IA

stichting
mathematisch
centrum

MC

AFDELING INFORMATICA IA IW 5/73 JULY

D. GRUNE, L.G.L.TH. MEERTENS and J.C. VAN VLIET
GRAMMAR-HANDLING TOOLS APPLIED TO ALGOL 68

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

AMS/MOS 68B10

ACM - Computing Reviews-category : 5.23

Abstract

In this report the methods are explained that are employed by our group in the construction of an ALGOL 68 compiler, in order to obtain from the syntax of ALGOL 68 as given in the Report a grammar which may serve as a base for mode-independent top-down parsing.
First, we relate how a Van Wijngaarden grammar is turned into a context-free grammar. Second, algorithms are given to determine which relations, such as "is the begin of", may hold between the terminal productions of notions and/or symbols of a given context-free grammar. At last, the process of transforming our context-free grammar into an LL(2) grammar is described, together with an algorithm to verify the LL(1)-ness of context-free grammars. The methods employed are all illustrated by examples taken from the syntax of ALGOL 68.

Contents

0. Preface. 1

1. Constructing an underlying context-free grammar. 2

   1.1. Syntax of ALGOL 68. 4

2. Context-free grammar of ALGOL 68. 34

3. Context-free grammar of 'collection list'. 44

4. Relations concerning the grammar. 47

   4.1. How to read the output. 49

   4.2. Examples. 50

5. Towards an LL(2)-grammar. 53

   5.1. LL(2)-grammar of ALGOL 68. 58

   5.2. Messages. 68

References 70

Appendix 72

0. Preface

In this report the methods are explained that are employed by our group in the construction of an ALGOL 68 compiler, in order to obtain from the syntax of ALGOL 68 as given in the Report a grammar which may serve as a base for mode-independent top-down parsing. As we have little confidence (partly based upon sad experience) in our ability to grasp the complex structure offered by the ALGOL 68 syntax, much of the effort has been invested in implementing algorithms to aid in this task. In view of the fact that the definition of ALGOL 68 is revised more or less continuously, while our aim is to implement the eventual revised language, this investment has in the meantime proved worthwhile. As a means to demonstrate the methods employed, the syntax of ALGOL 68 as given in [2], with some amendments, has been used.

Chapter 1 relates how this "point of departure", a Van Wijngaarden grammar, is turned into a context-free grammar of ALGOL 68 (with 'collection list' omitted), given in Chapter 2, and a context-free grammar of 'collection list', given in Chapter 3. In Chapter 4 algorithms are given to determine which relations, such as "is the begin of" or "contains", may hold between the terminal productions of notions and/or the symbols of a given context-free grammar. The result of turning this algorithm loose on the context-free grammar of 'collection list' is displayed in the Appendix, the result for the grammar of ALGOL 68, obsolete anyway, being too bulky to merit reproduction here. The process of transforming the latter grammar into an LL(2) grammar is described in Chapter 5, together with an algorithm to verify the LL(1)-ness of context-free grammars (in fact, a slightly stronger condition).

1. Constructing an underlying context-free grammar

ALGOL 68 is described in [1] by means of a two level Van Wijngaarden grammar, in which the rules as they are presented may contain "metanotions", which yet have to be replaced by one of their terminal productions. In this way, from a single given rule an infinity of new rules may be derived, corresponding to the infinite number of terminal productions for some metanotions, such as, e.g., "MOID". A typical rule where this mechanism is applied, is rule 7.1.1.aa:

'virtual procedure MOID plan: virtual MOID declarer.'

Because of this feature, Van Wijngaarden grammars are essentially more powerful than context-free grammars. The relationship between the "underlying" context-free grammar developed here, and ALGOL 68, is that the set of ALGOL-68-particular-programs is a strict subset of the set of programs generated by this grammar.

It is not simple to characterize the programs that are generated by the given context-free grammar, but are not ALGOL-68-particular-programs. Informally, they may be termed "programs with coercion errors", but, on the other hand, this terminology might lead one astray. It is better, perhaps, to give some typical examples of such programs:

begin ref int i = true; if (1,2) then skip:= j fi end;

begin real z:= to 3 do new line; 1 + par (2,3) end;

if v of "a" then (1 :=: goto m) [loc union (int)] fi.

It is possible, at the cost of considerable complications, to refine the grammar in such a way that many of the possible "errors" are prevented. From a more semantically oriented point of view, however, it is not al all clear that this would be advantageous. There seems to be no reason why, at this level, a condition like *(1,2)* should be excluded when [ ] *int r = (1,2);r* will still be generated.

Before the "removal" of the metanotions from the syntax of ALGOL 68 was undertaken, a slightly amended version of the grammar in [2]* has been constructed. The changes are of twofold nature:

1. Some notions for which a production rule is given, were considered primitive symbols, e.g., 'identifier', 'string item' and 'collection list', the reason being that the former two are treated by earlier scans than the mode-independent scan, whereas the latter is parsed in a completely different way (viz., bottom-up). A grammar for 'collection list' is given here, however, but treated separately from the remainder of ALGOL 68 (Chapter 3). Pragmats and comments are not taken into account.
2. The changes adopted during the Vienna meeting of WG 2.1 of IFIP have been incorporated in the syntax.

A print of this syntax is given in section 1.1.

In order to get rid of the metanotions, several strategies were applied. Some of the metanotions serve no other purpose than mere shortening of the syntax. In such cases, it is possible to write out the rule for the several terminal productions of the metanotion. In most of these cases, however, a new notion has been created, as in the case of

'enclosed clause: closed clause; collateral clause; choice clause.'.

The new notion 'enclosed clause' replaces, e.g., a hypernotion like 'SORTETY ENCLOSED MOID clause'.

Some metanotions, such as 'PRIORITY', exist only for semantical reasons. These, and their terminal productions, were simply struck out. The same fate befell the metanotions concerned with modes and coercions, such as 'MOID' and 'SORTETY', and their (possibly partial) terminal productions.

*) An advantage, from our point of view, of this grammar over previous versions, is the fact that the so-called extensions have been brought there into the strict language. That this can hardly be underestimated is evidenced by the fact that even the most meticulously constructed context-free grammar known to us contained errors in this respect, albeit that the particular combinations of extensions which were inadvertently excluded, may well have been unforeseen by the authors of the original Report themselves.

Thus, a rule like

'reference to MODE assignation: reference to MODE destination,
becomes token, MODE source.'

was reduced by this process to

'assignation: destination, becomes token, source.'.

A special role is played by the metanotion 'NOTION'. There is no one simple means by which the rules containing 'NOTION' can be brought into context-free form. For notions of the form 'NOTION list', e.g., the production rule has been written out in most of the cases. In order to treat notions of the form 'NOTION option', the syntactical description mechanism has been enriched with a new construct: a list of notions enclosed between the syntactic marks "(" and ")" may be used as one member in another list of notions. The direct productions of such a member are: empty, and the enclosed list of notions. It goes without saying that this does not enlarge the expressive power of the syntactical description mechanism, but is merely an expedient to shorten the syntax.

As a last step, many shortcuts have been made where notions were introduced in the ALGOL 68 syntax in behalf of the semantics. Thus, the two production rules

'variable point numeral: integral part option, fractional part.' and
'integral part: integral denotation.'

were combined into one rule:

'variable point numeral: (integral denotation), fractional part.'.

1.1. Syntax of ALGOL 68

1.2.1. Metaproduction Rules of Modes

a) MODE :: MOOD ; UNITED.

b) MOOD :: PLAIN ; format ; PROCEDURE ; reference to MODE ; structured with FIELDS ; ROWS of MODE.

d) PLAIN :: INTREAL ; boolean ; character.

e) INTREAL :: SHONGSETY integral ; SHONGSETY real.

ha) SHONGSETY :: long LONGSETY ; short SHORTSETY ; EMPTY.

hb) LONGSETY :: long LONGSETY ; EMPTY.

hc) SHORTSETY :: short SHORTSETY ; EMPTY.

i) EMPTY ::.

j) PROCEDURE :: procedure PARAMETY MOID.

k) PARAMETY :: with PARAMETERS ; EMPTY.

l) PARAMETERS :: PARAMETER ; PARAMETERS and PARAMETER.

m) PARAMETER :: MODE parameter.

n) MOID :: MODE ; void.

p) FIELDS :: FIELD ; FIELDS and FIELD.

q) FIELD :: MODE field TAG.

r) TAG :: LETTER ; TAG LETTER ; TAG DIGIT.

s) LETTER :: letter ALPHA ; letter aleph.

t) ALPHA :: a ; b ; c ; d ; e ; f ; g ; h ; i ; j ; k ; l ; m ; n ; o ; p ; q ; r ; s ; t ; u ; v ; w ; x ; y ; z.

u) DIGIT :: digit FIGURE.

v) FIGURE :: zero ; one ; two ; three ; four ; five ; six ; seven ; eight ; nine.

vb) ROWS :: row ; ROWS row.

w) UNITED :: union of LMOODS MOOD mode.

x) LMOODS :: LMOOD ; LMOODS LMOOD.

y) LMOOD :: MOOD and.

1.2.2. Metaproduction Rules Associated with Modes

bb) ROW :: row ; row of.

c) ROWSETY :: ROWS ; EMPTY.

e) NONROW :: PLAIN ; format ; PROCEDURE ; reference to MODE ; UNITED ; structured with FIELDS.

ea) NONREF :: UNITED ; PLAIN ; format ; PROCEDURE ; structured with FIELDS ; ROWS of MODE.

g) REFETY :: reference to ; EMPTY.

h) NONPROC :: PLAIN ; format ; procedure with PARAMETERS MOID ; reference to NONPROC ; UNITED ; structured with FIELDS ; row of MODE.

i) PRAM :: procedure with MODE parameter MOID ; procedure with MODE1 parameter and MODE2 parameter MOID.

ib) PARAMS :: parameter and PARAMETERS ; parameter.

m) LMOODSETY :: MOOD and LMOODSETY ; EMPTY.

n) RMOODSETY :: RMOODSETY and MOOD ; EMPTY.

p) BOX :: LMOODSETY box.

q) LFIELDSETY :: FIELDS and ; EMPTY.

r) RFIELDSETY :: and FIELDS ; EMPTY.

rb) FOLDS :: field TAG and FIELDS ; field TAG.

s) COMPLEX :: structured with real field letter r letter e and real field letter i letter m.

t) BITS :: structured with row of boolean field SHONGTHETY letter aleph.

tb) SHONG :: short ; long.

ub) LENGTHETY :: LENGTH LENGTHETY ; EMPTY.

uc) SHORTHETY :: SHORT SHORTHETY ; EMPTY.

va) LENGTH :: letter l letter o letter n letter g.

vb) SHORT :: letter s letter h letter o letter r letter t.

vc) SHONGTHETY :: LENGTH LENGTHETY ; SHORT SHORTHETY ; EMPTY.

w) BYTES :: structured with row of character field SHONGTHETY letter aleph.

x) STRING :: row of character ; character.

y) MABEL :: MODE mode ; label.

1.2.3. Metaproduction Rules Associated with Phrases and Coercion

cb) SINGLE :: unitary ; ENCLOSED.

d) ENCLOSED :: closed ; collateral ; CHOICE.

ea) CHOICE :: condition ; case ; conformity.

cb) UNETY :: UNITED ; EMPTY.

ec) CONFETY :: UNITED conformity ; EMPTY.

c) SOME :: SORT MOID.

g) SORT :: strong ; firm ; meek ; weak ; soft.

i) STRONG :: FIRM ; widened to ; rowed to ; voided to.

l) FIRM :: MEEK ; united to.

lb) MEEK :: unchanged from ; deprocedured to ; dereferenced to.

jb) STIRM :: strong ; firm.

o) FROBYT :: from ; by ; to.

1.2.4. Metaproduction Rules Associated with Coercends

a) COERCEND :: MOID FORM.

b) FORM :: MORF ; COMORF.

bb) FORMSPEC :: FORM ; specification.

ca) MORF :: routine text ; PRIETY ADIC formula ; selection ; multiple selection ; mode identifier ; slice ; call.

cb) COMORF :: assignation ; cast ; identity relation ; generator ; denotation.

d) ADIC :: dyadic ; monadic.

ea) PRIORITY :: priority NUMBER.

ec) PRIETY :: PRIORITY ; EMPTY.

f) NUMBER :: one ; TWO ; THREE ; FOUR ; FIVE ; SIX ; SEVEN ; EIGHT ; NINE.

g) TWO :: one plus one.

h) THREE :: TWO plus one.

i) FOUR :: THREE plus one.

j) FIVE :: FOUR plus one.

k) SIX :: FIVE plus one.

l) SEVEN :: SIX plus one.

m) EIGHT :: SEVEN plus one.

n) NINE :: EIGHT plus one.

1.2.5. Other Metaproduction Rules

a) VICTAL :: VIRACT ; formal.

ba) VIRACT :: virtual ; actual.

bb) VIRMAL :: virtual ; formal.

c) LOWPER :: lower ; upper.

d) ANY :: KIND ; suppressible KIND ; replicatable KIND ; replicatable suppressible KIND.

e) KIND :: sign ; zero ; digit ; point ; exponent ; complex ; string ; character.

f) NOTION :: ALPHA ; NOTION ALPHA.

i) PACK :: pack ; package.

j) BRACKET :: bracket ; packet ; pack.

k) RADIX :: radix two ; radix four ; radix eight ; radix sixteen.

l) MATCH :: strop ; brief.

n) SHONGABLE :: integral ; real ; complex ; bits ; bytes.

o) HOMETY :: homogeneous ; EMPTY.

p) LEAP :: local ; heap.

2. The Computer and the Program

2.1. Syntax

a) program : open symbol, standard prelude, library prelude option, particular program pack, exit, library postlude option, standard postlude, close symbol.

b) standard prelude : declaration prologue series.

c) library prelude : declaration prologue series.

d) particular program : label sequence option, strong void ENCLOSED CLAUSE.

e) exit : go on token, letter e letter x letter i letter t, label token.

f) library postlude : strong void labelled unit series.

g) standard postlude : strong void train.

3.0.1. General Constructions

b) NOTION option : NOTION ; EMPTY.

c) NOTION sequence : NOTION ; NOTION, NOTION sequence.

d) NOTION list : NOTION ; NOTION, comma token, NOTION list.

g) NOTION list proper : NOTION, comma token, NOTION list.

gb) NOTION sequence proper : NOTION, NOTION sequence.

gc) NOTION series : NOTION, go on token ; NOTION, go on token, NOTION series.

h) NOTION pack : open token, NOTION, close token.

i) NOTION package : begin token, NOTION, end token.

j) NOTION bracket : sub token, NOTION, bus token.

k) NOTION packet : alternate sub token, NOTION, alternate bus token.

l) NOTION token : pragment sequence option, NOTION symbol.

3.0.2. Letter Tokens

b) LETTER : LETTER token.

3.0.3. Digit Tokens

c) digit cypher : DIGIT.

d) DIGIT : DIGIT token.

3.0.9. Comments and pragmats

aa) pragment : pragmat ; comment.

ab) pragmat : MATCH pragmat symbol, comment item sequence option, MATCH pragmat symbol.

b) comment : MATCH comment symbol, comment item sequence option, MATCH comment symbol.

c) comment item : character glyph ; other comment item.

d) character glyph : LETTER symbol ; DIGIT symbol ; point symbol ; times ten to the power symbol ; plus i times symbol ; open symbol ; close symbol ; comma symbol ; space symbol ; plus symbol ; minus symbol ; other character glyph.

3.1. Symbols

3.1.1. Representations

a) Letter symbols

symbols	representation
letter a symbol	*a*
letter b symbol	*b*
letter c symbol	*c*
letter d symbol	*d*
letter e symbol	*e*
letter f symbol	*f*
letter g symbol	*g*
letter h symbol	*h*
letter i symbol	*i*
letter j symbol	*j*
letter k symbol	*k*
letter l symbol	*l*
letter m symbol	*m*
letter n symbol	*n*
letter o symbol	*o*
letter p symbol	*p*
letter q symbol	*q*
letter r symbol	*r*
letter s symbol	*s*
letter t symbol	*t*
letter u symbol	*u*
letter v symbol	*v*
letter w symbol	*w*
letter x symbol	*x*
letter y symbol	*y*
letter z symbol	*z*

b) Denotation symbols

symbol	representation
digit zero symbol	*0*
digit one symbol	*1*
digit two symbol	*2*
digit three symbol	*3*
digit four symbol	*4*
digit five symbol	*5*
digit six symbol	*6*
digit seven symbol	*7*
digit eight symbol	*8*
digit nine symbol	*9*
point symbol	.
times ten to the power symbol	$_{10}$ \
true symbol	TRUE
false symbol	FALSE
formatter symbol	$

c) Action symbols

symbol	representation
plus symbol	+
minus symbol	−
is defined as symbol	=
becomes symbol	:= ..= .=
is symbol	:=: IS
is not symbol	:≠: ISNT
routine symbol	:

d) Declaration symbols

symbol	representation
void symbol	VOID
long symbol	LONG
short symbol	SHORT
structure symbol	STRUCT
reference to symbol	REF
flexible symbol	FLEX
either symbol	EITHER
procedure symbol	PROC
union of symbol	UNION
integral symbol	INT
real symbol	REAL
boolean symbol	BOOL
character symbol	CHAR
format symbol	FORMAT
complex symbol	COMPL
bits symbol	BITS
bytes symbol	BYTES
string symbol	STRING
sema symbol	SEMA
file symbol	FILE
mode symbol	MODE
priority symbol	PRIORITY
local symbol	LOC
heap symbol	HEAP
operation symbol	OP

e) Syntactic symbols

symbol	representation
open symbol	(
close symbol	)

<table>
<tr><td>begin symbol</td><td>BEGIN</td></tr>
<tr><td>end symbol</td><td>END</td></tr>
<tr><td>comma symbol</td><td>,</td></tr>
<tr><td>parallel symbol</td><td>PAR</td></tr>
<tr><td>sub symbol</td><td>[</td></tr>
<tr><td>bus symbol</td><td>]</td></tr>
<tr><td>alternate sub symbol</td><td>(/</td></tr>
<tr><td>alternate bus symbol</td><td>/)</td></tr>
<tr><td>up to symbol</td><td>: ..</td></tr>
<tr><td>at symbol</td><td>@ AT</td></tr>
<tr><td>brief case start symbol</td><td>(</td></tr>
<tr><td>brief case in symbol</td><td>|</td></tr>
<tr><td>brief case again symbol</td><td>|:</td></tr>
<tr><td>brief case out symbol</td><td>|</td></tr>
<tr><td>brief case finish symbol</td><td>)</td></tr>
<tr><td>strop case start symbol</td><td>CASE</td></tr>
<tr><td>strop case in symbol</td><td>IN</td></tr>
<tr><td>strop case again symbol</td><td>OUSE</td></tr>
<tr><td>strop case out symbol</td><td>OUT</td></tr>
<tr><td>strop case finish symbol</td><td>ESAC</td></tr>
<tr><td>brief condition start symbol</td><td>(</td></tr>
<tr><td>brief condition in symbol</td><td>|</td></tr>
<tr><td>brief condition again symbol</td><td>|:</td></tr>
<tr><td>brief condition out symbol</td><td>|</td></tr>
<tr><td>brief condition finish symbol</td><td>)</td></tr>
<tr><td>strop condition start symbol</td><td>IF</td></tr>
<tr><td>strop condition in symbol</td><td>THEN</td></tr>
<tr><td>strop condition again symbol</td><td>ELSF</td></tr>
<tr><td>strop condition out symbol</td><td>ELSE</td></tr>
<tr><td>strop condition finish symbol</td><td>FI</td></tr>
<tr><td>brief conformity start symbol</td><td>(</td></tr>
<tr><td>brief conformity in symbol</td><td>|</td></tr>
<tr><td>brief conformity again symbol</td><td>|:</td></tr>
<tr><td>brief conformity out symbol</td><td>|</td></tr>
</table>

brief conformity finish symbol	)
strop conformity start symbol	CASE
strop conformity in symbol	IN
strop conformity again symbol	OUSE
strop conformity out symbol	OUT
strop conformity finish symbol	ESAC
alternate symbol	:
of symbol	→ OF
label symbol	:

f) Sequencing symbols

symbol	representation
go on symbol	; .,
completion symbol	. EXIT
go to symbol	GOTO GO TO

g) Hip symbols

symbol	representation
skip symbol	~ SKIP
nil symbol	○ NIL

h) Loop symbols

symbol	representation
for symbol	FOR
from symbol	FROM
by symbol	BY
to symbol	TO
while symbol	WHILE
do symbol	DO

i) Special symbols

symbol	representation
quote symbol	"
brief comment symbol	¢ #
strop comment symbol	CO COMMENT
strop pragmat symbol	PR PRAGMAT

4.1. Identifiers

4.1.1. Syntax

a)* identifier : MABEL identifier.

b) MABEL identifier : TAG.

c) TAG LETTER : TAG, LETTER.

d) TAG DIGIT : TAG, DIGIT.

e)* range : program ; SOME closed clause ;
SOME MATCH CHOICE chooser clause ; SOME MATCH condition in clause ;
SOME MATCH CHOICE out clause ; SOME UNITED conformity unit ;
PROCEDURE routine text.

