}}
\{[\langle ; \rangle \text{<page-limit-clause>} \}
\]
D33 \( \text{<report-name-declaration>} ::= \text{<non-reserved-alpha-word>} \)

D34 \( \text{<report-name>} ::= \text{<non-reserved-alpha-word>} \text{which appears as a} \text{<report-name-declaration>} \)
FILE AND SORT FILE RECORD DESCRIPTION SKELETON

D35  <non-ms-record-description> ::=  
      ← {Ø ! 0} 1 # ↑  
      <non-elem-non-ms-record-name-declaration>  
      {<;} <copy-clause> <.,> !  
      <non-elem-record-spec>  
      ← {Ø ! 0} 1 # ↑  
      <elem-non-ms-record-name-declaration>  
      {<;} <copy-clause> <.,> !  
      <elem-record-spec>  

D36  <ms-record-description> ::=  
      ← {Ø ! 0} 1 # ↑  
      <non-elem-ms-record-name-declaration>  
      {<;} <copy-clause> <.,> !  
      <non-elem-record-spec>  
      ← {Ø ! 0} 1 # ↑  
      <elem-ms-record-name-declaration>  
      {<;} <copy-clause> <.,> !  
      <elem-record-spec>  

D37  <sort-record-description> ::=  
      ← {Ø ! 0} 1 # ↑  
      <non-elem-sort-record-name-declaration>  
      {<;} <copy-clause> <.,> !  
      <non-elem-record-spec>  
      ← {Ø ! 0} 1 # ↑  
      <elem-sort-record-name-declaration>  
      {<;} <copy-clause> <.,> !  
      <elem-record-spec>
D38  \( \langle \text{non-elem-record-spec} \rangle ::= \)
\( \langle \text{non-elem-01-clauses} \rangle \langle \star \rangle \)
\( [\langle \text{88-entry} \rangle]... \)
\( \langle \text{subordinate-entries} \rangle \)
\( [\langle \text{66-entry} \rangle]... ! \)
\( \langle ; \rangle \langle \text{usage-is-index-clause} \rangle \langle \star \rangle \)
\( \langle \text{subordinate-entries} \rangle \)
\( [\langle \text{66-entry} \rangle]... \)

D39  \( \langle \text{elem-record-spec} \rangle ::= \)
\( \langle \text{elem-01-77-clauses} \rangle \langle \star \rangle \)
\( [\langle \text{88-entry} \rangle]... ! \)
\( \langle ; \rangle \langle \text{usage-is-index-clause} \rangle \langle \star \rangle \)
D40 \texttt{<66-entry> ::=}
\texttt{\triangleright 66 \#} \uparrow
\texttt{<non-elem-66-item-name-declaration>}
\texttt{;} \texttt{<non-elem-renames-clause>} \!
\texttt{\triangleright 66 \#} \uparrow
\texttt{<elem-66-item-name-declaration>}
\texttt{;} \texttt{<elem-renames-clause>}

D41 \texttt{<88-entry> ::=}
\texttt{\triangleright 88 \#} \uparrow
\texttt{<condition-name-declaration>}
\texttt{;} \texttt{<88-value-clause>} \langle.\rangle
D42  \( \text{subordinate-entries} ::= \)
    \( \leftarrow \# \text{ sub-number} \# \uparrow \)
    \( \text{sub-spec} \)
    \( \leftarrow \# \text{ sub-number} \# \uparrow \)
    \( \{ \text{sub-spec} \} ! \)
    \( \text{redefining-sub-spec} \} \)...

D43  \( \text{sub-spec} ::= \)
    \( \text{non-elem-spec} \)
    \( \text{subordinate-entries} ! \)
    \( \text{index-non-elem-spec} \)
    \( \text{subordinate-entries} ! \)
    \( \text{elem-spec} ! \)
    \( \text{index-elem-spec} \)

D44  \( \text{redefining-sub-spec} ::= \)
    \( \text{redefining-non-elem-spec} \)
    \( \text{subordinate-entries} ! \)
    \( \text{redefining-index-non-elem-spec} \)
    \( \text{subordinate-entries} ! \)
    \( \text{redefining-elem-spec} ! \)
    \( \text{redefining-index-elem-spec} \)

D45  \( \text{sub-number} ::= \)
    \( \text{level-number} \)
    with a value increased with respect to the entry to
    which the \( \text{subordinate-entries} \) are subordinate

D46  \( \text{level-number} ::= \)
    \( \{ 1 \ | \ 2 \ | \ 3 \ | \ 4 \} \) \( \text{digit} ! \)
    \( \{ \# \ | \ 0 \} \) \( \text{non-zero-digit} \)
\[D47\] \textit{non-elem-spec} ::= \\
\hspace{1em} \textit{non-elem-02-48-item-name-declaration} \\
\hspace{2em} \textit{non-elem-clauses} \langle . \rangle \\
\hspace{1em} [\langle 88\text{-entry} \rangle] \ldots

\[D48\] \textit{redefining-non-elem-spec} ::= \\
\hspace{1em} \textit{non-elem-02-48-item-name-declaration} \\
\hspace{2em} ; \langle \textit{redefines-clause} \rangle \\
\hspace{2em} \textit{non-elem-red-clauses} \langle . \rangle \\
\hspace{1em} [\langle 88\text{-entry} \rangle] \ldots

\[D49\] \textit{elem-spec} ::= \\
\hspace{1em} \textit{02-49-name-declaration} \\
\hspace{2em} \textit{elem-clauses} \langle . \rangle \\
\hspace{1em} [\langle 88\text{-entry} \rangle] \ldots

\[D50\] \textit{redefining-elem-spec} ::= \\
\hspace{1em} \textit{02-49-name-declaration} \\
\hspace{2em} ; \langle \textit{redefines-clause} \rangle \\
\hspace{2em} \textit{elem-red-clauses} \langle . \rangle \\
\hspace{1em} [\langle 88\text{-entry} \rangle] \ldots

\[D51\] \textit{index-non-elem-spec} ::= \\
\hspace{1em} \textit{non-elem-02-48-item-name-declaration} \\
\hspace{2em} [\langle ; \rangle \textit{usage-is-index-clause}] \langle . \rangle

\[D52\] \textit{redefining-index-non-elem-spec} ::= \\
\hspace{1em} \textit{non-elem-02-48-item-name-declaration} \\
\hspace{2em} ; \langle \textit{redefines-clause} \rangle \\
\hspace{2em} [\langle ; \rangle \textit{usage-is-index-clause}] \langle . \rangle

\[D53\] \textit{index-elem-spec} ::= \\
\hspace{1em} \textit{02-49-name-declaration} \\
\hspace{2em} [\langle ; \rangle \textit{usage-is-index-clause}] \langle . \rangle

\[D54\] \textit{redefining-index-elem-spec} ::= \\
\hspace{1em} \textit{02-49-name-declaration} \\
\hspace{2em} ; \langle \textit{redefines-clause} \rangle \\
\hspace{2em} [\langle ; \rangle \textit{usage-is-index-clause}] \langle . \rangle
D55  \text{\textless\textit{non-elem-clauses}\textgreater\textcolon\textbar
\textless\textit{non-elem-01-clauses}\textgreater\textbar
\textless\textit{non-elem-red-clauses}\textgreater\textbar
\textless\textit{var-occurs-non-elem-clauses}\textgreater}

D56  \text{\textless\textit{non-elem-01-clauses}\textgreater\textcolon\textbar
\{\text{\textless\textbar\textit{usage-clause}\textbar}\textup{\downarrow}
\text{\textless\textbar\textit{value-clause}\textup{\downarrow}}\}

D57  \text{\textless\textit{non-elem-red-clauses}\textgreater\textcolon\textbar
\{\text{\textless\textbar\textit{usage-clause}\textbar}\textup{\downarrow}
\text{\textless\textbar\textit{fixed-occurs-clause}\textbar}\}

D58  \text{\textless\textit{var-occurs-non-elem-clauses}\textgreater\textcolon\textbar
\{\text{\textless\textbar\textit{usage-clause}\textbar}\textup{\downarrow}
\text{\textless\textbar\textit{variable-occurs-clause}\textbar}\}

D59  \text{\textless\textit{elem-clauses}\textgreater\textcolon\textbar
\textless\textit{elem-01-77-clauses}\textgreater\textbar
\textless\textit{elem-red-clauses}\textgreater\textbar
\textless\textit{var-occurs-elem-clauses}\textgreater}

D60  \text{\textless\textit{elem-01-77-clauses}\textgreater\textcolon\textbar
\{\text{\textless\textbar\textit{usage-clause}\textbar}\textup{\downarrow}
\text{\textless\textbar\textit{picture-clause}\textup{\downarrow}
\text{\textless\textbar\textit{justified-clause}\textup{\downarrow}
\text{\textless\textbar\textit{blank-when-zero-clause}\textup{\downarrow}
\text{\textless\textbar\textit{synchronized-clause}\textup{\downarrow}
\text{\textless\textbar\textit{value-clause}\textup{\downarrow}}\}

D61  \text{\textless\textit{elem-red-clauses}\textgreater\textcolon\textbar
\{\text{\textless\textbar\textit{usage-clause}\textbar}\textup{\downarrow}
\text{\textless\textbar\textit{picture-clause}\textup{\downarrow}
\text{\textless\textbar\textit{justified-clause}\textup{\downarrow}
\text{\textless\textbar\textit{blank-when-zero-clause}\textup{\downarrow}
\text{\textless\textbar\textit{synchronized-clause}\textup{\downarrow}
\text{\textless\textbar\textit{fixed-occurs-clause}\textup{\downarrow}}\}

D62  \text{\textless\textit{var-occurs-elem-clauses}\textgreater\textcolon\textbar
\{\text{\textless\textbar\textit{usage-clause}\textbar}\textup{\downarrow}
\text{\textless\textbar\textit{picture-clause}\textup{\downarrow}
\text{\textless\textbar\textit{justified-clause}\textup{\downarrow}
\text{\textless\textbar\textit{blank-when-zero-clause}\textup{\downarrow}
\text{\textless\textbar\textit{synchronized-clause}\textup{\downarrow}
\text{\textless\textbar\textit{variable-occurs-clause}\textup{\downarrow}}\}
D63 <non-elem-non-ms-record-name-declaration> ::= 
      <non-reserved-alpha-word>

D64 <non-elem-non-ms-record-name> ::= 
     <non-reserved-alpha-word>
which appears as a
     <non-elem-non-ms-record-name-declaration>

D65 <elem-non-ms-record-name-declaration> ::= 
     <non-reserved-alpha-word>

D66 <elem-non-ms-record-name> ::= 
     <non-reserved-alpha-word>
which appears as a
     <elem-non-ms-record-name-declaration>

D67 <non-elem-ms-record-name-declaration> ::= 
     <non-reserved-alpha-word>

D68 <non-elem-ms-record-name> ::= 
     <non-reserved-alpha-word>
which appears as a
     <non-elem-ms-record-name-declaration>

D69 <elem-ms-record-name-declaration> ::= 
     <non-reserved-alpha-word>

D70 <elem-ms-record-name> ::= 
     <non-reserved-alpha-word>
which appears as a
     <elem-ms-record-name-declaration>

D71 <non-elem-sort-record-name-declaration> ::= 
     <non-reserved-alpha-word>

D72 <non-elem-sort-record-name> ::= 
     <non-reserved-alpha-word>
which appears as a
     <non-elem-sort-record-name-declaration>

D73 <elem-sort-record-name-declaration> ::= 
     <non-reserved-alpha-word>

D74 <elem-sort-record-name> ::= 
     <non-reserved-alpha-word>
which appears as a
     <elem-sort-record-name-declaration>

D75 <non-elem-02-48-item-name-declaration> ::= 
     <non-reserved-alpha-word>
D76 \( <\text{non-elem-02-48-item-name}> ::= <\text{non-reserved-alpha-word}> \) which appears as a \( <\text{non-elem-02-48-item-name-declaration}> \)

D77 \( <\text{02-49-name-declaration}> ::= <\text{elem-02-49-item-name-declaration}> \) | FILLER

D78 \( <\text{elem-02-49-item-name-declaration}> ::= <\text{non-reserved-alpha-word}> \)

D79 \( <\text{elem-02-49-item-name}> ::= <\text{non-reserved-alpha-word}> \) which appears as a \( <\text{elem-02-49-item-name-declaration}> \)

D80 \( <\text{non-elem-66-item-name-declaration}> ::= <\text{non-reserved-alpha-word}> \)

D81 \( <\text{non-elem-66-item-name}> ::= <\text{non-reserved-alpha-word}> \) which appears as a \( <\text{non-elem-66-item-name-declaration}> \)

D82 \( <\text{elem-66-item-name-declaration}> ::= <\text{non-reserved-alpha-word}> \)

D83 \( <\text{elem-66-item-name}> ::= <\text{non-reserved-alpha-word}> \) which appears as a \( <\text{elem-66-item-name-declaration}> \)

D84 \( <\text{condition-name-declaration}> ::= <\text{non-reserved-alpha-word}> \)

D85 \( <\text{condition-name}> ::= <\text{non-reserved-alpha-word}> \) which appears as a \( <\text{condition-name-declaration}> \)
WORKING-STORAGE DATA DESCRIPTION SKELETON

D86 \(\langle 77\text{-}description\rangle \ ::= \)
\(-\langle 77\text{-}item-name-declaration\rangle \)
\(-\langle elem-01\text{-}77\text{-}clauses\rangle \langle .\rangle \)
\([-\langle 88\text{-}entry\rangle ]\ldots \rangle \)
\(-\langle 77\text{-}item-name-declaration\rangle \)
\(-\langle usage\text{-}is\text{-}index\text{-}clause\rangle \langle .\rangle \)

D87 \(\langle redefining\text{-}77\text{-}description\rangle \ ::= \)
\(-\langle 77\text{-}item-name-declaration\rangle \)
\(-\langle 77\text{-}redefines\text{-}clause\rangle \)
\(-\langle elem-01\text{-}77\text{-}red\text{-}clauses\rangle \langle .\rangle \)
\([-\langle 88\text{-}entry\rangle ]\ldots \rangle \)
\(-\langle 77\text{-}item-name-declaration\rangle \)
\(-\langle 77\text{-}redefines\text{-}clause\rangle \)
\(-\langle usage\text{-}is\text{-}index\text{-}clause\rangle \langle .\rangle \)

D88 \(\langle ws\text{-}record\text{-}description\rangle \ ::= \)
\(-\{\emptyset \mid 0\} \ 1 \ # \uparrow \)
\(-\langle ncn\text{-}elem\text{-}ws\text{-}record\text{-}name\text{-}declaration\rangle \)
\(-\langle ;\rangle \langle copy\text{-}clause\rangle \langle .\rangle \mid \)
\(-\langle non\text{-}elem\text{-}record\text{-}spec\rangle \rangle \mid \)
\(-\{\emptyset \mid 0\} \ 1 \ # \uparrow \)
\(-\langle elem\text{-}ws\text{-}record\text{-}name\text{-}declaration\rangle \)
\(-\langle ;\rangle \langle copy\text{-}clause\rangle \langle .\rangle \mid \)
\(-\langle elem\text{-}record\text{-}spec\rangle \rangle \)

D89 \(\langle redefining\text{-}ws\text{-}record\text{-}description\rangle \ ::= \)
\(-\{\emptyset \mid 0\} \ 1 \ # \uparrow \)
\(-\langle ncn\text{-}elem\text{-}ws\text{-}record\text{-}name\text{-}declaration\rangle \)
\(-\langle ;\rangle \langle copy\text{-}clause\rangle \langle .\rangle \mid \)
\(-\langle redefining\text{-}non\text{-}elem\text{-}record\text{-}spec\rangle \rangle \mid \)
\(-\{\emptyset \mid 0\} \ 1 \ # \uparrow \)
\(-\langle elem\text{-}ws\text{-}record\text{-}name\text{-}declaration\rangle \)
\(-\langle ;\rangle \langle copy\text{-}clause\rangle \langle .\rangle \mid \)
\(-\langle redefining\text{-}elem\text{-}record\text{-}spec\rangle \rangle \)
D90 \labeled{redefining-non-elem-record-spec} ::= 
  \labeled{redefines-record-clause} 
  [\labeled{usage-clause}] .
  [\labeled{eq-entry}] ... 
  \labeled{subordinate-entries} 
  [\labeled{eq-entry}] ... ! 
  \labeled{redefines-record-clause} 
  \labeled{usage-is-index-clause} .
  \labeled{subordinate-entries} 
  [\labeled{eq-entry}] ...

D91 \labeled{redefining-elem-record-spec} ::= 
  \labeled{redefines-record-clause} 
  \labeled{elem-01-77-red-clauses} .
  [\labeled{eq-entry}] ... ! 
  \labeled{redefines-record-clause} 
  \labeled{usage-is-index-clause} .
D92 \( \langle \text{elem-01-77-red-clauses} \rangle ::= \)
\( \{ \langle ; \rangle \langle \text{usage-clause} \rangle \} \downarrow \)
\( \langle ; \rangle \langle \text{picture-clause} \rangle \downarrow \)
\( \langle ; \rangle \langle \text{justified-clause} \rangle \downarrow \)
\( \langle ; \rangle \langle \text{blank-when-zero-clause} \rangle \downarrow \)
\( \langle ; \rangle \langle \text{synchronized-clause} \rangle \} \uparrow \)
D93 \( \langle 77\text{-item-name-declaration} \rangle ::= \langle \text{non-reserved-alpha-word} \rangle \)

D94 \( \langle 77\text{-item-name} \rangle ::= \langle \text{non-reserved-alpha-word} \rangle \) which appears as a
\( \langle 77\text{-item-name-declaration} \rangle \)

D95 \( \langle \text{non-elem-ws-record-name-declaration} \rangle ::= \langle \text{non-reserved-alpha-word} \rangle \)

D96 \( \langle \text{non-elem-ws-record-name} \rangle ::= \langle \text{non-reserved-alpha-word} \rangle \) which appears as a
\( \langle \text{non-elem-ws-record-name-declaration} \rangle \)

D97 \( \langle \text{elem-ws-record-name-declaration} \rangle ::= \langle \text{non-reserved-alpha-word} \rangle \)

D98 \( \langle \text{elem-ws-record-name} \rangle ::= \langle \text{non-reserved-alpha-word} \rangle \) which appears as a
\( \langle \text{elem-ws-record-name-declaration} \rangle \)
REPORT-GROUP DESCRIPTION SKELETON

D99  \( \langle \text{report-group-description} \rangle \) ::= 
     {  \[ \{ \} \] 0 \} 1 \# \uparrow 
     [ \langle \text{non-elem-report-group-name-declaration} \rangle ] 
     {  \{ ; \}  \langle \text{copy-clause} \rangle  \langle . \rangle  \downarrow }  
     \langle \text{non-elem-report-group-spec} \rangle  \downarrow 
     {  \{ \} \{ 0 \} \} 1 \# \uparrow 
     [ \langle \text{sum-report-group-name-declaration} \rangle ] 
     {  \{ ; \}  \langle \text{copy-clause} \rangle  \langle . \rangle  \downarrow }  
     \langle \text{sum-report-group-spec} \rangle  \downarrow 
     {  \{ \} \{ 0 \} \} 1 \# \uparrow 
     [ \langle \text{elem-non-sum-report-group-name-declaration} \rangle ] 
     {  \{ ; \}  \langle \text{copy-clause} \rangle  \langle . \rangle  \downarrow }  
     \langle \text{elem-non-sum-report-group-spec} \rangle
D100 \text{<non-elem-report-group-spec>} ::= \\
\text{<non-elem-group-clauses> <.*> \\
<subordinate-report-entries>}

D101 \text{<sum-report-group-spec>} ::= \\
\text{<sum-group-clauses> <.*>}

D102 \text{<elem-non-sum-report-group-spec>} ::= \\
\text{<elem-non-sum-group-clauses> <.*>
D103 <subordinate-report-entries> ::= 
  {<sub-number> \# ↑ 
  <sub-report-spec>}"...

D104 <sub-report-spec> ::= 
  <non-elem-report-spec> 
  <subordinate-report-entries> ! 
  <elem-report-spec>
D105 ⟨non-elem-report-spec⟩ ::=  
  ⟨⟨non-elem-field-name-declaration⟩⟩  
  ⟨⟨⟨;> ⟨line-number-clause⟩⟩⟩  
  ⟨⟨⟨;> ⟨usage-is-display-clause⟩⟩⟩  

D106 ⟨elem-report-spec⟩ ::=  
  ⟨sum-spec⟩  
  !  
  ⟨elem-non-sum-spec⟩

D107 ⟨sum-spec⟩ ::=  
  ⟨⟨sum-field-name-declaration⟩⟩  
  ⟨⟨⟨;> ⟨line-number-clause⟩⟩⟩  
  ⟨⟨⟨;> ⟨column-number-clause⟩⟩⟩  
  ⟨⟨⟨;> ⟨usage-is-display-clause⟩⟩⟩  
  ⟨⟨⟨;> ⟨picture-clause⟩⟩⟩  
  ⟨⟨⟨;> ⟨justified-clause⟩⟩⟩  
  ⟨⟨⟨;> ⟨blank-when-zero-clause⟩⟩⟩  
  ⟨⟨⟨;> ⟨sum-clause⟩⟩⟩  
  ⟨⟨⟨;> ⟨reset-clause⟩⟩⟩  
  ⟨⟨⟩⟩  
  !  
  ⟨sum-field-name-declaration⟩  
  ⟨⟨⟨;> ⟨line-number-clause⟩⟩⟩  
  ⟨⟨⟨;> ⟨usage-is-display-clause⟩⟩⟩  
  ⟨⟨⟨;> ⟨picture-clause⟩⟩⟩  
  ⟨⟨⟨;> ⟨sum-clause⟩⟩⟩  
  ⟨⟨⟨;> ⟨reset-clause⟩⟩⟩  

D108 ⟨elem-non-sum-spec⟩ ::=  
  ⟨⟨elem-non-sum-field-name-declaration⟩⟩  
  ⟨⟨⟨;> ⟨line-number-clause⟩⟩⟩  
  ⟨⟨⟨;> ⟨column-number-clause⟩⟩⟩  
  ⟨⟨⟨;> ⟨group-indicate-clause⟩⟩⟩  
  ⟨⟨⟨;> ⟨usage-is-display-clause⟩⟩⟩  
  ⟨⟨⟨;> ⟨picture-clause⟩⟩⟩  
  ⟨⟨⟨;> ⟨blank-when-zero-clause⟩⟩⟩  
  !  
  ⟨⟨;> ⟨source-clause⟩⟩  
  !  
  ⟨⟨⟨;> ⟨value-clause⟩⟩⟩  
  ⟨⟨⟩⟩  
  !  
  ⟨⟨elem-non-sum-field-name-declaration⟩⟩  
  ⟨⟨;> ⟨source-clause⟩⟩
D109 \langle\text{non-elem-group-clauses}\rangle ::= \begin{align*}
\langle\cdot\rangle \langle\text{type-clause}\rangle \\
\langle\cdot\rangle \langle\text{next-group-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{line-number-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{usage-is-display-clause}\rangle [\cdot]
\end{align*}

D110 \langle\text{sum-group-clauses}\rangle ::= \begin{align*}
\langle\cdot\rangle \langle\text{type-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{next-group-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{line-number-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{column-number-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{usage-is-display-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{picture-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{justified-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{blank-when-zero-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{sum-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{reset-clause}\rangle [\cdot] ! \\
\langle\cdot\rangle \langle\text{type-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{next-group-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{line-number-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{usage-is-display-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{picture-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{sum-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{reset-clause}\rangle [\cdot]
\end{align*}

D111 \langle\text{elem-non-sum-group-clauses}\rangle ::= \begin{align*}
\langle\cdot\rangle \langle\text{type-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{next-group-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{line-number-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{column-number-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{group-indicate-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{usage-is-display-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{picture-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{justified-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{blank-when-zero-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{source-clause}\rangle ! \\
\langle\cdot\rangle \langle\text{value-clause}\rangle [\cdot] ! \\
\langle\cdot\rangle \langle\text{type-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{next-group-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{line-number-clause}\rangle [\cdot] \\
\langle\cdot\rangle \langle\text{source-clause}\rangle [\cdot]
\end{align*}
D112 <non-elem-report-group-name-declaration> ::= <non-reserved-alpha-word>

D113 <non-elem-report-group-name> ::= <non-reserved-alpha-word> which appears as a <non-elem-report-group-name-declaration>