4.2. Indications

4.2.1. Syntax

a)* indication : MODE mode indication ; PRIETY ADIC indication.

g)* adic indication : PRIETY ADIC indication.

4.3. Operators

4.3.1. Syntax

a)* operator : PRAM PRIETY ADIC operator.

b) procedure with MODE1 parameter and MODE2 parameter MOID PRIORITY dyadic operator : PRIORITY dyadic indication.

c) procedure with MODE parameter MOID monadic operator : monadic indication.

d)* ADIC operator : PRAM PRIETY ADIC operator.

5. Denotations

5.0.1. Syntax

a)* denotation : PLAIN denotation ; BITS denotation ; row of character denotation ; format denotation.

5.1. Plain Denotations

5.1.0.1. Syntax

a)* plain denotation : PLAIN denotation.

b) SHONG INTREAL denotation : SHONG token, INTREAL denotation.

5.1.1. Integral Denotations

5.1.1.1. Syntax

a) integral denotation : digit cypher sequence.

5.1.2. Real Denotations

5.1.2.1. Syntax

a) real denotation : variable point numeral ; floating point numeral.

b) variable point numeral : integral part option, fractional part.

c) integral part ; integral denotation.

d) fractional part : point token, integral denotation.

e) floating point numeral : stagnant part, exponent part.

f) stagnant part : integral denotation ; variable point numeral.

g) exponent part : times ten to the power choice, power of ten.

h) times ten to the power choice : times ten to the power token ; letter e.

i) power of ten : plusminus option, integral denotation.

j) plusminus : plus token ; minus token.

5.1.3. Boolean Denotations

5.1.3.1. Syntax

a) boolean denotation : true token ; false token.

5.1.4. Character Denotations

5.1.4.1. Syntax

a) character denotation : quote token, string item, quote symbol.

b) string item : character glyph ; quote image ; other string item.

c) quote image : quote symbol, quote symbol.

5.2. Bits Denotations

5.2.1. Syntax

a)* bits denotation : BITS denotation.

ba) structured with row of boolean field
LENGTH LENGTHETY letter aleph denotation :
long token, structured with row of boolean field LENGTHETY letter aleph denotation.

bb) structured with row of boolean field
SHORT SHORTHETY letter aleph denotation :
short token, structured with row of boolean field SHORTHETY letter aleph denotation.

c) structured with row of boolean field letter aleph denotation :
RADIX, letter r, RADIX digit sequence.

d) radix two : digit two.

e) radix four : digit four.

f) radix eight : digit eight.

g) radix sixteen : digit one, digit six.

h)* radix digit : RADIX digit.

i) radix two digit : digit zero ; digit one.

j) radix four digit ; radix two digit ; digit two ; digit three.

k) radix eight digit : radix four digit ; digit four ; digit five ; digit six ; digit seven.

l) radix sixteen digit : radix eight digit ; digit eight ; digit nine ; letter a ; letter b ; letter c ; letter d ; letter e ; letter f.

5.3. String Denotations

5.3.1. Syntax

a)* string denotation : row of character denotation.

b) row of character denotation : quote token, string item sequence proper option, quote symbol.

5.5. Format Denotations

5.5.1. Syntax

a) format denotation : formatter token, collection list, formatter token.

b) collection : picture ; insertion option, replicator, collection list pack, insertion option.

c) picture : MODE pattern option, insertion option; general pattern option, insertion option.

d) insertion : literal option, insert sequence ; literal.

e) insert : replicator, alignment, literal option.

f) replicator : replication option.

g) replication : dynamic replication ; integral denotation.

h) dynamic replication : letter n, meek integral ENCLOSED clause.

i) alignment : letter k ; letter x ; letter y ; letter l ; letter p.

j) literal : replicator, STRING denotation, replicated literal sequence option.

k) replicated literal : replication, STRING denotation.

l) sign mould : loose replicatable zero frame, sign frame ; loose sign frame.

m) loose ANY frame : insertion option, ANY frame.

n) replicatable ANY frame : replicator, ANY frame.

o) zero frame : letter z.

p) sign frame : plusminus.

q) suppressible ANY frame : letter s option, ANY frame.

r)* frame : ANY frame.

5.5.2. Syntax of Integral Patterns

a) integral pattern : sign mould option, integral mould ; integral choice pattern.

d) integral mould : loose replicatable suppressible digit frame sequence.

e) digit frame : zero frame ; letter d.

f) integral choice pattern : insertion option, letter c, literal list pack.

5.5.3. Syntax of Real Patterns

a) real pattern : sign mould option, real mould ; floating point mould.

b) real mould : integral mould , loose suppressible point frame, integral mould option ; loose suppressible point frame, integral mould.

c) point frame : point token.

d) floating point mould : stagnant mould, loose suppressible exponent frame, sign mould option, integral mould.

e) stagnant mould : sign mould option, INTREAL mould.

f) exponent frame : letter e.

5.5.4. Syntax of Boolean Patterns

a) boolean pattern : insertion option, letter b, boolean choice mould option.

b) boolean choice mould : open token, literal, comma token, literal, close token.

5.5.5. Syntax of Character Patterns

a) character pattern : loose suppressible character frame.

b) character frame : letter a.

5.5.6. Syntax of Complex Patterns

a)* complex pattern : COMPLEX pattern.

b) COMPLEX pattern : real pattern, loose suppressible complex frame, real pattern.

c) complex frame : letter i.

5.5.7. Syntax of String Patterns

a)* string pattern : row of character pattern.

b) row of character pattern : loose string frame ; loose replicatable suppressible character frame sequence proper ; insertion option, replication, suppressible character frame.

string frame : letter t.

5.5.8. Syntax of Format Patterns

a) format pattern : loose inclusion frame, meek ENCLOSED format clause.

b) inclusion frame : letter f.

5.5.9. Syntax of General Patterns

a) general pattern : loose general frame, actual specification list pack option.

b) general frame : letter g.

c) actual specification : minus token option, integral denotation.

5.5.7A. Syntax of Bits Patterns

a)* bits pattern : structured with row of boolean field letter aleph pattern.

b) structured with row of boolean field letter aleph pattern : radix mould, integral mould.

c) radix mould : insertion option, RADIX, letter r.

6. Phrases

6.0.1. Syntax

a)* SINGLE phrase : SINGLE declaration ; SOME SINGLE clause.

b)* SORT MODE SINGLE expression : SORT MODE SINGLE clause.

c)* SINGLE statement : strong void SINGLE clause.

d)* MODE constant : MODE FORM.

e)* MODE variable : reference to MODE FORM.

f)* procedure : REFETY PROCEDURE FORM.

g)* structure display : strong structured with FIELDS and FIELD collateral clause.

h)* row display : STIRM ROW MODE collateral clause.

6.1. Serial Clauses

6.1.1. Syntax

a) SOME serial clause : declaration prologue series option, SOME parade.

b) declaration prologue : strong void unit series option, SINGLE declaration.

e) SOME unit : SOME unitary clause.

g) SORT MOID parade : SORT MOID train ; SORT MOID train, completion token, label, strong MOID parade ; strong MOID train, completion token, label, SORT MOID parade.

h) SOME train : strong void labelled unit series option, SOME labelled unit.

j) SOME labelled unit : label sequence option, SOME unit.

k) label : label identifier, label token.

6.2. Collateral Phrases

6.2.1. Syntax

a) collateral declaration : unitary declaration list proper.

b) strong void collateral clause : parallel token option, strong void unit list proper PACK.

c) STIRM ROW MODE collateral clause : STIRM MODE balance PACK.

e) SORT MOID CONFETY balance : SORT MOID CONFETY unit, comma token, strong MOID CONFETY unit list ; strong MOID CONFETY unit, comma token, SORT MOID CONFETY unit ; strong MOID CONFETY unit, comma token, SORT MOID CONFETY balance.

f) strong structured with FIELDS and FIELD collateral clause : FIELDS and FIELD portrait PACK.

g) FIELDS and FIELD portrait : FIELDS portrait, comma token, FIELD portrait.

h) MODE field TAG portrait : strong MODE unit.

6.3. Closed Clauses

6.3.1. Syntax

a) SOME closed clause : SOME serial clause PACK.

6.4. Choice Clauses

6.4.1. Syntax

aa)* choice clause : SOME CHOICE clause.

ab) SOME CHOICE clause : MATCH CHOICE start token, SOME MATCH CHOICE chooser clause, MATCH CHOICE finish token.

ac) SOME MATCH CHOICE chooser clause : UNETY CHOICE, SOME MATCH UNETY CHOICE alternate clause.

ba) condition : meek boolean serial clause.

bb) case : meek integral serial clause.

bc) UNITED conformity : meek UNITED serial clause.

c) SORT MOID MATCH UNETY CHOICE alternate clause : SORT MOID MATCH UNETY CHOICE in clause, strong MOID MATCH CHOICE out clause option ; strong MOID MATCH UNETY CHOICE in clause, SORT MOID MATCH CHOICE out clause.

ea) SOME MATCH condition in clause : MATCH condition in token, SOME serial clause.

eb) SOME MATCH case in clause : MATCH case in token, SOME balance.

ec) SOME MATCH UNITED conformity in clause : MATCH conformity in token, SOME UNITED conformity unit ; MATCH conformity in token, SOME UNITED conformity balance.

ed) SOME UNITED conformity unit : united to UNITED specification, SOME unit.

ef) meek MODE specification : open token, formal MODE declarer, MODE mode identifier option, close token, alternate token.

eg) SOME MATCH CHOICE out clause : MATCH CHOICE out token, SOME serial clause ; MATCH CHOICE again token, SOME MATCH CHOICE chooser clause.

7. Unitary Declarations

7.0.1. Syntax

a) unitary declaration : mode declaration ; priority declaration ; identifier declaration ; operation declaration.

7.1. Declarers

7.1.1. Syntax

a)* declarer : VICTAL MODE declarer.

b) VICTAL MODE declarer : VICTAL MODE declarator ; MODE mode indication.

ea) VICTAL structured with FIELDS declarator : structure token, VICTAL FIELDS portrayer pack.

g)* field portrayer : VICTAL FIELD portrayer.

ha) VICTAL reference to MODE FOLDS portrayer : virtual reference to MODE declarer, VICTAL reference to MODE FOLDS HOMETY continuation.

hb) VICTAL NONREF FOLDS portrayer : VICTAL NONREF declarer, VICTAL NONREF FOLDS HOMETY continuation.

hc) VICTAL MODE field TAG and MODE FOLDS homegeneous continuation : MODE field TAG selector, comma token, VICTAL MODE FOLDS HOMETY continuation.

hd) VICTAL MODE field TAG HOMETY continuation : mode field TAG SELECTOR.

hc) VICTAL MODE1 field TAG and MODE2 FOLDS continuation : MODE1 field TAG selector, comma token, VICTAL MODE2 FOLDS portrayer.

i)* field selector : FIELD selector.

j) MODE field TAG selector : TAG.

l) VIRACT reference to MODE declarator : reference to token, virtual MODE declarer.

m) formal reference to reference to MODE declarator : reference to token, virtual reference to MODE declarer.

n) formal reference to NONREF declarator : reference to token, formal NONREF declarer.

o) VICTAL ROWS of reference to MODE declarator : VICTAL fleither option, VICTAL ROWS rower BRACKET, virtual reference to MODE declarer.

p) VICTAL ROWS of NONREF declarator : VICTAL fleither option, VICTAL ROWS rower BRACKET, VICTAL NONREF declarer.

pb) formal fleither : flexible token ; either token.

pc) actual fleither : flexible token.

q) VICTAL row ROWS rower : VICTAL row rower, comma token, VICTAL ROWS rower.

ra) actual row rower : lower part option, upper bound.

rb) VIRMAL row rower : up to token option.

s) lower part : lower bound, up to token.

t) LOWPER bound : meek integral unit.

w) VICTAL PROCEDURE declarator : procedure token, virtual PROCEDURE plan.

y) virtual MODE parameter : virtual MODE declarer.

z) virtual void declarer : void token.

bb)* parameters pack : VICTAL PARAMETERS pack.

cc) VICTAL union of LMOODS MOOD mode declarator : union of token, LMOODS MOOD and open box pack.

dd) LMOODSETY LMOOD open BOX : LMOODSETY closed LMOOD end BOX.

ee) LMOODSETY1 closed LMOODSETY2 LMOOD end BOX : LMOODSETY1 closed LMOODSETY2 LMOOD LMOOD end BOX ; LMOODSETY1 open LMOODSETY2 LMOOD BOX.

ff) LMOODSETY1 closed LMOODSETY2 LMOOD1 end LMOOD2 BOX : LMOODSETY1 closed LMOODSETY2 LMOOD2 LMOOD1 end BOX.

gg) open LMOODS LMOOD BOX : LMOODS LMOOD BOX ; open LMOODS box, comma token, LMOOD BOX.

hh) open LMOOD box : LMOOD box.

ii) LMOODS MOOD and box : union of LMOODS MOOD mode mode indication ; union of token, open LMOODS MOOD and box pack.

jj) MOOD and box : virtual MOOD declarer.

7.2. Mode Declarations

7.2.1. Syntax

a) mode declaration : mode token, mode definition list.

b) mode definition : MODE mode indication, is defined as token, actual MODE declarer.

7.3. Priority Declarations

7.3.1. Syntax

aa) priority declaration : priority token, priority definition list.

ab) priority definition : priority NUMBER dyadic indication, is defined as token, NUMBER cypher.

b) one cypher : digit one.

c) TWO cypher : digit two.

d) THREE cypher : digit three.

e) FOUR cypher : digit four.

f) FIVE cypher : digit five.

g) SIX cypher : digit six.

h) SEVEN cypher : digit seven.

i) EIGHT cypher : digit eight.

j) NINE cypher : digit nine.

7.4. Identifier Declarations

7.4.1. Syntax

a) identifier declaration : identity declaration ; variable declaration.

b) identity declaration : MODE identity declaration ; procedure identity declaration.

c) MODE identity declaration : formal MODE declarer, MODE identity definition list.

d) MODE identity definition : MODE mode identifier, is defined as token, strong MODE unit.

e) procedure identity declaration : procedure token, PROCEDURE mode identifier, is defined as token, PROCEDURE routine text.

f) variable declaration : MODE variable declaration ; procedure variable declaration.

g) MODE variable declaration : LEAP token option, actual MODE declarer, MODE variable definition list.

h) MODE variable definition : reference to MODE mode identifier, MODE initialization option.

i) MODE initialization : becomes token, MODE source.

j) procedure variable declaration : LEAP token option, procedure token, reference to PROCEDURE mode identifier, becomes token, PROCEDURE routine text.

7.5. Operation Declarations

7.5.1. Syntax

a) operation declaration : operation token, operator definition.

b) operator definition : PRAM ADIC operator, is defined as token, PRAM routine text ; virtual PRAM plan, PRAM ADIC operator, is defined as token, strong PRAM unit.

8.1. Unitary Clauses

8.1.1. Syntax

a) SOME unitary clause : SOME loop ; SOME routine text ; SOME assignation ; SOME identity relation ; SOME tertiary.

b) SOME tertiary : SOME PRIETY ADIC formula ; SOME secondary.

c) SOME secondary : SOME LEAP generator ; SOME selection ; SOME multiple selection ; SOME primary.

d) SOME primary : SOME denotation ; SOME mode identifier ; SOME slice ; SOME call ; SOME hip ; SOME ENCLOSED clause ; SOME cast.

8.2. Coercends

8.2.0.1. Syntax

a)* coercend : COERCEND.

b)* STRONG coercend : STRONG COERCEND.

d) strong COERCEND : STRONG COERCEND.

e) firm COERCEND : FIRM COERCEND.

fa) meek COERCEND : MEEK COERCEND.

fb) weak reference to MODE FORM : meek reference to MODE FORM.

fc) weak NONREF FORM : unchanged from NONREF FORM ; deprocedured to NONREF FORM.

g) soft reference to MODE FORM : unchanged from reference to MODE FORM ; only deprocedured to reference to MODE FORM.

h) unchanged from MODE FORM : MODE FORM.

8.2.1. Dereferenced Coercends

8.2.1.1. Syntax

a) dereferenced to MODE FORM : MEEK reference to MODE FORM.

8.2.2. Deprocedured Coercends

8.2.2.1. Syntax

aa) deprocedured to MOID MORF : MEEK procedure MOID MORF.

ab) deprocedured to MODE COMORF : MEEK procedure MODE COMORF.

c) only deprocedured to MODE FORM : unchanged from procedure MODE FORM ; only deprocedured to procedure MODE FORM.

8.2.4. United Coercends

8.2.4.1. Syntax

a) united to union of LMOODS MOOD mode FORMSPEC : one out of LMOODS MOOD mode FORMSPEC ; some of LMOODS MOOD and but not FORMSPEC.

b) one out of LMOODSETY2 MOOD RMOODSETY mode FORMSPEC : meek MOOD FORMSPEC.

c) some of LMOODSETY1 MOOD and RMOODSETY but not LMOODSETY2 FORMSPEC : some of LMOODSETY1 and MOOD RMOODSETY but not LMOODSETY2 FORMSPEC ; some of LMOODSETY1 RMOODSETY but not MOOD and LMOODSETY2 FORMSPEC.

d) some of EMPTY and LMOOD MOOD RMOODSETY but not LMOOD2 LMOODSETY2 FORMSPEC : meek union of LMOOD MOOD RMOODSETY mode FORMSPEC.

8.2.5. Widened Coercends

8.2.5.1. Syntax

a) widened to SHONGSETY real FORM : meek SHONGSETY integral FORM.

b) widened to structured with SHONGSETY real field letter r letter e and SHONGSETY real field letter i letter m FORM : meek SHONGSETY real FORM ; widened to SHONGSETY real FORM.

c) widened to row of boolean FORM : meek BITS FORM.

d) widened to row of character FORM : meek BYTES FORM.

e) widened to structured with row of character field letter aleph digit one FORM : meek format FORM.

8.2.6. Rowed Coercends

8.2.6.1. Syntax

aa) rowed to REFETY row ROWS of MODE FORM : strong REFETY ROWS of MODE FORM.

ab) rowed to REFETY row of MODE FORM : strong REFETY MODE FORM.

8.2.8. Voided Coercends

8.2.8.1. Syntax

a) voided to void COMORF : unchanged from MODE COMORF.

b) voided to void MORF : deprocedured to NONPROC MORF ; unchanged from NONPROC MORF.

8.3A. Loops

8.3A.1. Syntax

a) strong void loop : for part option, from part option, by part option, to part option, while part option, do part.

b) for part : for token, integral mode identifier.

c) FROBYT part : FROBYT token, meek integral unit.

d) while part : while token, meek boolean serial clause.

db) do part : do token, strong void unit.

8.3A.2. Routine texts

8.3A.2. Syntax

ba) procedure PARAMETY MOID routine text : formal procedure PARAMETY MOID plan, routine token, strong MOID unit.

bb) VICTAL procedure PARAMETERS MOID plan : VICTAL PARAMETERS pack, virtual MOID declarer.

bc) VICTAL procedure MOID plan ; virtual MOID declarer.

c) VIRACT PARAMETERS and PARAMETER : VIRACT PARAMETERS, comma token, VIRACT PARAMETER.

ea) formal MODE PARAMS : formal MODE declarer, formal MODE PARAMS HOMETY definition.

eb) formal MODE parameter and MODE PARAMS homogeneous definition : MODE mode identifier, comma token, formal MODE PARAMS HOMETY definition.

ec) formal MODE parameter HOMETY definition : mode mode identifier.

ed) formal MODE1 parameter and MODE2 PARAMS definition : MODE1 mode identifier, comma token, formal MODE2 PARAMS.

f)* VICTAL parameters pack : VICTAL PARAMETERS pack.

8.3.1. Assignations

8.3.1.1. Syntax

a) reference to MODE assignation : reference to MODE destination, becomes token, MODE source.

b) reference to MODE destination : soft reference to MODE tertiary.

c) MODE source : strong MODE unit.

8.3.3. Identity Relations

8.3.3.1. Syntax

a) boolean identity relation : soft reference to MODE tertiary, identity relator, strong reference to MODE tertiary ; strong reference to MODE tertiary, identity relator, soft reference to MODE tertiary.

b) identity relator : is token ; is not token.

8.3.4. Casts

8.3.4.1. Syntax

a) MOID cast : virtual MOID declarer, strong MOID ENCLOSED clause.

8.4. Formulas

8.4.1. Syntax

a)* SORT formula : SOME PRIETY ADIC formula.

b) MOID PRIORITY dyadic formula : MODE1 PRIORITY operand, procedure with MODE1 parameter and MODE2 parameter MOID PRIORITY dyadic operator, MODE2 PRIORITY plus one operand.

c)* operand : MODE PRIETY operand.

d) MODE PRIORITY operand : firm MODE PRIORITY dyadic formula ; MODE PRIORITY plus one operand.

e) MODE priority NINE plus one operand : firm MODE monadic formula ; firm MODE secondary.

g) MOID monadic formula : procedure with MODE parameter MOID monadic operator, MODE priority NINE plus one operand.

h)* dyadic formula : MOID PRIORITY dyadic formula.

8.5.1. Generators

8.5.1.1. Syntax

a)* MODE generator : MODE LEAP generator.

b) reference to MODE LEAP generator : LEAP token, actual MODE declarer.

8.5.2. Selections

8.5.2.1. Syntax

a) REFETY MODE selection : MODE field TAG selector, of token, weak REFETY structured with LFIELDSETY MODE field TAG RFIELDSETY secondary.

b) REFETY ROWS of MODE multiple selection : MODE field TAG selector, of token, weak REFETY ROWS of structured with LFIELDSETY MODE field TAG RFIELDSETY secondary.

8.6.1. Slices

8.6.1.1. Syntax

aa) REFETY ROWS of MODE slice : weak REFETY ROWSETY ROWS of MODE primary, ROWSETY ROWS leaving ROWS indexer BRACKET ; weak REFETY ROWS2 of ROWS of MODE primary, ROWS2 leaving EMPTY indexer BRACKET.

ab) REFETY NONROW slice : weak REFETY ROWS2 of NONROW primary, ROWS2 leaving EMPTY indexer BRACKET.

b) row ROWS leaving row ROWSETY indexer : trimmer, comma token, ROWS leaving ROWSETY indexer ; subscript, comma token, ROWS leaving row ROWSETY indexer.

c) row ROWS leaving EMPTY indexer : subscript, comma token, ROWS leaving EMPTY indexer.

d) row leaving row indexer : trimmer.

e) row leaving EMPTY indexer : subscript.

f) trimmer : lower bound option, up to token, upper bound option, new lower bound part option ; new lower bound part option.

g) new lower bound part : at token, new lower bound.

h) new lower bound : meek integral unit.

i) subscript : meek integral unit.

j)* trimscript : trimmer ; subscript.

k)* indexer : ROWS leaving ROWSETY indexer.

l)* boundscript : LOWPER bound ; new lower bound ; subscript.

8.6.2. Calls

8.6.2.1. Syntax

a) MOID call : meek procedure with PARAMETERS MOID primary, actual PARAMETERS pack.

b) actual MODE parameter : strong MODE unit.

8.6.3. Hips

8.6.3.1. Syntax

a) strong MOID hip : MOID skip ; MOID jump ; MOID nihil ; MOID vacuum.

b) MOID skip : skip token.

c) MOID jump : go to token option, label identifier.

d) reference to MODE nihil : nil token.

e) ROWS of MODE vacuum : open token, close token.

```
[ 2. CONTEXT-FREE GRAMMAR OF ALGOL 68. ]


[3,1.TOKENS]

[3,1.1.A. LETTER TOKENS]
    LETTER E TOKEN;

[3,1.1.B. DENOTATION TOKENS]
    DIGIT CYPHER;
    POINT TOKEN;                                    [.]
    TIMES TEN TO THE POWER TOKEN;                   [⏨]
    TRUE TOKEN;                                     [TRUE]
    FALSE TOKEN;                                    [FALSE]
    FORMATTER TOKEN;

[3,1.1.C. ACTION TOKENS]
    PLUS TOKEN;                                     [+]
    MINUS TOKEN;                                    [-]
    BECOMES TOKEN;                                  [:=]
    IS TOKEN;                                       [:=:]
    IS NOT TOKEN;                                   [:≠:]
    ROUTINE TOKEN;                                  [:]

[3,1.1.D. DECLARATION TOKENS]
    VOID TOKEN;                                     [VOID]
    LONG TOKEN;                                     [LONG]
    SHORT TOKEN;                                    [SHORT]
    STRUCTURE TOKEN;                                [STRUCT]
    REFERENCE TO TOKEN;                             [REF]
    FLEXIBLE TOKEN;                                 [FLEX]
    EITHER TOKEN;                                   [EITHER]
    PROCEDURE TOKEN;                                [PROC]
    UNION OF TOKEN;                                 [UNION]
    MODE TOKEN;                                     [MODE]
    PRIORITY TOKEN;                                 [PRIORITY]
    LOCAL TOKEN;                                    [LOC]
    HEAP TOKEN;                                     [HEAP]
    OPERATION TOKEN;                                [OP]
    IS DEFINED AS TOKEN;                            [=]

[3,1.1.E. SYNTACTIC TOKENS]
    OPEN TOKEN;                                     [(]
    CLOSE TOKEN;                                    [)]
    BEGIN TOKEN;                                    [BEGIN]
    END TOKEN;                                      [END]
    COMMA TOKEN;                                    [,]
    PARALLEL TOKEN;                                 [PAR]
    SUB TOKEN;                                      [[]
    BUS TOKEN;                                      []]
    ALTERNATE SUB TOKEN;                            [(/]
    ALTERNATE BUS TOKEN;                            [/)]
    UP TO TOKEN;                                    [:]
    AT TOKEN;                                       [AT]
    BRIEF CASE START TOKEN;                         [(]
    BRIEF CASE IN TOKEN;                            [|]
    BRIEF CASE AGAIN TOKEN;                        [|:]
    BRIEF CASE OUT TOKEN;                          [|]
    BRIEF CASE FINISH TOKEN;                       [)]
    STROP CASE START TOKEN;                        [CASE]
    STROP CASE IN TOKEN;                           [IN]
    STROP CASE AGAIN TOKEN;                        [OUSE]
    STROP CASE OUT TOKEN;                          [OUT]
    STROP CASE FINISH TOKEN;                       [ESAC]
    BRIEF CONDITION START TOKEN;                   [(]
    BRIEF CONDITION IN TOKEN;                      [|]
    BRIEF CONDITION AGAIN TOKEN;                   [|:]
    BRIEF CONDITION OUT TOKEN;                     [|]
    BRIEF CONDITION FINISH TOKEN;                  [)]
    STROP CONDITION START TOKEN;                   [IF]
    STROP CONDITION IN TOKEN;                      [THEN]
    STROP CONDITION AGAIN TOKEN;                   [ELSE]
    STROP CONDITION OUT TOKEN;                     [ELSE]
    STROP CONDITION FINISH TOKEN;                  [FI]
    BRIEF CONFORMITY START TOKEN;                  [(]
    BRIEF CONFORMITY IN TOKEN;                     [|]
    BRIEF CONFORMITY AGAIN TOKEN;                  [|:]
    BRIEF CONFORMITY OUT TOKEN;                    [|]
    BRIEF CONFORMITY FINISH TOKEN;                 [)]
    STROP CONFORMITY START TOKEN;                  [CASE]
    STROP CONFORMITY IN TOKEN;                     [IN]
    STROP CONFORMITY AGAIN TOKEN;                  [OUSE]
    STROP CONFORMITY OUT TOKEN;                    [OUT]
    STROP CONFORMITY FINISH TOKEN;                 [ESAC]
    ALTERNATE TOKEN;                               [:]
    OF TOKEN;                                      [OF]
    LABEL TOKEN;                                   [:]

[3,1.1.F. SEQUENCING TOKENS]
    GO ON TOKEN;                                   [;]
    COMPLETION TOKEN;                              [EXIT]
    GO TO TOKEN;                                   [GOTO]

[3,1.1.G. HIP TOKENS]
    SKIP TOKEN;                                    [SKIP]
    NIL TOKEN;                                     [NIL]

[3,1.1.H. LOOP TOKENS]
    FOR TOKEN;                                     [FOR]
    FROM TOKEN;                                    [FROM]
    BY TOKEN;                                      [BY]
    TO TOKEN;                                      [TO]
    WHILE TOKEN;                                   [WHILE]
    DO TOKEN;                                      [DO]

[3,1.1.I. SPECIAL TOKENS]
    QUOTE TOKEN;                                   ["]

[NOT-DEFINED]
    STRING ITEM;
    BITS DENOTATION;
    COLLECTION LIST;
    IDENTIFIER;
    MODE INDICATION;
    MONADIC INDICATION;
    DYADIC INDICATION.
```

[5. DENOTATIONS]

```
DENOTATION:
    PLAIN DENOTATION; CAPITAL BITS DENOTATION; STRING DENOTATION;
    FORMAT DENOTATION.

PLAIN DENOTATION:
    (SHONG TOKEN SEQUENCE), INTEGRAL DENOTATION;
    (SHONG TOKEN SEQUENCE), REAL DENOTATION;
    BOOLEAN DENOTATION; CHARACTER DENOTATION.
SHONG TOKEN SEQUENCE:
    SHORT TOKEN SEQUENCE; LONG TOKEN SEQUENCE.
SHORT TOKEN SEQUENCE:
    SHORT TOKEN, (SHORT TOKEN SEQUENCE).
LONG TOKEN SEQUENCE:
    LONG TOKEN, (LONG TOKEN SEQUENCE).

INTEGRAL DENOTATION:
    DIGIT CYPHER, (INTEGRAL DENOTATION).

REAL DENOTATION:
    VARIABLE POINT NUMERAL; FLOATING POINT NUMERAL.
VARIABLE POINT NUMERAL:
    (INTEGRAL DENOTATION), FRACTIONAL PART.
FRACTIONAL PART:
    POINT TOKEN, INTEGRAL DENOTATION.
FLOATING POINT NUMERAL:
    STAGNANT PART, EXPONENT PART.
STAGNANT PART:
    INTEGRAL DENOTATION; VARIABLE POINT NUMERAL.
EXPONENT PART:
    TIMES TEN TO THE POWER CHOICE, POWER OF TEN.
TIMES TEN TO THE POWER CHOICE:
    TIMES TEN TO THE POWER TOKEN; LETTER E TOKEN.
POWER OF TEN:
    (PLUSMINUS), INTEGRAL DENOTATION.
PLUSMINUS:
    PLUS TOKEN; MINUS TOKEN,

BOOLEAN DENOTATION:
    TRUE TOKEN; FALSE TOKEN,

CHARACTER DENOTATION:
    QUOTE TOKEN, STRING ITEM, QUOTE TOKEN.

CAPITAL BITS DENOTATION:
    (SHONG TOKEN SEQUENCE), BITS DENOTATION.

STRING DENOTATION:
    QUOTE TOKEN, (STRING ITEM SEQUENCE PROPER), QUOTE TOKEN.
STRING ITEM SEQUENCE PROPER:
    STRING ITEM, STRING ITEM;
    STRING ITEM, STRING ITEM SEQUENCE PROPER.

FORMAT DENOTATION:
    FORMATTER TOKEN, COLLECTION LIST, FORMATTER TOKEN,
[6. PHRASES]

ENCLOSED CLAUSE:
    CLOSED CLAUSE; COLLATERAL CLAUSE; CHOICE CLAUSE.


SERIAL CLAUSE:
    (DECLARATION PROLOGUE SERIES), PARADE.
DECLARATION PROLOGUE SERIES:
    DECLARATION PROLOGUE, GO ON TOKEN,
    (DECLARATION PROLOGUE SERIES).
DECLARATION PROLOGUE:
    (UNIT SERIES), SINGLE DECLARATION.
UNIT SERIES:
    UNIT, GO ON TOKEN, (UNIT SERIES).
PARADE:
    TRAIN, (COMPLETION TOKEN, LABEL, PARADE).
TRAIN:
    (LABELLED UNIT SERIES), LABELLED UNIT.
LABELLED UNIT SERIES:
    LABELLED UNIT, GO ON TOKEN, (LABELLED UNIT SERIES).
LABELLED UNIT:
    (LABEL SEQUENCE), UNIT.
LABEL SEQUENCE:
    LABEL, (LABEL SEQUENCE).
LABEL:
    IDENTIFIER, LABEL TOKEN.

CLOSED CLAUSE:
    OPEN TOKEN, SERIAL CLAUSE, CLOSE TOKEN;
    BEGIN TOKEN, SERIAL CLAUSE, END TOKEN.

COLLATERAL CLAUSE:
    (PARALLEL TOKEN), UNIT LIST PROPER CAPITAL PACK.
UNIT LIST PROPER CAPITAL PACK:
    OPEN TOKEN, UNIT LIST PROPER, CLOSE TOKEN;
    BEGIN TOKEN, UNIT LIST PROPER, END TOKEN.
UNIT LIST PROPER:
    UNIT, COMMA TOKEN, UNIT LIST.
UNIT LIST:
    UNIT, (COMMA TOKEN, UNIT LIST).

CHOICE CLAUSE:
    STROP CONDITION CHOICE CLAUSE; BRIEF CONDITION CHOICE CLAUSE;
    STROP CASE CHOICE CLAUSE; BRIEF CASE CHOICE CLAUSE;
    STROP CONFORMITY CHOICE CLAUSE; BRIEF CONFORMITY CHOICE CLAUSE.

STROP CONDITION CHOICE CLAUSE:
    STROP CONDITION START TOKEN, STROP CONDITION CHOOSER CLAUSE,
    STROP CONDITION FINISH TOKEN.
BRIEF CONDITION CHOICE CLAUSE:
    BRIEF CONDITION START TOKEN, BRIEF CONDITION CHOOSER CLAUSE,
    BRIEF CONDITION FINISH TOKEN.
STROP CASE CHOICE CLAUSE:
    STROP CASE START TOKEN, STROP CASE CHOOSER CLAUSE,
    STROP CASE FINISH TOKEN.
```

```
BRIEF CASE CHOICE CLAUSE:
    BRIEF CASE START TOKEN, BRIEF CASE CHOOSER CLAUSE,
    BRIEF CASE FINISH TOKEN.
STROP CONFORMITY CHOICE CLAUSE:
    STROP CONFORMITY START TOKEN, STROP CONFORMITY CHOOSER CLAUSE,
    STROP CONFORMITY FINISH TOKEN.
BRIEF CONFORMITY CHOICE CLAUSE:
    BRIEF CONFORMITY START TOKEN, BRIEF CONFORMITY CHOOSER CLAUSE,
    BRIEF CONFORMITY FINISH TOKEN.

STROP CONDITION CHOOSER CLAUSE:
    CONDITION, STROP CONDITION ALTERNATE CLAUSE.
BRIEF CONDITION CHOOSER CLAUSE:
    CONDITION, BRIEF CONDITION ALTERNATE CLAUSE.
STROP CASE CHOOSER CLAUSE:
    CASE, STROP CASE ALTERNATE CLAUSE.
BRIEF CASE CHOOSER CLAUSE:
    CASE, BRIEF CASE ALTERNATE CLAUSE.
STROP CONFORMITY CHOOSER CLAUSE:
    CONFORMITY, STROP CONFORMITY ALTERNATE CLAUSE.
BRIEF CONFORMITY CHOOSER CLAUSE:
    CONFORMITY, BRIEF CONFORMITY ALTERNATE CLAUSE.

CONDITION:
    SERIAL CLAUSE.
CASE:
    SERIAL CLAUSE.
CONFORMITY:
    SERIAL CLAUSE.

STROP CONDITION ALTERNATE CLAUSE:
    STROP CONDITION IN CLAUSE, (STROP CONDITION OUT CLAUSE).
BRIEF CONDITION ALTERNATE CLAUSE:
    BRIEF CONDITION IN CLAUSE, (BRIEF CONDITION OUT CLAUSE).
STROP CASE ALTERNATE CLAUSE:
    STROP CASE IN CLAUSE, (STROP CASE OUT CLAUSE).
BRIEF CASE ALTERNATE CLAUSE:
    BRIEF CASE IN CLAUSE, (BRIEF CASE OUT CLAUSE).
STROP CONFORMITY ALTERNATE CLAUSE:
    STROP CONFORMITY IN CLAUSE, (STROP CONFORMITY OUT CLAUSE).
BRIEF CONFORMITY ALTERNATE CLAUSE:
    BRIEF CONFORMITY IN CLAUSE, (BRIEF CONFORMITY OUT CLAUSE).

STROP CONDITION IN CLAUSE:
    STROP CONDITION IN TOKEN, SERIAL CLAUSE.
BRIEF CONDITION IN CLAUSE:
    BRIEF CONDITION IN TOKEN, SERIAL CLAUSE.
STROP CASE IN CLAUSE:
    STROP CASE IN TOKEN, BALANCE.
BRIEF CASE IN CLAUSE:
    BRIEF CASE IN TOKEN, BALANCE.
STROP CONFORMITY IN CLAUSE:
    STROP CONFORMITY IN TOKEN, CONFORMITY UNIT LIST.
BRIEF CONFORMITY IN CLAUSE:
    BRIEF CONFORMITY IN TOKEN, CONFORMITY UNIT LIST.

BALANCE:
    UNIT LIST PROPER.
CONFORMITY UNIT LIST:
    CONFORMITY UNIT, (COMMA TOKEN, CONFORMITY UNIT LIST).
CONFORMITY UNIT:
    SPECIFICATION, UNIT.
SPECIFICATION:
    OPEN TOKEN, DECLARER, (IDENTIFIER), CLOSE TOKEN,
    ALTERNATE TOKEN.

STROP CONDITION OUT CLAUSE:
    STROP CONDITION OUT TOKEN, SERIAL CLAUSE;
    STROP CONDITION AGAIN TOKEN, STROP CONDITION CHOOSER CLAUSE.
BRIEF CONDITION OUT CLAUSE:
    BRIEF CONDITION OUT TOKEN, SERIAL CLAUSE;
    BRIEF CONDITION AGAIN TOKEN, BRIEF CONDITION CHOOSER CLAUSE.
STROP CASE OUT CLAUSE:
    STROP CASE OUT TOKEN, SERIAL CLAUSE;
    STROP CASE AGAIN TOKEN, STROP CASE CHOOSER CLAUSE.
BRIEF CASE OUT CLAUSE:
    BRIEF CASE OUT TOKEN, SERIAL CLAUSE;
    BRIEF CASE AGAIN TOKEN, BRIEF CASE CHOOSER CLAUSE.
STROP CONFORMITY OUT CLAUSE:
    STROP CONFORMITY OUT TOKEN, SERIAL CLAUSE;
    BRIEF CONFORMITY AGAIN TOKEN, BRIEF CONFORMITY CHOOSER CLAUSE.
BRIEF CONFORMITY OUT CLAUSE:
    BRIEF CONFORMITY OUT TOKEN, SERIAL CLAUSE;
    STROP CONFORMITY AGAIN TOKEN, STROP CONFORMITY CHOOSER CLAUSE.



[7. DECLARATIONS]

SINGLE DECLARATION:
    UNITARY DECLARATION LIST.
UNITARY DECLARATION LIST:
    UNITARY DECLARATION, (COMMA TOKEN, UNITARY DECLARATION LIST).

UNITARY DECLARATION:
    MODE DECLARATION; PRIORITY DECLARATION; IDENTIFIER DECLARATION;
    OPERATION DECLARATION.


DECLARER:
    MODE INDICATION; STRUCTURED WITH FIELDS DECLARATOR;
    REFERENCE TO DECLARATOR; ROWS OF DECLARATOR;
    PROCEDURE DECLARATOR; UNION OF DECLARATOR.

STRUCTURED WITH FIELDS DECLARATOR:
    STRUCTURE TOKEN, PORTRAYER PACK.
PORTRAYER PACK:
    OPEN TOKEN, PORTRAYER, CLOSE TOKEN.
PORTRAYER:
    DECLARER, CONTINUATION.
CONTINUATION:
    IDENTIFIER, (COMMA TOKEN, CONTINUATION OR PORTRAYER).
CONTINUATION OR PORTRAYER:
    CONTINUATION; PORTRAYER.

REFERENCE TO DECLARATOR:
    REFERENCE TO TOKEN, DECLARER.

ROWS OF DECLARATOR:
    (FLEITHER), ROWER BRACKET, DECLARER.
FLEITHER:
    FLEXIBLE TOKEN; EITHER TOKEN.
ROWER BRACKET:
    SUB TOKEN, ROWER, BUS TOKEN;
    ALTERNATE SUB TOKEN, ROWER, ALTERNATE BUS TOKEN;
    OPEN TOKEN, ROWER, CLOSE TOKEN.
ROWER:
    ROW ROWER, (COMMA TOKEN, ROWER).
ROW ROWER:
    (LOWER PART), LOWPER BOUND; (UP TO TOKEN).
LOWER PART:
    LOWPER BOUND, UP TO TOKEN.
LOWPER BOUND:
    UNIT.

PROCEDURE DECLARATOR:
    PROCEDURE TOKEN, PROCEDURE PLAN.
PROCEDURE PLAN:
    (VIRTUAL PARAMETERS PACK), MOID DECLARER.
VIRTUAL PARAMETERS PACK:
    OPEN TOKEN, VIRTUAL PARAMETERS LIST, CLOSE TOKEN.
VIRTUAL PARAMETERS LIST:
    DECLARER, (COMMA TOKEN, VIRTUAL PARAMETERS LIST).
MOID DECLARER:
    VOID TOKEN; DECLARER.

UNION OF DECLARATOR:
    UNION OF TOKEN, OPEN BOX PACK.
OPEN BOX PACK:
    OPEN TOKEN, OPEN BOX LIST, CLOSE TOKEN.
OPEN BOX LIST:
    OPEN BOX, (COMMA TOKEN, OPEN BOX LIST).
OPEN BOX:
    DECLARER.


MODE DECLARATION:
    MODE TOKEN, MODE DEFINITION LIST.
MODE DEFINITION LIST:
    MODE DEFINITION, (COMMA TOKEN, MODE DEFINITION LIST).
MODE DEFINITION:
    MODE INDICATION, IS DEFINED AS TOKEN, DECLARER.

PRIORITY DECLARATION:
    PRIORITY TOKEN, PRIORITY DEFINITION LIST.
PRIORITY DEFINITION LIST:
    PRIORITY DEFINITION, (COMMA TOKEN, PRIORITY DEFINITION LIST).
PRIORITY DEFINITION:
    DYADIC INDICATION, IS DEFINED AS TOKEN, DIGIT CYPHER.

IDENTIFIER DECLARATION:
    IDENTITY DECLARATION; VARIABLE DECLARATION.

IDENTITY DECLARATION:
    MODE IDENTITY DECLARATION; PROCEDURE IDENTITY DECLARATION.
MODE IDENTITY DECLARATION:
    DECLARER, MODE IDENTITY DEFINITION LIST.
MODE IDENTITY DEFINITION LIST:
    MODE IDENTITY DEFINITION,
    (COMMA TOKEN, MODE IDENTITY DEFINITION LIST).
MODE IDENTITY DEFINITION:
    IDENTIFIER, IS DEFINED AS TOKEN, UNIT.
PROCEDURE IDENTITY DECLARATION:
    PROCEDURE TOKEN, IDENTIFIER, IS DEFINED AS TOKEN, ROUTINE TEXT.

VARIABLE DECLARATION:
    MODE VARIABLE DECLARATION; PROCEDURE VARIABLE DECLARATION.
MODE VARIABLE DECLARATION:
    (LEAP TOKEN), DECLARER, MODE VARIABLE DEFINITION LIST.
LEAP TOKEN:
    LOCAL TOKEN; HEAP TOKEN.
MODE VARIABLE DEFINITION LIST:
    MODE VARIABLE DEFINITION,
    (COMMA TOKEN, MODE VARIABLE DEFINITION LIST).
MODE VARIABLE DEFINITION:
    IDENTIFIER, (MODE INITIALIZATION).
MODE INITIALIZATION:
    BECOMES TOKEN, SOURCE.
PROCEDURE VARIABLE DECLARATION:
    (LEAP TOKEN), PROCEDURE TOKEN, IDENTIFIER, BECOMES TOKEN,
    ROUTINE TEXT.

OPERATION DECLARATION:
    OPERATION TOKEN, OPERATION DEFINITION.
OPERATION DEFINITION:
    OPERATOR, IS DEFINED AS TOKEN, PRAM ROUTINE TEXT;
    VIRTUAL PRAM PLAN, OPERATOR, IS DEFINED AS TOKEN, UNIT.
OPERATOR:
    MONADIC OPERATOR; DYADIC OPERATOR.
MONADIC OPERATOR:
    MONADIC INDICATION.
DYADIC OPERATOR:
    DYADIC INDICATION.
PRAM ROUTINE TEXT:
    FORMAL PRAM PLAN, ROUTINE TOKEN, UNIT.
VIRTUAL PRAM PLAN:
    VIRTUAL PRAM PACK, MOID DECLARER.
VIRTUAL PRAM PACK:
    OPEN TOKEN, DECLARER, (COMMA TOKEN, DECLARER), CLOSE TOKEN.
FORMAL PRAM PLAN:
    FORMAL PRAM PACK, MOID DECLARER.
FORMAL PRAM PACK:
    OPEN TOKEN, DECLARER, IDENTIFIER,
    (COMMA TOKEN, SECOND FORMAL PARAMETER), CLOSE TOKEN.
SECOND FORMAL PARAMETER:
    (DECLARER), IDENTIFIER.
```