D114 <elem-non-sum-report-group-name-declaration> ::= <non-reserved-alpha-word>

D115 <elem-non-sum-report-group-name> ::= <non-reserved-alpha-word> which appears as a <elem-non-sum-report-group-name-declaration>

D116 <sum-report-group-name-declaration> ::= <non-reserved-alpha-word>

D117 <sum-report-group-name> ::= <non-reserved-alpha-word> which appears as a <sum-report-group-name-declaration>

D118 <non-elem-field-name-declaration> ::= <non-reserved-alpha-word>

D119 <non-elem-field-name> ::= <non-reserved-alpha-word> which appears as a <non-elem-field-name-declaration>

D120 <elem-non-sum-field-name-declaration> ::= <non-reserved-alpha-word>

D121 <sum-field-name-declaration> ::= <non-reserved-alpha-word>

D122 <sum-field-name> ::= <non-reserved-alpha-word> which appears as a <sum-field-name-declaration>
BLANK WHEN ZERO CLAUSE

D123 \[ \langle \text{blank-when-zero-clause} \rangle ::= \]
\[ \text{BLANK } \text{[# WHEN]} \text{ [# ZERO} \]
D124  \( \langle \text{block-clause} \rangle ::= \)
    \( \text{BLOCK} \ # \ [\text{CONTAINS} \ # ] \)
    \( [\langle \text{positive-integer} \rangle \ # \ \text{TO} \ # ] \)
    \( \langle \text{positive-integer} \rangle \)
    \( \{ [\langle \# \ \text{CHARACTERS} \rangle \ | \ \# \ \text{RECORDS} \} \)
CODE CLAUSE

D125  \( \langle \text{code-clause} \rangle ::= \)

\text{CODE} \ #

\( \langle \text{mnemonic-name-for-code-for-report-groups} \rangle \)
COLUMN NUMBER CLAUSE

D126  \langle\text{column-number-clause}\rangle ::= 
    \text{COLUMN} \# \ [\text{NUMBER} \#] \ [\text{IS} \#] 
    \langle\text{positive-integer}\rangle
CONTROL CLAUSE

D127 \( \langle \text{control-clause} \rangle ::= \)
\( \{ \text{CONTROL} \ \# \ [\text{IS} \ \#] \} \)
\( \{ \text{CONTROLS} \ \# \ [\text{ARE} \ \#] \} \)
\( \{ \text{FINAL} \ ! \)
\( \langle \text{item-name-identifier} \rangle \)
\[ \langle, \rangle \langle \text{item-name-identifier} \rangle \ldots \)
DATA RECORDS CLAUSE

D123 <data-records-clause> ::= 
  DATA #
  {RECORD # [IS #] | RECORDS # [ARE #]}
  <non-ws-record-name>
  [{,} <non-ws-record-name>]*
GROUP INDICATE CLAUSE

D129 <group-indicate-clause> ::= GROUP [# INDICATE]
JUSTIFIED CLAUSE

D130  \langle justified-clause \rangle ::= 
       \{JUSTIFIED ! JUST\} [# RIGHT]
LABEL RECORDS CLAUSE

D131 \( \langle \text{label-records-clause} \rangle ::= \)
\( \quad \text{LABEL} \# \)
\( \quad \langle \text{RECORD} \# \ [\text{IS} \#] \rangle \)
\( \quad \langle \text{RECORDS} \# \ [\text{ARE} \#] \rangle \)
\( \quad \langle \text{OMITTED} ! \text{STANDARD} ! \rangle \)
\( \quad \langle \text{non-ws-record-name} \rangle \)
\( \quad [<,> \langle \text{non-ws-record-name} \rangle ] . . . ] \)
LINE NUMBER CLAUSE

D132  <line-number-clause> ::= 
    LINE # [NUMBER #] [IS #]
    {<positive-integer> | 
    PLUS # <positive-integer> | 
    NEXT # PAGE}
NEXT GROUP CLAUSE

D133 \( \langle \text{next-group-clause} \rangle ::= \)
\( \text{NEXT} \# \text{GROUP} \# [\text{IS} \#] \)
\( \{ \langle \text{positive-integer} \rangle \} \)
\( \text{PLUS} \# \langle \text{positive-integer} \rangle \)
\( \text{NEXT} \# \text{PAGE} \)
OCCURS CLAUSE

D134  \(<\text{fixed-occurs-clause}\> ::= \<\text{OCCURS}\> \<\text{#}\> \<\text{positive-integer}\> \[\# \<\text{TIMES}\>]\[\# \<\text{key-option}\>]\[\# \<\text{index-option}\>\]

D135  \(<\text{variable-occurs-clause}\> ::= \<\text{OCCURS}\> \<\text{#}\> \<\text{integer}\> \# \<\text{TO}\> \<\text{positive-integer}\> \[\# \<\text{TIMES}\>]\[\# \<\text{DEPENDING}\> \[\# \<\text{ON}\>]\] \<\text{elem-item-name-qualified}\>\[\# \<\text{key-option}\>]\[\# \<\text{index-option}\>\]

D136  \(<\text{key-option}\> ::= \{\<\text{ASCENDING} ! \text{DESCENDING}\> \<\text{#}\> \<\text{KEY}\> \[\#\]> \<\text{IS}\> \[\#\]> \<\text{elem-item-name-qualified}\> \[\langle,\> \<\text{elem-item-name-qualified}\>]\[\langle,\> \<\text{elem-item-name-qualified}\>\]

D137  \(<\text{index-option}\> ::= \<\text{INDEXED}\> \[\#\]> \<\text{BY}\> \[\#\]> \<\text{index-name-declaration}\> \[\langle,\> \<\text{index-name-declaration}\>\[\langle,\> \<\text{index-name-declaration}\>\]
D138  \textit{index-name-declaration} ::= \\
  \textit{non-reserved-alpha-word} \\

D139  \textit{index-name} ::= \\
  \textit{non-reserved-alpha-word} \\
  \textit{which appears as a} \\
  \textit{index-name-declaration}
PAGE LIMIT CLAUSE

D140  

\[
\text{\texttt{<page-limit-clause> ::= PAGE \#}} \n\quad \text{([LIMIT \#] [IS \#] | [LIMITS \#] [ARE \#])} \n\quad \text{<positive-integer>} \# \n\quad \text{<LINE | LINES>} \n\quad [\text{<,> HEADING \# \n}\quad \text{<positive-integer>}] \n\quad [\text{<,> FIRST \# DETAIL \# \n}\quad \text{<positive-integer>}] \n\quad [\text{<,> LAST \# DETAIL \# \n}\quad \text{<positive-integer>}] \n\quad [\text{<,> FOOTING \# \n}\quad \text{<positive-integer>}] \n\]

\]
PICTURE CLAUSE

D141  \langle picture-clause \rangle ::=  
       \{ PICTURE ! PIC \} \# [IS \#]  
       \langle picture \rangle

D142  \langle picture \rangle ::=  
       \langle numeric-picture \rangle !  
       \langle numeric-edited-picture \rangle  
       not ending with a period or a comma !  
       \langle nonnumeric-picture \rangle

D143  \langle numeric-picture \rangle ::=  
       [S] \langle dig-seq \rangle  
       [V] [\langle dig-seq \rangle] !  
       [S] \langle dig-seq \rangle  
       [\langle p \rangle]...[V] !  
       [S] [V] [\langle p \rangle]...  
       \langle dig-seq \rangle

D144  \langle numeric-edited-picture \rangle ::=  
       \langle sign-float-pict \rangle !  
       \langle cs-float-pict \rangle !  
       \langle supp-pict \rangle !  
       \langle fixed-insert-pict \rangle

D145  \langle nonnumeric-picture \rangle ::=  
       [\langle b09 \rangle]...  
       \{\langle ax \rangle...[\langle b09 \rangle]...\}...
D146 \langle sign-float-pict \rangle ::= 
[\langle cs \rangle] \langle + - seq \rangle 
\langle point \rangle \langle + - seq \rangle ! 
[\langle cs \rangle] \langle - - seq \rangle 
\langle point \rangle \langle - - seq \rangle ! 
[\langle cs \rangle] \langle sign-float \rangle 
[\langle 9 - seq \rangle] \langle point \rangle 
[\langle 9 - seq - or - bo \rangle] ! 
[\langle cs \rangle] \langle sign-float \rangle 
[\langle 9 - seq \rangle] \langle \langle p \rangle \rangle \ldots \langle V \rangle ! 
\langle V \rangle \langle \langle p \rangle \rangle \ldots 
[\langle cs \rangle] \langle sign-float \rangle 

D147 \langle cs-float-pict \rangle ::= 
[+ ! -] \langle cs - seq \rangle 
\langle point \rangle \langle cs - seq \rangle ! 
[+ ! -] \langle cs - float \rangle 
[\langle 9 - seq \rangle] \langle point \rangle 
[\langle 9 - seq - or - bo \rangle] ! 
[+ ! -] \langle cs - float \rangle 
[\langle 9 - seq \rangle] \langle \langle p \rangle \rangle \ldots \langle V \rangle ! 
\langle V \rangle \langle \langle p \rangle \rangle \ldots 
[+ ! -] \langle cs - float \rangle ! 
\langle cs - seq \rangle \langle point \rangle 
\langle cs - seq \rangle \langle + ! - \rangle \langle CR \rangle \langle DB \rangle ! 
\langle cs - float \rangle \langle \langle 9 - seq \rangle \rangle 
\langle point \rangle \langle \langle 9 - seq - or - bo \rangle \rangle 
\langle + ! - \rangle \langle CR \rangle \langle DB \rangle ! 
\langle cs - float \rangle \langle \langle 9 - seq \rangle \rangle 
\langle + ! - \rangle \langle \langle p \rangle \rangle \ldots \langle V \rangle ! 
\langle CR \rangle \langle DB \rangle \langle \langle p \rangle \rangle \ldots ! 
\langle V \rangle \langle \langle p \rangle \rangle \ldots 
\langle cs - float \rangle \langle + ! - \rangle \langle CR \rangle \langle DB \rangle !

D148 \langle supp-pict \rangle ::= 
[+ ! -] \langle \langle cs \rangle \rangle \langle z - seq \rangle 
\langle point \rangle \langle \langle z - seq \rangle \rangle ! 
[+ ! -] \langle \langle cs \rangle \rangle \langle * - seq \rangle 
\langle point \rangle \langle * - seq \rangle ! 
[+ ! -] \langle \langle cs \rangle \rangle \langle z - or - * - seq \rangle 
[\langle 9 - seq \rangle] \langle point \rangle 
[\langle 9 - seq - or - bo \rangle] ! 
[+ ! -] \langle \langle cs \rangle \rangle \langle z - or - * - seq \rangle 
[\langle 9 - seq \rangle] \langle \langle p \rangle \rangle \ldots \langle V \rangle ! 
\langle V \rangle \langle \langle p \rangle \rangle \ldots 
[+ ! -] \langle \langle cs \rangle \rangle \langle z - or - * - seq \rangle ! 
[\langle cs \rangle] \langle z - seq \rangle \langle point \rangle 
\langle z - seq \rangle \langle + ! - \rangle \langle CR \rangle \langle DB \rangle ! 
[\langle cs \rangle] \langle * - seq \rangle \langle point \rangle 
\langle * - seq \rangle \langle + ! - \rangle \langle CR \rangle \langle DB \rangle ! 
[\langle cs \rangle] \langle z - or - * - seq \rangle \langle \langle 9 - seq \rangle \rangle 
\langle point \rangle \langle \langle 9 - seq - or - bo \rangle \rangle 
\langle + ! - \rangle \langle CR \rangle \langle DB \rangle !
continued...  

\[
\langle \text{cs} \rangle \text{ } \langle z \text{-} or \text{-} *\text{-}seq \rangle \text{ } [\langle 9 \text{-} seq \rangle ]
\]
\[
+ \text{! -} \quad [\langle p \rangle]... \quad [V] \text{!}
\]
\[
[\langle \text{cs} \rangle ] \text{ } \langle z \text{-} or \text{-} *\text{-}seq \rangle \text{ } [\langle 9 \text{-} seq \rangle ]
\]
\[
\{ \text{CR }! \text{DB} \} \text{ } [\langle p \rangle]... \quad !
\]
\[
[V] \text{ } [\langle p \rangle]... \quad [\langle \text{cs} \rangle ]
\]
\[
\langle z \text{-} or \text{-} *\text{-}seq \rangle \quad + \quad ! \quad - \quad ! \text{CR} \quad \text{DB}
\]

D149 \text{ } \langle \text{fixed-insert-pict} \rangle \text{ } ::= \text{ }
\[
+ \text{! -} \quad [\langle \text{cs} \rangle ] \text{ } \langle 9 \text{-} seq \rangle
\]
\[
\langle \text{point} \rangle \text{ } [\langle 9 \text{-} seq-or-b0,\rangle ] \text{!}
\]
\[
+ \text{! -} \quad [\langle \text{cs} \rangle ] \text{ } \langle 9 \text{-} seq \rangle
\]
\[
[\langle p \rangle]... \quad [V] \text{!}
\]
\[
[\langle \text{cs} \rangle ] \text{ } \langle 9 \text{-} seq \rangle
\]
\[
+ \text{! -} \quad [\langle p \rangle]... \quad [V] \text{!}
\]
\[
[\langle \text{cs} \rangle ] \text{ } \langle 9 \text{-} seq \rangle
\]
\[
\{ \text{CR }! \text{DB} \} \text{ } [\langle p \rangle]... \quad !
\]
\[
[V] \text{ } [\langle p \rangle]... \quad [\langle \text{cs} \rangle ]
\]
\[
\langle 9 \text{-} seq \rangle \quad + \quad ! \quad - \quad ! \text{CR} \quad \text{DB}
\]

D150 \text{ } \langle \text{sign-float} \rangle \text{ } ::= \text{ }
\[
[\langle b0,\rangle ]... \quad + \quad \langle +\text{-seq} \rangle \text{!}
\]
\[
[\langle b0,\rangle ]... \quad - \quad \langle -\text{-seq} \rangle \text{!}
\]
\[
[\langle b0,\rangle ]... \quad + \quad (\langle \text{positive-integer} \rangle)
\]
\[
[\langle +\text{-seq-or-b0,\rangle ] \text{!}
\]
\[
[\langle b0,\rangle ]... \quad - \quad (\langle \text{positive-integer} \rangle)
\]
\[
[\langle -\text{-seq-or-b0,\rangle ]
\]

D151 \text{ } \langle \text{cs-float} \rangle \text{ } ::= \text{ }
\[
[\langle b0,\rangle ]... \quad \langle \text{cs} \rangle \text{ } \langle \text{cs-seq} \rangle \text{!}
\]
\[
[\langle b0,\rangle ]... \quad \langle \text{cs} \rangle \text{ } (\langle \text{positive-integer} \rangle)
\]
\[
[\langle \text{cs-seq-or-b0,\rangle ]
\]
D152 \langle \text{z-or-*-seq} \rangle ::=
\langle \text{z-seq} \rangle \!
\langle \text{*-seq} \rangle

D153 \langle \text{g-seq-or-b0,} \rangle ::=
\langle \text{g-seq} \rangle \!
\langle \text{b0,} \rangle\ldots

D154 \langle \text{+-seq-or-b0,} \rangle ::=
\langle \text{+-seq} \rangle \!
\langle \text{b0,} \rangle\ldots

D155 \langle \text{+-seq-or-b0,} \rangle ::=
\langle \text{+-seq} \rangle \!
\langle \text{b0,} \rangle\ldots

D156 \langle \text{cs-seq-or-b0,} \rangle ::=
\langle \text{cs-seq} \rangle \!
\langle \text{b0,} \rangle\ldots

D157 \langle \text{dig-seq} \rangle ::=
\langle \langle 0 \rangle \rangle \ldots
\langle \langle 9 \rangle \ldots \langle 0 \rangle \ldots \rangle \ldots

D158 \langle \text{g-seq} \rangle ::=
\langle \langle \text{b0,} \rangle \rangle \ldots
\langle \langle 9 \rangle \ldots \langle \text{b0,} \rangle \ldots \rangle \ldots

D159 \langle \text{+-seq} \rangle ::=
\langle \langle \text{b0,} \rangle \rangle \ldots
\langle \langle \text{+-} \rangle \ldots \langle \text{b0,} \rangle \ldots \rangle \ldots

D160 \langle \text{+-seq} \rangle ::=
\langle \langle \text{b0,} \rangle \rangle \ldots
\langle \langle \text{-} \rangle \ldots \langle \text{b0,} \rangle \ldots \rangle \ldots

D161 \langle \text{cs-seq} \rangle ::=
\langle \langle \text{b0,} \rangle \rangle \ldots
\langle \langle \text{9} \rangle \ldots \langle \text{b0,} \rangle \ldots \rangle \ldots

D162 \langle \text{z-seq} \rangle ::=
\langle \langle \text{b0,} \rangle \rangle \ldots
\langle \langle \text{z} \rangle \ldots \langle \text{b0,} \rangle \ldots \rangle \ldots

D163 \langle \text{*-seq} \rangle ::=
\langle \langle \text{b0,} \rangle \rangle \ldots
\langle \langle \text{*} \rangle \ldots \langle \text{b0,} \rangle \ldots \rangle \ldots

D164 \langle \text{b0,} \rangle ::=
\{ \text{B} \ 1 \ 0 \ \langle \text{digit-separator} \rangle \}
\langle \langle \text{positive-integer} \rangle \rangle
D165 \( \langle b09 \rangle ::= \)
\[ \{B \mid 0 \mid 9\} \]
\[ \{(\text{positive-integer})\} \]

D166 \( \langle ax \rangle ::= \)
\[ \{A \mid X\} \]
\[ \{(\text{positive-integer})\} \]

D167 \( \langle 9 \rangle ::= \)
\[ 9 \]
\[ \{(\text{positive-integer})\} \]

D168 \( \langle 0 \rangle ::= \)
\[ 0 \]
\[ \{(\text{positive-integer})\} \]

D169 \( \langle p \rangle ::= \)
\[ P \]
\[ \{(\text{positive-integer})\} \]

D170 \( \langle + \rangle ::= \)
\[ \dagger \]
\[ \{(\text{positive-integer})\} \]

D171 \( \langle - \rangle ::= \)
\[ \_ \]
\[ \{(\text{positive-integer})\} \]

D172 \( \langle \$ \rangle ::= \)
\[ \langle \text{cs} \rangle \]
\[ \{(\text{positive-integer})\} \]

D173 \( \langle z \rangle ::= \)
\[ Z \]
\[ \{(\text{positive-integer})\} \]

D174 \( \langle * \rangle ::= \)
\[ \* \]
\[ \{(\text{positive-integer})\} \]

D175 \( \langle \text{point} \rangle ::= \)
\[ V \mid \}
\[ \langle \text{decimal-point} \rangle \]
RECORD CONTAINS CLAUSE

D176 <record-contains-clause> ::=  
    RECORD # [CONTAINS #] 
    [<positive-integer>] # [TO #] 
    <positive-integer> 
    [# CHARACTERS]
REDEFINES CLAUSE

D177 <redefines-clause> ::= 
    REDEFINES # 
    <02-49-item-name-qualified>

D178 <redefines-record-clause> ::= 
    REDEFINES # 
    <ws-record-name>

D179 <77-redefines-clause> ::= 
    REDEFINES # 
    <77-item-name>
RENAME CLAUSE

D180  \langle\text{elem-renames-clause}\rangle ::= 
      \text{RENAME}\ #
      \langle\text{elem-item-name-qualified}\rangle

D181  \langle\text{non-elem-renames-clause}\rangle ::= 
      \text{RENAME}\ #
      \langle\text{item-name-qualified}\rangle
      [\# \{\text{THROUGH} \mid \text{TTHRU}\}\#
      \langle\text{item-name-qualified}\rangle]
REPORT CLAUSE

D182 〈report-clause〉 ::=  
   {REPORT # [IS #] !  
    REPORTS # [ARE #]}  
   〈report-name〉  
   [‹;› 〈report-name〉]...
RESET CLAUSE

D183 \(<\text{reset-clause}\> ::=\n  \text{RESET} \ [# [\text{ON} \ #]]\n  \text{\textbackslash{FINAL}} !\n  \langle\text{item-name-identifier}\rangle\n  [\langle,\rangle \langle\text{item-name-identifier}\rangle]...\n
SOURCE CLAUSE

D184  \langle source-clause \rangle ::= 
    \langle source-item-name-identifier \rangle 
    \langle source-clause \rangle 
    \langle source-item-name-identifier \rangle
SUM CLAUSE

D185 \( \langle \text{sum-clause} \rangle ::= \)
\( \text{SUM} \# \)
\( \langle \text{summed-item-name-identifier} \rangle \)
\( [\langle,\rangle \langle \text{summed-item-name-identifier} \rangle] \ldots \)
\( [\# \text{ UPON } \# \)
\( \langle \text{report-group-name-qualified} \rangle ] \)
SYNCHRONIZED CLAUSE

D186  \langle\text{synchronized-clause}\rangle ::= 
    \{\text{SYNCHRONIZED} \mid \text{SYNC}\} 
    \{\# \{\text{LEFT} \mid \text{RIGHT}\}\}
TYPE CLAUSE

D187 \(<\text{type-clause}\> ::=\)
\begin{verbatim}
TYPE # [IS #]
{REPORT # HEADING !
RH !
PAG # HEADING !
PH !
CONTROL # HEADING #
{FINAL !
<item-name-identifier>!}
CH #
{FINAL !
<item-name-identifier>!}
DETAIL !
DE !
CONTROL # FOOTING #
{FINAL !
<item-name-identifier>!}
CF #
{FINAL !
<item-name-identifier>!}
PAG # FOOTING !
PF !
REPORT # FOOTING !
RF}
\end{verbatim}
USAGE CLAUSE

D188 <usage-clause> ::= 
  [USAGE # [IS #]]
  {COMPUTATIONAL |
    COMP |
    DISPLAY}

D189 <usage-is-display-clause> ::= 
  [USAGE # [IS #]] DISPLAY

D190 <usage-is-index-clause> ::= 
  [USAGE # [IS #]] INDEX
VALUE CLAUSE

D191  \(<\text{value-clause}> ::= \<\text{value-clause}> \<\text{literal}>\)

D192  \(<\text{88-value-clause}> ::= \<\text{value-clause}> \<\text{values}> \<\text{numeric-literal}>\)
          \[\# \<\text{through} > \<\text{thru}> \# \<\text{numeric-literal}>\]
          \[\<\text{>,} \<\text{numeric-literal}>\]
          \[\# \<\text{through} > \<\text{thru}> \# \<\text{numeric-literal}>\]
          \[\<\text{>,} \<\text{nonnumeric-literal}>\]
          \[\# \<\text{through} > \<\text{thru}> \# \<\text{nonnumeric-literal}>\]
          \[\<\text{>,} \<\text{nonnumeric-literal}>\]
          \[\# \<\text{through} > \<\text{thru}> \# \<\text{nonnumeric-literal}>\]
          ...}
VALUE OF CLAUSE

D193  \langle\text{value-of-clause}\rangle ::= 
VALUE # OF #
\langle\text{elem-item-name-qualified}\rangle #
[IS #]
\{\langle\text{elem-item-name-qualified}\rangle !\langle\text{literal}\rangle\}\ 
[<,> \langle\text{elem-item-name-qualified}\rangle
[IS #]
\{\langle\text{elem-item-name-qualified}\rangle !\langle\text{literal}\rangle\}]...
IDENTIFIERS

D194 \(<\text{non-ws-record-name}>\) ::= 
\(<\text{non-elem-non-ms-record-name}>\) | 
\(<\text{non-elem-ms-record-name}>\) | 
\(<\text{non-elem-sort-record-name}>\) | 
\(<\text{elem-non-ms-record-name}>\) | 
\(<\text{elem-ms-record-name}>\) | 
\(<\text{elem-sort-record-name}>\)

D195 \(<\text{ws-record-name}>\) ::= 
\(<\text{non-elem-ws-record-name}>\) | 
\(<\text{elem-ws-record-name}>\)
D196 <in-of> ::=  
   # IN # ! # OF #

D197 <non-elem-non-ms-record-name-qualified> ::= 
   <non-elem-non-ms-record-name> 
   [<in-of> 
    <non-ms-file-name>] 

D198 <elem-non-ms-record-name-qualified> ::= 
   <elem-non-ms-record-name> 
   [<in-of> 
    <non-ms-file-name>] 

D199 <non-elem-ms-record-name-qualified> ::= 
   <non-elem-ms-record-name> 
   [<in-of> 
    <ms-file-name>] 

D200 <elem-ms-record-name-qualified> ::= 
   <elem-ms-record-name> 
   [<in-of> 
    <ms-file-name>] 

D201 <non-elem-sort-record-name-qualified> ::= 
   <non-elem-sort-record-name> 
   [<in-of> 
    <sort-file-name>] 

D202 <elem-sort-record-name-qualified> ::= 
   <elem-sort-record-name> 
   [<in-of> 
    <sort-file-name>]

D203 \(<\text{sum-report-group-name-qualified}>=\)<br>\([\langle\text{in-of}\rangle\]<br>\(<\text{report-name}\>]\)

D204 \(<\text{non-ms-record-name-qualified}>=\)<br>\(<\text{non-elem-non-ms-record-name-qualified}>=\) !
\(<\text{elem-non-ms-record-name-qualified}>=\)

D205 \(<\text{ms-record-name-qualified}>=\)<br>\(<\text{non-elem-ms-record-name-qualified}>=\) !
\(<\text{elem-ms-record-name-qualified}>=\)

D206 \(<\text{sort-record-name-qualified}>=\)<br>\(<\text{non-elem-sort-record-name-qualified}>=\) !
\(<\text{elem-sort-record-name-qualified}>=\)

D207 \(<\text{non-elem-record-name-qualified}>=\)<br>\(<\text{non-elem-non-ms-record-name-qualified}>=\) !
\(<\text{non-elem-ms-record-name-qualified}>=\) !
\(<\text{non-elem-sort-record-name-qualified}>=\) !
\(<\text{non-elem-ws-record-name}>=\)

D208 \(<\text{elem-record-name-qualified}>=\)<br>\(<\text{elem-non-ms-record-name-qualified}>=\) !
\(<\text{elem-ms-record-name-qualified}>=\) !
\(<\text{elem-sort-record-name-qualified}>=\) !
\(<\text{elem-ws-record-name}>=\)

D209 \(<\text{non-elem-report-group-name-qualified}>=\)<br>\([\langle\text{in-of}\rangle\]<br>\(<\text{report-name}\>]\)

D210 \(<\text{elem-non-sum-report-group-name-qualified}>=\)<br>\([\langle\text{in-of}\rangle\]<br>\(<\text{report-name}\>]\)

D211 \(<\text{report-group-name-qualified}>=\)<br>\(<\text{non-elem-report-group-name-qualified}>=\) !
\(<\text{elem-non-sum-report-group-name-qualified}>=\) !
\(<\text{sum-report-group-name-qualified}>=\)
D212  \texttt{<non-elem-02-48-item-name-qualified>} ::=  
      \texttt{<non-elem-02-48-item-name>}  
      [\texttt{<in-of>}  
      \texttt{<non-elem-02-48-item-name>}] ...  
      [\texttt{<in-of>}  
      \texttt{<non-elem-record-name-qualified>}]  

D213  \texttt{<elem-02-49-item-name-qualified>} ::=  
      \texttt{<elem-02-49-item-name>}  
      [\texttt{<in-of>}  
      \texttt{<non-elem-02-48-item-name-qualified>}]  

D214  \texttt{<02-49-item-name-qualified>} ::=  
      \texttt{<non-elem-02-48-item-name-qualified>} !  
      \texttt{<elem-02-49-item-name-qualified>}  

D215  \texttt{<non-elem-66-item-name-qualified>} ::=  
      \texttt{<non-elem-66-item-name>}  
      [\texttt{<n-of>}  
      \texttt{<non-elem-record-name-qualified>}]  

D216  \texttt{<elem-66-item-name-qualified>} ::=  
      \texttt{<elem-66-item-name>}  
      [\texttt{<in-of>}  
      \texttt{<non-elem-record-name-qualified>}]
D217 <sum-field-name-qualified> ::= 
   <sum-field-name> 
   [{<in-of>
     <non-elem-field-name>}]... 
   [{<non-elem-report-group-name-qualified>}] 

D218 <non-elem-item-name-qualified> ::= 
   <non-elem-record-name-qualified> | 
   <non-elem-02-49-item-name-qualified> | 
   <non-elem-66-item-name-qualified> 

D219 <elem-item-name-qualified> ::= 
   <elem-record-name-qualified> | 
   <elem-02-49-item-name-qualified> | 
   <elem-66-item-name-qualified> | 
   <77-item-name> | 
   <special-item-name-qualified> 

D220 <item-name-qualified> ::= 
   <non-elem-item-name-qualified> | 
   <elem-item-name-qualified> 

D221 <sum-item-name-qualified> ::= 
   <sum-report-group-name-qualified> | 
   <sum-field-name-qualified> 

D222 <special-item-name-qualified> ::= 
   TALLY ! 
   LINE-COUNTER 
   [{<in-of> <report-name>}] ! 
   PAGE-COUNTER 
   [{<in-of> <report-name>}] 

D223 <condition-name-qualified> ::= 
   <condition-name> 
   [{<in-of> 
     <02-49-item-name-qualified>}] 

D224 <index-name-qualified> ::= 
   <index-name> 
   [{<in-of> 
     <02-49-item-name-qualified>}]
\[\text{D225}\ <\text{non-elem-02-48-item-name-identifier}> ::=\\\ \text{<non-elem-02-48-item-name-qualified}>\\\ [(\text{subscripts})]]\]

\[\text{D226}\ <\text{elem-02-49-item-name-identifier}> ::=\\\ <\text{elem-02-49-item-name-qualified}>\\\ [(\text{subscripts})]]\]

\[\text{D227}\ <\text{non-elem-item-name-identifier}> ::=\\\ <\text{non-elem-record-name-qualified}> !\\\ <\text{non-elem-02-48-item-name-identifier}> !\\\ <\text{non-elem-66-item-name-qualified}> !\]

\[\text{D228}\ <\text{elem-item-name-identifier}> ::=\\\ <\text{elem-record-name-qualified}> !\\\ <\text{elem-02-49-item-name-identifier}> !\\\ <\text{elem-66-item-name-qualified}> !\\\ <\text{77-item-name}> !\\\ <\text{special-item-name-qualified}> !\]

\[\text{D229}\ <\text{item-name-identifier}> ::=\\\ <\text{non-elem-item-name-identifier}> !\\\ <\text{elem-item-name-identifier}> !\]

\[\text{D230}\ <\text{source-item-name-identifier}> ::=\\\ <\text{item-name-identifier}> !\\\ <\text{sum-item-name-qualified}> !\]

\[\text{D231}\ <\text{summed-item-name-identifier}> ::=\\\ <\text{elem-item-name-identifier}> !\\\ <\text{sum-item-name-qualified}> !\]

\[\text{D232}\ <\text{non-elem-data-name-identifier}> ::=\\\ <\text{non-elem-item-name-identifier}> !\]

\[\text{D233}\ <\text{elem-data-name-identifier}> ::=\\\ <\text{elem-item-name-identifier}> !\\\ <\text{sum-item-name-qualified}> !\]

\[\text{D234}\ <\text{data-name-identifier}> ::=\\\ <\text{non-elem-data-name-identifier}> !\\\ <\text{elem-data-name-identifier}> !\]

\[\text{D235}\ <\text{condition-name-identifier}> ::=\\\ <\text{condition-name-qualified}>\\\ [(\text{subscripts})]]\]
D236 ⟨subscripts⟩ ::= 
 ⟨elem-item-name-qualified⟩
 [⟨,⟩
 ⟨elem-item-name-qualified⟩
 ]...!
 ⟨relative-index-name-qualified⟩
 [⟨,⟩
 ⟨relative-index-name-qualified⟩
 ]...

D237 ⟨relative-index-name-qualified⟩ ::= 
 ⟨index-name-qualified⟩
 [# {+1 -1} #
 ⟨integer⟩]
P. PROCEDURE DIVISION

PROCEDURE DIVISION STRUCTURE

P1 \[\text{procedure-division} ::= \]
\[\text{PROCEDURE DIVISION} \text{ } \cdot\]
\[\text{procedure-division-body}\]

P2 \[\text{procedure-division-body} ::= \]
\[\{\text{declarative-portion}\}\]
\[\text{non-declarative-portion}\]
\[\text{paragraph}\]...
DECLARATIVE PORTION

P3  ⟨declarative-portion⟩ ::= 
    <!DECLARATIVES ⟨.⟩
    ⟨declarative-section⟩...
    <!END # DECLARATIVES ⟨.⟩

NON-DECLARATIVE PORTION

P4 \langle non-declarative-portion \rangle ::=
\langle non-declarative-section \rangle...
SECTIONS

P5  \langle declarative-section \rangle ::=  
    \langle section-name-declaration \rangle # SECTION \langle . \rangle  
    \langle copy-sentence \rangle !  
    \langle declarative-sentence \rangle \langle section-body \rangle  

P6  \langle non-declarative-section \rangle ::=  
    \langle section-name-declaration \rangle # SECTION  
    [\# \langle priority-number \rangle] \langle . \rangle  
    \langle copy-sentence \rangle !  
    \langle section-body \rangle  

P7  \langle priority-number \rangle ::=  
    [0]...  \langle empty \rangle ! \langle non-zero-digit \rangle \langle digit \rangle
SECTION NAME

P8 \( \langle \text{section-name-declaration} \rangle ::= \langle \text{non-reserved-word} \rangle \)

P9 \( \langle \text{section-name} \rangle ::= \langle \text{non-reserved-word} \rangle \text{ which appears as a } \langle \text{section-name-declaration} \rangle \)
SECTION BODY

P10 ⟨section-body⟩ ::= ⟨paragraph⟩...
<paragraph> ::= 
  ← <paragraph-name-declaration> ⟨.⟩ ↑
  ⟨<copy-sentence>⟩!
  ⟨<paragraph-body⟩⟩
PARAGRAPHS NAME

P12  \textlangle \text{paragraph-name-declaration} \textrangle ::= \\
  \textlangle \text{non-reserved-word} \textrangle

P13  \textlangle \text{paragraph-name} \textrangle ::= \\
  \textlangle \text{non-reserved-word} \textrangle \text{ which appears as a} \\
  \textlangle \text{paragraph-name-declaration} \textrangle

P14  \textlangle \text{paragraph-name-qualified} \textrangle ::= \\
  \textlangle \text{paragraph-name} \textrangle [\textlangle \text{in-of} \textrangle \textlangle \text{section-name} \textrangle]
PROCEDURE NAME

P15 \langle procedure-name \rangle ::= \\
\langle section-name \rangle ! \\
\langle paragraph-name-qualified \rangle
PARAGRAPH BODY

P16 \( \langle \text{paragraph-body} \rangle \) ::= 
\( \langle \text{note-paragraph-body} \rangle \) !
\( \langle \text{exit-paragraph-body} \rangle \) !
\( \langle \text{alterable-go-to-paragraph-body} \rangle \) !
\( \langle \text{regular-paragraph-body} \rangle \)

P17 \( \langle \text{note-paragraph-body} \rangle \) ::= 
\( \langle \text{note-sentence} \rangle \) [\( \langle \text{comment-sentence} \rangle \)]...

P18 \( \langle \text{exit-paragraph-body} \rangle \) ::= 
\( \langle \text{exit-sentence} \rangle \)

P19 \( \langle \text{alterable-go-to-paragraph-body} \rangle \) ::= 
\( \langle \text{alterable-go-to-sentence} \rangle \)

P20 \( \langle \text{regular-paragraph-body} \rangle \) ::= 
\{\( \langle \text{regular-sentence} \rangle \) !
\( \langle \text{other-language-block}\rangle \}\)
[\( \langle \text{regular-sentence} \rangle \) !
\( \langle \text{other-language-block}\rangle \) !
\( \langle \text{note-sentence} \rangle \) ...]
P21 \(<\text{regular-sentence}\> ::= <\text{regular-imperative-sentence}> | <\text{conditional-sentence}> | <\text{enter-routine-sentence}>

P22 \(<\text{other-language-block}\> ::= <\text{enter-other-language-sentence}> <\text{other-language-string}> <\text{enter-cobol-sentence}>\)
IMPERATIVE SENTENCES

P23  \(<exit\text{-sentence}>::=\)
     \(<exit\text{-statement}>\ <.>\)

P24  \(<\text{alterable-go-to-sentence}>::=\)
     \(<\text{alterable-go-to-statement}>\ <.>\)

P25  \(<\text{regular-imperative-sentence}>::=\)
     \(<\text{regular-imperative-statement}>\ <.>\)
CONDITIONAL SENTENCE

P26  \( \langle \text{conditional-sentence} \rangle ::= 
\quad [\langle \text{continuable-imperative-statement} \rangle \langle ; \rangle ] 
\quad \langle \text{conditional-statement} \rangle \langle . \rangle \)
COMPILER DIRECTING SENTENCES

P27  \(\text{<note-sentence>} ::= \text{<note-statement>} <.>\)

P28  \(\text{<comment-sentence>} ::= \text{<comment-string>} <.>\)

P29  \(\text{<enter-routine-sentence>} ::= \text{<enter-routine-statement>} <.>\)

P30  \(\text{<enter-other-language-sentence>} ::= \text{<enter-other-language-statement>} <.>\)

P31  \(\text{<enter-cobol-sentence>} ::= \text{<enter-cobol-statement>} <.>\)
DEclarative Sentence

P32  \langle\text{declarative-sentence}\rangle::=\\n    \langle\text{declarative-statement}\rangle\.\rangle
IMPERATIVE STATEMENTS

P33  <regular-imperative-statement> ::=  
    <continuable-imperative-statement>  
    [ ; > <terminating-imp-verb-statement> ] !  
    <terminating-imp-verb-statement>

P34  <continuable-imperative-statement> ::=  
    <continuable-imp-verb-statement>  
    [ <continuable-imp-verb-statement> ] ...  

P35  <continuable-imp-verb-statement> ::=  
    <imp-arithmetic-statement> !  
    <move-statement> !  
    <examine-statement> !  
    <alter-statement> !  
    <go-to-depending-statement> !  
    <perform-statement> !  
    <stop-literal-statement> !  
    <set-statement> !  
    <imperative-i-o-statement> !  
    <release-statement> !  
    <sort-statement> !  
    <generate-statement> !  
    <initiate-statement> !  
    <terminate-statement>

P36  <imp-arithmetic-statement> ::=  
    <imp-add-statement> !  
    <imp-subtract-statement> !  
    <imp-multiply-statement> !  
    <imp-divide-statement> !  
    <imp-compute-statement>

P37  <imperative-i-o-statement> ::=  
    <accept-statement> !  
    <display-statement> !  
    <open-statement> !  
    <close-statement> !  
    <imperative-write-statement> !  
    <seek-statement>

P38  <terminating-imp-verb-statement> ::=  
    <simple-go-to-statement> !  
    <stop-run-statement>
P39  \[\text{conditional-statement} \ ::= \]
\[\text{if-statement} \]
\[\text{imp-arithmetic-statement} \]
\[\text{;} \quad \text{size-error-phrase} \]
\[\text{search-statement} \]
\[\text{read-statement} \]
\[\text{write-invalid-key-statement} \]
\[\text{return-statement} \]
P40 \[
\text{<declarative-statement> ::=}
\text{<use-error-statement> !}
\text{<use-label-statement> !}
\text{<use-before-reporting-statement>}
\]
COMMON OPTIONS

P41  \(<\text{size-error-phrase}\> ::= \[\text{ON \#} \] \text{SIZE \# ERROR \#} \<\text{regular-imperative-statement}\>

P42  \(<\text{at-end-phrase}\> ::= \[\text{AT \#} \] \text{END \#} \<\text{regular-imperative-statement}\>

P43  \(<\text{invalid-key-phrase}\> ::= \text{INVALID \# [KEY \#]} \<\text{regular-imperative-statement}\>
COMMON TERMS

P44 \( \langle \text{result} \rangle ::= \langle \text{elem-item-name-identifier} \rangle \ [# \langle \text{rounded} \rangle \] \\
P45 \( \langle \text{rounded} \rangle ::= \) \\
    \text{ROUNDED} \\
P46 \( \langle \text{numeric-operand} \rangle ::= \) \\
    \langle \text{elem-item-name-identifier} \rangle ! \\
    \langle \text{numeric-literal} \rangle \\
P47 \( \langle \text{literal-for-display-stop} \rangle ::= \) \\
    \langle \text{simple-figurative-constant} \rangle ! \\
    \langle \text{proper-numeric-literal} \rangle ! \\
    \langle \text{proper-nonnumeric-literal} \rangle
ARITHMETIC EXPRESSION

P48  \(<\text{arithmetic-expression}\> ::= \<\text{term}\> [\# \{+ \mid -\} \# \<\text{term}\>]... 

P49  \(<\text{term}\> ::= \<\text{factor}\> [\# \{\ast \mid /\} \# \<\text{factor}\>]...

P50  \(<\text{factor}\> ::= \<\text{primary}\> [\# \{\ast \mid \# \# \<\text{primary}\>]\...

P51  \(<\text{primary}\> ::= [\{+ \mid -\} \#] \<\text{unsigned-primary}\>

P52  \(<\text{unsigned-primary}\> ::= \<\text{numeric-operand}\> \mid \(<\text{arithmetic-expression}\>)}
CONDITIONS

P53  ⟨condition⟩ ::=  
    ⟨condition-term⟩ [# OR # ⟨condition-term⟩]...

P54  ⟨condition-term⟩ ::=  
    ⟨condition-factor⟩ [# AND # ⟨condition-factor⟩]...

P55  ⟨condition-factor⟩ ::=  
    [⟨not⟩ #] ⟨condition-primary⟩

P56  ⟨not⟩ ::=  
    NOT

P57  ⟨condition-primary⟩ ::=  
    ⟨simple-condition⟩ !  
    ⟨abbreviated-relation-condition⟩ !  
    ⟨condition⟩)

P58  ⟨simple-condition⟩ ::=  
    ⟨relation-condition⟩ !  
    ⟨class-condition⟩ !  
    ⟨condition-name-condition⟩ !  
    ⟨switch-status-condition⟩ !  
    ⟨sign-condition⟩

P59  ⟨abbreviated-relation-condition⟩ ::=  
    ⟨relation-condition⟩  
    [# AND # ⟨abbreviation⟩]...  
    [# OR # ⟨abbreviation⟩]...  
    [# AND # ⟨abbreviation⟩]...[...]

P60  ⟨abbreviation⟩ ::=  
    [[⟨not⟩ #] ⟨relational-operator⟩ #]  
    ⟨relation-operand⟩

P61  ⟨relation-condition⟩ ::=  
    ⟨relation-operand⟩ #  
    ⟨relational-operator⟩ #  
    ⟨relation-operand⟩

P62  ⟨relation-operand⟩ ::=  
    ⟨item-name-identifier⟩ !  
    ⟨index-name-qualified⟩ !  
    ⟨nonnumeric-literal⟩ !  
    ⟨arithmetic-expression⟩

P63  ⟨relational-operator⟩ ::=  
    [IS #] [⟨not⟩ #] {>} ! GREATER [# THAN]!  
    [IS #] [⟨not⟩ #] {<} ! LESS [# THAN]!  
    [IS #] [⟨not⟩ #] {=} ! EQUAL [# TO]!
P64 \( \langle \text{class-condition} \rangle ::= \\langle \text{item-name-identifier} \rangle \# \)  
\[ \text{IS} \# \] \[ \langle \text{not} \# \rangle \{ \text{NUMERIC} \land \text{ALPHABETIC} \} \]

P65 \( \langle \text{condition-name-condition} \rangle ::= \langle \text{condition-name-identifier} \rangle \)

P66 \( \langle \text{switch-status-condition} \rangle ::= \langle \text{switch-status-name} \rangle \)

P67 \( \langle \text{sign-condition} \rangle ::= \langle \text{elem-item-name-identifier} \rangle \# \)  
\[ \text{IS} \# \] \[ \langle \text{not} \# \rangle \{ \text{POSITIVE} \land \text{NEGATIVE} \land \text{ZERO} \} \)
ACCEPT STATEMENT

P68 \langle accept-statement\rangle ::= 
  ACCEPT # \langle item-name-identifier\rangle 
  [# FROM #
  \{\langle mnemonic-name-for-individual-io-unit\rangle !
  \langle mnemonic-name-for-type-of-io-unit\rangle\}]
ADD STATEMENT

P69  <imp-add-statement> ::=  
    ADD # <numeric-operand>  
    [{<,> <numeric-operand>}]... #  
    TO # <result>  
    [{<,> <result>}]... !  
    ADD # <numeric-operand>  
    [{<,> <numeric-operand>}]... #  
    GIVING # <result> !  
    ADD # {CORRESPONDING ! CORR} #  
    <non-elem-data-name-identifier> #  
    TO # <non-elem-data-name-identifier> [# <rounded>]
ALTER STATEMENT

P70 \(<alter-statement> ::= \<ALTER \# \<alteration> [\<,\> \<alteration>]\)...\n
P71 \(<alteration> ::= \<paragraph-name-qualified> \# TO \# \[\<PROCEED \# TO \#]\) \<destination>\)
CLOSE STATEMENT

P72  \langle\text{close-statement}\rangle ::= 
    \text{CLOSE} \ # \ \langle\text{closure}\rangle \ [\langle,\rangle \ \langle\text{closure}\rangle]\ldots

P73  \langle\text{closure}\rangle ::= 
    \langle\text{non-ms-file-name}\rangle 
    \[\# \ \langle\text{reel}\rangle\] \[[\# \ \text{WITH}] \ # \ \{\langle\text{no-rewind}\rangle \ | \ \langle\text{lock}\rangle\}\] !
    \langle\text{sequential-ms-file-name}\rangle 
    \[\# \ \langle\text{unit}\rangle\] \[[\# \ \text{WITH}] \ # \ \langle\text{lock}\rangle\] !
    \langle\text{random-ms-file-name}\rangle 
    \[[\# \ \text{WITH}] \ # \ \langle\text{lock}\rangle\]

P74  \langle\text{reel}\rangle ::= 
    \text{REEL}

P75  \langle\text{unit}\rangle ::= 
    \text{UNIT}

P76  \langle\text{no-rewind}\rangle ::= 
    \text{NO} \ # \ \text{REWIND}

P77  \langle\text{lock}\rangle ::= 
    \text{LOCK}
COMPUTE STATEMENT

P78  \texttt{(imp-compute-statement)} ::= \\
    \texttt{COMPUTE} \# \texttt{<result>} \# = \# \texttt{<arithmetic-expression>}

DISPLAY STATEMENT

P79  \(<display\text{-}statement> ::= \\
   \text{DISPLAY} \ # \ <display\text{-}operand> \\
   [\langle,\rangle \ <display\text{-}operand>]... \\
   [\# \ UPON \ # \\
    \{\langle\text{mnemonic\text{-}name\text{-}for\text{-}individual\text{-}io\text{-}unit}\rangle \mid \\
    \langle\text{mnemonic\text{-}name\text{-}for\text{-}type\text{-}of\text{-}io\text{-}unit}\rangle\}]\\n\)

P80  \(<display\text{-}operand> ::= \\
   \langle\text{item\text{-}name\text{-}identifier}\rangle \mid \\
   \langle\text{literal\text{-}for\text{-}display\text{-}stop}\rangle
DIVIDE STATEMENT

P81  <imp-divide-statement> ::=  
    DIVIDE # <numeric-operand> #  
    INTO # <result> !  
    DIVIDE # <numeric-operand> #  
    INTO # <numeric-operand> # GIVING # <result>  
    [ # REMAINDER # <elem-item-name-identifier> ] !  
    DIVIDE # <numeric-operand> #  
    BY # <numeric-operand> # GIVING # <result>  
    [ # REMAINDER # <elem-item-name-identifier> ]
ENTER STATEMENT

P82  \langle enter-routine-statement\rangle ::=
      ENTER # \langle other-language-name\rangle # \langle routine-name\rangle

P83  \langle enter-other-language-statement\rangle ::=
      ENTER # \langle other-language-name\rangle

P84  \langle enter-cobol-statement\rangle ::=
      ENTER # COBOL

P85  \langle routine-name\rangle ::=
      \langle non-reserved-word\rangle
EXAMINE STATEMENT

P86  <examine-statement> ::= 
    EXAMINE # <item-name-identifier> # 
    TALLYING # 
    <ALL | LEADING | UNTIL | FIRST> # 
    <one-character-literal> 
    [# REPLACING # BY # 
    <one-character-literal>] ! 
    EXAMINE # <item-name-identifier> # 
    REPLACING # 
    <ALL | LEADING | FIRST | UNTIL | FIRST> # 
    <one-character-literal> # 
    BY # 
    <one-character-literal>