[8, UNITARY CLAUSES.]

```
UNIT:
    LOOP; ROUTINE TEXT; ASSIGNATION; IDENTITY RELATION; TERTIARY.
TERTIARY:
    FORMULA; SECONDARY.
SECONDARY:
    LEAP GENERATOR; SELECTION; PRIMARY.
PRIMARY:
        DENOTATION; IDENTIFIER; SLICE; CALL; HIP; CAST;
        ENCLOSED CLAUSE.


    LOOP:
        (FOR PART), (FROM PART), (BY PART), (TO PART), (WHILE PART),
        DO PART.
    FOR PART:
        FOR TOKEN, IDENTIFIER.
    FROM PART:
        FROM TOKEN, UNIT.
    BY PART:
        BY TOKEN, UNIT.
    TO PART:
        TO TOKEN, UNIT.
    WHILE PART:
        WHILE TOKEN, SERIAL CLAUSE.
    DO PART:
        DO TOKEN, UNIT.

    ROUTINE TEXT:
        FORMAL PROCEDURE PLAN, ROUTINE TOKEN, UNIT.
    FORMAL PROCEDURE PLAN:
        (FORMAL PARAMETERS PACK), MOID DECLARER.
    FORMAL PARAMETERS PACK:
        OPEN TOKEN, FORMAL PARAMETERS, CLOSE TOKEN.
    FORMAL PARAMETERS:
        DECLARER, DEFINITION.
    DEFINITION:
        IDENTIFIER, (COMMA TOKEN, DEFINITION OR PARAMETERS).
    DEFINITION OR PARAMETERS:
        DEFINITION; FORMAL PARAMETERS.

    ASSIGNATION:
        TERTIARY, BECOMES TOKEN, SOURCE.
    SOURCE:
        UNIT.

    IDENTITY RELATION:
        TERTIARY, IDENTITY RELATOR, TERTIARY.
    IDENTITY RELATOR:
        IS TOKEN; IS NOT TOKEN.

    FORMULA:
        MONADIC OPERATOR, OPERAND;
        OPERAND, DYADIC OPERATOR, OPERAND.
    OPERAND:
        FORMULA; SECONDARY.

    LEAP GENERATOR:
        LEAP TOKEN, DECLARER.

    SELECTION:
        IDENTIFIER, OF TOKEN, SECONDARY.

    SLICE:
        PRIMARY, INDEXER BRACKET.
    INDEXER BRACKET:
        SUB TOKEN, INDEXER, BUS TOKEN;
        ALTERNATE SUB TOKEN, INDEXER, ALTERNATE BUS TOKEN;
    OPEN TOKEN, INDEXER, CLOSE TOKEN.
INDEXER:
    TRIMSCRIPT, (COMMA TOKEN, INDEXER).
TRIMSCRIPT:
    (LOWPER BOUND), UP TO TOKEN, (LOWPER BOUND),
    (AT TOKEN, LOWPER BOUND);
    (AT TOKEN, LOWPER BOUND);
    UNIT.

CALL:
    PRIMARY, ACTUAL PARAMETERS PACK.
ACTUAL PARAMETERS PACK:
    OPEN TOKEN, UNIT LIST, CLOSE TOKEN.

HIP:
    SKIP TOKEN; (GO TO TOKEN), IDENTIFIER; NIL TOKEN;
    OPEN TOKEN, CLOSE TOKEN.

CAST:
    MOID DECLARER, ENCLOSED CLAUSE.
```

```
[3. CONTEXT-FREE GRAMMAR OF COLLECTION-LIST. ]

[NOT-DEFINED-NOTIONS]

    LETTER A TOKEN;
    LETTER B TOKEN;
    LETTER C TOKEN;
    LETTER D TOKEN;
    LETTER E TOKEN;
    LETTER F TOKEN;
    LETTER G TOKEN;
    LETTER I TOKEN;
    LETTER K TOKEN;
    LETTER L TOKEN;
    LETTER N TOKEN;
    LETTER P TOKEN;
    LETTER R TOKEN;
    LETTER S TOKEN;
    LETTER T TOKEN;
    LETTER X TOKEN;
    LETTER Y TOKEN;
    LETTER Z TOKEN;

    DIGIT TWO TOKEN;
    DIGIT FOUR TOKEN;
    DIGIT EIGHT TOKEN;
    DIGIT ONE TOKEN;
    DIGIT SIX TOKEN;

    PLUS TOKEN;
    MINUS TOKEN;
    POINT TOKEN;
    OPEN TOKEN;
    CLOSE TOKEN;
    BEGIN TOKEN;
    COMMA TOKEN;
    END TOKEN;
    STROP CASE START TOKEN;                 [CASE]
    STROP CASE FINISH TOKEN;                [ESAC]
    STROP CONDITION START TOKEN;            [IF]
    STROP CONDITION FINISH TOKEN;           [FI]
    INNER CLAUSE;
          [I.E., THE BUNCH BETWEEN THE OUTERMOST BRACKETS
           OF THE ENCLOSED-CLAUSE.]
    INTEGRAL DENOTATION;
    CHARACTER DENOTATION;
    ROW OF CHARACTER DENOTATION.


[FORMAT DENOTATION:
    FORMATTER TOKEN, COLLECTION LIST, FORMATTER TOKEN.]

COLLECTION LIST:
    COLLECTION, (COMMA TOKEN, COLLECTION LIST).
COLLECTION:
    PICTURE;
    (INSERTION), REPLICATOR, COLLECTION LIST PACK, (INSERTION).
COLLECTION LIST PACK:
    OPEN TOKEN, COLLECTION LIST, CLOSE TOKEN.
PICTURE:
    (PATTERN), (INSERTION).
INSERTION:
    (LITERAL), INSERT SEQUENCE; LITERAL.
INSERT SEQUENCE:
    INSERT, (INSERT SEQUENCE).
INSERT:
    REPLICATOR, ALIGNMENT, (LITERAL).
REPLICATOR:
    (REPLICATION).
REPLICATION:
    DYNAMIC REPLICATION; INTEGRAL DENOTATION.
DYNAMIC REPLICATION:
    LETTER N TOKEN, ENCLOSED CLAUSE.
ENCLOSED CLAUSE:
    BEGIN TOKEN, INNER CLAUSE, END TOKEN;
    OPEN TOKEN, INNER CLAUSE, CLOSE TOKEN;
    STROP CASE START TOKEN, INNER CLAUSE, STROP CASE FINISH TOKEN;
    STROP CONDITION START TOKEN, INNER CLAUSE,
    STROP CONDITION FINISH TOKEN.
ALIGNMENT:
    LETTER K TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
    LETTER L TOKEN; LETTER P TOKEN.
LITERAL:
    REPLICATOR, STRING DENOTATION, (REPLICATED LITERAL SEQUENCE).
REPLICATED LITERAL SEQUENCE: REPLICATED LITERAL,
    (REPLICATED LITERAL SEQUENCE).
REPLICATED LITERAL:
    REPLICATION, STRING DENOTATION.
STRING DENOTATION:
    CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
SIGN MOULD:
    (INSERTION), REPLICATOR, LETTER Z TOKEN, SIGN FRAME;
    (INSERTION), SIGN FRAME.
SIGN FRAME:
    PLUS TOKEN; MINUS TOKEN.


PATTERN:
    INTEGRAL PATTERN; REAL PATTERN; BOOLEAN PATTERN;
    CHARACTER PATTERN; COMPLEX PATTERN; STRING PATTERN;
    BITS PATTERN; GENERAL PATTERN; INCLUDED PATTERN.

INTEGRAL PATTERN:
    (SIGN MOULD), INTEGRAL MOULD; INTEGRAL CHOICE PATTERN.
INTEGRAL MOULD:
    (INSERTION), REPLICATOR, (LETTER S TOKEN), DIGIT FRAME,
    (INTEGRAL MOULD).
DIGIT FRAME:
    LETTER Z TOKEN; LETTER D TOKEN.
INTEGRAL CHOICE PATTERN:
    (INSERTION), LETTER C TOKEN, LITERAL LIST PACK.
LITERAL LIST PACK:
    OPEN TOKEN, LITERAL LIST, CLOSE TOKEN.
LITERAL LIST:
    LITERAL, (COMMA TOKEN, LITERAL LIST).

REAL PATTERN:
    (SIGN MOULD), REAL MOULD; FLOATING POINT MOULD.
```

```
REAL MOULD:
    INTEGRAL MOULD, (INSERTION), (LETTER S TOKEN), POINT TOKEN,
    (INTEGRAL MOULD);
    (INSERTION), (LETTER S TOKEN), POINT TOKEN, INTEGRAL MOULD.
FLOATING POINT MOULD:
    STAGNANT MOULD, (INSERTION), (LETTER S TOKEN), LETTER E TOKEN,
    (SIGN MOULD), INTEGRAL MOULD.
STAGNANT MOULD:
    (SIGN MOULD), INTEGRAL MOULD; (SIGN MOULD), REAL MOULD.

BOOLEAN PATTERN:
    (INSERTION), LETTER B TOKEN, (BOOLEAN CHOICE MOULD).
BOOLEAN CHOICE MOULD:
    OPEN TOKEN, LITERAL, COMMA TOKEN, LITERAL, CLOSE TOKEN.

CHARACTER PATTERN:
    (INSERTION), (LETTER S TOKEN), LETTER A TOKEN.

COMPLEX PATTERN:
    REAL PATTERN, (INSERTION), (LETTER S TOKEN), LETTER I TOKEN,
    REAL PATTERN.

STRING PATTERN:
    (INSERTION), LETTER T TOKEN;
    LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME,
    LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
    (INSERTION), REPLICATION, (LETTER S TOKEN), LETTER A TOKEN.
LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE:
    LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME,
    (LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE).
LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME:
    (INSERTION), REPLICATOR, (LETTER S TOKEN), LETTER A TOKEN.

BITS PATTERN:
    RADIX MOULD, INTEGRAL MOULD.
RADIX MOULD:
    (INSERTION), RADIX, LETTER R TOKEN.
RADIX:
    DIGIT TWO TOKEN; DIGIT FOUR TOKEN; DIGIT EIGHT TOKEN;
    DIGIT ONE TOKEN, DIGIT SIX TOKEN.

GENERAL PATTERN:
    (INSERTION), LETTER G TOKEN, (ACTUAL SPECIFICATION PACK).
ACTUAL SPECIFICATION PACK:
    OPEN TOKEN, ACTUAL SPECIFICATION LIST, CLOSE TOKEN.
ACTUAL SPECIFICATION LIST:
    ACTUAL SPECIFICATION, (COMMA TOKEN, ACTUAL SPECIFICATION LIST).
ACTUAL SPECIFICATION:
    (MINUS TOKEN), INTEGRAL DENOTATION.

INCLUDED PATTERN:
    (INSERTION), LETTER F TOKEN, ENCLOSED CLAUSE.
```

4. Relations concerning the grammar

In this Chapter we shall deal with some relations between notions or symbols, viz., "may contain", "may be contained in", "may begin with", "may be the begin of", "may end with", "may be the end of", all of which are transitive, and "may follow" and "may precede". If, in the sequel, the term "notion" is used, this is meant to include symbols; the terminal production of a notion which is a symbol is that symbol. We say that "α may contain β" if there exists some program in which terminal productions ξ an η occur of the notions α and β, such that the occurrence of ξ contains the occurrence of η (where, in the case that the occurrences coincide, ξ contains η if and only if β is a descendent of α in the production tree of ξ). Similar definitions may be given for the other relations. A more extensive treatment can be found, e.g., in Knuth [7] or Gries [8]; here, only the algorithms to determine these relations will be outlined.

First, we need some definitions:

the <u>product</u> of two relations P and R:
$\alpha PR\beta$ if and only if there exists a γ such that $\alpha P\gamma$ and $\gamma R\beta$.

the <u>power</u> of a relation:
$R^1 = R$, $R^2 = RR, \dots R^n = RR^{n-1}$ $(n>1)$.

the <u>transitive closure</u> R^+ of a relation:
$\alpha R^+\beta$ if and only if $\alpha R^n\beta$ for some $n > 0$.

the <u>reflexive transitive closure</u> R^* of a relation:
$\alpha R^*\beta$ if and only if $\alpha = \beta$ or $\alpha R^+\beta$.

the <u>transpose</u> R^T of a relation:
$\alpha R^T\beta$ if and only if $\beta R\alpha$.

The first relation we want to look at, is "may contain". We say α "WITHIN" β if β occurs in the right hand side of the production rule for α. "may contain" may then be computed by: may contain = WITHIN$^+$.
We say α "FIRST" β if β occurs as the first member of some direct production of α. "may begin with" is then given by: may begin with = FIRST$^+$.

Similarly, we define "LAST", so that: may end with = LAST^+. We say α "FOLLOW" β if, in some production rule the succession of members β, α occurs. "may follow" can then be computed as follows:
may follow = FIRST^{*T} FOLLOW LAST^*, that is, α may follow β if there exist γ and δ such that γ FOLLOW δ, γ = α or γ may begin with α, and δ = β or δ may end with β.

The other relations are simply the transpose of one of the relations treated above.

In our program, the notions are represented by numbers and the relations by Boolean matrices, where B represents the relation R if B[i,j] = notion_i R notion_j.

The relations WITHIN, FIRST, LAST and FOLLOW are constructed by scanning the grammar. The transitive closure is computed by Warshall's algorithm [9], which can be expressed in ALGOL 68 as follows:

```
proc transitive closure = (ref [,] bool r) void:
begin int n = upb r;
      for i to n do
      for j to n do
      if r[j,i] then
          for k to n do
          if r[i,k] then r[j,k]:= true fi
      fi
end
```

The reflexive transitive closure is then simply constructed by setting the elements of the diagonal true.

The product R = $F^T VL$, for three relations F, V and L, is computed by:

```
proc mul3 = (ref [,] bool f,v,l,r) void:
begin int n = upb f;
      for a to n do for b to n do r[a,b]:= false;
      for c to n do
      for d to n do
      if v[c,d] then
          for a to n do
          if f[c,a] then
              for b to n do
              if l[d,b] then r[a,b]:= true fi
          fi
      fi
end
```

Although this process is of order n^4 (as compared to the n^3 for taking the product of two matrices), in practice it is much more efficient than first computing, for example, $S = F^TV$ and then $R = SL$, due to the sparseness of the matrix *v*, and to some lesser extent, *f* and *l*.

4.1. <u>How to read the output</u>

In the Appendix prints are given for the various relations defined above, applied to the context-free grammar of collection-list. For each relation R a list is given of entries for each of the notions, where an entry of the form "notion-s_1;...;s_n." indicates that $\{s \mid \text{notion } Rs\} = \{s_1,\ldots,s_n\}$. An entry of the form "notion$_1$,...,notion$_m$-s_1;...;s_n." is an abbreviation of m entries of the form 'notion$_i$-s_1;...;s_n.", $1 \leq i \leq m$.

In each entry, the part "s_1;...;s_n." is split up into two parts, the first one of the form "s_1;...;s_j.", containing only terminals (and, therefore, preceded by the string (terminals:)), the second one of the form "s_{j+1};...;s_n.", containing only non-terminals (and preceded by "(non-terminals:)"). If one of those parts is empty, the string preceding it is also omitted.

To give an example, for the simple grammar

```
ab : a,b.
```

the output would be:

```
may contain              (MC):   ab - a;b.
may be contained in      (CI):   a,b - ab.
may begin with           (BW):   ab - a.
may be the begin of      (BO):   a - ab.
may end with             (EW):   ab - b.
may be the end of        (EO):   b - ab.
may follow               (MF):   b - a.
may precede              (MP):   a - b.
```

On the output, exactly the above lines are used to indicate the corresponding relation. In front of each entry, also a number is given, which serves as a reference number for an alphabetical listing of notions given at the end.

In an entry, one or more of the s_i are marked with an asterisk. If, for a relation R, some s_j is unmarked, this implies that there is a marked s_i such that $s_i Q s_j$, where Q is R (is "may begin with", is "may end with") if R is a transitive relation (is "may follow", is "may precede"). Also, one or more of the notion_i may be marked with a plus. If some notion_i is marked, this implies that this relation R is reflexive with respect to notion_i, i.e., notion_i R notion_i holds.

4.2. Examples

In this section, examples are given how some conclusions can be drawn from the output for a given grammar.

Example 1.
In a certain context (see [10]) the problem arose if it was possible to determine whether a given do-symbol was the first symbol of a loop or not.

The conjecture was that this can be determined by inspecting the symbol immediately preceding the do-symbol (with pragmats not taken into account).

Case a: The do-symbol is the first symbol of a loop. The set of symbols that may precede such a do-symbol is the set of symbols that may precede a loop. For our context-free grammar, we obtained:

:=[*]   :   =   (*begin*   ,   [[*]   (/[*]
@[*] | |:   *case*   *in*   *ouse*   *out*   *if*
then   *elsf*   *else*   ;   *from*[*]   *by*[*]   *to*[*]   *while*   *do*.