P87  <one-character-literal> ::= 
    " <computer-character> diff <quotation-mark> " ! 
    <digit> ! 
    <simple-figurative-constant>
EXIT STATEMENT

P88 \<exit-statement> ::= EXIT
GENERATE STATEMENT

P89  <generate-statement> ::= 
    GENERATE # <report-name> !
    GENERATE # <report-group-name-qualified>
GO TO STATEMENT

P90  \langle simple-go-to-statement \rangle ::=  
      GO # TO # \langle destination \rangle

P91  \langle alterable-go-to-statement \rangle ::=  
      GO # TO [ # \langle destination \rangle ]

P92  \langle go-to-depending-statement \rangle ::=  
      GO # TO # \langle destination \rangle \{ \langle, \rangle \langle destination \rangle \}... #  
      DEPENDING # [ ON # ] \langle elem-item-name-identifier \rangle

P93  \langle destination \rangle ::=  
      \langle procedure-name \rangle
IF STATEMENT

P94  <if-statement> ::=  
\quad IF # <condition> 
\quad <;;> {<statement> ! NEXT # SENTENCE} 
\quad <;;> ELSE {<statement> ! NEXT # SENTENCE}

P95  <statement> ::=  
\quad <regular-imperative-statement> ! 
\quad [[<continuable-imperative-statement> <;;>]] 
\quad <conditional-statement>
INITIATE STATEMENT

P96  \( \langle \text{initiate-statement} \rangle ::= \)
     \( \text{INITIATE} \ # \ \langle \text{report-name} \rangle \ [\langle, \rangle \ \langle \text{report-name} \rangle] \ldots \)
MOVE STATEMENT

P97 <move-statement> ::= 
   MOVE # `{literal} ! <item-name-identifier>\} # 
   TO # <item-name-identifier> 
   [\<,> <item-name-identifier>],.. ! 
   MOVE # \{CORRESPONDING | CORR\} # 
   <non-elem-item-name-identifier> # 
   TO # <non-elem-item-name-identifier>
MULTIPLY STATEMENT

P98  <imp-multiply-statement> ::= 
    MULTIPLY # <numeric-operand> # BY # <result> !
    MULTIPLY # <numeric-operand> #
    BY # <numeric-operand> # GIVING # <result>
NOTE STATEMENT

P99 \( \langle \text{note-statement} \rangle ::= \)
\[
\text{NOTE} \ # \ \langle \text{comment-string} \rangle
\]
OPEN STATEMENT

P100 <open-statement> ::= OPEN <open-options>

P101 <open-options> ::= {<input-option> ↓
[<output-option>] ↓
[<i-o-option>]!}
{<output-option> ↓
[<i-o-option>]!}
<i-o-option>

P102 <input-option> ::= # INPUT # <input-file> [','] <input-file>]

P103 <output-option> ::= # OUTPUT # <output-file> [','] <output-file>

P104 <i-o-option> ::= # I-O # <i-o-file> [','] <i-o-file>

P105 <input-file> ::= <ms-file-name> |
[<non-ms-file-name> [# <reversed> | [# WITH] # <no-rewind>]]

P106 <output-file> ::= <ms-file-name> |
[<non-ms-file-name> [# WITH] # <no-rewind>]

P107 <i-o-file> ::= <ms-file-name>

P108 <reversed> ::= REVERSED
PERFORM STATEMENT

P109  <perform-statement> ::= 
      PERFORM # <range> 
      [# <times-option> ] 
      [# <until-option> ] 
      [# <varying-option> ]

P110  <range> ::= 
      <procedure-name> 
      [# <THROUGH | THRU> # <procedure-name> ]

P111  <times-option> ::= 
      <integral-operand> # TIMES

P112  <until-option> ::= 
      UNTIL # <condition>

P113  <varying-option> ::= 
      VARYING # <varying-control-phrase> 
      [# AFTER # <varying-control-phrase> 
      [# AFTER # <varying-control-phrase> ]]

P114  <varying-control-phrase> ::= 
      <control-variable> # 
      FROM # 
      {<index-name-qualified> ! <numeric-operand>} # 
      BY # 
      <numeric-operand> # 
      UNTIL # <condition>

P115  <integral-operand> ::= 
      <integer> ! 
      <elem-item-name-identifier>

P116  <control-variable> ::= 
      <elem-item-name-identifier> ! 
      <index-name-qualified>
READ STATEMENT

P117 \( \langle \text{read-statement} \rangle ::= \)
\[
\text{READ} \# \langle \text{sequential-file-name} \rangle [\# \text{RECORD}]
[\# \text{INTO} \# \langle \text{item-name-identifier} \rangle ]
\langle ; \rangle \langle \text{at-end-phrase} \rangle !
\text{READ} \# \langle \text{random-ms-file-name} \rangle [\# \text{RECORD}]
[\# \text{INTO} \# \langle \text{item-name-identifier} \rangle ]
\langle ; \rangle \langle \text{invalid-key-phrase} \rangle
\]
RELEASE STATEMENT

P118 \( \langle \text{release-statement} \rangle ::= \)
    \( \text{RELEASE} \# \langle \text{sort-record-name-qualified} \rangle \)
    \( [\# \text{FROM} \# \langle \text{item-name-identifier} \rangle] \)
RETURN STATEMENT

P119 \textless{}\textit{return-statement}\textgreater{} ::= 
\textsc{RETURN} # \textless{}\textit{sort-file-name}\textgreater{} [\# \textsc{RECORD}] 
[# INTO # \textless{}\textit{item-name-identifier}\textgreater{}] 
\text{";"} \textless{}\textit{at-end-phrase}\textgreater{}}
SEARCH STATEMENT

P120  <search-statement> ::=  
      SEARCH # <02-49-item-name-qualified>  
      [# VARYING # <control-variable>]  
      [\>; <at-end-phrase>]  
      [\>; <when-phrase>]... !  
      SEARCH # ALL # <02-49-item-name-qualified>  
      [\>; <at-end-phrase>]  
      [\>; <when-phrase>]

P121  <when-phrase> ::=  
      WHEN # <condition> #  
      {<regular-imperative-statement> !  
      NEXT # SENTENCE}
SEEK STATEMENT

P122  \textless \textit{seek-statement} \textgreater : =
        \texttt{SEEK \# \langle random-ms-file-name \rangle [\# RECORD]}
SET STATEMENT

P123  <set-statement> ::=  
    SET # <control-variable>  
    [<> <control-variable>]... #  
    TO #  
    {! <integral-operand> ! <index-name-qualified> } !  
    SET # <index-name-qualified>  
    [<> <index-name-qualified>]... #  
    {UP | DOWN} # BY # <integral-operand>
SORT STATEMENT

P124  \[sort\text{-}statement\] ::= 
  \(\text{SORT \# \langle\text{sort\text{-}file\text{-}name}\rangle \#} \)
  \(\langle\text{key\text{-}clause}\rangle \ [\langle;\rangle \ \langle\text{key\text{-}clause}\rangle]\ldots \# \)
  \(\langle\text{sort\text{-}input\text{-}specification}\rangle \# \)
  \(\langle\text{sort\text{-}output\text{-}specification}\rangle \)

P125  \(\langle\text{key\text{-}clause}\rangle ::= \)
  \([\text{ON \#}] \ \langle\text{ASCENDING} \mid \text{DESCENDING}\rangle \# \ [\text{KEY \#}] \)
  \(\langle\text{sort\text{-}key\text{-}identifier}\rangle \)
  \(\[\langle,\rangle \ \langle\text{sort\text{-}key\text{-}identifier}\rangle\ldots \]

P126  \(\langle\text{sort\text{-}input\text{-}specification}\rangle ::= \)
  \(\langle\text{USING \# \langle\text{non\text{-}sort\text{-}file\text{-}name}\rangle \rangle ! \)
  \(\langle\text{INPUT \# \langle\text{PROCEDURE \# \ [IS \#] \langle\text{sort\text{-}procedure\text{-}range}\rangle \rangle \rangle \rangle \}

P127  \(\langle\text{sort\text{-}output\text{-}specification}\rangle ::= \)
  \(\langle\text{GIVING \# \langle\text{non\text{-}sort\text{-}file\text{-}name}\rangle \rangle ! \)
  \(\langle\text{OUTPUT \# \langle\text{PROCEDURE \# \ [IS \#] \langle\text{sort\text{-}procedure\text{-}range}\rangle \rangle \rangle \rangle \}

P128  \(\langle\text{sort\text{-}procedure\text{-}range}\rangle ::= \)
  \(\langle\text{section\text{-}name}\rangle \)
  \([\# \ \langle\text{THROUGH} \mid \text{THRU}\rangle \# \langle\text{section\text{-}name}\rangle\]

P129  \(\langle\text{sort\text{-}key\text{-}identifier}\rangle ::= \)
  \(\langle\text{sort\text{-}record\text{-}name\text{-}qualified} \rangle ! \)
  \(\langle\text{non\text{-}elem\text{-}02\text{-}48\text{-}item\text{-}name\text{-}identifier} \rangle ! \)
  \(\langle\text{elem\text{-}02\text{-}49\text{-}item\text{-}name\text{-}identifier} \rangle \)
STOP STATEMENT

P130  \[ \text{stop-literal-statement} ::= \]
      \hspace{1em} \text{STOP} \# \text{literal-for-display-stop} \\

P131  \[ \text{stop-run-statement} ::= \]
      \hspace{1em} \text{STOP} \# \text{RUN}
SUBTRACT STATEMENT

P132 \langle imp-subtract-statement \rangle \ ::= \\
\hspace{1em} \text{SUBTRACT} \ # \ \langle \text{numeric-operand} \rangle \\
\hspace{2em} [\langle , \rangle \ \langle \text{numeric-operand} \rangle] \ldots \ # \\
\hspace{1em} \text{FROM} \ # \ \langle \text{result} \rangle \\
\hspace{2em} [\langle , \rangle \ \langle \text{result} \rangle] \ldots \ ! \\
\hspace{1em} \text{SUBTRACT} \ # \ \langle \text{numeric-operand} \rangle \\
\hspace{2em} [\langle , \rangle \ \langle \text{numeric-operand} \rangle] \ldots \ # \\
\hspace{1em} \text{FROM} \ # \ \langle \text{numeric-operand} \rangle \ # \ \text{GIVING} \ # \ \langle \text{result} \rangle \ ! \\
\hspace{1em} \text{SUBTRACT} \ # \ \langle \text{CORRESPONDING} \ 1 \ \text{CORR} \rangle \ # \\
\hspace{1em} \langle \text{non-elem-item-name-identifier} \rangle \ # \\
\hspace{1em} \text{FROM} \ # \ \langle \text{non-elem-item-name-identifier} \rangle \\
[\# \ \langle \text{rounded} \rangle \]
TERMINATE STATEMENT

P133 <terminate-statement> ::= 
    TERMINATE # <report-name> [<,> <report-name>]...
USE STATEMENT

P134  \textlt{use-error-statement} ::= \\
\hspace{1em} \textsl{USE} \# \textsl{AFTER} \# \textsl{[STANDARD \#]} \textsl{ERROR} \# \\
\hspace{1em} \textsl{PROCEDURE} \# \textsl{[ON \#]} \\
\hspace{2em} \{\textlt{non-sort-file-name} \\
\hspace{3em} \{\textlt{,} \textlt{non-sort-file-name}\} \ldots \! \\
\hspace{2em} \textsl{INPUT} \! \\
\hspace{2em} \textsl{OUTPUT} \! \\
\hspace{2em} \textsl{I-O} \}

P135  \textlt{use-label-statement} ::= \\
\hspace{1em} \textsl{USE} \# \{\textsl{BEFORE} \! \textsl{! AFTER}\! \# \textsl{[STANDARD \#]} \\
\hspace{2em} \{\textlt{beginning} \# \! \textlt{ending} \# \! \\
\hspace{3em} \{\textlt{reel} \# \! \textlt{file} \# \} \textsl{LABEL} \# \textsl{PROCEDURE} \# \textsl{[ON \#]} \\
\hspace{4em} \{\textlt{non-ms-file-name} \\
\hspace{5em} \{\textlt{,} \textlt{non-ms-file-name}\} \ldots \! \\
\hspace{2em} \textsl{INPUT} \! \\
\hspace{2em} \textsl{OUTPUT} \! \\
\hspace{2em} \{\textlt{unit} \# \! \textlt{file} \# \} \textsl{LABEL} \# \textsl{PROCEDURE} \# \textsl{[ON \#]} \\
\hspace{3em} \{\textlt{sequential-ms-file-name} \\
\hspace{4em} \{\textlt{,} \textlt{sequential-ms-file-name}\} \ldots \! \\
\hspace{2em} \textsl{INPUT} \! \\
\hspace{2em} \textsl{OUTPUT} \! \\
\hspace{2em} \textsl{I-O} \! \\
\hspace{2em} \{\textlt{file} \# \} \textsl{LABEL} \# \textsl{PROCEDURE} \# \textsl{[ON \#]} \\
\hspace{3em} \{\textlt{random-ms-file-name} \\
\hspace{4em} \{\textlt{,} \textlt{random-ms-file-name}\} \ldots \! \\
\hspace{2em} \textsl{INPUT} \! \\
\hspace{2em} \textsl{OUTPUT} \! \\
\hspace{2em} \textsl{I-O}\!

P136  \textlt{use-before-reporting-statement} ::= \\
\hspace{1em} \textsl{USE} \# \textsl{BEFORE} \# \textsl{REPORTING} \# \\
\hspace{2em} \textsl{[report-group-name-qualified]}

P137  \textlt{file} ::= \\
\hspace{1em} \textsl{FILE}

P138  \textlt{beginning} ::= \\
\hspace{1em} \textsl{BEGINNING}

P139  \textlt{ending} ::= \\
\hspace{1em} \textsl{ENDING}
WRITE STATEMENT

P140  <imperative-write-statement> ::=  
WRITE # <non-ms-record-name-qualified>  
[ # FROM # <item-name-identifier> ]  
[ # <advancing-phrase> ]

P141  <write-invalid-key-statement> ::=  
WRITE # <ms-record-name-qualified>  
[ # FROM # <item-name-identifier> ]  
<;> <invalid-key-phrase>

P142  <advancing-phrase> ::=  
{ BEFORE | AFTER } # ADVANCING #  
{ <integral-operand> [# LINES] !  
<mnemonic-name-for-paper-advance> }
L. COBOL LIBRARY

STRUCTURE OF LIBRARY CALLS

L1  \(<\text{copy-entry}\> ::= \<\text{copy-clause}\> \<,\>

L2  \(<\text{copy-clause}\> ::= \<\text{library-call}\>

L3  \(<\text{copy-sentence}\> ::= \<\text{copy-statement}\> \<,\>

L4  \(<\text{copy-statement}\> ::= \<\text{library-call}\>

L5  \(<\text{library-call}\> ::= \COPY \# \<\text{library-name}\>
[# \text{REPLACING}
\# \<\text{word}\> \# \text{BY} \# \<\text{replacement}\>
[\<,\> \<\text{word}\> \# \text{BY} \# \<\text{replacement}\>]]

L6  \(<\text{replacement}\> ::= \<\text{word}\> \! \<\text{literal}\> \!
\<\text{data-name-identifier}\> \!
\<\text{procedure-name}\>
LIBRARY NAME

L7 \langle library-name \rangle ::= \\
\langle non-reserved-word \rangle
R. RESERVED WORDS

R1  <reserved-word> ::=  
  ACCEPT | ACCESS | ACTUAL | ADD | ADDRESS | 
  ADVANCING | ATER | ALL | ALPHABETIC | ALTER | 
  ALTERNATE | AND | ARE | AREA | AREAS | 
  ASCENDING | ASSIGN | AT | AUTHOR | BEFORE | 
  BEGINNING | BLANK | BLOCK | BY | CF | CH | 
  CHARACTERS | CLOCK-UNITS | CLOSE | COBOL | CODE | 
  COLUMN | COMMA | COMP | COMPUTATIONAL | COMPUTE | 
  CONFIGURATION | CONTAINS | CONTROL | CONTROLS | 
  COPY | CORR | CORRESPONDING | CURRENCY | DATA | 
  DATE-COMPILED | DATE-WRITTEN | DE | 
  DECIMAL-POINT | DECLARATIVES | DEPENDING | 
  DESCENDING | DETAIL | DISPLAY | DIVIDE | 
  DIVISION | DOWN | ELSE | END | ENDING | ENTER | 
  ENVIRONMENT | EQUAL | ERROR | EVERY | EXAMINE | 
  EXIT | FD | FILE | FILE-CONTROL | FILE-LIMIT | 
  FILE-LIMITS | FILLER | FINAL | FIRST | FOOTING | 
  FOR | FROM | GENERATE | GIVING | GO | GREATER | 
  GROUP | HEADING | HIGH-VALUE | HIGH-VALUES | 
  I-O | I-O-CONTROL | IDENTIFICATION | IF | IN | 
  INDEX | INDEXED | INDICATE | INITIATE | INPUT | 
  INPUT-OUTPUT | INSTALLATION | INTO | INVALID | 
  IS | JUST | JUSTIFIED | KEY | KEYS | LABEL | 
  LAST | LEADING | LEFT | LESS | LIMIT | LIMITS | 
  LINE | LINE-COUNTER | LINES | LOCK | LOW-VALUE | 
  LOW-VALUES | MEMORY | MODE | MODULES | MOVE | 
  MULTIPLE | MULTIPLY | NEGATIVE | NEXT | NO | 
  NOT | NOTE | NUMBER | NUMERIC | OBJECT-COMPUTER | 
  OCCURS | OF | OFF | OMITTED | ON | OPEN | 
  OPTIONAL | OR | OUTPUT | PAGE | PAGE-COUNTER | 
  PERFORM | PF | PH | PIC | PICTURE | PLUS | 
  POSITION | POSITIVE | PROCEDURE | PROCEED | 
  PROCESSING | PROGRAM-ID | QUOTE | QUOTES | 
  RANDOM | RD | READ | RECORD | RECORDS | 
  REDEFINED | REEL | RELEASE | REMARKS | RENAMES | 
  REPLACING | REPORT | REPORTING | REPORTS | 
  RERUN | RESERVE | RESET | RETURN | REVERSED | 
  REWIND | RF | RH | RIGHT | ROUNDED | RUN | SAME | 
  SD | SEARCH | SECTION | SECURITY | SEEK | 
  SEGMENT-LIMIT | SELECT | SENTENCE | SEQUENTIAL | 
  SET | SIGN | SIZE | SORT | SOURCE | 
  SOURCE-COMPUTER | SPACE | SPACES | 
  SPECIAL-NAME | STANDARD | STATUS | STOP | 
  SUBTRACT | SUM | SYNC | SYNCHRONIZED | TALLY | 
  TALLYING | TAPE | TERMINATE | THAN | THROUGH | 
  THRU | TIMES | TO | TYPE | UNIT | UNTIL | UP | 
  UPON | USAGE | USE | USING | VALUE | VALUES | 
  VARYING | WHEN | WITH | WORDS | WORKING-STORAGE | 
  WRITE | ZERO | ZEROES | ZEROS |
INDEX
OF THE FORMAL DEFINITION
OF COBOL SYNTAX
### INDEX OF THE ECMA TC6 SYNTAX DEFINITION OF COBOL.

<table>
<thead>
<tr>
<th>META-VARIABLE</th>
<th>DEFN</th>
<th>USED</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;02-49-item-name-qualified&gt;</td>
<td>D214</td>
<td>D177 D223 D224 P120</td>
</tr>
<tr>
<td>&lt;02-49-name-declaration&gt;</td>
<td>D77</td>
<td>D49 D50 D53 D54</td>
</tr>
<tr>
<td>&lt;0&gt;</td>
<td>D168</td>
<td>D157 D38 D90</td>
</tr>
<tr>
<td>&lt;66-entry&gt;</td>
<td>D40</td>
<td>D86 D9 D8</td>
</tr>
<tr>
<td>&lt;77-description&gt;</td>
<td>D54</td>
<td>D179 D219 D228</td>
</tr>
<tr>
<td>&lt;77-descriptione&gt;</td>
<td>D93</td>
<td>D86 D87 D94</td>
</tr>
<tr>
<td>&lt;77-item-name&gt;</td>
<td>D179</td>
<td>D87 D38 D39 D47 D48</td>
</tr>
<tr>
<td>&lt;77-item-name-declaration&gt;</td>
<td>D41</td>
<td>D49 D50 D86 D87</td>
</tr>
<tr>
<td>&lt;77-redefines-clause&gt;</td>
<td></td>
<td>D90 D91</td>
</tr>
<tr>
<td>&lt;88-entry&gt;</td>
<td></td>
<td>D41</td>
</tr>
<tr>
<td>&lt;88-value-clause&gt;</td>
<td>D192</td>
<td>D157 D158</td>
</tr>
<tr>
<td>&lt;9&gt;</td>
<td>D167</td>
<td>D146 D147 D148 D149</td>
</tr>
<tr>
<td>&lt;9-seq&gt;</td>
<td>D158</td>
<td>D153</td>
</tr>
<tr>
<td>&lt;9-seq-or-b0,&gt;</td>
<td></td>
<td>D146 D147 D148 D149</td>
</tr>
<tr>
<td>&lt;;&gt;</td>
<td></td>
<td>T1 E72 E73 E74</td>
</tr>
<tr>
<td>&lt;*&gt;</td>
<td></td>
<td>E75 D14 D15 D16</td>
</tr>
<tr>
<td>&lt;*-seq&gt;</td>
<td></td>
<td>D27 D28 D31 D32</td>
</tr>
<tr>
<td>&lt;+&gt;</td>
<td></td>
<td>D35 D36 D37 D38</td>
</tr>
<tr>
<td>&lt;+*seq&gt;</td>
<td></td>
<td>D39 D40 D41 D48</td>
</tr>
<tr>
<td>&lt;+*seq-or-b0,&gt;</td>
<td></td>
<td>D50 D51 D52 D53</td>
</tr>
<tr>
<td>&lt;,&gt;</td>
<td></td>
<td>D54 D55 D57 D58</td>
</tr>
<tr>
<td>&lt;*&gt;</td>
<td></td>
<td>D60 D61 D62 D86</td>
</tr>
<tr>
<td>&lt;*-seq&gt;</td>
<td></td>
<td>D87 D88 D89 D90</td>
</tr>
<tr>
<td>&lt;+&gt;</td>
<td></td>
<td>D91 D92 D99 D105</td>
</tr>
<tr>
<td>&lt;+*seq&gt;</td>
<td></td>
<td>D107 D108 D109 D110</td>
</tr>
<tr>
<td>&lt;+*seq-or-b0,&gt;</td>
<td></td>
<td>D111 D182 P26 P33</td>
</tr>
<tr>
<td>&lt;,&gt;</td>
<td></td>
<td>P39 P94 P95 P117</td>
</tr>
<tr>
<td>&lt;*&gt;</td>
<td></td>
<td>P119 P120 P124 P141</td>
</tr>
<tr>
<td>&lt;*-seq&gt;</td>
<td></td>
<td>D174 D163</td>
</tr>
<tr>
<td>&lt;+&gt;</td>
<td></td>
<td>D148 D152</td>
</tr>
<tr>
<td>&lt;+*seq&gt;</td>
<td></td>
<td>D170 D159</td>
</tr>
<tr>
<td>&lt;+*seq-or-b0,&gt;</td>
<td></td>
<td>D146 D150 D154</td>
</tr>
<tr>
<td>&lt;,&gt;</td>
<td></td>
<td>D154 D150</td>
</tr>
<tr>
<td>&lt;*&gt;</td>
<td></td>
<td>T1 E12 E18 E19</td>
</tr>
<tr>
<td>&lt;*-seq&gt;</td>
<td></td>
<td>E22 E32 E48 E49</td>
</tr>
<tr>
<td>&lt;+&gt;</td>
<td></td>
<td>E50 E59 E60 E64</td>
</tr>
<tr>
<td>&lt;+*seq&gt;</td>
<td></td>
<td>E85 E86 E87 E88</td>
</tr>
<tr>
<td>&lt;+*seq-or-b0,&gt;</td>
<td></td>
<td>D127 D128 D131 D136</td>
</tr>
<tr>
<td>&lt;,&gt;</td>
<td></td>
<td>D137 D140 D183 D185</td>
</tr>
<tr>
<td>&lt;*&gt;</td>
<td></td>
<td>D192 D193 D236 P69</td>
</tr>
<tr>
<td>&lt;*-seq&gt;</td>
<td></td>
<td>P70 P72 P79 P92</td>
</tr>
<tr>
<td>&lt;+&gt;</td>
<td></td>
<td>P96 P97 P102 P103</td>
</tr>
<tr>
<td>&lt;+*seq&gt;</td>
<td></td>
<td>P104 P123 P125 P132</td>
</tr>
<tr>
<td>&lt;+*seq-or-b0,&gt;</td>
<td></td>
<td>P133 P134 P135 L5</td>
</tr>
<tr>
<td>META-VARIABLE</td>
<td>DEFN</td>
<td>USED</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td><code>&lt;-&gt;</code></td>
<td>D171</td>
<td>D160</td>
</tr>
<tr>
<td><code>&lt;--seq&gt;</code></td>
<td>D160</td>
<td>D146 D150 D155</td>
</tr>
<tr>
<td><code>&lt;--seq-or-bo,&gt;</code></td>
<td>D155</td>
<td>D150</td>
</tr>
<tr>
<td><code>&lt;.</code></td>
<td>G20</td>
<td>G47</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E48</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E49</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E51</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E70</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E72</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D36</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D37</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D39</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D41</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D47</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D48</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D49</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D51</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D52</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D53</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D86</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D88</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D89</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D90</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D91</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D101</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D102</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D105</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D107</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D108</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P29</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>L1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>L3</td>
</tr>
<tr>
<td><code>&lt;g&gt;</code></td>
<td>D172</td>
<td>D161</td>
</tr>
<tr>
<td><code>&lt;abbreviated-relation-condition&gt;</code></td>
<td>P59</td>
<td>P57</td>
</tr>
<tr>
<td><code>&lt;abbreviation&gt;</code></td>
<td>P60</td>
<td>P59</td>
</tr>
<tr>
<td><code>&lt;accept-statement&gt;</code></td>
<td>P68</td>
<td>P37</td>
</tr>
<tr>
<td><code>&lt;access-mode-random-clause&gt;</code></td>
<td>E67</td>
<td>E50</td>
</tr>
<tr>
<td><code>&lt;access-mode-sequential-clause&gt;</code></td>
<td>E66</td>
<td>E49</td>
</tr>
<tr>
<td><code>&lt;actual-key-clause&gt;</code></td>
<td>E69</td>
<td>E49</td>
</tr>
<tr>
<td><code>&lt;additional-data-character&gt;</code></td>
<td>N6</td>
<td>G16</td>
</tr>
<tr>
<td><code>&lt;advancing-phrase&gt;</code></td>
<td>P142</td>
<td>P140</td>
</tr>
<tr>
<td><code>&lt;alpha-word&gt;</code></td>
<td>G26</td>
<td>G28</td>
</tr>
<tr>
<td><code>&lt;alter-statement&gt;</code></td>
<td>P70</td>
<td>P35</td>
</tr>
<tr>
<td><code>&lt;alterable-go-to-paragraph-body&gt;</code></td>
<td>P19</td>
<td>P16</td>
</tr>
<tr>
<td><code>&lt;alterable-go-to-sentence&gt;</code></td>
<td>P24</td>
<td>P19</td>
</tr>
<tr>
<td><code>&lt;alterable-go-to-statement&gt;</code></td>
<td>P91</td>
<td>P24</td>
</tr>
<tr>
<td><code>&lt;alteration&gt;</code></td>
<td>P71</td>
<td>P70</td>
</tr>
<tr>
<td><code>&lt;alternate-area-clause&gt;</code></td>
<td>E63</td>
<td>E48</td>
</tr>
<tr>
<td><code>&lt;arithmetic-expression&gt;</code></td>
<td>P48</td>
<td>P52</td>
</tr>
<tr>
<td><code>&lt;arithmetic-expression-character&gt;</code></td>
<td>G7</td>
<td>G14</td>
</tr>
<tr>
<td><code>&lt;arithmetic-operator&gt;</code></td>
<td>G44</td>
<td>T4</td>
</tr>
<tr>
<td><code>&lt;assign-clause&gt;</code></td>
<td>E57</td>
<td>E48</td>
</tr>
<tr>
<td><code>&lt;assign-individual-units-clause&gt;</code></td>
<td>E59</td>
<td>E57</td>
</tr>
<tr>
<td><code>&lt;assign-type-clause&gt;</code></td>
<td>E58</td>
<td>E57</td>
</tr>
<tr>
<td><code>&lt;at-end-phrase&gt;</code></td>
<td>P42</td>
<td>P117</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P119</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P120</td>
</tr>
<tr>
<td>META-VARIABLE</td>
<td>DEFN</td>
<td>USED</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>&lt;author-paragraph&gt;</td>
<td>I8</td>
<td>I2</td>
</tr>
<tr>
<td>&lt;ax&gt;</td>
<td>D166</td>
<td>D145</td>
</tr>
<tr>
<td>&lt;b09&gt;</td>
<td>D165</td>
<td>D145</td>
</tr>
<tr>
<td>&lt;b0&gt;</td>
<td>D164</td>
<td>D150 D151 D153 D154</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D155 D156 D158 D159</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D160 D161 D162 D163</td>
</tr>
<tr>
<td>&lt;beginning&gt;</td>
<td>P138</td>
<td>P135</td>
</tr>
<tr>
<td>&lt;blank-when-zero-clause&gt;</td>
<td>D123</td>
<td>D60 D61 D62 D92</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D107 D108 D110 D111</td>
</tr>
<tr>
<td>&lt;block-clause&gt;</td>
<td>D124</td>
<td>D16</td>
</tr>
<tr>
<td>&lt;class-condition&gt;</td>
<td>P64</td>
<td>P58</td>
</tr>
<tr>
<td>&lt;clock-units-rerun-condition&gt;</td>
<td>E82</td>
<td>E80</td>
</tr>
<tr>
<td>&lt;close-statement&gt;</td>
<td>P72</td>
<td>P37</td>
</tr>
<tr>
<td>&lt;closure&gt;</td>
<td>P73</td>
<td>P72</td>
</tr>
<tr>
<td>&lt;cobol-character&gt;</td>
<td>G15</td>
<td>G16</td>
</tr>
<tr>
<td>&lt;cobol-program&gt;</td>
<td>C1</td>
<td></td>
</tr>
<tr>
<td>&lt;cobol-text&gt;</td>
<td>T9</td>
<td></td>
</tr>
<tr>
<td>&lt;code-clause&gt;</td>
<td>D125</td>
<td>D32</td>
</tr>
<tr>
<td>&lt;column-number-clause&gt;</td>
<td>D126</td>
<td>D107 D108 D110 D111</td>
</tr>
<tr>
<td>&lt;comment-entry&gt;</td>
<td>I14</td>
<td>I13</td>
</tr>
<tr>
<td>&lt;comment-paragraph-body&gt;</td>
<td>I13</td>
<td>I7 I8 I9 I10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I11 I12</td>
</tr>
<tr>
<td>&lt;comment-sentence&gt;</td>
<td>P28</td>
<td>P17</td>
</tr>
<tr>
<td>&lt;comment-string&gt;</td>
<td>G47</td>
<td>T5 I14 P28 P99</td>
</tr>
<tr>
<td>&lt;compound-figurative-constant&gt;</td>
<td>G38</td>
<td>G39</td>
</tr>
<tr>
<td>&lt;computer-character&gt;</td>
<td>G16</td>
<td>G29 G47 G48 E39</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P87</td>
</tr>
<tr>
<td>&lt;computer-name&gt;</td>
<td>N1</td>
<td>E9 E12</td>
</tr>
<tr>
<td>&lt;condition&gt;</td>
<td>P53</td>
<td>P57 P94 P112 P114</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P121</td>
</tr>
<tr>
<td>&lt;condition-factor&gt;</td>
<td>P55</td>
<td>P54</td>
</tr>
<tr>
<td>&lt;condition-name&gt;</td>
<td>D85</td>
<td>D223</td>
</tr>
<tr>
<td>&lt;condition-name-condition&gt;</td>
<td>P65</td>
<td>P58</td>
</tr>
<tr>
<td>&lt;condition-name-declaration&gt;</td>
<td>D84</td>
<td>D41 D85</td>
</tr>
<tr>
<td>&lt;condition-name-identifier&gt;</td>
<td>D235</td>
<td>P65</td>
</tr>
<tr>
<td>&lt;condition-name-qualified&gt;</td>
<td>D223</td>
<td>D235</td>
</tr>
<tr>
<td>&lt;condition-primary&gt;</td>
<td>P57</td>
<td>P55</td>
</tr>
<tr>
<td>&lt;condition-term&gt;</td>
<td>P54</td>
<td>P53</td>
</tr>
<tr>
<td>&lt;conditional-sentence&gt;</td>
<td>P26</td>
<td>P21</td>
</tr>
<tr>
<td>&lt;conditional-statement&gt;</td>
<td>P39</td>
<td>P26 P95</td>
</tr>
<tr>
<td>&lt;configuration-section&gt;</td>
<td>E3</td>
<td>E2</td>
</tr>
<tr>
<td>&lt;configuration-section-body&gt;</td>
<td>E4</td>
<td>E3</td>
</tr>
<tr>
<td>&lt;continuable-imp-verb-statement&gt;</td>
<td>P35</td>
<td>P34</td>
</tr>
<tr>
<td>&lt;continuable-imperative-statement&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;control-clause&gt;</td>
<td>D127</td>
<td>D32</td>
</tr>
<tr>
<td>&lt;control-variable&gt;</td>
<td>P116</td>
<td>P114 P120 P123</td>
</tr>
<tr>
<td>META-VARIABLE</td>
<td>DEFN</td>
<td>USED</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>&lt;copy-clause&gt;</td>
<td>L2</td>
<td>D14 D15 D27 D31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D35 D36 D37 D88</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D89 D99 L1</td>
</tr>
<tr>
<td>&lt;copy-entry&gt;</td>
<td>L1</td>
<td>E7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E10 E16 E45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E70</td>
</tr>
<tr>
<td>&lt;copy-sentence&gt;</td>
<td>L3</td>
<td>P5</td>
</tr>
<tr>
<td>&lt;copy-statement&gt;</td>
<td>L4</td>
<td>P6</td>
</tr>
<tr>
<td></td>
<td>L3</td>
<td>P11</td>
</tr>
<tr>
<td>&lt;cs&gt;</td>
<td>E41</td>
<td>D146 D148 D149 D151</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D172</td>
</tr>
<tr>
<td>&lt;cs-float&gt;</td>
<td>D151</td>
<td>D147</td>
</tr>
<tr>
<td>&lt;cs-float-pict&gt;</td>
<td>D147</td>
<td>D144</td>
</tr>
<tr>
<td>&lt;cs-seq&gt;</td>
<td>D161</td>
<td>D147 D151 D156</td>
</tr>
<tr>
<td>&lt;cs-seq-or-b0&gt;</td>
<td>D156</td>
<td>D151</td>
</tr>
<tr>
<td>&lt;currency-sign&gt;</td>
<td>G9</td>
<td>G14</td>
</tr>
<tr>
<td>&lt;currency-sign-clause&gt;</td>
<td>E37</td>
<td>E18</td>
</tr>
<tr>
<td>&lt;currency-sign-declaration&gt;</td>
<td>E38</td>
<td>E37 E40</td>
</tr>
<tr>
<td>&lt;currency-symbol&gt;</td>
<td>E40</td>
<td>E41</td>
</tr>
<tr>
<td></td>
<td>D1</td>
<td>C1</td>
</tr>
<tr>
<td>&lt;data-division&gt;</td>
<td>D2</td>
<td>D1</td>
</tr>
<tr>
<td>&lt;data-division-body&gt;</td>
<td>D234</td>
<td>D65 D69 L6</td>
</tr>
<tr>
<td>&lt;data-name-identifier&gt;</td>
<td>D128</td>
<td>D16 D28</td>
</tr>
<tr>
<td>&lt;data-records-clause&gt;</td>
<td>I7</td>
<td>I2</td>
</tr>
<tr>
<td>&lt;date-compiled-paragraph&gt;</td>
<td>I10</td>
<td>I2</td>
</tr>
<tr>
<td>&lt;date-written-paragraph&gt;</td>
<td>G32</td>
<td>G33</td>
</tr>
<tr>
<td>&lt;decimal-fraction&gt;</td>
<td>E43</td>
<td>G32 D175</td>
</tr>
<tr>
<td>&lt;decimal-point&gt;</td>
<td>E42</td>
<td>E18 E43 E44</td>
</tr>
<tr>
<td>&lt;decimal-point-clause&gt;</td>
<td>P3</td>
<td>P2</td>
</tr>
<tr>
<td>&lt;declarative-portion&gt;</td>
<td>P5</td>
<td>P3</td>
</tr>
<tr>
<td>&lt;declarative-section&gt;</td>
<td>P32</td>
<td>P5</td>
</tr>
<tr>
<td>&lt;declarative-sentence&gt;</td>
<td>P40</td>
<td>P32</td>
</tr>
<tr>
<td>&lt;declarative-statement&gt;</td>
<td>P93</td>
<td>P71 P90 P91 P92</td>
</tr>
<tr>
<td>&lt;destination&gt;</td>
<td>D157</td>
<td>D143</td>
</tr>
<tr>
<td>&lt;digit-seq&gt;</td>
<td>G4</td>
<td>G15 G23 G24 G31</td>
</tr>
<tr>
<td>&lt;digit&gt;</td>
<td></td>
<td>E15 E39 D46 P7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P87</td>
</tr>
<tr>
<td>&lt;digit-separator&gt;</td>
<td>E44</td>
<td>D164</td>
</tr>
<tr>
<td>&lt;display-operand&gt;</td>
<td>P80</td>
<td>P79</td>
</tr>
<tr>
<td>&lt;display-statement&gt;</td>
<td>P79</td>
<td>P37</td>
</tr>
<tr>
<td>&lt;elem-01-77-clauses&gt;</td>
<td>D60</td>
<td>D39 D59 D86</td>
</tr>
<tr>
<td>&lt;elem-01-77-red-clauses&gt;</td>
<td>D92</td>
<td>D87 D91</td>
</tr>
<tr>
<td>&lt;elem-02-49-item-name&gt;</td>
<td>D79</td>
<td>D213</td>
</tr>
<tr>
<td>&lt;elem-02-49-item-name-</td>
<td>D78</td>
<td>D77 D79</td>
</tr>
<tr>
<td>declaration&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;elem-02-49-item-name-</td>
<td>D226</td>
<td>D228 P129</td>
</tr>
<tr>
<td>identifier&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;elem-02-49-item-name-</td>
<td>D213</td>
<td>D214 D219 D226</td>
</tr>
<tr>
<td>qualified&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;elem-66-item-name&gt;</td>
<td>D83</td>
<td>D216</td>
</tr>
<tr>
<td>META-VARIABLE</td>
<td>DEFN</td>
<td>USED</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>&lt;elem-66-item-name-declaration&gt;</td>
<td>D82</td>
<td>D40 D83</td>
</tr>
<tr>
<td>&lt;elem-66-item-name-qualified&gt;</td>
<td>D216</td>
<td>D219 D228</td>
</tr>
<tr>
<td>&lt;elem-clauses&gt;</td>
<td>D59</td>
<td>D49</td>
</tr>
<tr>
<td>&lt;elem-data-name-identifier&gt;</td>
<td>D233</td>
<td>D234</td>
</tr>
<tr>
<td>&lt;elem-item-name-identifier&gt;</td>
<td>D228</td>
<td>D229 D231 D233 P44 P46 P67 P81 P92 P115 P116</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D135 D136 D180 D193 D220 D236</td>
</tr>
<tr>
<td>&lt;elem-item-name-qualified&gt;</td>
<td>D219</td>
<td></td>
</tr>
<tr>
<td>&lt;elem-ms-record-name&gt;</td>
<td>D70</td>
<td>D194 D200</td>
</tr>
<tr>
<td></td>
<td>D69</td>
<td>D36 D70</td>
</tr>
<tr>
<td>&lt;elem-ms-record-name-qualified&gt;</td>
<td>D200</td>
<td>D205 D208</td>
</tr>
<tr>
<td>&lt;elem-non-ms-record-name&gt;</td>
<td>D66</td>
<td>D194 D198</td>
</tr>
<tr>
<td>&lt;elem-non-ms-record-name-qualified&gt;</td>
<td>D65</td>
<td>D35 D66</td>
</tr>
<tr>
<td>&lt;elem-non-sum-field-name-declaration&gt;</td>
<td>D198</td>
<td>D204 D208</td>
</tr>
<tr>
<td>&lt;elem-non-sum-field-name-qualified&gt;</td>
<td>D120</td>
<td>D108</td>
</tr>
<tr>
<td>&lt;elem-non-sum-group-clauses&gt;</td>
<td>D111</td>
<td>D102</td>
</tr>
<tr>
<td>&lt;elem-non-sum-report-group-name&gt;</td>
<td>D115</td>
<td>D210</td>
</tr>
<tr>
<td>&lt;elem-non-sum-report-group-name-declaration&gt;</td>
<td>D114</td>
<td>D99 D115</td>
</tr>
<tr>
<td>&lt;elem-non-sum-report-group-name-qualified&gt;</td>
<td>D210</td>
<td>D211</td>
</tr>
<tr>
<td>&lt;elem-non-sum-report-group-spec&gt;</td>
<td>D102</td>
<td>D99</td>
</tr>
<tr>
<td>&lt;elem-non-sum-spec&gt;</td>
<td>D108</td>
<td>D106</td>
</tr>
<tr>
<td>&lt;elem-record-name-qualified&gt;</td>
<td>D208</td>
<td>D219 D228</td>
</tr>
<tr>
<td>&lt;elem-record-spec&gt;</td>
<td>D39</td>
<td>D35 D36 D37 D88 D88</td>
</tr>
<tr>
<td>&lt;elem-red-clauses&gt;</td>
<td>D61</td>
<td>D59</td>
</tr>
<tr>
<td>&lt;elem-renames-clause&gt;</td>
<td>D180</td>
<td>D40</td>
</tr>
<tr>
<td>&lt;elem-report-spec&gt;</td>
<td>D106</td>
<td>D104</td>
</tr>
<tr>
<td>&lt;elem-sort-record-name&gt;</td>
<td>D74</td>
<td>D194 D202</td>
</tr>
<tr>
<td>&lt;elem-sort-record-name-declaration&gt;</td>
<td>D73</td>
<td>D37 D74</td>
</tr>
<tr>
<td>&lt;elem-sort-record-name-qualified&gt;</td>
<td>D202</td>
<td>D206 D208</td>
</tr>
<tr>
<td>&lt;elem-spec&gt;</td>
<td>D49</td>
<td>D43</td>
</tr>
<tr>
<td>&lt;elem-ws-record-name&gt;</td>
<td>D98</td>
<td>D195 D208</td>
</tr>
<tr>
<td>&lt;elem-ws-record-name-declaration&gt;</td>
<td>D97</td>
<td>D88 D89 D98</td>
</tr>
<tr>
<td>&lt;empty&gt;</td>
<td>G1</td>
<td>T2 E15 P7</td>
</tr>
<tr>
<td>&lt;end-of-reel-rerun-condition&gt;</td>
<td>E79</td>
<td>E78</td>
</tr>
<tr>
<td>&lt;ending&gt;</td>
<td>P139</td>
<td>P135</td>
</tr>
<tr>
<td>&lt;enter-cobol-sentence&gt;</td>
<td>P31</td>
<td>P22</td>
</tr>
<tr>
<td>&lt;enter-cobol-statement&gt;</td>
<td>P84</td>
<td>P31</td>
</tr>
<tr>
<td>META-VARIABLE</td>
<td>DEFN</td>
<td>USED</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>enter-other-language-sentence</td>
<td>P30</td>
<td>P22</td>
</tr>
<tr>
<td>enter-other-language-statement</td>
<td>P83</td>
<td>P30</td>
</tr>
<tr>
<td>enter-routine-sentence</td>
<td>P29</td>
<td>P21</td>
</tr>
<tr>
<td>enter-routine-statement</td>
<td>P82</td>
<td>P29</td>
</tr>
<tr>
<td>environment-division</td>
<td>E1</td>
<td>C1</td>
</tr>
<tr>
<td>environment-division-body</td>
<td>E2</td>
<td>E1</td>
</tr>
<tr>
<td>examine-statement</td>
<td>P86</td>
<td>P35</td>
</tr>
<tr>
<td>exit-paragraph-body</td>
<td>P18</td>
<td>P16</td>
</tr>
<tr>
<td>exit-sentence</td>
<td>P23</td>
<td>P18</td>
</tr>
<tr>
<td>exit-statement</td>
<td>P88</td>
<td>P23</td>
</tr>
<tr>
<td>factor</td>
<td>P50</td>
<td>P49</td>
</tr>
<tr>
<td>fd-clauses</td>
<td>D16</td>
<td>D14</td>
</tr>
<tr>
<td>figurative-constant</td>
<td>G39</td>
<td>G41</td>
</tr>
<tr>
<td>file</td>
<td>P137</td>
<td>P135</td>
</tr>
<tr>
<td>file-control-entry</td>
<td>E47</td>
<td>E46</td>
</tr>
<tr>
<td>file-control-entry-for-non-ms-file</td>
<td>E48</td>
<td>E47</td>
</tr>
<tr>
<td>file-control-entry-for-random-ms-file</td>
<td>E50</td>
<td>E47</td>
</tr>
<tr>
<td>file-control-entry-for-sequential-ms-file</td>
<td>E49</td>
<td>E47</td>
</tr>
<tr>
<td>file-control-entry-for-sort-file</td>
<td>E51</td>
<td>E47</td>
</tr>
<tr>
<td>file-control-paragraph</td>
<td>E45</td>
<td>E6</td>
</tr>
<tr>
<td>file-control-paragraph-body</td>
<td>E46</td>
<td>E45</td>
</tr>
<tr>
<td>file-limit</td>
<td>E65</td>
<td>E64</td>
</tr>
<tr>
<td>file-limit-clause</td>
<td>E64</td>
<td>E49</td>
</tr>
<tr>
<td>file-name</td>
<td>D26</td>
<td>E85</td>
</tr>
<tr>
<td>file-section</td>
<td>D3</td>
<td>D2</td>
</tr>
<tr>
<td>file-section-body</td>
<td>D4</td>
<td>D3</td>
</tr>
<tr>
<td>file-specification</td>
<td>D5</td>
<td>D4</td>
</tr>
<tr>
<td>fixed-insert-pict</td>
<td>D149</td>
<td>D144</td>
</tr>
<tr>
<td>fixed-occurs-clause</td>
<td>D134</td>
<td>D57</td>
</tr>
<tr>
<td>generalized-character-string</td>
<td>T6</td>
<td>T7</td>
</tr>
<tr>
<td>generalized-character-string-type-one</td>
<td>T4</td>
<td>T6</td>
</tr>
<tr>
<td>generalized-character-string-type-two</td>
<td>T5</td>
<td>T6</td>
</tr>
<tr>
<td>generalized-separator</td>
<td>T3</td>
<td>T7</td>
</tr>
<tr>
<td>generate-statement</td>
<td>P89</td>
<td>P35</td>
</tr>
<tr>
<td>go-to-depending-statement</td>
<td>P92</td>
<td>P35</td>
</tr>
<tr>
<td>group-indicate-clause</td>
<td>D129</td>
<td>D108</td>
</tr>
<tr>
<td>i-o-control-entry</td>
<td>E72</td>
<td>E71</td>
</tr>
<tr>
<td>i-o-control-paragraph</td>
<td>E70</td>
<td>E6</td>
</tr>
<tr>
<td>i-o-control-paragraph-body</td>
<td>E71</td>
<td>E70</td>
</tr>
<tr>
<td>i-o-file</td>
<td>P107</td>
<td>P104</td>
</tr>
<tr>
<td>META-VARIABLE</td>
<td>DEFN</td>
<td>USED</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>&lt;i-o-option&gt;</td>
<td>P104</td>
<td></td>
</tr>
<tr>
<td>&lt;identification-division&gt;</td>
<td>I1</td>
<td>C1</td>
</tr>
<tr>
<td>&lt;identification-division-body&gt;</td>
<td>T2</td>
<td>I1</td>
</tr>
<tr>
<td>&lt;if-statement&gt;</td>
<td>P94</td>
<td>P39</td>
</tr>
<tr>
<td>&lt;imp-add-statement&gt;</td>
<td>P69</td>
<td>P36</td>
</tr>
<tr>
<td>&lt;imp-arithmetic-statement&gt;</td>
<td>P36</td>
<td>P35</td>
</tr>
<tr>
<td>&lt;imp-compute-statement&gt;</td>
<td>P78</td>
<td>P36</td>
</tr>
<tr>
<td>&lt;imp-divide-statement&gt;</td>
<td>P81</td>
<td>P36</td>
</tr>
<tr>
<td>&lt;imp-multiply-statement&gt;</td>
<td>P98</td>
<td>P36</td>
</tr>
<tr>
<td>&lt;imp-subtract-statement&gt;</td>
<td>P132</td>
<td>P36</td>
</tr>
<tr>
<td>&lt;imperative-i-o-statement&gt;</td>
<td>P37</td>
<td>P35</td>
</tr>
<tr>
<td>&lt;imperative-write-statement&gt;</td>
<td>P140</td>
<td>P37</td>
</tr>
<tr>
<td>&lt;implementor-name-for-code-for-report-groups&gt;</td>
<td>N8</td>
<td>E21</td>
</tr>
<tr>
<td>&lt;implementor-name-for-individual&gt;</td>
<td>N3</td>
<td>E21</td>
</tr>
<tr>
<td>-i-o-unit&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;implementor-name-for-individual&gt;</td>
<td>N5</td>
<td>E22</td>
</tr>
<tr>
<td>-switch&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;implementor-name-for-paper-advance&gt;</td>
<td>N7</td>
<td>E21</td>
</tr>
<tr>
<td>&lt;implementor-name-for-rerun-medium&gt;</td>
<td>N4</td>
<td>E77</td>
</tr>
<tr>
<td>&lt;implementor-name-for-type-of-i-o&gt;</td>
<td>N2</td>
<td>E21</td>
</tr>
<tr>
<td>-unit&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;in-of&gt;</td>
<td>D196</td>
<td></td>
</tr>
</tbody>
</table>