Case b: The do-symbol is not the first symbol of a loop. In this case it is preceded by a for-part, a from-part, a by-part, a to-part or a while-part. The set of symbols that may precede the do-symbol is the union of the sets of symbols with which these parts may end:

0...9   *true*   *false* $ *void*   )   *end*   ]   /)
esac   *fi*   *skip*   ~   *nil*   °   "   *a...z*
mode-indication.

It turns out that these sets are disjoint indeed.

Example 2.

In our compiler, the scan immediately preceding the mode-independent scan, proceeds from left to right. One of the main tasks of this scan is to bring formulas in prefix form. When, scanning from right to left, a dyadic-operator is met, this operator has to be stacked, and inserted just before the operand preceding it. This operand is either a secondary, or a formula,

[*] These symbols appear here due to the fact that the context-free grammar encompasses a somewhat larger language.

and then this formula constitutes other secondaries separated by operators with a priority less than the one at hand. In both cases, we have to know where a secondary ends (if parsed from right to left). If looked at from the other side, this exactly covers the set of symbols that may precede a secondary. This set constitutes:

:= :=: :≠: : = (*begin* , [(/
@ | |: *if* *then* *elsf* *else* *case* *in* *ouse*
out *of* ; *from* *by* *to* *while* *do*
indication.

The set of symbols that may be contained in a secondary comprises all ALGOL 68 symbols, because a secondary may contain an enclosed-clause. This can be solved by leaving all constituent enclosed parts (viz. enclosed-clause, indexer-bracket, actual-parameters-pack, portrayer-pack, virtual-parameters-pack, open-box-pack and base-vacuum) out of consideration. This set can be found by determining the relation "may contain", with the above notions defined as symbols. We obtained the following set:

0...9 . $_{10}$ \ *true* *false* + -
void *long* *short* *struct* *ref* *flex* *either* *proc* *union* *loc*
heap *of* *goto* *skip* ~ *nil* ∘ " a...z
string-item format-denotation mode-indication.

As these sets are disjoint, the beginning of a secondary may be determined by simply inspecting symbols.

5. Towards an LL(2)-grammar

The context-free grammar, given in Chapter 2, is by no means locally unambiguous, and, therefore, difficult to parse. Consider, e.g., the syntax of identifier-declarations. Only after having inspected the symbol immediately following the first constituent identifier, it is decidable which alternative should be applied. If this symbol is an is-defined-as-symbol, the identifier-declaration is an identity-declaration, if it is a becomes-symbol, a comma-symbol or a semicolon-symbol, the identifier-declaration is a variable-declaration. However, the declarer which is the beginning of an identity-declaration needs a different parsing from the one a variable-declaration starts with. (The declarer of an identity-relation is virtual, so the bounds of constituent rows-of-declarators should be empty, whereas the declarer of a variable-declaration is actual, so those bounds should not be empty.) Consequently, one has to look forward over an undeterminable (because of the variable size of the declarer) length to decide which case should be applied. This moving up and down the input stream makes a top-down parser very inefficient, opaque, and easily leads to errors.

It is much more elegant to construct a parser, emanating from a grammar which only leads to limited back-up, i.e., a grammar for which it is decidable which path to take by looking at most k symbols ahead, for a given k. Such a grammar is called of type LL(k) (see Knuth [7]). For k = 1, there is no back-up problem at all, at any moment is is decidable from the next symbol of the input stream which rule of the syntax should be applied. The necessary requirements for a grammar* without useless non-terminals (i.e., without non-terminals which either do not produce any finite string, or do not depend on the root of the grammar) to be of type LL(1), are:

1. for any rule of the form $A : a_1;\ldots;a_n.$, the sets first(a_1),..., first(a_n), where first(a_i) = $\{s \mid a_i \text{ FIRST}^* s\}$, are mutually disjoint;

* The grammar then should be written in a form without optional parts, i.e., each part of our grammar written between the braces "(" and ")", should be given a name, which either produces empty, or the enclosed list of notions.

2. at most one of the $a_1,\ldots,a_n$ can produce the null string (ε);
3. if a_p produces ε, then first (A) has no elements in common with follow(A), where follow(A) = $\{s \mid s$ "may follow" A$\}$.

This third requirement is slightly different from the one, given by Knuth [7], but somewhat easier to check. Knuth's version runs as follows:

3'. if a_p produces ε, then first(a_q) has no elements in common with follow(A), $1 \leq q \leq n$, $q \neq p$.

Those two requirements can proved to be equivalent as follows:

a. first(A) $\supseteq$ first(a_q), because a_q is one of the alternatives of A. So, first(A) $\cap$ follow(A) = $\emptyset$ implies that first(a_q) $\cap$ follow(A) = $\emptyset$.

b. Suppose that requirements 1, 2 and 3' hold, and requirement 3 does not. Then there is some rule A : $a_1;\ldots;a_n$, for which some alternative, say a_p, produces empty. Also, there is some s, s $\in$ first(a_p), s $\in$ follow(A), s $\notin$ first(a_q), $q \neq p$. Suppose one of the alternatives of a_p is of the form $m_1,\ldots,m_n$, m_i produces empty ($1 \leq i \leq n$), and there exists some j, with s $\in$ first(m_j) (there is at least one alternative for which this is possible). But also, s $\in$ follow(m_j). Now, A may be replaced by m_j, and case b may be applied again. This goes on for ever, but as there is a finite number of non-terminals, we must get in a loop somewhere. But then, one of the non-terminals is left-recursive, and since requirement 1 still holds, this non-terminal cannot possibly produce any finite string, and is, therefore, useless, which was excluded in the first place. So, requirement 3' also implies requirement 3, and the two requirements are proved equivalent.

Our context-free grammar certainly is not of type LL(1). Requirement 1 is violated quite often. We have used several strategies to transform the context-free grammar such that it satisfies this requirement.

Case a: The rule is left-recursive. Apart from the violation of requirement 1, these rules have to be rewritten anyway. For, if some left-recursive rule is called, this results in calling the same rule without reading a symbol from the input stream, and, consequently, gets the compiler into a loop. Left-recursiveness occurs at two places in our grammar, to wit, primary and formula. As we already wanted to have formulas in prefix form, this problem is solved by putting the operator in front of the formula. The method used for this has been described in Example 2 of section 4.2.

The syntax of primary can easily be transformed by writing it thus:

'primary: non call non slice primary,
(indexer bracket or actual parameters pack sequence).',

where non-call-non-slice-primary stands for any primary, except a call or a slice. However, by using this trick, primaries are going to associate the wrong way around. In the context-free grammar, the last indexer-bracket or actual-parameters-pack belongs to the primary preceding it, whereas in the above version there simply is some non-call-non-slice-primary, followed by a sequence of indexer-brackets and/or actual-parameters-packs, without any relation to one another. To enable the next scan to recover from this trick, some necessary information concerning slices and calls is put in a list, which then can be consulted by the next scan.

Case b: The problem can be solved by simply rewriting the grammar, as is, e.g., the case with the rules for serial-clause or intreal-denotation.

Case c: It is advantageous to advance information through the backward scan immediately preceding the mode-independent scan. This method is, e.g., applied to some rules that may start with a declarer, viz., the rules for identity-declaration, variable-declaration, cast and routine-text. The distinction between the first two possibilities has been described at the beginning of this Chapter. If the declarer is followed by an open-sumbol, it is the beginning of a cast, if it is followed by a colon-symbol, it is the beginning of a routine-text. This information, fixing the kind of construction (in the grammar of section 5.1 termed: follows-identity-decl, follows-variable-decl, follows-cast and follows-routine-text, respectively), is picked up and plugged in just in front of the declarer by the scan proceeding from right to left (backward scan).

If, in some declarer, a procedure-symbol is met, it may be followed by an open-symbol. If this procedure has parameters, the open-symbol is the beginning of a formal-parameters-pack, otherwise it is the beginning of a rows-of-declarator. The first one consists of one or more declarers, the second one consists of zero or more units. Proceeding from right to left, those two can be distinguished, and, therefore, the notion follows-rower has been inserted, indicating the beginning of a rows-of-declarator.

In a similar way, the distinction between the various kinds of enclosed-clauses is made. E.g., if an enclosed-clause (with the exclusion of the eventually constituent enclosed parts) contains no stick-symbols ("|"), no semicolon-symbols, and at least one comma-symbol, it is a collateral-clause, if it contains no stick-symbols, and at least one semicolon-symbol (or neither a semicolon-symbol nor a comma-symbol), it is a closed clause.

As a last step, assignations and identity-relations are told apart by using the same algorithm as applied to formulas, thus bringing them in prefix form too.

In this way, requirement one holds for almost all rules of the grammar. The exceptions are:

1. if an identifier is met, one sometimes has to look forward over one symbol to decide whether it is a label (and then it is followed by a colon-symbol), a selector (then it is followed by an of-symbol), or (part of) a unit. This is an example of very limited back-up, and it is not worthwhile removing those local ambiguities.
2. if, in some unitary-declaration, a comma-symbol is met, it is not possible to the compiler to decide whether it has to continue with the single-declaration at hand, or start a new one (a unitary-declaration is a "list of lists"). Although this local ambiguity cannot be removed from the syntax, this problem can also be solved by looking one symbol ahead. If the next symbol is a mode-symbol, a priority-symbol, an operation-symbol, or one of the inserted notions follows-identity-decl or follows-variable-decl, the parser has to start a new single-declaration, otherwise it simply continues the one at hand.

As for requirement 2, there are four rules that may produce empty: rower, row-rower, indexer and trimscript. By inspecting the rules for these nonterminals it can easy be checked that each of them has only one alternative that may produce empty. For these rules, requirement 3 also holds. By comparing the sets first(A) and follow(A), where A stands for one of the notions mentioned above, it shows that those sets are disjoint.

To ease the checking of whether a given grammar is of type LL(1), a program to perform this task has been written. With the help of the outcome of this program for the context-free grammar of Chapter 2, we have bootstrapped this context-free grammar. The final version of this grammar is given in section 5.1. Section 5.2 shows the messages, given by the program which checks a grammar for LL(1)-ness, for the grammar of section 5.1.

```
[ 5,1. LL(2) GRAMMAR OF ALGOL 68 ]


[ TERMINALS ]


[5.1.TOKENS]

[5.1.1.A. LETTER TOKENS]
    LETTER E TOKEN;

[5.1.1.B. DENOTATION TOKENS]
    DIGIT CYPHER;
    POINT TOKEN;                                   [.]
    TIMES TEN TO THE POWER TOKEN;                  [⏨]
    TRUE TOKEN;                                    [TRUE]
    FALSE TOKEN;                                   [FALSE]
    FORMATTER TOKEN;

[5.1.1.C. ACTION TOKENS]
    PLUS TOKEN;                                    [+]
    MINUS TOKEN;                                   [-]
    BECOMES TOKEN;                                 [:=]
    IS TOKEN;                                      [:=:]
    IS NOT TOKEN;                                  [:≠:]
    ROUTINE TOKEN;                                 [:]

[5.1.1.D. DECLARATION TOKENS]
    VOID TOKEN;                                    [VOID]
    LONG TOKEN;                                    [LONG]
    SHORT TOKEN;                                   [SHORT]
    STRUCTURE TOKEN;                               [STRUCT]
    REFERENCE TO TOKEN;                            [REF]
    FLEXIBLE TOKEN;                                [FLEX]
    EITHER TOKEN;                                  [EITHER]
    PROCEDURE TOKEN;                               [PROC]
    UNION OF TOKEN;                                [UNION]
    MODE TOKEN;                                    [MODE]
    PRIORITY TOKEN;                                [PRIORITY]
    LOCAL TOKEN;                                   [LOC]
    HEAP TOKEN;                                    [HEAP]
    OPERATION TOKEN;                               [OP]
    IS DEFINED AS TOKEN;                           [=]

[5.1.1.E. SYNTACTIC TOKENS]
    OPEN TOKEN;                                    [(]
    CLOSE TOKEN;                                   [)]
    BEGIN TOKEN;                                   [BEGIN]
    END TOKEN;                                     [END]
    COMMA TOKEN;                                   [,]
    PARALLEL TOKEN;                                [PAR]
    SUB TOKEN;                                     [[]
    BUS TOKEN;                                     []
    ALTERNATE SUB TOKEN;                           [(/]
    ALTERNATE BUS TOKEN;                           [/)]
    UP TO TOKEN;                                   [:]
    AT TOKEN;                                      [AT]
    BRIEF CASE START TOKEN;                        [(]
    BRIEF CASE IN TOKEN;                             [|]
    BRIEF CASE AGAIN TOKEN;                          [|:]
    BRIEF CASE OUT TOKEN;                            [|]
    BRIEF CASE FINISH TOKEN;                         [)]
    STROP CASE START TOKEN;                          [CASE]
    STROP CASE IN TOKEN;                             [IN]
    STROP CASE AGAIN TOKEN;                          [OUSE]
    STROP CASE OUT TOKEN;                            [OUT]
    STROP CASE FINISH TOKEN;                         [ESAC]
    BRIEF CONDITION START TOKEN;                     [(]
    BRIEF CONDITION IN TOKEN;                        [|]
    BRIEF CONDITION AGAIN TOKEN;                     [|:]
    BRIEF CONDITION OUT TOKEN;                       [|]
    BRIEF CONDITION FINISH TOKEN;                    [)]
    STROP CONDITION START TOKEN;                     [IF]
    STROP CONDITION IN TOKEN;                        [THEN]
    STROP CONDITION AGAIN TOKEN;                     [ELSE]
    STROP CONDITION OUT TOKEN;                       [ELSE]
    STROP CONDITION FINISH TOKEN;                    [FI]
    BRIEF CONFORMITY START TOKEN;                    [(]
    BRIEF CONFORMITY IN TOKEN;                       [|]
    BRIEF CONFORMITY AGAIN TOKEN;                    [|:]
    BRIEF CONFORMITY OUT TOKEN;                      [|]
    BRIEF CONFORMITY FINISH TOKEN;                   [)]
    STROP CONFORMITY START TOKEN;                    [CASE]
    STROP CONFORMITY IN TOKEN;                       [IN]
    STROP CONFORMITY AGAIN TOKEN;                    [OUSE]
    STROP CONFORMITY OUT TOKEN;                      [OUT]
    STROP CONFORMITY FINISH TOKEN;                   [ESAC]
    ALTERNATE TOKEN;                                 [:]
    OF TOKEN;                                        [OL]
    LABEL TOKEN;                                     [:]

[3,1.1.F. SEQUENCING TOKENS]
    GO ON TOKEN;                                     [;]
    COMPLETION TOKEN;                                [EXIT]
    GO TO TOKEN;                                     [GOTO]

[3,1.1.G. HIP TOKENS]
    SKIP TOKEN;                                      [SKIP]
    NIL TOKEN;                                       [NIL]

[3,1.1.H. LOOP TOKENS]
    FOR TOKEN;                                       [FOR]
    FROM TOKEN;                                      [FROM]
    BY TOKEN;                                        [BY]
    TO TOKEN;                                        [TO]
    WHILE TOKEN;                                     [WHILE]
    DO TOKEN;                                        [DO]

[3,1.1.I. SPECIAL TOKENS]
    QUOTE TOKEN;                                     ["]

[SYMBOLS, INSERTED BY A PREVIOUS SCAN ]
    FOLLOWS CLOSED CLAUSE;
    FOLLOWS COLLATERAL CLAUSE;
    FOLLOWS ROUTINE;
    FOLLOWS CAST;
    FOLLOWS IDENTITY DECL;
    FOLLOWS VARIABLE DECL;
    FOLLOWS ROWER;
    FOLLOWS FORMULA;            [ SOME OPERATOR ]
    FOLLOWS ASSIGNATION;        [ := ]
    FOLLOWS IDENTITY RELATION;    [ :=: OR :≠: ]

[NOT-DEFINED]
    STRING ITEM;
    BITS DENOTATION;
    COLLECTION LIST;
    IDENTIFIER;
    MODE INDICATION;
    MONADIC INDICATION;
    DYADIC INDICATION.




[5, DENOTATIONS]

DENOTATION:
    BOOLEAN DENOTATION; STRING OR CHAR DENOTATION;
    FORMAT DENOTATION;
    (SHONG TOKEN SEQUENCE), INTREAL OR BITS DENOTATION.

BOOLEAN DENOTATION:
    TRUE TOKEN; FALSE TOKEN.

STRING OR CHAR DENOTATION:
    QUOTE TOKEN, (STRING ITEM SEQUENCE), QUOTE TOKEN.
STRING ITEM SEQUENCE:
    STRING ITEM, (STRING ITEM SEQUENCE).

FORMAT DENOTATION:
    FORMATTER TOKEN, COLLECTION LIST, FORMATTER TOKEN.

SHONG TOKEN SEQUENCE:
    SHORT TOKEN SEQUENCE; LONG TOKEN SEQUENCE.
SHORT TOKEN SEQUENCE:
    SHORT TOKEN, (SHORT TOKEN SEQUENCE).
LONG TOKEN SEQUENCE:
    LONG TOKEN, (LONG TOKEN SEQUENCE).

INTREAL OR BITS DENOTATION:
    INTREAL DENOTATION; BITS DENOTATION.
INTREAL DENOTATION:
    INTEGRAL DENOTATION, (FRACTIONAL PART), (EXPONENT PART);
    FRACTIONAL PART, (EXPONENT PART).

INTEGRAL DENOTATION:
    DIGIT CYPHER, (INTEGRAL DENOTATION).

FRACTIONAL PART:
    POINT TOKEN, INTEGRAL DENOTATION.
EXPONENT PART:
    TIMES TEN TO THE POWER CHOICE, POWER OF TEN.
TIMES TEN TO THE POWER CHOICE:
    TIMES TEN TO THE POWER TOKEN; LETTER E TOKEN.
POWER OF TEN:
    (PLUSMINUS), INTEGRAL DENOTATION.
```

```
PLUSMINUS:
    PLUS TOKEN; MINUS TOKEN.




[6. PHRASES]

ENCLOSED CLAUSE:
    CLOSED CLAUSE; COLLATERAL CLAUSE; CHOICE CLAUSE.


SERIAL CLAUSE:
    UNIT, (POST UNIT);
    SINGLE DECLARATION, GO ON TOKEN, SERIAL CLAUSE;
    LABEL, PARADE.
POST UNIT:
    COMPLETER PARADE; GO ON TOKEN, SERIAL CLAUSE.
PARADE:
    LABELLED UNIT, (POST LABELLED UNIT).
POST LABELLED UNIT:
    COMPLETER PARADE; GO ON TOKEN, PARADE.
COMPLETER PARADE:
    COMPLETION TOKEN, LABEL, PARADE.
LABELLED UNIT:
    (LABEL SEQUENCE), UNIT.
LABEL SEQUENCE:
    LABEL, (LABEL SEQUENCE).
LABEL:
    IDENTIFIER, LABEL TOKEN.

CLOSED CLAUSE:
    FOLLOWS CLOSED CLAUSE, REAL CLOSED CLAUSE.
REAL CLOSED CLAUSE:
    OPEN TOKEN, SERIAL CLAUSE, CLOSE TOKEN;
    BEGIN TOKEN, SERIAL CLAUSE, END TOKEN.

COLLATERAL CLAUSE:
    FOLLOWS COLLATERAL CLAUSE, (PARALLEL TOKEN),
    UNIT LIST PROPER CAPITAL PACK.
UNIT LIST PROPER CAPITAL PACK:
    OPEN TOKEN, UNIT LIST PROPER, CLOSE TOKEN;
    BEGIN TOKEN, UNIT LIST PROPER, END TOKEN.
UNIT LIST PROPER:
    UNIT, COMMA TOKEN, UNIT LIST.
UNIT LIST:
    UNIT, (COMMA TOKEN, UNIT LIST).

CHOICE CLAUSE:
    STROP CONDITION CHOICE CLAUSE; BRIEF CONDITION CHOICE CLAUSE;
    STROP CASE CHOICE CLAUSE; BRIEF CASE CHOICE CLAUSE;
    STROP CONFORMITY CHOICE CLAUSE; BRIEF CONFORMITY CHOICE CLAUSE.

STROP CONDITION CHOICE CLAUSE:
    STROP CONDITION START TOKEN, STROP CONDITION CHOOSER CLAUSE,
    STROP CONDITION FINISH TOKEN.
BRIEF CONDITION CHOICE CLAUSE:
    BRIEF CONDITION START TOKEN, BRIEF CONDITION CHOOSER CLAUSE,
    BRIEF CONDITION FINISH TOKEN.
```

```
STROP CASE CHOICE CLAUSE:
    STROP CASE START TOKEN, STROP CASE CHOOSER CLAUSE,
    STROP CASE FINISH TOKEN.
BRIEF CASE CHOICE CLAUSE:
    BRIEF CASE START TOKEN, BRIEF CASE CHOOSER CLAUSE,
    BRIEF CASE FINISH TOKEN.
STROP CONFORMITY CHOICE CLAUSE:
    STROP CONFORMITY START TOKEN, STROP CONFORMITY CHOOSER CLAUSE,
    STROP CONFORMITY FINISH TOKEN.
BRIEF CONFORMITY CHOICE CLAUSE:
    BRIEF CONFORMITY START TOKEN, BRIEF CONFORMITY CHOOSER CLAUSE,
    BRIEF CONFORMITY FINISH TOKEN.

STROP CONDITION CHOOSER CLAUSE:
    CONDITION, STROP CONDITION ALTERNATE CLAUSE.
BRIEF CONDITION CHOOSER CLAUSE:
    CONDITION, BRIEF CONDITION ALTERNATE CLAUSE.
STROP CASE CHOOSER CLAUSE:
    CASE, STROP CASE ALTERNATE CLAUSE.
BRIEF CASE CHOOSER CLAUSE:
    CASE, BRIEF CASE ALTERNATE CLAUSE.
STROP CONFORMITY CHOOSER CLAUSE:
    CONFORMITY, STROP CONFORMITY ALTERNATE CLAUSE.
BRIEF CONFORMITY CHOOSER CLAUSE:
    CONFORMITY, BRIEF CONFORMITY ALTERNATE CLAUSE.

CONDITION:
    SERIAL CLAUSE.
CASE:
    SERIAL CLAUSE.
CONFORMITY:
    SERIAL CLAUSE.

STROP CONDITION ALTERNATE CLAUSE:
    STROP CONDITION IN CLAUSE, (STROP CONDITION OUT CLAUSE).
BRIEF CONDITION ALTERNATE CLAUSE:
    BRIEF CONDITION IN CLAUSE, (BRIEF CONDITION OUT CLAUSE).
STROP CASE ALTERNATE CLAUSE:
    STROP CASE IN CLAUSE, (STROP CASE OUT CLAUSE).
BRIEF CASE ALTERNATE CLAUSE;
    BRIEF CASE IN CLAUSE, (BRIEF CASE OUT CLAUSE).
STROP CONFORMITY ALTERNATE CLAUSE:
    STROP CONFORMITY IN CLAUSE, (STROP CONFORMITY OUT CLAUSE),
BRIEF CONFORMITY ALTERNATE CLAUSE:
    BRIEF CONFORMITY IN CLAUSE, (BRIEF CONFORMITY OUT CLAUSE).

STROP CONDITION IN CLAUSE:
    STROP CONDITION IN TOKEN, SERIAL CLAUSE.
BRIEF CONDITION IN CLAUSE:
    BRIEF CONDITION IN TOKEN, SERIAL CLAUSE.
STROP CASE IN CLAUSE:
    STROP CASE IN TOKEN, BALANCE.
BRIEF CASE IN CLAUSE:
    BRIEF CASE IN TOKEN, BALANCE.
STROP CONFORMITY IN CLAUSE:
    STROP CONFORMITY IN TOKEN, CONFORMITY UNIT LIST.
BRIEF CONFORMITY IN CLAUSE:
    BRIEF CONFORMITY IN TOKEN, CONFORMITY UNIT LIST.

BALANCE:
    UNIT LIST PROPER.
CONFORMITY UNIT LIST:
    CONFORMITY UNIT, (COMMA TOKEN, CONFORMITY UNIT LIST).
CONFORMITY UNIT:
    SPECIFICATION, UNIT.
SPECIFICATION:
    OPEN TOKEN, DECLARER, (IDENTIFIER), CLOSE TOKEN,
    ALTERNATE TOKEN.

STROP CONDITION OUT CLAUSE:
    STROP CONDITION OUT TOKEN, SERIAL CLAUSE;
    STROP CONDITION AGAIN TOKEN, STROP CONDITION CHOOSER CLAUSE.
BRIEF CONDITION OUT CLAUSE:
    BRIEF CONDITION OUT TOKEN, SERIAL CLAUSE;
    BRIEF CONDITION AGAIN TOKEN, BRIEF CONDITION CHOOSER CLAUSE.
STROP CASE OUT CLAUSE:
    STROP CASE OUT TOKEN, SERIAL CLAUSE;
    STROP CASE AGAIN TOKEN, STROP CASE CHOOSER CLAUSE.
BRIEF CASE OUT CLAUSE:
    BRIEF CASE OUT TOKEN, SERIAL CLAUSE;
    BRIEF CASE AGAIN TOKEN, BRIEF CASE CHOOSER CLAUSE.
STROP CONFORMITY OUT CLAUSE:
    STROP CONFORMITY OUT TOKEN, SERIAL CLAUSE;
    BRIEF CONFORMITY AGAIN TOKEN, BRIEF CONFORMITY CHOOSER CLAUSE.
BRIEF CONFORMITY OUT CLAUSE:
    BRIEF CONFORMITY OUT TOKEN, SERIAL CLAUSE;
    STROP CONFORMITY AGAIN TOKEN, STROP CONFORMITY CHOOSER CLAUSE.



[/, DECLARATIONS]

SINGLE DECLARATION:
    UNITARY DECLARATION LIST.
UNITARY DECLARATION LIST:
    UNITARY DECLARATION, (COMMA TOKEN, UNITARY DECLARATION LIST).

UNITARY DECLARATION:
    MODE DECLARATION; PRIORITY DECLARATION;
    IDENTIFIER DECLARATION; OPERATION DECLARATION.


DECLARER:
    NON PROC DECLARER; PROCEDURF DECLARATOR.
NON PROC DECLARER:
    MODE INDICATION; STRUCTURED WITH FIELDS DECLARATOR;
    REFERENCE TO DECLARATOR; ROWS OF DECLARATOR;
    UNION OF DECLARATOR.

STRUCTURED WITH FIELDS DECLARATOR:
    STRUCTURE TOKEN, PORTRAYER PACK.
PORTRAYER PACK:
    OPEN TOKEN, PORTRAYER, CLOSE TOKEN.
PORTRAYER:
    DECLARER, CONTINUATION.
CONTINUATION:
    IDENTIFIER, (COMMA TOKEN, CONTINUATION OR PORTRAYER).
CONTINUATION OR PORTRAYER:
    CONTINUATION; PORTRAYER.
```

```
REFERENCE TO DECLARATOR:
    REFERENCE TO TOKEN, DECLARER.

ROWS OF DECLARATOR:
    FOLLOWS ROWER, (FLEITHER), ROWER BRACKET, DECLARER.
FLEITHER:
    FLEXIBLE TOKEN; EITHER TOKEN.
ROWER BRACKET:
    SUB TOKEN, ROWER, BUS TOKEN;
    ALTERNATE SUB TOKEN, ROWER, ALTERNATE BUS TOKEN;
    OPEN TOKEN, ROWER, CLOSE TOKEN.
ROWER:
    ROW ROWER, (COMMA TOKEN, ROWER).
ROW ROWER:
    LOWPER BOUND, (UP TO TOKEN, LOWPER BOUND); (UP TO TOKEN).
LOWPER BOUND:
    UNIT.

PROCEDURE DECLARATOR:
    PROCEDURE TOKEN, PROCEDURE PLAN.
PROCEDURE PLAN:
    (DECLARER LIST PACK), MOID DECLARER.
DECLARER LIST PACK:
    OPEN TOKEN, DECLARER LIST, CLOSE TOKEN,
DECLARER LIST:
    DECLARER, (COMMA TOKEN, DECLARER LIST).
MOID DECLARER:
    VOID TOKEN; DECLARER.

UNION OF DECLARATOR:
    UNION OF TOKEN, DECLARER LIST PACK.


MODE DECLARATION:
    MODE TOKEN, MODE DEFINITION LIST.
MODE DEFINITION LIST:
    MODE DEFINITION, (COMMA TOKEN, MODE DEFINITION LIST).
MODE DEFINITION:
    MODE INDICATION, IS DEFINED AS TOKEN, DECLARER.

PRIORITY DECLARATION:
    PRIORITY TOKEN, PRIORITY DEFINITION LIST.
PRIORITY DEFINITION LIST:
    PRIORITY DEFINITION, (COMMA TOKEN, PRIORITY DEFINITION LIST).
PRIORITY DEFINITION:
    DYADIC INDICATION, IS DEFINED AS TOKEN, DIGIT CYPHER,

IDENTIFIER DECLARATION:
    IDENTITY DECLARATION; VARIABLE DECLARATION.

IDENTITY DECLARATION:
    FOLLOWS IDENTITY DECL, REST IDENTITY DECL.
REST IDENTITY DECL:
    PROCEDURE TOKEN, AFTER PROC IDENTITY DECL;
    NON PROC DECLARER, MODE IDENTITY DEFINITION LIST.
AFTER PROC IDENTITY DECL:
    PROCEDURE PLAN, MODE IDENTITY DEFINITION LIST;
    IDENTIFIER, IS DEFINED AS TOKEN, ROUTINE TEXT.
```

```
VARIABLE DECLARATION:
    FOLLOWS VARIABLE DECL, (LEAP TOKEN), REST VARIABLE DECL.
REST VARIABLE DECL:
    PROCEDURE TOKEN, AFTER PROC VARIABLE DECL;
    NON PROC DECLARER, IDENTIFIER, (MODE INITIALIZATION),
    (COMMA TOKEN, MODE VARIABLE DEFINITION LIST).
AFTER PROC VARIABLE DECL:
    PROCEDURE PLAN, IDENTIFIER, (MODE INITIALIZATION),
    (COMMA TOKEN, MODE VARIABLE DEFINITION LIST);
    IDENTIFIER, BECOMES TOKEN, ROUTINE TEXT.

MODE IDENTITY DEFINITION LIST:
    MODE IDENTITY DEFINITION,
    (COMMA TOKEN, MODE IDENTITY DEFINITION LIST).
MODE IDENTITY DEFINITION:
    IDENTIFIER, IS DEFINED AS TOKEN, UNIT.
MODE VARIABLE DEFINITION LIST:
    MODE VARIABLE DEFINITION,
    (COMMA TOKEN, MODE VARIABLE DEFINITION LIST).
MODE VARIABLE DEFINITION:
    IDENTIFIER, (MODE INITIALIZATION).
MODE INITIALIZATION:
    BECOMES TOKEN, SOURCE.
LEAP TOKEN:
    LOCAL TOKEN; HEAP TOKEN.

OPERATION DECLARATION:
    OPERATION TOKEN, OPERATION DEFINITION.
OPERATION DEFINITION:
    OPERATOR, IS DEFINED AS TOKEN, PRAM ROUTINE TEXT;
    VIRTUAL PRAM PLAN, OPERATOR, IS DEFINED AS TOKEN, UNIT.
OPERATOR:
    MONADIC OPERATOR; DYADIC OPERATOR.
MONADIC OPERATOR:
    MONADIC INDICATION.
DYADIC OPERATOR:
    DYADIC INDICATION.
PRAM ROUTINE TEXT:
    FOLLOWS ROUTINE, FORMAL PRAM PLAN, ROUTINE TOKEN, UNIT.
VIRTUAL PRAM PLAN:
    VIRTUAL PRAM PACK, MOID DECLARER.
VIRTUAL PRAM PACK:
    OPEN TOKEN, DECLARER, (COMMA TOKEN, DECLARER), CLOSE TOKEN.
FORMAL PRAM PLAN:
    FORMAL PRAM PACK, MOID DECLARER.
FORMAL PRAM PACK:
    OPEN TOKEN, DECLARER, IDENTIFIER,
    (COMMA TOKEN, SECOND FORMAL PARAMETER), CLOSE TOKEN.
SECOND FORMAL PARAMETER:
    (DECLARER), IDENTIFIER.



[8. UNITARY CLAUSES.]

UNIT:
    LOOP; ROUTINE TEXT; ASSIGNATION; IDENTITY RELATION; TERTIARY.
TERTIARY:
    FORMULA; SECONDARY.
```

```
FORMULA:
    FOLLOWS FORMULA, REST FORMULA.
REST FORMULA:
    MONADIC OPERATOR, OPERAND;
    OPERAND, DYADIC OPERATOR, OPERAND,
OPERAND:
    TERTIARY,
SECONDARY:
    LEAP GENERATOR; SELECTION; PRIMARY.
PRIMARY:
    NON CALL NON SLICE PRIMARY,
    (INDEXER BRACKET OR ACTUAL PARAMETERS PACK SEQUENCE),


LOOP:
    (FOR PART), (FROM PART), (BY PART), (TO PART), (WHILE PART),
    DO PART.
FOR PART:
    FOR TOKEN, IDENTIFIER.
FROM PART:
    FROM TOKEN, UNIT.
BY PART:
    BY TOKEN, UNIT.
TO PART:
    TO TOKEN, UNIT.
WHILE PART:
    WHILE TOKEN, SERIAL CLAUSE.
DO PART:
    DO TOKEN, UNIT.

ROUTINE TEXT:
    FOLLOWS ROUTINE, FORMAL PROCEDURE PLAN, ROUTINE TOKEN, UNIT.
FORMAL PROCEDURE PLAN:
    (FORMAL PARAMETERS PACK), MOID DECLARER.
FORMAL PARAMETERS PACK:
    OPEN TOKEN, FORMAL PARAMETERS, CLOSE TOKEN.
FORMAL PARAMETERS:
    DECLARER, DEFINITION.
DEFINITION:
    IDENTIFIER, (COMMA TOKEN, DEFINITION OR PARAMETERS).
DEFINITION OR PARAMETERS:
    DEFINITION; FORMAL PARAMETERS.

ASSIGNATION:
    FOLLOWS ASSIGNATION, TERTIARY, BECOMES TOKEN, SOURCE,
SOURCE:
    UNIT.

IDENTITY RELATION:
    FOLLOWS IDENTITY RELATION,
    TERTIARY, IDENTITY RELATOR, TERTIARY.
IDENTITY RELATOR:
    IS TOKEN; IS NOT TOKEN.

LEAP GENERATOR:
    LEAP TOKEN, DECLARER.

SELECTION:
    IDENTIFIER, OF TOKEN, SECONDARY.
```

```
NON CALL NON SLICE PRIMARY:
    DENOTATION; IDENTIFIER; UNAMBIGUOUS HIP; CAST; ENCLOSED CLAUSE.
INDEXER BRACKET OR ACTUAL PARAMETERS PACK SEQUENCE:
    INDEXER BRACKET OR ACTUAL PARAMETERS PACK,
    (INDEXER BRACKET OR ACTUAL PARAMETERS PACK SEQUENCE).
INDEXER BRACKET OR ACTUAL PARAMETERS PACK:
    OPEN TOKEN, INDEXER OR PARAMETERS, CLOSE TOKEN;
    SUB TOKEN, INDEXER, BUS TOKEN;
    ALTERNATE SUB TOKEN, INDEXER, ALTERNATE BUS TOKEN.
INDEXER OR PARAMETERS:
    INDEXER.
INDEXER:
    TRIMSCRIPT, (COMMA TOKEN, INDEXER).
TRIMSCRIPT:
    UNIT, (FROM UP TO TOKEN);
    FROM UP TO TOKEN;
    (AT TOKEN, LOWPER BOUND).
FROM UP TO TOKEN:
    UP TO TOKEN, (LOWPER BOUND), (AT TOKEN, LOWPER BOUND).

UNAMBIGUOUS HIP:
    SKIP TOKEN; GO TO TOKEN, IDENTIFIER; NIL TOKEN;
    OPEN TOKEN, CLOSE TOKEN.

CAST:
    FOLLOWS CAST, MOID DECLARER, ENCLOSED CLAUSE.
```

5.2. Messages

```
THE FOLLOWING NOTIONS MAY PRODUCE EMPTY:

ROWER
ROWROWER
INDEXERORPARAMETERS
INDEXER
TRIMSCRIPT




NO RULE IS LEFT-RECURSIVE



FOR THE FOLLOWING NOTIONS, MORE THAN ONE ALTERNATIVE MAY
     START WITH A GIVEN NOTION:


SERIALCLAUSE-
        IDENTIFIER

LABELLEDUNIT-
        IDENTIFIER

SECONDARY-
        IDENTIFIER
```

```
VIOLATIONS OF REQUIREMENT 3:


IN LABELLEDUNIT THE POSSIBLY EMPTY OR OPTIONAL NOTION LABELSEQUENCE
     MAY BE FOLLOWED BY UNIT;
     BOTH LABELSEQUENCE AND UNIT MAY START WITH IDENTIFIER

THE NOTION LABELSEQUENCE IS THE BEGINNING OF THE POSSIBLE EMPTY
     OR OPTIONAL LAST MEMBER OF LABELSEQUENCE
     (VIA LABELSEQUENCE)
     AND MAY BE FOLLOWED BY UNIT;
     BOTH LABELSEQUENCE AND UNIT MAY BEGIN WITH IDENTIFIER

THE NOTION COMMATOKEN IS THE BEGINNING OF THE POSSIBLE EMPTY
     OR OPTIONAL LAST MEMBER OF UNITARYDECLARATION
     (VIA MODEDEFINITIONLIST)
     AND MAY BE FOLLOWED BY COMMATOKEN;
     BOTH COMMATOKEN AND COMMATOKEN MAY BEGIN WITH COMMATOKEN

THE NOTION COMMATOKEN IS THE BEGINNING OF THE POSSIBLE EMPTY
     OR OPTIONAL LAST MEMBER OF UNITARYDECLARATION
     (VIA PRIORITYDEFINITIONLIST)
     AND MAY BE FOLLOWED BY COMMATOKEN;
     BOTH COMMATOKEN AND COMMATOKEN MAY BEGIN WITH COMMATOKEN

THE NOTION COMMATOKEN IS THE BEGINNING OF THE POSSIBLE EMPTY
     OR OPTIONAL LAST MEMBER OF UNITARYDECLARATION
     (VIA RESTVARIABLEDECL)
     AND MAY BE FOLLOWED BY COMMATOKEN;
     BOTH COMMATOKEN AND COMMATOKEN MAY BEGIN WITH COMMATOKEN

THE NOTION COMMATOKEN IS THE BEGINNING OF THE POSSIBLE EMPTY
     OR OPTIONAL LAST MEMBER OF UNITARYDECLARATION
     (VIA AFTERPROCVARIABLEDECL)
     AND MAY BE FOLLOWED BY COMMATOKEN;
     BOTH COMMATOKEN AND COMMATOKEN MAY BEGIN WITH COMMATOKEN

THE NOTION COMMATOKEN IS THE BEGINNING OF THE POSSIBLE EMPTY
     OR OPTIONAL LAST MEMBER OF UNITARYDECLARATION
     (VIA MODEIDENTITYDEFINITIONLIST)
     AND MAY BE FOLLOWED BY COMMATOKEN;
     BOTH COMMATOKEN AND COMMATOKEN MAY BEGIN WITH COMMATOKEN

THE NOTION COMMATOKEN IS THE BEGINNING OF THE POSSIBLE EMPTY
     OR OPTIONAL LAST MEMBER OF UNITARYDECLARATION
     (VIA MODEVARIABLEDEFINITIONLIST)
     AND MAY BE FOLLOWED BY COMMATOKEN;
     BOTH COMMATOKEN AND COMMATCKEN MAY BEGIN WITH COMMATOKEN
```

References

[1] A. van Wijngaarden (Editor), "Report on the Algorithmic Language ALGOL 68", Numerische Mathematik, 14 (1969).

[2] A. van Wijngaarden (Editor), "Report on the Algorithmic Language ALGOL 68", Preliminary Draft version, presented during the Vienna meeting of W.G.2.1 of IFIP, August 1972.

[3] P. Branquart, J.P. Cardinael, J.P. Delescaille and J. Lewi, "A context-free syntax of ALGOL 68", Information Processing Letters 1 (1972), p. 141-148, North Holland Publishing Company.

[4] H.J. Bowlden, "ALGOL 68 structural flowchart", Westinghouse Research Laboratories, Research Report 69-1c4-comps-R2, October 1969.

[5] M. Simonet, "Une grammaire context-free d'ALGOL 68", Rapport IMAG, Grenoble, June 1969.

[6] G.S. Hodgson, "ALGOL 68 extended syntax", University of Manchester, March 1970.

[7] D.E. Knuth, "Top-down syntax analysis", Acta Informatica 1 (1971), p. 79-110, Springer-Verlag, Berlin.

[8] D. Gries, "Compiler Construction for digital computers", Wiley, New York, 1971.

[9] S. Warshall, "A theorem on Boolean matrices", J. ACM 9, January 1962, p. 11-12.

[10] L.G.L.Th. Meertens, J.C. van Vliet, "Repairing the parenthesis skeleton of ALGOL 68 programs", Report IW 2/73, Mathematical Centre, Amsterdam, February 1973.