<p>| &lt;index-elem-spec&gt;                                 | D53  | D197 | D198 | D199 | D200 |
| &lt;index-name&gt;                                      | D139 | D201 | D202 | D203 | D209 |
| &lt;index-name-declaration&gt;                          | D138 | D210 | D212 | D213 | D215 |
| &lt;index-name-qualified&gt;                            | D224 | D216 | D217 | D222 | D223 |
| &lt;index-non-elem-spec&gt;                             | D51  | D123 |
| &lt;index-option&gt;                                    | D137 | D134 | D135 |
| &lt;initiate-statement&gt;                              | P96  | P102 |
| &lt;input-file&gt;                                      | P105 |      |
| &lt;input-option&gt;                                    | P102 |      |
| &lt;input-output-section&gt;                            | E5   |      |
| &lt;input-output-section-body&gt;                       | E6   | E2   |
| &lt;installation-paragraph&gt;                          | I9   | T2   |
| &lt;integer&gt;                                         | G31  | G32  | G33  | G36  | E13  |
| &lt;integer-records-rerun-condition&gt;                 | E81  | E58  | E63  | E88  | D135 |
|                                                   | E237 | P115 |
|                                                   |      | E80  |</p>
<table>
<thead>
<tr>
<th>META-VARIABLE</th>
<th>DEFN</th>
<th>USED</th>
</tr>
</thead>
<tbody>
<tr>
<td>integral-operand</td>
<td>P115</td>
<td>P111 P123 P142</td>
</tr>
<tr>
<td>invalid-key-phrase</td>
<td>P43</td>
<td>P117 P141</td>
</tr>
<tr>
<td>item-name-identifier</td>
<td>D229</td>
<td>D127 D183 D187 D230</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P62 P64 P68 P80</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P86 P97 P117 P118</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P119 P140 P141</td>
</tr>
<tr>
<td>item-name-qualified</td>
<td>D220</td>
<td>D181</td>
</tr>
<tr>
<td>justified-clause</td>
<td>D130</td>
<td>D60 D61 D62 D92</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D107 D110 D111</td>
</tr>
<tr>
<td>key-clause</td>
<td>P125</td>
<td>P124</td>
</tr>
<tr>
<td>key-option</td>
<td>D136</td>
<td>D134 D135</td>
</tr>
<tr>
<td>label-records-clause</td>
<td>D131</td>
<td>D16</td>
</tr>
<tr>
<td>letter</td>
<td>G5</td>
<td>G15 G23 G24 G26</td>
</tr>
<tr>
<td>level-number</td>
<td>D46</td>
<td>D45</td>
</tr>
<tr>
<td>library-call</td>
<td>L5</td>
<td>L2 L4</td>
</tr>
<tr>
<td>library-name</td>
<td>L7</td>
<td>L5</td>
</tr>
<tr>
<td>line-number-clause</td>
<td>D132</td>
<td>D105 D107 D108 D109</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D110 D111</td>
</tr>
<tr>
<td>literal</td>
<td>G43</td>
<td>E65 E91 D193 P97</td>
</tr>
<tr>
<td>literal-for-display-stop</td>
<td>P47</td>
<td>P80 P130</td>
</tr>
<tr>
<td>literal-string</td>
<td>G29</td>
<td>G30</td>
</tr>
<tr>
<td>lock</td>
<td>P77</td>
<td>P73</td>
</tr>
<tr>
<td>memory-size-clause</td>
<td>E13</td>
<td>E12</td>
</tr>
<tr>
<td>mnemonic-name-declaration-for-code-for-report-groups</td>
<td>E26</td>
<td>E21 E31</td>
</tr>
<tr>
<td>mnemonic-name-declaration-for-individual-io-unit</td>
<td>E23</td>
<td>E21 E28</td>
</tr>
<tr>
<td>mnemonic-name-declaration-for-individual-switch</td>
<td>E27</td>
<td>E22</td>
</tr>
<tr>
<td>mnemonic-name-declaration-for-paper-advance</td>
<td>E25</td>
<td>E21 E30</td>
</tr>
<tr>
<td>mnemonic-name-declaration-for-type-of-io-unit</td>
<td>E24</td>
<td>E21 E29</td>
</tr>
<tr>
<td>mnemonic-name-for-code-for-report-groups</td>
<td>E31</td>
<td>D125</td>
</tr>
<tr>
<td>mnemonic-name-for-individual-io-unit</td>
<td>E28</td>
<td>P68 P79</td>
</tr>
<tr>
<td>mnemonic-name-for-paper-advance</td>
<td>E30</td>
<td>P142</td>
</tr>
<tr>
<td>mnemonic-name-for-type-of-io-unit</td>
<td>E29</td>
<td>P68 P79</td>
</tr>
<tr>
<td>move-statement</td>
<td>P97</td>
<td>P35</td>
</tr>
<tr>
<td>ms-file-description</td>
<td>D15</td>
<td>D5</td>
</tr>
<tr>
<td>ms-file-name</td>
<td>D24</td>
<td>D25 D199 D200 P105</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P106 P107</td>
</tr>
<tr>
<td>ms-record-description</td>
<td>D36</td>
<td>D5</td>
</tr>
<tr>
<td>META-VARIABLE</td>
<td>DEFN</td>
<td>USED</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td><code>&lt;ms-record-name-qualified&gt;</code></td>
<td>D205</td>
<td>P141</td>
</tr>
<tr>
<td><code>&lt;multiple-file-clause&gt;</code></td>
<td>E88</td>
<td>E75</td>
</tr>
<tr>
<td><code>&lt;multiple-file-clauses&gt;</code></td>
<td>E75</td>
<td>E72</td>
</tr>
<tr>
<td><code>&lt;multiple-reel-clause&gt;</code></td>
<td>E61</td>
<td>E48</td>
</tr>
<tr>
<td><code>&lt;multiple-unit-clause&gt;</code></td>
<td>E62</td>
<td>E49</td>
</tr>
<tr>
<td><code>&lt;next-group-clause&gt;</code></td>
<td>D133</td>
<td>D109 D110 D111</td>
</tr>
<tr>
<td><code>&lt;no-rewind&gt;</code></td>
<td>P76</td>
<td>P73 P105 P106</td>
</tr>
<tr>
<td><code>&lt;non-declarative-portion&gt;</code></td>
<td>P4</td>
<td>P2</td>
</tr>
<tr>
<td><code>&lt;non-declarative-section&gt;</code></td>
<td>P6</td>
<td>P4</td>
</tr>
<tr>
<td><code>&lt;non-elem-01-clauses&gt;</code></td>
<td>D56</td>
<td>D38 D55</td>
</tr>
<tr>
<td><code>&lt;non-elem-02-48-item-name&gt;</code></td>
<td>D76</td>
<td>D212</td>
</tr>
<tr>
<td><code>&lt;non-elem-02-48-item-name-declaration&gt;</code></td>
<td>D75</td>
<td>D47 D48 D51 D52 D76</td>
</tr>
<tr>
<td><code>&lt;non-elem-02-48-item-name-identifier&gt;</code></td>
<td>D225</td>
<td>D227 P129</td>
</tr>
<tr>
<td><code>&lt;non-elem-02-48-item-name-qualified&gt;</code></td>
<td>D212</td>
<td>D213 D214 D218 D225</td>
</tr>
<tr>
<td><code>&lt;non-elem-66-item-name&gt;</code></td>
<td>D81</td>
<td>D215</td>
</tr>
<tr>
<td><code>&lt;non-elem-66-item-name-declaration&gt;</code></td>
<td>D80</td>
<td>D40 D81</td>
</tr>
<tr>
<td><code>&lt;non-elem-66-item-name-qualified&gt;</code></td>
<td>D215</td>
<td>D218 D227</td>
</tr>
<tr>
<td><code>&lt;non-elem-clauses&gt;</code></td>
<td>D55</td>
<td>D47</td>
</tr>
<tr>
<td><code>&lt;non-elem-data-name-identifier&gt;</code></td>
<td>D232</td>
<td>D234 P69</td>
</tr>
<tr>
<td><code>&lt;non-elem-field-name&gt;</code></td>
<td>D119</td>
<td>D217</td>
</tr>
<tr>
<td><code>&lt;non-elem-field-name-declaration&gt;</code></td>
<td>D118</td>
<td>D105 D119</td>
</tr>
<tr>
<td><code>&lt;non-elem-group-clauses&gt;</code></td>
<td>D109</td>
<td>D100</td>
</tr>
<tr>
<td><code>&lt;non-elem-item-name-identifier&gt;</code></td>
<td>D227</td>
<td>D229 D232 P97 P132</td>
</tr>
<tr>
<td><code>&lt;non-elem-item-name-qualified&gt;</code></td>
<td>D218</td>
<td>D220</td>
</tr>
<tr>
<td><code>&lt;non-elem-ms-record-name&gt;</code></td>
<td>D68</td>
<td>D194 D199</td>
</tr>
<tr>
<td><code>&lt;non-elem-ms-record-name-declaration&gt;</code></td>
<td>D67</td>
<td>D36 D68</td>
</tr>
<tr>
<td><code>&lt;non-elem-ms-record-name-qualified&gt;</code></td>
<td>D199</td>
<td>D205 D207</td>
</tr>
<tr>
<td><code>&lt;non-elem-non-ms-record-name&gt;</code></td>
<td>D64</td>
<td>D194 D197</td>
</tr>
<tr>
<td><code>&lt;non-elem-non-ms-record-name-declaration&gt;</code></td>
<td>D63</td>
<td>D35 D64</td>
</tr>
<tr>
<td><code>&lt;non-elem-non-ms-record-name-qualified&gt;</code></td>
<td>D197</td>
<td>D204 D207</td>
</tr>
<tr>
<td><code>&lt;non-elem-record-name-qualified&gt;</code></td>
<td>D207</td>
<td>D212 D215 D216 D218 D227</td>
</tr>
<tr>
<td><code>&lt;non-elem-record-spec&gt;</code></td>
<td>D38</td>
<td>D35 D36 D37 D38 D48 D55</td>
</tr>
<tr>
<td><code>&lt;non-elem-red-clauses&gt;</code></td>
<td>D57</td>
<td>D36 D55</td>
</tr>
<tr>
<td><code>&lt;non-elem-replaces-clause&gt;</code></td>
<td>D181</td>
<td>D40</td>
</tr>
<tr>
<td><code>&lt;non-elem-report-group-name&gt;</code></td>
<td>D113</td>
<td>D209</td>
</tr>
<tr>
<td><code>&lt;non-elem-report-group-name-declaration&gt;</code></td>
<td>D112</td>
<td>D99 D113</td>
</tr>
<tr>
<td>META-VARIABLE</td>
<td>DEFN</td>
<td>USED</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>⟨non-elem-report-group-name-qualified⟩</td>
<td>D209</td>
<td>D211</td>
</tr>
<tr>
<td>⟨non-elem-report-group-spec⟩</td>
<td>D100</td>
<td>D99</td>
</tr>
<tr>
<td>⟨non-elem-report-spec⟩</td>
<td>D105</td>
<td>D104</td>
</tr>
<tr>
<td>⟨non-elem-sort-record-name⟩</td>
<td>D72</td>
<td>D194</td>
</tr>
<tr>
<td>⟨non-elem-sort-record-name-declaration⟩</td>
<td>D71</td>
<td>D37</td>
</tr>
<tr>
<td>⟨non-elem-sort-record-name-qualified⟩</td>
<td>D201</td>
<td>D206</td>
</tr>
<tr>
<td>⟨non-elem-spec⟩</td>
<td>D47</td>
<td>D43</td>
</tr>
<tr>
<td>⟨non-elem ws-record-name⟩</td>
<td>D96</td>
<td>D195</td>
</tr>
<tr>
<td>⟨non-elem ws-record-name-declaration⟩</td>
<td>D95</td>
<td>D88</td>
</tr>
<tr>
<td>⟨non-ms-file-description⟩</td>
<td>D14</td>
<td>D5</td>
</tr>
<tr>
<td>⟨non-ms-file-name⟩</td>
<td>D18</td>
<td>E52</td>
</tr>
<tr>
<td>⟨non-ms-file-name-declaration⟩</td>
<td>D17</td>
<td>D23</td>
</tr>
<tr>
<td>⟨non-ms-record-description⟩</td>
<td>D35</td>
<td>D25</td>
</tr>
<tr>
<td>⟨non-ms-record-name-qualified⟩</td>
<td>D204</td>
<td>D197</td>
</tr>
<tr>
<td>⟨non-reserved-alpha-word⟩</td>
<td>G28</td>
<td>D18</td>
</tr>
<tr>
<td>⟨non-reserved-word⟩</td>
<td>G27</td>
<td>D17</td>
</tr>
<tr>
<td>⟨non-sort-file-name⟩</td>
<td>D25</td>
<td>D12</td>
</tr>
<tr>
<td>⟨non-switch-special-names-clause⟩</td>
<td>G2</td>
<td>G4</td>
</tr>
<tr>
<td>⟨non-null-record-name⟩</td>
<td>G41</td>
<td>G43</td>
</tr>
<tr>
<td>⟨nonzero-digit⟩</td>
<td>D145</td>
<td>D142</td>
</tr>
<tr>
<td>⟨numeric-literal⟩</td>
<td></td>
<td></td>
</tr>
<tr>
<td>⟨nonnumeric-picture⟩</td>
<td></td>
<td></td>
</tr>
<tr>
<td>META-VARIABLE</td>
<td>DEFN</td>
<td>USED</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>⟨not⟩</td>
<td>P56</td>
<td>P55</td>
</tr>
<tr>
<td>⟨note-paragraph-body⟩</td>
<td>P17</td>
<td>P16</td>
</tr>
<tr>
<td>⟨note-sentence⟩</td>
<td>P27</td>
<td>P17</td>
</tr>
<tr>
<td>⟨note-statement⟩</td>
<td>P99</td>
<td>P20</td>
</tr>
<tr>
<td>⟨numeric-edited-picture⟩</td>
<td>D144</td>
<td>D142</td>
</tr>
<tr>
<td>⟨numeric-literal⟩</td>
<td>G42</td>
<td>G43</td>
</tr>
<tr>
<td>⟨numeric-operand⟩</td>
<td>G46</td>
<td>D192</td>
</tr>
<tr>
<td>⟨numeric-picture⟩</td>
<td>D143</td>
<td>D142</td>
</tr>
<tr>
<td>⟨object-computer-entry⟩</td>
<td>E12</td>
<td>E11</td>
</tr>
<tr>
<td>⟨object-computer-paragraph⟩</td>
<td>E10</td>
<td>E4</td>
</tr>
<tr>
<td>⟨object-computer-paragraph-body⟩</td>
<td>E11</td>
<td>E10</td>
</tr>
<tr>
<td>⟨off-status⟩</td>
<td>E34</td>
<td>E32</td>
</tr>
<tr>
<td>⟨on-status⟩</td>
<td>E33</td>
<td>E32</td>
</tr>
<tr>
<td>⟨one-character-literal⟩</td>
<td>P87</td>
<td>P86</td>
</tr>
<tr>
<td>⟨open-options⟩</td>
<td>P101</td>
<td>P100</td>
</tr>
<tr>
<td>⟨open-statement⟩</td>
<td>P100</td>
<td>P37</td>
</tr>
<tr>
<td>⟨optional-phrase⟩</td>
<td>E56</td>
<td>E52</td>
</tr>
<tr>
<td>⟨other-language-block⟩</td>
<td>P22</td>
<td>P20</td>
</tr>
<tr>
<td>⟨other-language-name⟩</td>
<td>N9</td>
<td>P82</td>
</tr>
<tr>
<td>⟨other-language-string⟩</td>
<td>G48</td>
<td>T5</td>
</tr>
<tr>
<td>⟨other-language-string-terminator⟩</td>
<td>T2</td>
<td>T3</td>
</tr>
<tr>
<td>⟨output-file⟩</td>
<td>P106</td>
<td>P103</td>
</tr>
<tr>
<td>⟨output-option⟩</td>
<td>P103</td>
<td>P101</td>
</tr>
<tr>
<td>⟨p⟩</td>
<td>D169</td>
<td>D143</td>
</tr>
<tr>
<td>⟨page-limit-clause⟩</td>
<td>D140</td>
<td>D32</td>
</tr>
<tr>
<td>⟨paragraph⟩</td>
<td>P11</td>
<td>P2</td>
</tr>
<tr>
<td>⟨paragraph-body⟩</td>
<td>P16</td>
<td>P11</td>
</tr>
<tr>
<td>⟨paragraph-name⟩</td>
<td>P13</td>
<td>P14</td>
</tr>
<tr>
<td>⟨paragraph-name-declaration⟩</td>
<td>P12</td>
<td>P11</td>
</tr>
<tr>
<td>⟨paragraph-name-qualified⟩</td>
<td>P14</td>
<td>P13</td>
</tr>
<tr>
<td>⟨parenthesis⟩</td>
<td>G12</td>
<td>G13</td>
</tr>
<tr>
<td>⟨perform-statement⟩</td>
<td>P109</td>
<td>P35</td>
</tr>
<tr>
<td>⟨picture⟩</td>
<td>D142</td>
<td>D146</td>
</tr>
<tr>
<td>⟨picture-character-string⟩</td>
<td>G46</td>
<td>T5</td>
</tr>
<tr>
<td>⟨picture-clause⟩</td>
<td>D141</td>
<td>D141</td>
</tr>
<tr>
<td>⟨point⟩</td>
<td>D175</td>
<td></td>
</tr>
<tr>
<td>⟨positive-integer⟩</td>
<td>D36</td>
<td></td>
</tr>
</tbody>
</table>
META-VARIABLE  DEFN  USED