Appendix

```
MAY CONTAIN  (MC): AB - A; B.


  MC1 LETTER A TOKEN, LETTER B TOKEN, LETTER C TOKEN,
      LETTER D TOKEN, LETTER E TOKEN, LETTER F TOKEN,
      LETTER G TOKEN, LETTER I TOKEN, LETTER K TOKEN,
      LETTER L TOKEN, LETTER N TOKEN, LETTER P TOKEN,
      LETTER R TOKEN, LETTER S TOKEN, LETTER T TOKEN,
      LETTER X TOKEN, LETTER Y TOKEN, LETTER Z TOKEN,
      DIGIT TWO TOKEN, DIGIT FOUR TOKEN, DIGIT EIGHT TOKEN,
      DIGIT ONE TOKEN, DIGIT SIX TOKEN, PLUS TOKEN,
      MINUS TOKEN, POINT TOKEN, OPEN TOKEN, CLOSE TOKEN,
      BEGIN TOKEN, COMMA TOKEN, END TOKEN,
      STROP CASE START TOKEN, STROP CASE FINISH TOKEN,
      STROP CONDITION START TOKEN,
      STROP CONDITION FINISH TOKEN, INNER CLAUSE,
      INTEGRAL DENOTATION, CHARACTER DENOTATION,
      ROW OF CHARACTER DENOTATION-

  MC2 COLLECTION LIST+, COLLECTION+, COLLECTION LIST PACK*-
    (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
         LETTER C TOKEN; LETTER D TOKEN; LETTER E TOKEN;
         LETTER F TOKEN; LETTER G TOKEN; LETTER I TOKEN;
         LETTER K TOKEN; LETTER L TOKEN; LETTER N TOKEN;
         LETTER P TOKEN; LETTER R TOKEN; LETTER S TOKEN;
         LETTER T TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
         LETTER Z TOKEN; DIGIT TWO TOKEN; DIGIT FOUR TOKEN;
         DIGIT EIGHT TOKEN; DIGIT ONE TOKEN; DIGIT SIX TOKEN;
         PLUS TOKEN; MINUS TOKEN; POINT TOKEN; OPEN TOKEN*;
         CLOSE TOKEN*; BEGIN TOKEN; COMMA TOKEN*; END TOKEN;
         STROP CASE START TOKEN; STROP CASE FINISH TOKEN;
         STROP CONDITION START TOKEN;
         STROP CONDITION FINISH TOKEN; INNER CLAUSE;
         INTEGRAL DENOTATION; CHARACTER DENOTATION;
         ROW OF CHARACTER DENOTATION,
    (NONTERMINALS:) COLLECTION LIST*; COLLECTION*; PICTURE*;
         INSERTION*; REPLICATOR*; COLLECTION LIST PACK*;
         PATTERN; LITERAL; INSERT SEQUENCE; INSERT; ALIGNMENT;
         REPLICATION; DYNAMIC REPLICATION; ENCLOSED CLAUSE;
         STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
         REPLICATED LITERAL; SIGN MOULD; SIGN FRAME;
         INTEGRAL PATTERN; REAL PATTERN; BOOLEAN PATTERN;
         CHARACTER PATTERN; COMPLEX PATTERN; STRING PATTERN;
         BITS PATTERN; GENERAL PATTERN; INCLUDED PATTERN;
         INTEGRAL MOULD; INTEGRAL CHOICE PATTERN; DIGIT FRAME;
         LITERAL LIST PACK; LITERAL LIST; REAL MOULD;
         FLOATING POINT MOULD; STAGNANT MOULD;
         BOOLEAN CHOICE MOULD;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
         RADIX MOULD; RADIX; ACTUAL SPECIFICATION PACK;
         ACTUAL SPECIFICATION LIST; ACTUAL SPECIFICATION.

  MC3 PICTURE-
    (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
         LETTER C TOKEN; LETTER D TOKEN; LETTER E TOKEN;
         LETTER F TOKEN; LETTER G TOKEN; LETTER I TOKEN;
         LETTER K TOKEN; LETTER L TOKEN; LETTER N TOKEN;
```

```
MAY CONTAIN  (MC): AB - A; B.

          LETTER P TOKEN; LETTER R TOKEN; LETTER S TOKEN;
          LETTER T TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
          LETTER Z TOKEN; DIGIT TWO TOKEN; DIGIT FOUR TOKEN;
          DIGIT EIGHT TOKEN; DIGIT ONE TOKEN; DIGIT SIX TOKEN;
          PLUS TOKEN; MINUS TOKEN; POINT TOKEN; OPEN TOKEN;
          CLOSE TOKEN; BEGIN TOKEN; COMMA TOKEN; END TOKEN;
          STROP CASE START TOKEN; STROP CASE FINISH TOKEN;
          STROP CONDITION START TOKEN;
          STROP CONDITION FINISH TOKEN; INNER CLAUSE;
          INTEGRAL DENOTATION; CHARACTER DENOTATION;
          ROW OF CHARACTER DENOTATION.
     (NONTERMINALS:) INSERTION*; REPLICATOR; PATTERN*;
          LITERAL; INSERT SEQUENCE; INSERT; ALIGNMENT;
          REPLICATION; DYNAMIC REPLICATION; ENCLOSED CLAUSE;
          STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
          REPLICATED LITERAL; SIGN MOULD; SIGN FRAME;
          INTEGRAL PATTERN; REAL PATTERN; BOOLEAN PATTERN;
          CHARACTER PATTERN; COMPLEX PATTERN; STRING PATTERN;
          BITS PATTERN; GENERAL PATTERN; INCLUDED PATTERN;
          INTEGRAL MOULD; INTEGRAL CHOICE PATTERN; DIGIT FRAME;
          LITERAL LIST PACK; LITERAL LIST; REAL MOULD;
          FLOATING POINT MOULD; STAGNANT MOULD;
          BOOLEAN CHOICE MOULD;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
          RADIX MOULD; RADIX; ACTUAL SPECIFICATION PACK;
          ACTUAL SPECIFICATION LIST; ACTUAL SPECIFICATION.

   MC4 INSERTION, INSERT SEQUENCE+-
     (TERMINALS:) LETTER K TOKEN; LETTER L TOKEN;
          LETTER N TOKEN; LETTER P TOKEN; LETTER X TOKEN;
          LETTER Y TOKEN; OPEN TOKEN; CLOSE TOKEN; BEGIN TOKEN;
          END TOKEN; STROP CASE START TOKEN;
          STROP CASE FINISH TOKEN; STROP CONDITION START TOKEN;
          STROP CONDITION FINISH TOKEN; INNER CLAUSE;
          INTEGRAL DENOTATION; CHARACTER DENOTATION;
          ROW OF CHARACTER DENOTATION.
     (NONTERMINALS:) REPLICATOR; LITERAL*; INSERT SEQUENCE*;
          INSERT*; ALIGNMENT; REPLICATION; DYNAMIC REPLICATION;
          ENCLOSED CLAUSE; STRING DENOTATION;
          REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL.

   MC5 REPLICATOR-
     (TERMINALS:) LETTER N TOKEN; OPEN TOKEN; CLOSE TOKEN;
          BEGIN TOKEN; END TOKEN; STROP CASE START TOKEN;
          STROP CASE FINISH TOKEN; STROP CONDITION START TOKEN;
          STROP CONDITION FINISH TOKEN; INNER CLAUSE;
          INTEGRAL DENOTATION.
     (NONTERMINALS:) REPLICATION*; DYNAMIC REPLICATION;
          ENCLOSED CLAUSE.

   MC6 PATTERN-
     (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
          LETTER C TOKEN; LETTER D TOKEN; LETTER E TOKEN;
          LETTER F TOKEN; LETTER G TOKEN; LETTER I TOKEN;
          LETTER K TOKEN; LETTER L TOKEN; LETTER N TOKEN;
          LETTER P TOKEN; LETTER R TOKEN; LETTER S TOKEN;
```

```
MAY CONTAIN  (MC): AB - A; B.

          LETTER T TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
          LETTER Z TOKEN; DIGIT TWO TOKEN; DIGIT FOUR TOKEN;
          DIGIT EIGHT TOKEN; DIGIT ONE TOKEN; DIGIT SIX TOKEN;
          PLUS TOKEN; MINUS TOKEN; POINT TOKEN; OPEN TOKEN;
          CLOSE TOKEN; BEGIN TOKEN; COMMA TOKEN; END TOKEN;
          STROP CASE START TOKEN; STROP CASE FINISH TOKEN;
          STROP CONDITION START TOKEN;
          STROP CONDITION FINISH TOKEN; INNER CLAUSE;
          INTEGRAL DENOTATION; CHARACTER DENOTATION;
          ROW OF CHARACTER DENOTATION.
      (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
          INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
          DYNAMIC REPLICATION; ENCLOSED CLAUSE;
          STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
          REPLICATED LITERAL; SIGN MOULD; SIGN FRAME;
          INTEGRAL PATTERN*; REAL PATTERN*; BOOLEAN PATTERN*;
          CHARACTER PATTERN*; COMPLEX PATTERN*; STRING PATTERN*;
          BITS PATTERN*; GENERAL PATTERN*; INCLUDED PATTERN*;
          INTEGRAL MOULD; INTEGRAL CHOICE PATTERN; DIGIT FRAME;
          LITERAL LIST PACK; LITERAL LIST; REAL MOULD;
          FLOATING POINT MOULD; STAGNANT MOULD;
          BOOLEAN CHOICE MOULD;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
          RADIX MOULD; RADIX; ACTUAL SPECIFICATION PACK;
          ACTUAL SPECIFICATION LIST; ACTUAL SPECIFICATION.

   MC7 LITERAL-
      (TERMINALS:) LETTER N TOKEN; OPEN TOKEN; CLOSE TOKEN;
          BEGIN TOKEN; END TOKEN; STROP CASE START TOKEN;
          STROP CASE FINISH TOKEN; STROP CONDITION START TOKEN;
          STROP CONDITION FINISH TOKEN; INNER CLAUSE;
          INTEGRAL DENOTATION; CHARACTER DENOTATION;
          ROW OF CHARACTER DENOTATION.
      (NONTERMINALS:) REPLICATOR*; REPLICATION;
          DYNAMIC REPLICATION; ENCLOSED CLAUSE;
          STRING DENOTATION*; REPLICATED LITERAL SEQUENCE*;
          REPLICATED LITERAL.

   MC8 INSERT-
      (TERMINALS:) LETTER K TOKEN; LETTER L TOKEN;
          LETTER N TOKEN; LETTER P TOKEN; LETTER X TOKEN;
          LETTER Y TOKEN; OPEN TOKEN; CLOSE TOKEN; BEGIN TOKEN;
          END TOKEN; STROP CASE START TOKEN;
          STROP CASE FINISH TOKEN; STROP CONDITION START TOKEN;
          STROP CONDITION FINISH TOKEN; INNER CLAUSE;
          INTEGRAL DENOTATION; CHARACTER DENOTATION;
          ROW OF CHARACTER DENOTATION.
      (NONTERMINALS:) REPLICATOR*; LITERAL*; ALIGNMENT*;
          REPLICATION; DYNAMIC REPLICATION; ENCLOSED CLAUSE;
          STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
          REPLICATED LITERAL.

   MC9 ALIGNMENT-
      (TERMINALS:) LETTER K TOKEN*; LETTER L TOKEN*;
          LETTER P TOKEN*; LETTER X TOKEN*; LETTER Y TOKEN*.
```

```
MAY CONTAIN  (MC): AB - A; B.

 MC10 REPLICATION-
    (TERMINALS:) LETTER N TOKEN; OPEN TOKEN; CLOSE TOKEN;
        BEGIN TOKEN; END TOKEN; STROP CASE START TOKEN;
        STROP CASE FINISH TOKEN; STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION*.
    (NONTERMINALS:) DYNAMIC REPLICATION*; ENCLOSED CLAUSE.

 MC11 DYNAMIC REPLICATION-
    (TERMINALS:) LETTER N TOKEN*; OPEN TOKEN; CLOSE TOKEN;
        BEGIN TOKEN; END TOKEN; STROP CASE START TOKEN;
        STROP CASE FINISH TOKEN; STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE.
    (NONTERMINALS:) ENCLOSED CLAUSE*.

 MC12 ENCLOSED CLAUSE-
    (TERMINALS:) OPEN TOKEN*; CLOSE TOKEN*; BEGIN TOKEN*;
        END TOKEN*; STROP CASE START TOKEN*;
        STROP CASE FINISH TOKEN*;
        STROP CONDITION START TOKEN*;
        STROP CONDITION FINISH TOKEN*; INNER CLAUSE*.

 MC13 STRING DENOTATION-
    (TERMINALS:) CHARACTER DENOTATION*;
        ROW OF CHARACTER DENOTATION*.

 MC14 REPLICATED LITERAL SEQUENCE+-
    (TERMINALS:) LETTER N TOKEN; OPEN TOKEN; CLOSE TOKEN;
        BEGIN TOKEN; END TOKEN; STROP CASE START TOKEN;
        STROP CASE FINISH TOKEN; STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) REPLICATION; DYNAMIC REPLICATION;
        ENCLOSED CLAUSE; STRING DENOTATION;
        REPLICATED LITERAL SEQUENCE*; REPLICATED LITERAL*.

 MC15 REPLICATED LITERAL-
    (TERMINALS:) LETTER N TOKEN; OPEN TOKEN; CLOSE TOKEN;
        BEGIN TOKEN; END TOKEN; STROP CASE START TOKEN;
        STROP CASE FINISH TOKEN; STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) REPLICATION*; DYNAMIC REPLICATION;
        ENCLOSED CLAUSE; STRING DENOTATION*.

 MC16 SIGN MOULD-
    (TERMINALS:) LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER X TOKEN;
        LETTER Y TOKEN; LETTER Z TOKEN*; PLUS TOKEN;
        MINUS TOKEN; OPEN TOKEN; CLOSE TOKEN; BEGIN TOKEN;
        END TOKEN; STROP CASE START TOKEN;
        STROP CASE FINISH TOKEN; STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
```

```
MAY CONTAIN  (MC): AB - A; B.

    (NONTERMINALS:) INSERTION*; REPLICATOR*; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; SIGN FRAME*.

 MC17 SIGN FRAME-
    (TERMINALS:) PLUS TOKEN*; MINUS TOKEN*.

 MC18 INTEGRAL PATTERN-
    (TERMINALS:) LETTER C TOKEN; LETTER D TOKEN;
        LETTER K TOKEN; LETTER L TOKEN; LETTER N TOKEN;
        LETTER P TOKEN; LETTER S TOKEN; LETTER X TOKEN;
        LETTER Y TOKEN; LETTER Z TOKEN; PLUS TOKEN;
        MINUS TOKEN; OPEN TOKEN; CLOSE TOKEN; BEGIN TOKEN;
        COMMA TOKEN; END TOKEN; STROP CASE START TOKEN;
        STROP CASE FINISH TOKEN; STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; SIGN MOULD*; SIGN FRAME;
        INTEGRAL MOULD*; INTEGRAL CHOICE PATTERN*;
        DIGIT FRAME; LITERAL LIST PACK; LITERAL LIST.

 MC19 REAL PATTERN-
    (TERMINALS:) LETTER D TOKEN; LETTER E TOKEN;
        LETTER K TOKEN; LETTER L TOKEN; LETTER N TOKEN;
        LETTER P TOKEN; LETTER S TOKEN; LETTER X TOKEN;
        LETTER Y TOKEN; LETTER Z TOKEN; PLUS TOKEN;
        MINUS TOKEN; POINT TOKEN; OPEN TOKEN; CLOSE TOKEN;
        BEGIN TOKEN; END TOKEN; STROP CASE START TOKEN;
        STROP CASE FINISH TOKEN; STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; SIGN MOULD*; SIGN FRAME;
        INTEGRAL MOULD; DIGIT FRAME; REAL MOULD*;
        FLOATING POINT MOULD*; STAGNANT MOULD.

 MC20 BOOLEAN PATTERN-
    (TERMINALS:) LETTER B TOKEN*; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER X TOKEN; LETTER Y TOKEN; OPEN TOKEN;
        CLOSE TOKEN; BEGIN TOKEN; COMMA TOKEN; END TOKEN;
        STROP CASE START TOKEN; STROP CASE FINISH TOKEN;
        STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
```

```
MAY CONTAIN  (MC): AB - A; B.

    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; BOOLEAN CHOICE MOULD*.

 MC21 CHARACTER PATTERN,
     LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME-
    (TERMINALS:) LETTER A TOKEN*; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN*; LETTER X TOKEN; LETTER Y TOKEN;
        OPEN TOKEN; CLOSE TOKEN; BEGIN TOKEN; END TOKEN;
        STROP CASE START TOKEN; STROP CASE FINISH TOKEN;
        STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR*; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL.

 MC22 COMPLEX PATTERN-
    (TERMINALS:) LETTER D TOKEN; LETTER E TOKEN;
        LETTER I TOKEN*; LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER S TOKEN*;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
        PLUS TOKEN; MINUS TOKEN; POINT TOKEN; OPEN TOKEN;
        CLOSE TOKEN; BEGIN TOKEN; END TOKEN;
        STROP CASE START TOKEN; STROP CASE FINISH TOKEN;
        STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; SIGN MOULD; SIGN FRAME;
        REAL PATTERN*; INTEGRAL MOULD; DIGIT FRAME;
        REAL MOULD; FLOATING POINT MOULD; STAGNANT MOULD.

 MC23 STRING PATTERN-
    (TERMINALS:) LETTER A TOKEN*; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN*; LETTER T TOKEN*; LETTER X TOKEN;
        LETTER Y TOKEN; OPEN TOKEN; CLOSE TOKEN; BEGIN TOKEN;
        END TOKEN; STROP CASE START TOKEN;
        STROP CASE FINISH TOKEN; STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION*;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
```

```
MAY CONTAIN  (MC): AB - A; B.

        REPLICATED LITERAL;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE*.

 MC24 BITS PATTERN-
    (TERMINALS:) LETTER D TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER R TOKEN; LETTER S TOKEN; LETTER X TOKEN;
        LETTER Y TOKEN; LETTER Z TOKEN; DIGIT TWO TOKEN;
        DIGIT FOUR TOKEN; DIGIT EIGHT TOKEN; DIGIT ONE TOKEN;
        DIGIT SIX TOKEN; OPEN TOKEN; CLOSE TOKEN;
        BEGIN TOKEN; END TOKEN; STROP CASE START TOKEN;
        STROP CASE FINISH TOKEN; STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; INTEGRAL MOULD*; DIGIT FRAME;
        RADIX MOULD*; RADIX.

 MC25 GENERAL PATTERN-
    (TERMINALS:) LETTER G TOKEN*; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER X TOKEN; LETTER Y TOKEN; MINUS TOKEN;
        OPEN TOKEN; CLOSE TOKEN; BEGIN TOKEN; COMMA TOKEN;
        END TOKEN; STROP CASE START TOKEN;
        STROP CASE FINISH TOKEN; STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; ACTUAL SPECIFICATION PACK*;
        ACTUAL SPECIFICATION LIST; ACTUAL SPECIFICATION.

 MC26 INCLUDED PATTERN-
    (TERMINALS:) LETTER F TOKEN*; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER X TOKEN; LETTER Y TOKEN; OPEN TOKEN;
        CLOSE TOKEN; BEGIN TOKEN; END TOKEN;
        STROP CASE START TOKEN; STROP CASE FINISH TOKEN;
        STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE*;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL.

 MC27 INTEGRAL MOULD*-
```

```
MAY CONTAIN  (MC): AB - A; B.

    (TERMINALS:) LETTER D TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN*; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; OPEN TOKEN; CLOSE TOKEN; BEGIN TOKEN;
        END TOKEN; STROP CASE START TOKEN;
        STROP CASE FINISH TOKEN; STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR*; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; INTEGRAL MOULD*; DIGIT FRAME*.

 MC28 INTEGRAL CHOICE PATTERN-
    (TERMINALS:) LETTER C TOKEN*; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER X TOKEN; LETTER Y TOKEN; OPEN TOKEN;
        CLOSE TOKEN; BEGIN TOKEN; COMMA TOKEN; END TOKEN;
        STROP CASE START TOKEN; STROP CASE FINISH TOKEN;
        STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; LITERAL LIST PACK*; LITERAL LIST.

 MC29 DIGIT FRAME-
    (TERMINALS:) LETTER D TOKEN*; LETTER Z TOKEN*.

 MC30 LITERAL LIST PACK, LITERAL LIST*-
    (TERMINALS:) LETTER N TOKEN; OPEN TOKEN*; CLOSE TOKEN*;
        BEGIN TOKEN; COMMA TOKEN*; END TOKEN;
        STROP CASE START TOKEN; STROP CASE FINISH TOKEN;
        STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) REPLICATOR; LITERAL*; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; LITERAL LIST*.

 MC31 REAL MOULD-
    (TERMINALS:) LETTER D TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN*; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; POINT TOKEN*; OPEN TOKEN;
        CLOSE TOKEN; BEGIN TOKEN; END TOKEN;
        STROP CASE START TOKEN; STROP CASE FINISH TOKEN;
        STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
```

```
MAY CONTAIN  (MC): AB - A; B.

        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; INTEGRAL MOULD*; DIGIT FRAME.

 MC32 FLOATING POINT MOULD-
    (TERMINALS:) LETTER D TOKEN; LETTER E TOKEN*;
        LETTER K TOKEN; LETTER L TOKEN; LETTER N TOKEN;
        LETTER P TOKEN; LETTER S TOKEN*; LETTER X TOKEN;
        LETTER Y TOKEN; LETTER Z TOKEN; PLUS TOKEN;
        MINUS TOKEN; POINT TOKEN; OPEN TOKEN; CLOSE TOKEN;
        BEGIN TOKEN; END TOKEN; STROP CASE START TOKEN;
        STROP CASE FINISH TOKEN; STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; SIGN MOULD*; SIGN FRAME;
        INTEGRAL MOULD*; DIGIT FRAME; REAL MOULD;
        STAGNANT MOULD*.

 MC33 STAGNANT MOULD-
    (TERMINALS:) LETTER D TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; PLUS TOKEN; MINUS TOKEN; POINT TOKEN;
        OPEN TOKEN; CLOSE TOKEN; BEGIN TOKEN; END TOKEN;
        STROP CASE START TOKEN; STROP CASE FINISH TOKEN;
        STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; SIGN MOULD*; SIGN FRAME;
        INTEGRAL MOULD*; DIGIT FRAME; REAL MOULD*.

 MC34 BOOLEAN CHOICE MOULD-
    (TERMINALS:) LETTER N TOKEN; OPEN TOKEN*; CLOSE TOKEN*;
        BEGIN TOKEN; COMMA TOKEN*; END TOKEN;
        STROP CASE START TOKEN; STROP CASE FINISH TOKEN;
        STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) REPLICATOR; LITERAL*; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL.
```

```
MAY CONTAIN  (MC): AB - A; B.

 MC35
     LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE+-
    (TERMINALS:) LETTER A TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        OPEN TOKEN; CLOSE TOKEN; BEGIN TOKEN; END TOKEN;
        STROP CASE START TOKEN; STROP CASE FINISH TOKEN;
        STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE*.

 MC36 RADIX MOULD-
    (TERMINALS:) LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER R TOKEN*;
        LETTER X TOKEN; LETTER Y TOKEN; DIGIT TWO TOKEN;
        DIGIT FOUR TOKEN; DIGIT EIGHT TOKEN; DIGIT ONE TOKEN;
        DIGIT SIX TOKEN; OPEN TOKEN; CLOSE TOKEN;
        BEGIN TOKEN; END TOKEN; STROP CASE START TOKEN;
        STROP CASE FINISH TOKEN; STROP CONDITION START TOKEN;
        STROP CONDITION FINISH TOKEN; INNER CLAUSE;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; RADIX*.

 MC37 RADIX-
    (TERMINALS:) DIGIT TWO TOKEN*; DIGIT FOUR TOKEN*;
        DIGIT EIGHT TOKEN*; DIGIT ONE TOKEN*;
        DIGIT SIX TOKEN*.

 MC38 ACTUAL SPECIFICATION PACK-
    (TERMINALS:) MINUS TOKEN; OPEN TOKEN*; CLOSE TOKEN*;
        COMMA TOKEN; INTEGRAL DENOTATION.
    (NONTERMINALS:) ACTUAL SPECIFICATION LIST*;
        ACTUAL SPECIFICATION.

 MC39 ACTUAL SPECIFICATION LIST+-
    (TERMINALS:) MINUS TOKEN; COMMA TOKEN*;
        INTEGRAL DENOTATION.
    (NONTERMINALS:) ACTUAL SPECIFICATION LIST*;
        ACTUAL SPECIFICATION*.

 MC40 ACTUAL SPECIFICATION-
    (TERMINALS:) MINUS TOKEN*; INTEGRAL DENOTATION*.
```