<possible-character-for-currency> E39  E38  E40
  -sign>
  <primary>     P51  P50
  <priority-number> P7  P6
  <priority-number-limit> E15  E14
  <procedure-division> P1  C1
  <procedure-division-body> P2  P1
  <procedure-name> P15  P93  P110  L6
  <processing-mode-sequential-clause> E68  E49  E50
  <program-id-entry> I5  I4
  <program-id-paragraph> I3  I2
  <program-id-paragraph-body> I4  I3
  <program-name-declaration> I6  I5
  <proper-nonnemonic-literal> G30  G38  G41  T4  P47
  <proper-nemonic-literal> G35  G42  T4  P47
  <proper-punctuation-character> G13  G14
  <proper-relational-operator> G45  T4
  <quotation-mark> G11  G13  P87
  <random-ms-file-name> D22  E54  D24  P73  P117
  <random-ms-file-name-declaration> D21  D15  D22
  <range> P110  P109
  <rd-clauses> D32  D31
  <read-statement> P117  P39
  <record-contains-clause> D176  D16  D28
  <redefines-clause> D177  D48  D50  D52  D54
  <redefines-record-clause> D176  D90  D91
  <redefining-77-description> D87  D9
  <redefining-elem-record-spec> D91  D89
  <redefining-elem-spec> D50  D44
  <redefining-index-elem-spec> D54  D44
  <redefining-index-non-elem-spec> D52  D44
  <redefining-non-elem-record-spec> D90  D89
  <redefining-non-elem-spec> D48  D44
  <redefining-sub-spec> D44  D42
  <redefining-ws-record-description> D89  D10
  <reel> P74  P73  P135
  <regular-imperative-sentence> P25  P21
  <regular-imperative-statement> P33  P25  P41  P42  P43
  <regular-paragraph-body> P20  P16
  <regular-sentence> P21  P20
  <relation-character> G8  G14
<table>
<thead>
<tr>
<th>META-VARIABLE</th>
<th>DEFN</th>
<th>USED</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;relation-condition&gt;</td>
<td>P61</td>
<td>P58</td>
</tr>
<tr>
<td>&lt;relation-operand&gt;</td>
<td>P62</td>
<td>P60</td>
</tr>
<tr>
<td>&lt;relational-operator&gt;</td>
<td>P63</td>
<td>P60</td>
</tr>
<tr>
<td>&lt;relative-index-name-qualified&gt;</td>
<td>D237</td>
<td>D236</td>
</tr>
<tr>
<td>&lt;release-statement&gt;</td>
<td>P118</td>
<td>P35</td>
</tr>
<tr>
<td>&lt;remarks-paragraph&gt;</td>
<td>I12</td>
<td>I2</td>
</tr>
<tr>
<td>&lt;replacement&gt;</td>
<td>L6</td>
<td>L5</td>
</tr>
<tr>
<td>&lt;report-clause&gt;</td>
<td>D182</td>
<td>D16</td>
</tr>
<tr>
<td>&lt;report-description&gt;</td>
<td>D31</td>
<td>D13</td>
</tr>
<tr>
<td>&lt;report-group-description&gt;</td>
<td>D99</td>
<td>D13</td>
</tr>
<tr>
<td>&lt;report-group-name-qualified&gt;</td>
<td>D211</td>
<td>D185</td>
</tr>
<tr>
<td></td>
<td>D34</td>
<td>D182</td>
</tr>
<tr>
<td></td>
<td>D222</td>
<td>P89</td>
</tr>
<tr>
<td>&lt;report-name-declaration&gt;</td>
<td>D33</td>
<td>D31</td>
</tr>
<tr>
<td>&lt;report-section&gt;</td>
<td>D11</td>
<td>D2</td>
</tr>
<tr>
<td>&lt;report-section-body&gt;</td>
<td>D12</td>
<td>D11</td>
</tr>
<tr>
<td>&lt;report-specification&gt;</td>
<td>D13</td>
<td>D12</td>
</tr>
<tr>
<td>&lt;rerun-clause&gt;</td>
<td>E76</td>
<td>E73</td>
</tr>
<tr>
<td>&lt;rerun-clauses&gt;</td>
<td>E73</td>
<td>E72</td>
</tr>
<tr>
<td>&lt;rerun-condition-1&gt;</td>
<td>E78</td>
<td>E76</td>
</tr>
<tr>
<td>&lt;rerun-condition-2&gt;</td>
<td>E80</td>
<td>E76</td>
</tr>
<tr>
<td>&lt;rerun-medium&gt;</td>
<td>E77</td>
<td>E76</td>
</tr>
<tr>
<td>&lt;reserved-word&gt;</td>
<td>R1</td>
<td>G27</td>
</tr>
<tr>
<td>&lt;reset-clause&gt;</td>
<td>D183</td>
<td>D107</td>
</tr>
<tr>
<td>&lt;result&gt;</td>
<td>P44</td>
<td>P69</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;return-statement&gt;</td>
<td>P119</td>
<td>P39</td>
</tr>
<tr>
<td>&lt;reversed&gt;</td>
<td>P108</td>
<td>P105</td>
</tr>
<tr>
<td>&lt;rounded&gt;</td>
<td>P45</td>
<td>P44</td>
</tr>
<tr>
<td>&lt;routine-name&gt;</td>
<td>P85</td>
<td>P82</td>
</tr>
<tr>
<td>&lt;same-block-area-clause&gt;</td>
<td>E86</td>
<td>E84</td>
</tr>
<tr>
<td>&lt;same-clause&gt;</td>
<td>E84</td>
<td>E74</td>
</tr>
<tr>
<td>&lt;same-clauses&gt;</td>
<td>E74</td>
<td>E72</td>
</tr>
<tr>
<td>&lt;same-record-area-clause&gt;</td>
<td>E85</td>
<td>E84</td>
</tr>
<tr>
<td>&lt;same-sort-area-clause&gt;</td>
<td>E87</td>
<td>E84</td>
</tr>
<tr>
<td>&lt;sd-clauses&gt;</td>
<td>D28</td>
<td>D27</td>
</tr>
<tr>
<td>&lt;search-statement&gt;</td>
<td>P120</td>
<td>P39</td>
</tr>
<tr>
<td>&lt;section-body&gt;</td>
<td>P10</td>
<td>P5</td>
</tr>
<tr>
<td>&lt;section-name&gt;</td>
<td>P9</td>
<td>P14</td>
</tr>
<tr>
<td>&lt;section-name-declaration&gt;</td>
<td>P8</td>
<td>P5</td>
</tr>
<tr>
<td>&lt;security-paragraph&gt;</td>
<td>I11</td>
<td>I2</td>
</tr>
<tr>
<td>&lt;seek-statement&gt;</td>
<td>P122</td>
<td>P37</td>
</tr>
<tr>
<td>&lt;segment-limit-clause&gt;</td>
<td>E14</td>
<td>E12</td>
</tr>
<tr>
<td>&lt;select-clause-for-non-ms-file&gt;</td>
<td>E52</td>
<td>E48</td>
</tr>
<tr>
<td>&lt;select-clause-for-random-ms-file&gt;</td>
<td>E54</td>
<td>E50</td>
</tr>
<tr>
<td>META-VARIABLE</td>
<td>DEFN</td>
<td>USED</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>&lt;select-clause-for-sequential-ms-file&gt;</td>
<td>E53</td>
<td>E49</td>
</tr>
<tr>
<td>&lt;select-clause-for-sort-file&gt;</td>
<td>E55</td>
<td>E51</td>
</tr>
<tr>
<td>&lt;separator&gt;</td>
<td>T1</td>
<td>T3</td>
</tr>
<tr>
<td>&lt;sequential-file-name&gt;</td>
<td>D23</td>
<td>P117</td>
</tr>
<tr>
<td>&lt;sequential-ms-file-name&gt;</td>
<td>E53</td>
<td>D23</td>
</tr>
<tr>
<td>&lt;sequential-ms-file-name-declaration&gt;</td>
<td>D19</td>
<td>D15</td>
</tr>
<tr>
<td>&lt;set-statement&gt;</td>
<td>P123</td>
<td>P35</td>
</tr>
<tr>
<td>&lt;sign&gt;</td>
<td>G34</td>
<td>G35</td>
</tr>
<tr>
<td>&lt;sign-condition&gt;</td>
<td>P67</td>
<td>P58</td>
</tr>
<tr>
<td>&lt;sign-float&gt;</td>
<td>D150</td>
<td>D146</td>
</tr>
<tr>
<td>&lt;sign-float-pict&gt;</td>
<td>D146</td>
<td>D144</td>
</tr>
<tr>
<td>&lt;simple-condition&gt;</td>
<td>P58</td>
<td>P57</td>
</tr>
<tr>
<td>&lt;simple-figurative-constant&gt;</td>
<td>G37</td>
<td>G38</td>
</tr>
<tr>
<td>&lt;simple-go-to-statement&gt;</td>
<td>P90</td>
<td>P38</td>
</tr>
<tr>
<td>&lt;size-error-phrase&gt;</td>
<td>P41</td>
<td>P39</td>
</tr>
<tr>
<td>&lt;skip-into-area-b&gt;</td>
<td>G18</td>
<td>T7</td>
</tr>
<tr>
<td>&lt;sort-file-description&gt;</td>
<td>D27</td>
<td>D6</td>
</tr>
<tr>
<td>&lt;sort-file-name&gt;</td>
<td>D30</td>
<td>E55</td>
</tr>
<tr>
<td>&lt;sort-file-name-declaration&gt;</td>
<td>D29</td>
<td>D27</td>
</tr>
<tr>
<td>&lt;sort-file-specification&gt;</td>
<td>D6</td>
<td>D4</td>
</tr>
<tr>
<td>&lt;sort-input-specification&gt;</td>
<td>P126</td>
<td>P124</td>
</tr>
<tr>
<td>&lt;sort-key-identifier&gt;</td>
<td>P129</td>
<td>P125</td>
</tr>
<tr>
<td>&lt;sort-output-assign-clause&gt;</td>
<td>E60</td>
<td>E48</td>
</tr>
<tr>
<td>&lt;sort-output-specification&gt;</td>
<td>P127</td>
<td>E49</td>
</tr>
<tr>
<td>&lt;sort-procedure-range&gt;</td>
<td>P128</td>
<td>P124</td>
</tr>
<tr>
<td>&lt;sort-record-description&gt;</td>
<td>D37</td>
<td>D6</td>
</tr>
<tr>
<td>&lt;sort-record-name-qualified&gt;</td>
<td>D206</td>
<td>P118</td>
</tr>
<tr>
<td>&lt;sort-statement&gt;</td>
<td>P124</td>
<td>P35</td>
</tr>
<tr>
<td>&lt;source-clause&gt;</td>
<td>D184</td>
<td>D108</td>
</tr>
<tr>
<td>&lt;source-computer-entry&gt;</td>
<td>E9</td>
<td>E8</td>
</tr>
<tr>
<td>&lt;source-computer paragraph&gt;</td>
<td>E7</td>
<td>E4</td>
</tr>
<tr>
<td>&lt;source-computer-paragraph-body&gt;</td>
<td>E8</td>
<td>E7</td>
</tr>
<tr>
<td>&lt;source-item-name-identifier&gt;</td>
<td>D230</td>
<td>D184</td>
</tr>
<tr>
<td>&lt;space&gt;</td>
<td>G6</td>
<td>G15</td>
</tr>
<tr>
<td>&lt;spaces&gt;</td>
<td>G19</td>
<td>G19</td>
</tr>
<tr>
<td>&lt;special-character&gt;</td>
<td>G14</td>
<td>G15</td>
</tr>
<tr>
<td>&lt;special-item-name-qualified&gt;</td>
<td>D222</td>
<td>D219</td>
</tr>
<tr>
<td>&lt;special-names-clause&gt;</td>
<td>E20</td>
<td>E19</td>
</tr>
<tr>
<td>&lt;special-names-clauses&gt;</td>
<td>E19</td>
<td>E18</td>
</tr>
<tr>
<td>&lt;special-names-entry&gt;</td>
<td>E18</td>
<td>E17</td>
</tr>
<tr>
<td>&lt;special-names-paragraph&gt;</td>
<td>E16</td>
<td>E4</td>
</tr>
<tr>
<td>&lt;special-names-paragraph-body&gt;</td>
<td>E17</td>
<td>E16</td>
</tr>
<tr>
<td>&lt;statement&gt;</td>
<td>P95</td>
<td>P94</td>
</tr>
<tr>
<td>META-VARIABLE</td>
<td>DEFN</td>
<td>USED</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td><code>&lt;stop-literal-statement&gt;</code></td>
<td>P130</td>
<td></td>
</tr>
<tr>
<td><code>&lt;stop-run-statement&gt;</code></td>
<td>P131</td>
<td>P35</td>
</tr>
<tr>
<td><code>&lt;strophe&gt;</code></td>
<td>T8</td>
<td></td>
</tr>
<tr>
<td><code>&lt;strophe-mark&gt;</code></td>
<td>G17</td>
<td>T8</td>
</tr>
<tr>
<td><code>&lt;structure&gt;</code></td>
<td>T7</td>
<td>T7</td>
</tr>
<tr>
<td><code>&lt;sub-number&gt;</code></td>
<td>D45</td>
<td>D42</td>
</tr>
<tr>
<td><code>&lt;sub-report-spec&gt;</code></td>
<td>D104</td>
<td>D103</td>
</tr>
<tr>
<td><code>&lt;sub-spec&gt;</code></td>
<td>D43</td>
<td>D42</td>
</tr>
<tr>
<td><code>&lt;subordinate-entries&gt;</code></td>
<td>D42</td>
<td>D38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D43</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D90</td>
</tr>
<tr>
<td><code>&lt;subordinate-report-entries&gt;</code></td>
<td>D103</td>
<td>D100</td>
</tr>
<tr>
<td><code>&lt;subscripts&gt;</code></td>
<td>D236</td>
<td>D225</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D226</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D235</td>
</tr>
<tr>
<td><code>&lt;sum-clause&gt;</code></td>
<td>D185</td>
<td>D107</td>
</tr>
<tr>
<td><code>&lt;sum-field-name&gt;</code></td>
<td>D122</td>
<td>D217</td>
</tr>
<tr>
<td><code>&lt;sum-field-name-declaration&gt;</code></td>
<td>D121</td>
<td>D107</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D122</td>
</tr>
<tr>
<td><code>&lt;sum-field-name-qualified&gt;</code></td>
<td>D217</td>
<td>D221</td>
</tr>
<tr>
<td><code>&lt;sum-group-clauses&gt;</code></td>
<td>D110</td>
<td>D101</td>
</tr>
<tr>
<td><code>&lt;sum-item-name-qualified&gt;</code></td>
<td>D221</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>D230</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D231</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D233</td>
</tr>
<tr>
<td><code>&lt;sum-report-group-name&gt;</code></td>
<td>D117</td>
<td>D203</td>
</tr>
<tr>
<td><code>&lt;sum-report-group-name-declaration&gt;</code></td>
<td>D116</td>
<td>D99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D117</td>
</tr>
<tr>
<td><code>&lt;sum-report-group-name-qualified&gt;</code></td>
<td>D203</td>
<td>D211</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D221</td>
</tr>
<tr>
<td><code>&lt;sum-report-group-spec&gt;</code></td>
<td>D101</td>
<td>D99</td>
</tr>
<tr>
<td><code>&lt;sum-spec&gt;</code></td>
<td>D107</td>
<td>D106</td>
</tr>
<tr>
<td><code>&lt;summed-item-name-identifier&gt;</code></td>
<td>D231</td>
<td>D185</td>
</tr>
<tr>
<td><code>&lt;supp-pict&gt;</code></td>
<td>D148</td>
<td>D144</td>
</tr>
<tr>
<td><code>&lt;switch-re-run-condition&gt;</code></td>
<td>E83</td>
<td>E80</td>
</tr>
<tr>
<td><code>&lt;switch-special-names-clause&gt;</code></td>
<td>E22</td>
<td>E20</td>
</tr>
<tr>
<td><code>&lt;switch-status-condition&gt;</code></td>
<td>P66</td>
<td>P58</td>
</tr>
<tr>
<td><code>&lt;switch-status-name&gt;</code></td>
<td>E36</td>
<td>E83</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P66</td>
</tr>
<tr>
<td><code>&lt;switch-status-name-assignment&gt;</code></td>
<td>E32</td>
<td>E22</td>
</tr>
<tr>
<td><code>&lt;switch-status-name-declaration&gt;</code></td>
<td>E35</td>
<td>E33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E36</td>
</tr>
<tr>
<td><code>&lt;synchronized-clause&gt;</code></td>
<td>D186</td>
<td>D60</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D61</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D62</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D92</td>
</tr>
<tr>
<td><code>&lt;term&gt;</code></td>
<td>P49</td>
<td></td>
</tr>
<tr>
<td><code>&lt;terminate-statement&gt;</code></td>
<td>P133</td>
<td>P35</td>
</tr>
<tr>
<td><code>&lt;terminating-character&gt;</code></td>
<td>G10</td>
<td>G13</td>
</tr>
<tr>
<td><code>&lt;terminating-imp-verb-statement&gt;</code></td>
<td>P38</td>
<td>P33</td>
</tr>
<tr>
<td><code>&lt;times-option&gt;</code></td>
<td>P111</td>
<td>P109</td>
</tr>
<tr>
<td><code>&lt;type-clause&gt;</code></td>
<td>D187</td>
<td>D109</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D110</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D111</td>
</tr>
<tr>
<td><code>&lt;unit&gt;</code></td>
<td>P75</td>
<td></td>
</tr>
<tr>
<td><code>&lt;unsigned-primary&gt;</code></td>
<td>P52</td>
<td>P51</td>
</tr>
<tr>
<td><code>&lt;unsigned-proper-numeric-literal&gt;</code></td>
<td>G33</td>
<td>G35</td>
</tr>
<tr>
<td><code>&lt;until-option&gt;</code></td>
<td>P112</td>
<td>P109</td>
</tr>
<tr>
<td><code>&lt;usage-clause&gt;</code></td>
<td>D188</td>
<td>D56</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D57</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D58</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D60</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D61</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D62</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D90</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D92</td>
</tr>
<tr>
<td>META-VARIABLE</td>
<td>DEFN</td>
<td>USED</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>------</td>
<td>--------------</td>
</tr>
<tr>
<td>&lt;usage-is-display-clause&gt;</td>
<td>D189</td>
<td>D105 D107 D108 D109</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D110 D111</td>
</tr>
<tr>
<td>&lt;usage-is-index-clause&gt;</td>
<td>D190</td>
<td>D38 D39 D51 D52</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D53 D54 D86 D87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D90 D91</td>
</tr>
<tr>
<td>&lt;use-before-reporting-statement&gt;</td>
<td>P136</td>
<td>P40</td>
</tr>
<tr>
<td>&lt;use-error-statement&gt;</td>
<td>P134</td>
<td>P40</td>
</tr>
<tr>
<td>&lt;use-label-statement&gt;</td>
<td>P135</td>
<td>P40</td>
</tr>
<tr>
<td>&lt;value-clause&gt;</td>
<td>D191</td>
<td>D56 D60 D108 D111</td>
</tr>
<tr>
<td>&lt;value-of-clause&gt;</td>
<td>D193</td>
<td>D16</td>
</tr>
<tr>
<td>&lt;var-occurs-elem-clauses&gt;</td>
<td>D62</td>
<td>D59</td>
</tr>
<tr>
<td>&lt;var-occurs-non-elem-clauses&gt;</td>
<td>D58</td>
<td>D55</td>
</tr>
<tr>
<td>&lt;variable-occurs-clause&gt;</td>
<td>D135</td>
<td>D58 D62</td>
</tr>
<tr>
<td>&lt;varying-control-phrase&gt;</td>
<td>P114</td>
<td>P113</td>
</tr>
<tr>
<td>&lt;varying-option&gt;</td>
<td>P113</td>
<td>P109</td>
</tr>
<tr>
<td>&lt;when-phrase&gt;</td>
<td>P121</td>
<td>P120</td>
</tr>
<tr>
<td>&lt;word&gt;</td>
<td>G25</td>
<td>G26 G27 T4 N1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>N2 N3 N4 N5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>N7 N8 N9 L5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>L6</td>
</tr>
<tr>
<td>&lt;word-element&gt;</td>
<td>G23</td>
<td>G25</td>
</tr>
<tr>
<td>&lt;word-terminator&gt;</td>
<td>G24</td>
<td>G25</td>
</tr>
<tr>
<td>&lt;working-storage-section&gt;</td>
<td>D7</td>
<td>D2</td>
</tr>
<tr>
<td>&lt;working-storage-section-body&gt;</td>
<td>D8</td>
<td>D7</td>
</tr>
<tr>
<td>&lt;write-Invalid-key-statement&gt;</td>
<td>P141</td>
<td>P39</td>
</tr>
<tr>
<td>&lt;ws-record-description&gt;</td>
<td>D88</td>
<td>D10</td>
</tr>
<tr>
<td>&lt;ws-record-descriptions&gt;</td>
<td>D10</td>
<td>D8</td>
</tr>
<tr>
<td>&lt;ws-record-name&gt;</td>
<td>D195</td>
<td>D178</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D173 D162</td>
</tr>
<tr>
<td>&lt;z&gt;</td>
<td>D152</td>
<td>D148</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D148 D152</td>
</tr>
<tr>
<td>&lt;z-or-*=seq&gt;</td>
<td>D162</td>
<td>G3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>G4</td>
</tr>
<tr>
<td>&lt;z-seq&gt;</td>
<td></td>
<td>G42</td>
</tr>
<tr>
<td>&lt;zero-digit&gt;</td>
<td></td>
<td>G40 G42</td>
</tr>
</tbody>
</table>
EXPLANATORY NOTES
EXPLANATORY NOTES

G6

<space>

6 represents the cobol-character "space"

G8

<relation-character>

The symbols > (greater than), < (less than) are different from Backus brackets.

G9

<currency-sign>

$ represents the currency-sign defined by the implementor.

G17

<strophe-mark>

+ is a cobol-control-character.
In the cobol reference format the strophe-mark is represented by a new line without a hyphen in the continuation area and a skip into area A.

G18

<skip-into-area-b>

* is a cobol-control-character.
In the cobol reference format the skip-into-area-b is represented by a skip into area B, if the current position is in area A, otherwise by no skip.

G45

<proper-relational-operator>

The symbols > (greater than), < (less than) are different from Backus brackets.
APPENDIX

A METALANGUAGE FOR THE
DESCRIPTION OF PROGRAMMING LANGUAGES
# TABLE OF CONTENTS

1. Table of Contents .......................................................... 200
2. Introduction ...................................................................... 201
3. Terminal-Symbols .............................................................. 201
4. Strings ............................................................................. 201
   4.1 The Concept of a String ............................................... 201
   4.2 Concatenation of Strings .............................................. 201
   4.3 Substrings .................................................................... 202
5. Sets of Strings .................................................................... 202
   5.1 The Concept of a set of Strings .................................... 202
   5.2 The Name of a Set of Strings ....................................... 203
   5.3 Constructive Definition of Sets .................................... 204
      5.3.1 Terminal Sets .................................................... 204
      5.3.2 Steps of the Construction .................................... 204
   5.4 Union-Operation for Sets ............................................. 205
   5.5 Concatenation-Operation for Sets ............................... 206
   5.6 "Containing"-Operation for Sets ................................. 207
   5.7 "Not-Containing"-Operation for Sets ........................... 208
   5.8 Difference-Operation for Sets ...................................... 210
   5.9 Option-Operation for Sets .......................................... 210
   5.10 Repetition-Operation for Sets .................................... 212
   5.11 Operation of Permutations ........................................ 213
   5.12 Nested Operations .................................................... 214
6. English Language Extensions of the Metalanguage ............. 216
7. Sets of Connected Examples. ............................................ 216
2. INTRODUCTION

The syntax of a programming language can be described either with plain English, or in a more formal way, using a formalized metalanguage.

Below, the metalanguage is explained, using English language and examples. Connected examples, demonstrating the use of the metalanguage, can be found at the end of this appendix.

3. TERMINAL-SYMBOLS

The terminal-symbols are the irreducible elements of the language to be described by the metalanguage. Usually the terminal-symbols of a language are its character set and some additional symbols that are used to denote controls and other basic concepts of the language which cannot be defined in terms of the character set.

4. STRINGS

4.1 THE CONCEPT OF A STRING

A string is a sequence of terminal-symbols.

If A and B are terminal-symbols, then A AA BAB are strings.

Strings formed by terminal-symbols are written as such, terminated by space or any symbol that is not a terminal-symbol. ("b" is used to represent the terminal-symbol "space", and "space" is used to terminate strings.) The empty string is a sequence which does not consist of any terminal-symbol, that is, the string which is a sequence of 0 terminal-symbols.

4.2 CONCATENATION OF STRINGS

If A and B are strings, then a new string A B, the concatenation of A and B, is defined as the string, consisting of the ordered sequence of symbols comprising A followed by that comprising B.

Thus, if A is the string MN and B is the string PQR, then the concatenation of A and B is the string MNPQR.
Concatenation is associative: The string resulting from first concatenating A and B and then concatenating the string obtained with C is the same as that resulting from concatenating A with the string obtained by concatenating B and C. To concatenate three or more strings, no brackets of any kind are necessary to show any order of concatenation.

Concatenation is obviously not commutative: The concatenation of A and B is not necessarily the same as the concatenation of B and A. Generally the results will be different.

The empty string is the "identity" string with respect to concatenation: The concatenation of the empty string with any other string is a string identical to the latter; the concatenation of any string with the empty string is a string identical with the former.

4.3 SUBSTRINGS

The string A is a substring of the string C, if there exist strings X and Y such that C equals the concatenation of X, A and Y. It may be that either or both X and Y are empty strings.

Thus the string RST is a substring of the string PORSTUVW. There are strings PQ and UVW such that the string PORSTUVW is the concatenation of the strings PQ, RST and UVW.

Other examples: MN is a substring of MNPQR, FGH is a substring of EFGH.

The substring relation is reflexive: Any string A contains this same string A as a trivial substring.

The substring relation is transitive: If A is a substring of B, and B is a substring of C, then A is a substring of C.

The substring relation is not symmetric: If A is a substring of B, then B is not necessarily a substring of A. Generally B will not be a substring of A.

5. SETS OF STRINGS

5.1 THE CONCEPT OF A SET OF STRINGS

Any collection of strings is called a set of strings.

Thus each proper-numeric-literal, permitted in COBOL, is a string of terminals-symbols. All proper-numeric-literals can be grouped together conceptionally to form the set of all proper-numeric-literals.
Further example: Each of the 26 letters A, B, C ... Z is a string, consisting of exactly one terminal-symbol. The 26 letters can be grouped together to form the set of all letters.

The symbol "∈" will be used between a string and the name of a set, to indicate that the string on the left-hand side is contained in the set on the right-hand side. Thus "ABC∈<p>" is an abbreviation for "the string ABC is element of the set <p>". This will only be used in the description of the metalanguage.

The null set of strings is that set that does not contain any strings.

5.2 THE NAME OF A SET OF STRINGS

References to individual sets of strings will be by assigned names. The term meta-variable is used interchangeably with name of a set.

Names of sets will be chosen in such a way that they have some mnemonic relation to the specific set they are assigned to. Any convenient symbols can be used: Terminal-symbols (for instance: digits, period, comma, semi-colon, plus-sign, hyphen, asterisk; upper case letters) or any other additional symbols (for instance: lower case letters).

In order to rigorously indicate the begin and the end of the name, each name begins with a left Backus bracket (<) and ends with a right Backus bracket (>).

Examples of names for sets of strings are: <integer>, <data-division>.

Remark:

The use of Backus brackets is necessary, because it is frequently convenient to utilize terminal-symbols for the formation of names for sets, and further, because it is common practice not to introduce a concatenation operator for sets (similar to multiplication in ordinary arithmetic), thus eliminating the separator between the two operands of concatenation.

Terminal-symbols can be considered as a string consisting of exactly one terminal-symbol. And a set can be formed, consisting of exactly this single string. Though the terminal-symbol and the related set are conceptually different, the terminal-symbol will be used as name for the related one-element set, provided that no misunderstanding is possible.

This slightly extends the conventions for names of sets given above.
Thus B is considered as the name of the set consisting of the single one-character string B, 9 is considered as the name of the set consisting of the single one-character string 9.

Some names of sets have to be used frequently. In this case, unique additional symbols are introduced as abbreviations for the names of the set.

Again, this further extends the conventions for names of sets given above.

Thus # is used as abbreviation for <spaces>, representing the set consisting of strings of one or more spaces.

<null> will be used as name for the null set of strings.

<empty> will be used as name for the set, consisting of the empty string only.

5.3 CONSTRUCTIVE DEFINITION OF SETS

Sets of strings will be defined as intermediate steps for the definition of the set of all syntactically correct programs.

Sets chosen and defined will reflect the syntactic structure of the language being described, and possibly facilitate the description of its semantics in an accompanying English-language document. In other words, sets chosen usually will correspond to concepts or logical entities in the language.

5.3.1 TERMINAL SETS

The construction starts with sets consisting of exactly one string, which in turn consists of exactly one terminal-symbol.

As described above, the terminal-symbol will be used as name for the related terminal set.

5.3.2 STEPS OF THE CONSTRUCTION

Each individual step of the construction is the definition of a new set of strings, in terms of sets already known. It is represented by a meta-definition: The name of the set to be defined, followed by the definition-symbol (::=), which means "is defined as", followed by names of sets already defined and meta-operators representing the principles of construction to be applied.
Example:
<number> ::= <unsigned-number> | <signed-number>

The detail explanation of the operators representing the principles of construction follows below.

5.4 UNION-OPERATION FOR SETS

The union-operation permits the constructive definition of a new set in terms of two other sets already known. The principles of construction represented by the union-operator are explained below.

If <a> and <b> are sets of strings, then a new set <c>::=<a>|<b>, the union of <a> and <b>, is defined as the set consisting of exactly those strings, that are present either in <a> or in <b> or in both.

Or more formally:

X ∈ <a>|<b>, if and only if X ∈ <a> or X ∈ <b>

Thus, if <set-1> consists of the 3 strings AB, CDE, F and if <set-2> consists of the 2 strings X, YZ then

<new-set>::=<set-1>|<set-2>

defines a new set consisting of the following 5 strings:

AB  CDE  F
X  YZ

Further example:

<number> ::= <unsigned-number> | <signed-number>

The concept "number" is defined in terms of the concepts "unsigned-number" and "signed-number", that are assumed to be known. The principle of construction is determined by the union-operator: Any unsigned-number or any signed-number is to be considered as number, that is as a string of the set <number>.

The union-operation for sets is associative: The set resulting from first combining <a> and <b> and then combining the set obtained with <c> is the same as the set resulting from combining <a> with the set obtained by combining <b> and <c>. To combine three or more sets with the union-operation, no brackets of any kind are necessary to show any order of the union-operations.

The union-operation for sets is commutative: The set
resulting from combining <a> and <b> is the same as the set resulting from combining <b> and <a>.

The set <null> is the "identity" with respect to the union-operation: The union of <null> and any other set is a set identical to the latter.

Note. In the other parts of the present book, the union operator is represented by an exclamation mark ! for typographical reasons.

5.5 CONCATENATION-OPERATION FOR SETS

The concatenation-operation permits the constructive definition of a new set in terms of two other sets already known. The principles of construction represented by the concatenation-operator are explained below.

If <a> and <b> are sets of strings, then a new set <c>::=<a><b>, or abbreviated <c>::=<a><b>, the concatenation of <a> and <b>, is defined as the set consisting of exactly those strings, that can be formed by concatenation of any string of <a> with any string of <b>.

Or more formally:

Z ∈ <a><b>, if and only if there are strings X ∈ <a> and Y ∈ <b> such that Z=XY.

Thus, if <set-1> consists of the 3 strings AB, CDE, F and if <set-2> consists of the 2 strings X, YZ then

<new-set>::=<set-1><set-2>

defines a new set consisting of the following 6 strings:

<table>
<thead>
<tr>
<th>ABX</th>
<th>ABYZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDEX</td>
<td>CDEYZ</td>
</tr>
<tr>
<td>FX</td>
<td>FYZ</td>
</tr>
</tbody>
</table>

Further example:

<signed-number>::=<sign><unsigned-number>

The concept "signed-number" is defined in terms of the concepts "sign" and "unsigned-number", that are assumed to be known. The principle of construction is determined by the concatenation-operator: Any sign (+ or -) followed by any unsigned-number is a string that is to be considered as a signed-number, that is, as a string of the set <signed-number>.

The concatenation-operation for sets is associative: The set resulting from first concatenating <a> and <b> and then concatenating the set obtained with <c> is the same as the set resulting from concatenating <a> with the set obtained from concatenating <b> and <c>. 
To concatenate three or more sets, no brackets of any kind are necessary to show any order of concatenation.

Concatenation of sets is obviously not commutative: The concatenation of \(\langle a\rangle\) and \(\langle b\rangle\) is not necessarily the same as the concatenation of \(\langle b\rangle\) and \(\langle a\rangle\). Generally the results will be different.

The set \(\langle\text{empty}\rangle\) is the "identity" with respect to concatenation: The concatenation of \(\langle\text{empty}\rangle\) with any other set is a set identical to the latter; the concatenation of any set with \(\langle\text{empty}\rangle\) is a set identical to the former.

The set \(\langle\text{null}\rangle\) is the "zero" with respect to concatenation: The concatenation of \(\langle\text{null}\rangle\) with any other set is a set identical to \(\langle\text{null}\rangle\); the concatenation of any set with \(\langle\text{null}\rangle\) is a set identical to \(\langle\text{null}\rangle\).

Union-operation and concatenation-operation are distributive:

The set resulting from first forming the union of \(\langle a\rangle\) and \(\langle b\rangle\) and then concatenating the set obtained with \(\langle d\rangle\) is the same as the set resulting from forming the union of the set obtained from concatenating \(\langle a\rangle\) and \(\langle d\rangle\) and of the set obtained from concatenating \(\langle b\rangle\) and \(\langle d\rangle\).

Similarly the set resulting from first forming the union of \(\langle a\rangle\) and \(\langle b\rangle\) and then concatenating \(\langle d\rangle\) with the set obtained is the same as the set resulting from forming the union of the set obtained from concatenating \(\langle d\rangle\) and \(\langle a\rangle\) and of the set obtained from concatenating \(\langle d\rangle\) and \(\langle b\rangle\).

5.6 CONTAINING-OPERATION FOR SETS

The "containing"-operation permits the constructive definition of a new set in terms of two other sets already known. The principles of construction represented by the "containing"-operator are explained below:

If \(\langle a\rangle\) and \(\langle b\rangle\) are sets of strings, then a new set \(\langle c\rangle::=\langle a\rangle\text{\bf containing}\ \langle b\rangle\), or abbreviated \(\langle c\rangle::=\langle a\rangle\text{ containing}\ \langle b\rangle\), is defined as the set consisting of exactly those strings of \(\langle a\rangle\), that contain a substring that belongs to \(\langle b\rangle\).

Or more formally:

\[ X \in \langle a\rangle\text{ containing}\ \langle b\rangle, \text{ if and only if } X \in \langle a\rangle \text{ and there is a string } \gamma \text{ such that } \gamma \text{ substring of } X \text{ and } \gamma \in \langle b\rangle. \]
Thus, if \(<\text{set-1}>\) consists of the 5 strings \(AX, YBZ, 17YZ, X32, Z3\) and if \(<\text{set-2}>\) consists of the 2 strings \(X, YZ\) then

\[
<\text{new-set}>::=<\text{set-1}> \text{ containing } <\text{set-2}>
\]
defines a new set consisting of the following 3 strings

\(AX, 17YZ, X32\)

Further example:

\[
<\text{alphaword}>::=<\text{word}> \text{ containing } <\text{letter}>
\]

The concept "alphaword" is defined in terms of the concepts "word" and "letter", that are assumed to be known. The principle of construction is determined by the "containing" operator: Any word that contains a letter is to be considered as alpha-word, that is as a string of the set \(<\text{alpha-word}>\).

The example above is representative for the main application of the "containing" operator: To characterize subclasses of names in programming languages, the members of which are requested to contain for instance at least one letter, but not necessarily as the first or last symbol.

\(<x> \text{ containing } <\text{null}>\) equals \(<\text{null}>,\) for any set \(<x>\). The set \(<\text{null}>\) is a "right-hand zero" with respect to the "containing"-operation.

\(<\text{null}> \text{ containing } <x>\) equals \(<\text{null}>,\) for any set \(<x>\). The set \(<\text{null}>\) is a "left-hand zero" with respect to the "containing"-operation.

\(<x> \text{ containing } <\text{empty}>\) equals \(<x>,\) for any set \(<x>\). This is implied by the fact that every string contains the empty string as a substring. The set \(<\text{empty}>\) is a "right-hand identity" with respect to the "containing"-operation.

\(<\text{empty}> \text{ containing } <x>\) equals \(<\text{empty}>,\) if \(<x>\) contains the empty string.

\(<\text{empty}> \text{ containing } <x>\) equals \(<\text{null}>,\) if \(<x>\) does not contain the empty string.

5.7 NOT CONTAINING-OPERATION FOR SETS

The "not-containing"-operation permits the constructive definition of a new set in terms of two other sets already known. The principles of construction represented by the "not-containing"-operator are explained below.
If \(<a>\) and \(<b>\) are sets of strings, then a new set
\(<c>:=<a> \text{ not-containing } <b>\), abbreviated
\(<c>:=<a> \text{ not-containing } <b>\), is defined as the set
consisting of exactly those strings of \(<a>\), that do
not contain a substring that belongs to \(<b>\).

Or more formally:

\(X \in <a> \text{ not-containing } <b>, \text{ if and only if } X \in <a>
\text{ and there is no string } Y \text{ such that } Y \text{ substring of } X \text{ and }
Y \in <b>\).

Thus, if \(<\text{set-1}>\) consists of the 5 strings \(AX, YBZ, 1YZ, X32, Z3\), and if \(<\text{set-2}>\) consists of the 2 strings \(X, YZ\)
then

\(<\text{new-set}>:=<\text{set-1}> \text{ not-containing } <\text{set-2}>\)
defines a new set consisting of the following 2 strings:

\(YBZ, Z3\)

Further example:

\(<\text{special-label}>:=<\text{word}> \text{ not-containing } <\text{letter}>\)

The concept "special-label" is defined in terms of the
concepts "word" and "letter", that are assumed to be
known. The principle of construction is determined by the
"not-containing"-operator: Any word that does not contain
a letter is to be considered as special-label, that is as
a string of the set \(<\text{special-label}>\).

The main application of the "not-containing"-operator is
to characterize strings, that are terminated by a defin-
ite symbol, or a definite sequence of symbols (for in-
stance comments or non-numeric literals).

\(<x> \text{ not-containing } <\text{null}>\) equals \(<x>\), for any set \(<x>\). The
set \(<\text{null}>\) is a "right-hand identity" with respect to
the "not-containing"-operation.

\(<\text{null}> \text{ not-containing } <x>\) equals \(<\text{null}>\), for any set
\(<x>\). The set \(<\text{null}>\) is a "left-hand identity" with
respect to the "not-containing"-operation.

\(<x> \text{ not-containing } <\text{empty}>\) equals \(<\text{null}>\), for any set \(<x>\).
This is implied by the fact that every string contains
the empty string as a substring.

\(<\text{empty}> \text{ not-containing } <x>\) equals \(<\text{null}>\), if \(<x>
contains the empty string.

\(<\text{empty}> \text{ not-containing } <x>\) equals \(<\text{empty}>\), if \(<x>
does not contain the empty string.
5.8 DIFFERENCE-OPERATION FOR SETS

The difference-or "different-from"-operation permits the constructive definition of a new set in terms of two other sets already known. The principles of contructions represented by the difference-operator are explained below.

If <a> and <b> are sets of strings, then a new set 
\[ <c> ::= <a> \textbf{diff} <b>, \text{ or abbreviated } <c> ::= <a> \text{ diff } <b>, \]
is defined as the set consisting of exactly those strings of <a>, that are not identical with any string of <b>.

Or more formally:
\[ x \in <a> \textbf{diff} <b>, \text{ if and only if } x \in <a> \text{ and } x \notin <b>. \]

Thus, if <set-1> consists of the 3 strings AX, YBZ, 1YZ and if <set-2> consists of the 2 strings AX, YZ then
\[ <\text{new-set}> ::= <\text{set-1} \textbf{diff} <\text{set-2}> \]
defines a new set consisting of the following 2 strings:

YBZ, 1YZ

Further example:
\[ <\text{non-reserved-word}> ::= <\text{word} \textbf{diff} <\text{reserved-word}> \]

The concept "user-defined-name" is defined in terms of the concepts "word" and "reserved-word", that are assumed to be known. The principle of construction is determined by the difference-operator: Any word that is not identical with a reserved-word can be utilized as a user-defined-name, that is it is a string of the set <user-defined-name>.

The main application of the difference-operator is demonstrated by the example above.

\[ <x> \textbf{diff} <\text{null}> \text{ equals } <x>, \text{ for any set } <x>. \text{ The set } <\text{null}> \text{ is a "right-hand identity". With respect to the difference operation.} \]

\[ <\text{null}> \textbf{diff} <x> \text{ equals } <\text{null}>, \text{ for any set } <x>. \text{ The set } <\text{null}> \text{ is a "left-hand zero" with respect to the difference operation.} \]

5.9 OPTION-OPERATION FOR SETS

The option-operation permits the constructive definition of a new set in terms of another set already known. The principles of construction represented by the option-operator are explained below.

If <a> is a set of strings, then a new set 
\[ <c> ::= [ <a> ], \]
is defined as the set consisting of the
strings of <a> and the empty string.

Or more formally:

\( \times \in [a] \), if and only if \( \times \in a \) or \( \times \) is the empty string.

Thus, if <set-1> consists of the 3 strings AB, CDE, F then

<new-set>::=[<set-1>]

defines a new set consisting of the following 4 strings:

AB, CDE, F, empty string

Further example:

<optional-sign>::=[<sign>]

The concept "optional-sign" is defined in terms of the concept "sign", that is assumed to be known.

The principle of construction is determined by the option-operator: Any sign, omitted or optionally present, is to be considered as optional-sign, that is a string of the set <optional-sign>.

Combined example:

<number>::=[<sign>] <unsigned-number>

The concept "number" is defined in terms of the concepts "sign" and "unsigned-number", that are assumed to be known. The principle of construction is determined by the option-operator and the concatenation-operator: Any unsigned-number optionally preceded by a sign is to be considered as number, that is as a string of the set <number>.

The following 3 constructions are equivalent:

a) <number>::=<optional-sign><unsigned-number>
   <optional-sign>::=<sign>|<empty>

b) <number>::=
   <sign><unsigned-number>|<unsigned-number>

c) <number>::=[<sign>]<unsigned-number>

It is obvious, that construction c) is most convenient for the reader. It is very close to the natural language ("optionally preceded by").

It might be useful to note, that the unary option-operation, an operation or the function with a single
operand or argument, is represented in a way different from the one conventionally used in algebra.

In algebra, the unary operators + and - usually refer to the number following the operator, as for instance in -112. If the operators are intended to refer to a more extended expression, the scope of the operator has to be indicated by parentheses, supplementing the operator, as for instance in -(112+0.5).

The same method would be possible for the option-operation as well, and an operator-symbol "opt" could be introduced. If required, the scope could be indicated by some type of meta-brackets.

However, it seems to be more convenient for the reader to use option-brackets. In that way, the scope of the operation is indicated for all cases, even for the simple ones, and the form of the brackets defines the type of operation to be performed (that is the option-operation and not any other unary set operation).

5.10 REPETITION-OPERATION FOR SETS

The repetition-operation permits the constructive definition of a new set in terms of another set already known. The principles of construction represented by the repetition-operator are explained below.

If \(<a>\) is a set of strings, then a new set \(<c>::=<a>\cdot\cdot\cdot\) is defined as the set consisting of the string of \(<a>\) and all those strings, which can be formed by concatenation of two or more strings, each of them belonging to the set \(<a>\).

Or more formally:

\[ X \in <a> \cdot\cdot\cdot \text{, if and only if there is an integer } n \geq 1 \text{ and strings } X_1, X_2, \ldots, X_n \text{ such that } X_1 \in <a>, X_2 \in <a>, \ldots, X_n \in <a> \text{ and } X = X_1 X_2 \ldots X_n. \]

It should be noted that the set \(<a>\cdot\cdot\cdot\) is identical to the set

\[ <a> | <a>a | <a>a <a> | \text{etc.} \]

Thus, if \(<\text{set-1}>\) consists of the 2 strings AB, XYZ then \(<\text{new-set}>::=<\text{set-1}>\cdot\cdot\cdot\)

defines a new set consisting of the following strings:

<table>
<thead>
<tr>
<th>AB</th>
<th>XYZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABAB</td>
<td>ABXYZ</td>
</tr>
<tr>
<td>ABABAB</td>
<td>ABABXYZ</td>
</tr>
</tbody>
</table>
Further example:

\[ \text{integer} ::= \text{digit}^+ \]

The concept "integer" is defined in terms of the concept "digit," that is assumed to be known. The principle of construction is determined by the repetition-operator: A sequence of one or more digits is to be considered as integer, that is as a string of the set \text{integer}.

It is important to realize, that \text{digit}^+ consists of all digit strings, not of just strings of all zeros, all ones, etc.. Further it should be noted, that \text{digit}^+ does not include the empty string.

The following 3 constructions are equivalent:

a) \text{integer} ::= \text{digit} | \text{integer} \text{digit}

b) \text{integer} ::= \text{digit} | \text{digit} \text{digit} | \text{digit} \text{digit} \text{digit} | \text{etc}.

c) \text{integer} ::=: \text{digit}^+

It is obvious, that construction c) is most convenient for the reader. It is very close to the natural language ("any number of"; "a sequence of"). Construction b) is inconvenient because of the "etc", construction a) because of the recursivity.

It might be useful to note, that the unary repetition-operation, an operation or function with a single operand or argument is represented in a way slightly different from the one conventionally used in algebra.

In algebra, the unary operators + and - usually refer to the number following the operator, as for instance in \(-112\) (pre-fixed operator).

The same method would be possible for the repetition-operation as well. However, for the convenience of the reader, the unary repetition-operator \text{digit}^+ is defined to refer to the set preceding the operator, as for instance <\text{set-1}>^+ (post-fixed operator).

5.11 OPERATION OF PERMUTATIONS

Concatenated sets can be permuted and the permutations can be connected with the union-operator. This operation permits the construction of new sets in terms of sets already known.

The operation will be explained for permutations of two
sets and for permutations of three sets. Analogously, the operation can be defined for permutations of any number of sets.

If \(<a>\) and \(<b>\) are sets of strings then a new set
\[
<\text{p}> ::= \{ \langle a \rangle \downarrow \langle b \rangle \}
\]
is defined as being equivalent to
\[
<\text{p}> ::= \langle a \rangle <\text{b}> | <\text{b}> <\text{a}>
\]
If \(<a>\), \(<b>\) and \(<c>\) are sets of strings then a new set
\[
<\text{p}> ::= \{ \langle a \rangle \downarrow \langle b \rangle \downarrow \langle c \rangle \}
\]
is defined as being equivalent to
\[
<\text{p}> ::= \langle a \rangle <\text{b}> <\text{c}> | <\text{a}> <\text{c}> <\text{b}> |
\langle b \rangle <\text{a}> <\text{c}> | <\text{b}> <\text{c}> <\text{a}> |
\langle c \rangle <\text{a}> <\text{b}> | <\text{c}> <\text{b}> <\text{a}>
\]
The meta-symbols \(\{,\}\) are called permutation-brackets, and the meta-symbol \(\downarrow\) is called permutation-separator.

It might be useful to note that
\[
<\text{p}> ::= \{ \langle a \rangle \uparrow \langle b \rangle \downarrow \langle c \rangle \}
\]
includes 6 triples and 2 doubles \(<b><c>\) and \(<c><b>\), that is, the set being defined is equivalent to
\[
<\text{p}> ::= \langle a \rangle <\text{b}> <\text{c}> | <\text{a}> <\text{c}> <\text{b}> |
\langle b \rangle <\text{a}> <\text{c}> | <\text{b}> <\text{c}> <\text{a}> |
\langle c \rangle <\text{a}> <\text{b}> | <\text{c}> <\text{b}> <\text{a}> |
\langle b \rangle <\text{c}> | <\text{c}> <\text{b}>
\]

5.12 NESTED OPERATIONS

All the operations that have been defined for sets in the previous chapters are complete. That is any set of strings can be used as operand. The operation will always be defined and there will always be a unique set of strings that is the result of the operation.

Consequently, any result can be used as operand for another operation, and any level of nesting is possible.

For instance,
\(<a>\) can be concatenated with \(<b>\), giving \(<a><b>\). The
result can be combined by the union-operator with \( \langle c \rangle \), thus giving \( \langle a \rangle \langle b \rangle | \langle c \rangle \).

It becomes immediately clear, that meta-brackets are required to define the sequence of the operations and to avoid ambiguities. The meta-symbols \{'and'} will be used for that purpose.

Thus the combined operation mentioned above would have to be described by

\[ \langle \langle a \rangle \langle b \rangle \rangle | \langle c \rangle \]

The 3 operands could however have been combined as follows:

\[ \langle a \rangle \langle \langle b \rangle | \langle c \rangle \rangle \]

If \( \langle a \rangle \) consists of a single string AXA, \( \langle b \rangle \) of the single string BYB, and \( \langle c \rangle \) of the single string CZC, then it is obvious, that the two results are different from each other: The first result is a set consisting of the 2 strings AXABYB and CZC, the second result is a set consisting of the 2 strings AXABYB and AXACZC.

Meta-brackets can be avoided to some extent, by assigning priorities to the operators, which determine the sequence of operations, in case meta-brackets are not present.

For the meta-operators defined in the preceding chapters, the following sequence of priorities shall be valid (highest priority first, lowest priority last):

repetition-operator \( \ldots \)
concatenation-operator \( || \), abbreviated by concatenating operands.
union-operator \( | \)
containing-operator \( \text{containing} \), abbreviated by "containing".
not-containing-operator \( \text{not-containing} \), abbreviated by "not-containing".
not-identical-operator \( \text{diff} \), abbreviated by "diff".

The following convention determines the sequence of operations if meta-brackets are not present:

1. If there is a conflict between unary and binary operations, the unary operation is executed first. Thus \( \langle a \rangle | \langle b \rangle \ldots \) is equivalent to \( \langle a \rangle (\langle b \rangle \ldots ) \) and \( \langle a \rangle \langle b \rangle \ldots \) is equivalent to \( \langle a \rangle (\langle b \rangle \ldots ) \).

2. If two binary operations have different priority, the operation with the higher priority is executed first. Thus \( \langle a \rangle | \langle b \rangle \langle c \rangle \) is equivalent to \( \langle a \rangle (\langle b \rangle \langle c \rangle ) \).
3. If two binary operations have the same priority, the order of execution is from left to right. Thus \(<a><b><c>\) is equivalent to \(\{a<b>\}<c>\).

No priorities are assigned to the following operators:

- option-operator
- operators for permutation

Their operands are clearly located in the string of operators and operand-names. First the operands have to be evaluated, afterwards the indicated operation has to be executed. Because of the structure of the operator symbols no meta-brackets are required to enclose the operands.

Thus, \([<a><b>]\) requires first the evaluation of \(<a><b>\), that is the evaluation of the operand of the option-operation, and then adding the empty string to the set, that is the execution of the option-operation.

Similarly, \(\{<a><b>|<x><y>\}\) requires first the evaluation of \(<a><b>\) and of \(<x><y>\), that is the evaluation of the operands of the permutation-operation, and then the execution of the permutation-operation.

That is

\[
<\text{result}>::=\{<a><b>|<x><y>\}
\]

is equivalent to

\[
<\text{result}>::=\{<a><b>\}\{<x><y>\}|\{<x><y>\}\{<a><b>\}
\]

ENGLISH LANGUAGE EXTENSIONS OF THE METALANGUAGE

If required, the metalanguage described above is extended by the use of the English language.

SETS OF CONNECTED EXAMPLES

The following three sets of examples illustrate the utilization of the metalanguage.

Example 1

**Formal Syntactic Definition of the Concept "Possibly-Signed-Integer"**

<table>
<thead>
<tr>
<th>Formal Definition</th>
<th>English Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;\text{digit}&gt;::=0</td>
<td>1</td>
</tr>
</tbody>
</table>
<integer>::=
  <digit>•••
  An integer is a sequence of one or more digits.
<sign>::=
  +|-••
  A sign is either the symbol "+" or the symbol "-".
<possibly-signed-integer>::=
  [<sign>]<integer>
  A possibly-signed-integer is an integer, optionally preceded by a sign.

Example 2

Formal Syntactic Definition of the Concept "Signed-Positive-Integer"

Formal Definition
<digit>::=
  0|1|2|3|4|5|6|7|8|9
  A digit is any one of the listed symbols.

<non-zero-digit>::=
  <digit> diff 0
  A non-zero-digit is a digit different from "0".
<integer>::=
  <digit>•••
  An integer is a sequence of one or more digits.
<positive-integer>::=
  <integer>
  containing<non-zero-digit>
  A positive-integer is an integer that does contain a non-zero-digit.
<signed-positive-integer>::=
  +<positive-integer>
  A signed-positive-integer is a positive-integer, preceded by the symbol "+".

Example 3

Formal Syntactic Definition of the Concept "Comment-Sentence"

Formal Definition

..........

<computer-character>::=
  <cobol-character>|<!additional-data-character>

<delimiting-period>::=
  .(<space>•••)

<comment-string>::=
  (<computer-character>•••) not containing
  <delimiting-period>

<comment-entry>::=
  <comment-string>
  <delimiting-period>

       A comment-string is a sequence of one or more computer-characters, that does not contain a delimiting-period.

       A comment-entry is a comment-string followed by a delimiting-period.