MAY BE CONTAINED IN (CI): A, B - AB.

CI1 LETTER A TOKEN-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN; CHARACTER PATTERN*;
STRING PATTERN*;
LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE.

CI2 LETTER B TOKEN, BOOLEAN CHOICE MOULD-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN; BOOLEAN PATTERN*.

CI3 LETTER C TOKEN, LITERAL LIST PACK-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN; INTEGRAL PATTERN;
INTEGRAL CHOICE PATTERN*.

CI4 LETTER D TOKEN-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN; INTEGRAL PATTERN;
REAL PATTERN; COMPLEX PATTERN; BITS PATTERN;
INTEGRAL MOULD; DIGIT FRAME*; REAL MOULD;
FLOATING POINT MOULD; STAGNANT MOULD.

CI5 LETTER E TOKEN, STAGNANT MOULD-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN; REAL PATTERN;
COMPLEX PATTERN; FLOATING POINT MOULD*.

CI6 LETTER F TOKEN-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN; INCLUDED PATTERN*.

CI7 LETTER G TOKEN, ACTUAL SPECIFICATION PACK-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN; GENERAL PATTERN*.

CI8 LETTER I TOKEN, REAL PATTERN-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN*; COMPLEX PATTERN*.

CI9 LETTER K TOKEN, LETTER L TOKEN, LETTER P TOKEN,
LETTER X TOKEN, LETTER Y TOKEN-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
INSERTION; COLLECTION LIST PACK; PATTERN;
INSERT SEQUENCE; INSERT; ALIGNMENT*; SIGN MOULD;
INTEGRAL PATTERN; REAL PATTERN; BOOLEAN PATTERN;
CHARACTER PATTERN; COMPLEX PATTERN; STRING PATTERN;
BITS PATTERN; GENERAL PATTERN; INCLUDED PATTERN;
INTEGRAL MOULD; INTEGRAL CHOICE PATTERN; REAL MOULD;
FLOATING POINT MOULD; STAGNANT MOULD;
LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
RADIX MOULD.

CI10 LETTER N TOKEN, ENCLOSED CLAUSE-

```
MAY BE CONTAINED IN  (CI): A, B - AB.

    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        INSERTION; REPLICATOR; COLLECTION LIST PACK; PATTERN;
        LITERAL; INSERT SEQUENCE; INSERT; REPLICATION;
        DYNAMIC REPLICATION*; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; SIGN MOULD; INTEGRAL PATTERN;
        REAL PATTERN; BOOLEAN PATTERN; CHARACTER PATTERN;
        COMPLEX PATTERN; STRING PATTERN; BITS PATTERN;
        GENERAL PATTERN; INCLUDED PATTERN*; INTEGRAL MOULD;
        INTEGRAL CHOICE PATTERN; LITERAL LIST PACK;
        LITERAL LIST; REAL MOULD; FLOATING POINT MOULD;
        STAGNANT MOULD; BOOLEAN CHOICE MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
        RADIX MOULD.

 CI11 LETTER R TOKEN, RADIX-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        COLLECTION LIST PACK; PATTERN; BITS PATTERN;
        RADIX MOULD*.

 CI12 LETTER S TOKEN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        COLLECTION LIST PACK; PATTERN; INTEGRAL PATTERN;
        REAL PATTERN; CHARACTER PATTERN*; COMPLEX PATTERN*;
        STRING PATTERN*; BITS PATTERN; INTEGRAL MOULD*;
        REAL MOULD*; FLOATING POINT MOULD*; STAGNANT MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE.

 CI13 LETTER T TOKEN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        COLLECTION LIST PACK; PATTERN; STRING PATTERN*.

 CI14 LETTER Z TOKEN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        COLLECTION LIST PACK; PATTERN; SIGN MOULD*;
        INTEGRAL PATTERN; REAL PATTERN; COMPLEX PATTERN;
        BITS PATTERN; INTEGRAL MOULD; DIGIT FRAME*;
        REAL MOULD; FLOATING POINT MOULD; STAGNANT MOULD.

 CI15 DIGIT TWO TOKEN, DIGIT FOUR TOKEN, DIGIT EIGHT TOKEN,
     DIGIT ONE TOKEN, DIGIT SIX TOKEN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        COLLECTION LIST PACK; PATTERN; BITS PATTERN;
        RADIX MOULD; RADIX*.

 CI16 PLUS TOKEN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        COLLECTION LIST PACK; PATTERN; SIGN MOULD;
        SIGN FRAME*; INTEGRAL PATTERN; REAL PATTERN;
        COMPLEX PATTERN; FLOATING POINT MOULD;
        STAGNANT MOULD.

 CI17 MINUS TOKEN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        COLLECTION LIST PACK; PATTERN; SIGN MOULD;
        SIGN FRAME*; INTEGRAL PATTERN; REAL PATTERN;
```

MAY BE CONTAINED IN (CI): A, B - AB.

COMPLEX PATTERN; GENERAL PATTERN; FLOATING POINT MOULD; STAGNANT MOULD; ACTUAL SPECIFICATION PACK; ACTUAL SPECIFICATION LIST; ACTUAL SPECIFICATION*.

CI18 POINT TOKEN-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE; COLLECTION LIST PACK; PATTERN; REAL PATTERN; COMPLEX PATTERN; REAL MOULD*; FLOATING POINT MOULD; STAGNANT MOULD.

CI19 OPEN TOKEN, CLOSE TOKEN-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE; INSERTION; REPLICATOR; COLLECTION LIST PACK*; PATTERN; LITERAL; INSERT SEQUENCE; INSERT; REPLICATION; DYNAMIC REPLICATION; ENCLOSED CLAUSE*; REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL; SIGN MOULD; INTEGRAL PATTERN; REAL PATTERN; BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN; STRING PATTERN; BITS PATTERN; GENERAL PATTERN; INCLUDED PATTERN; INTEGRAL MOULD; INTEGRAL CHOICE PATTERN; LITERAL LIST PACK*; LITERAL LIST; REAL MOULD; FLOATING POINT MOULD; STAGNANT MOULD; BOOLEAN CHOICE MOULD*; LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME; LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE; RADIX MOULD; ACTUAL SPECIFICATION PACK*.

CI20 BEGIN TOKEN, END TOKEN, STROP CASE START TOKEN, STROP CASE FINISH TOKEN, STROP CONDITION START TOKEN, STROP CONDITION FINISH TOKEN, INNER CLAUSE-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE; INSERTION; REPLICATOR; COLLECTION LIST PACK; PATTERN; LITERAL; INSERT SEQUENCE; INSERT; REPLICATION; DYNAMIC REPLICATION; ENCLOSED CLAUSE*; REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL; SIGN MOULD; INTEGRAL PATTERN; REAL PATTERN; BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN; STRING PATTERN; BITS PATTERN; GENERAL PATTERN; INCLUDED PATTERN; INTEGRAL MOULD; INTEGRAL CHOICE PATTERN; LITERAL LIST PACK; LITERAL LIST; REAL MOULD; FLOATING POINT MOULD; STAGNANT MOULD; BOOLEAN CHOICE MOULD; LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME; LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE; RADIX MOULD.

CI21 COMMA TOKEN-
(NONTERMINALS:) COLLECTION LIST*; COLLECTION; PICTURE; COLLECTION LIST PACK; PATTERN; INTEGRAL PATTERN; BOOLEAN PATTERN; GENERAL PATTERN; INTEGRAL CHOICE PATTERN; LITERAL LIST PACK; LITERAL LIST*; BOOLEAN CHOICE MOULD*; ACTUAL SPECIFICATION PACK; ACTUAL SPECIFICATION LIST*.

CI22 INTEGRAL DENOTATION-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;

```
MAY BE CONTAINED IN  (CI): A, B - AB,

        INSERTION; REPLICATOR; COLLECTION LIST PACK; PATTERN;
        LITERAL; INSERT SEQUENCE; INSERT; REPLICATION*;
        REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL;
        SIGN MOULD; INTEGRAL PATTERN; REAL PATTERN;
        BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
        STRING PATTERN; BITS PATTERN; GENERAL PATTERN;
        INCLUDED PATTERN; INTEGRAL MOULD;
        INTEGRAL CHOICE PATTERN; LITERAL LIST PACK;
        LITERAL LIST; REAL MOULD; FLOATING POINT MOULD;
        STAGNANT MOULD; BOOLEAN CHOICE MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
        RADIX MOULD; ACTUAL SPECIFICATION PACK;
        ACTUAL SPECIFICATION LIST; ACTUAL SPECIFICATION*.

 CI23 CHARACTER DENOTATION, ROW OF CHARACTER DENOTATION-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        INSERTION; COLLECTION LIST PACK; PATTERN; LITERAL;
        INSERT SEQUENCE; INSERT; STRING DENOTATION*;
        REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL;
        SIGN MOULD; INTEGRAL PATTERN; REAL PATTERN;
        BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
        STRING PATTERN; BITS PATTERN; GENERAL PATTERN;
        INCLUDED PATTERN; INTEGRAL MOULD;
        INTEGRAL CHOICE PATTERN; LITERAL LIST PACK;
        LITERAL LIST; REAL MOULD; FLOATING POINT MOULD;
        STAGNANT MOULD; BOOLEAN CHOICE MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
        RADIX MOULD.

 CI24 COLLECTION LIST+, COLLECTION+, PICTURE,
     COLLECTION LIST PACK+-
    (NONTERMINALS:) COLLECTION LIST*; COLLECTION*;
        COLLECTION LIST PACK*.

 CI25 INSERTION-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION*; PICTURE*;
        COLLECTION LIST PACK; PATTERN; SIGN MOULD*;
        INTEGRAL PATTERN; REAL PATTERN; BOOLEAN PATTERN*;
        CHARACTER PATTERN*; COMPLEX PATTERN*; STRING PATTERN*;
        BITS PATTERN; GENERAL PATTERN*; INCLUDED PATTERN*;
        INTEGRAL MOULD*; INTEGRAL CHOICE PATTERN*;
        REAL MOULD*; FLOATING POINT MOULD*; STAGNANT MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
        RADIX MOULD*.

 CI26 REPLICATOR-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION*; PICTURE;
        INSERTION; COLLECTION LIST PACK; PATTERN; LITERAL*;
        INSERT SEQUENCE; INSERT*; SIGN MOULD*;
        INTEGRAL PATTERN; REAL PATTERN; BOOLEAN PATTERN;
        CHARACTER PATTERN; COMPLEX PATTERN; STRING PATTERN;
        BITS PATTERN; GENERAL PATTERN; INCLUDED PATTERN;
        INTEGRAL MOULD*; INTEGRAL CHOICE PATTERN;
        LITERAL LIST PACK; LITERAL LIST; REAL MOULD;
```

```
MAY BE CONTAINED IN  (C1): A, B - AB.

          FLOATING POINT MOULD; STAGNANT MOULD;
          BOOLEAN CHOICE MOULD;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
          RADIX MOULD.

 C127 PATTERN-
    (NONTERMINALS;) COLLECTION LIST; COLLECTION; PICTURE*;
          COLLECTION LIST PACK.

 C128 LITERAL-
    (NONTERMINALS;) COLLECTION LIST; COLLECTION; PICTURE;
          INSERTION*; COLLECTION LIST PACK; PATTERN;
          INSERT SEQUENCE; INSERT*; SIGN MOULD;
          INTEGRAL PATTERN; REAL PATTERN; BOOLEAN PATTERN;
          CHARACTER PATTERN; COMPLEX PATTERN; STRING PATTERN;
          BITS PATTERN; GENERAL PATTERN; INCLUDED PATTERN;
          INTEGRAL MOULD; INTEGRAL CHOICE PATTERN;
          LITERAL LIST PACK; LITERAL LIST*; REAL MOULD;
          FLOATING POINT MOULD; STAGNANT MOULD;
          BOOLEAN CHOICE MOULD*;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
          RADIX MOULD.

 C129 INSERT SEQUENCE*, INSERT-
    (NONTERMINALS;) COLLECTION LIST; COLLECTION; PICTURE;
          INSERTION*; COLLECTION LIST PACK; PATTERN;
          INSERT SEQUENCE*; SIGN MOULD; INTEGRAL PATTERN;
          REAL PATTERN; BOOLEAN PATTERN; CHARACTER PATTERN;
          COMPLEX PATTERN; STRING PATTERN; BITS PATTERN;
          GENERAL PATTERN; INCLUDED PATTERN; INTEGRAL MOULD;
          INTEGRAL CHOICE PATTERN; REAL MOULD;
          FLOATING POINT MOULD; STAGNANT MOULD;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
          RADIX MOULD.

 C130 ALIGNMENT-
    (NONTERMINALS;) COLLECTION LIST; COLLECTION; PICTURE;
          INSERTION; COLLECTION LIST PACK; PATTERN;
          INSERT SEQUENCE; INSERT*; SIGN MOULD;
          INTEGRAL PATTERN; REAL PATTERN; BOOLEAN PATTERN;
          CHARACTER PATTERN; COMPLEX PATTERN; STRING PATTERN;
          BITS PATTERN; GENERAL PATTERN; INCLUDED PATTERN;
          INTEGRAL MOULD; INTEGRAL CHOICE PATTERN; REAL MOULD;
          FLOATING POINT MOULD; STAGNANT MOULD;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
          RADIX MOULD.

 C131 REPLICATION-
    (NONTERMINALS;) COLLECTION LIST; COLLECTION; PICTURE;
          INSERTION; REPLICATOR*; COLLECTION LIST PACK;
          PATTERN; LITERAL; INSERT SEQUENCE; INSERT;
          REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL*;
          SIGN MOULD; INTEGRAL PATTERN; REAL PATTERN;
```

```
MAY BE CONTAINED IN  (CI): A, B - AB.

          BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
          STRING PATTERN*; BITS PATTERN; GENERAL PATTERN;
          INCLUDED PATTERN; INTEGRAL MOULD;
          INTEGRAL CHOICE PATTERN; LITERAL LIST PACK;
          LITERAL LIST; REAL MOULD; FLOATING POINT MOULD;
          STAGNANT MOULD; BOOLEAN CHOICE MOULD;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
          RADIX MOULD.

 C132 DYNAMIC REPLICATION-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
          INSERTION; REPLICATOR; COLLECTION LIST PACK; PATTERN;
          LITERAL; INSERT SEQUENCE; INSERT; REPLICATION*;
          REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL;
          SIGN MOULD; INTEGRAL PATTERN; REAL PATTERN;
          BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
          STRING PATTERN; BITS PATTERN; GENERAL PATTERN;
          INCLUDED PATTERN; INTEGRAL MOULD;
          INTEGRAL CHOICE PATTERN; LITERAL LIST PACK;
          LITERAL LIST; REAL MOULD; FLOATING POINT MOULD;
          STAGNANT MOULD; BOOLEAN CHOICE MOULD;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
          RADIX MOULD.

 C133 STRING DENOTATION-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
          INSERTION; COLLECTION LIST PACK; PATTERN; LITERAL*;
          INSERT SEQUENCE; INSERT; REPLICATED LITERAL SEQUENCE;
          REPLICATED LITERAL*; SIGN MOULD; INTEGRAL PATTERN;
          REAL PATTERN; BOOLEAN PATTERN; CHARACTER PATTERN;
          COMPLEX PATTERN; STRING PATTERN; BITS PATTERN;
          GENERAL PATTERN; INCLUDED PATTERN; INTEGRAL MOULD;
          INTEGRAL CHOICE PATTERN; LITERAL LIST PACK;
          LITERAL LIST; REAL MOULD; FLOATING POINT MOULD;
          STAGNANT MOULD; BOOLEAN CHOICE MOULD;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
          RADIX MOULD.

 C134 REPLICATED LITERAL SEQUENCE+, REPLICATED LITERAL-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
          INSERTION; COLLECTION LIST PACK; PATTERN; LITERAL*;
          INSERT SEQUENCE; INSERT; REPLICATED LITERAL SEQUENCE*;
          SIGN MOULD; INTEGRAL PATTERN; REAL PATTERN;
          BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
          STRING PATTERN; BITS PATTERN; GENERAL PATTERN;
          INCLUDED PATTERN; INTEGRAL MOULD;
          INTEGRAL CHOICE PATTERN; LITERAL LIST PACK;
          LITERAL LIST; REAL MOULD; FLOATING POINT MOULD;
          STAGNANT MOULD; BOOLEAN CHOICE MOULD;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
          RADIX MOULD.

 C135 SIGN MOULD-
```

MAY BE CONTAINED IN (C1): A, B - AB.

(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN; INTEGRAL PATTERN*;
REAL PATTERN*; COMPLEX PATTERN; FLOATING POINT MOULD*;
STAGNANT MOULD*.

C136 SIGN FRAME-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN; SIGN MOULD*;
INTEGRAL PATTERN; REAL PATTERN; COMPLEX PATTERN;
FLOATING POINT MOULD; STAGNANT MOULD.

C137 INTEGRAL PATTERN, BOOLEAN PATTERN, CHARACTER PATTERN,
COMPLEX PATTERN, STRING PATTERN, BITS PATTERN,
GENERAL PATTERN, INCLUDED PATTERN-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN*.

C138 INTEGRAL MOULD+, DIGIT FRAME-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN; INTEGRAL PATTERN*;
REAL PATTERN; COMPLEX PATTERN; BITS PATTERN*;
INTEGRAL MOULD*; REAL MOULD*; FLOATING POINT MOULD*;
STAGNANT MOULD*.

C139 INTEGRAL CHOICE PATTERN-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN; INTEGRAL PATTERN*.

C140 LITERAL LIST+-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN; INTEGRAL PATTERN;
INTEGRAL CHOICE PATTERN; LITERAL LIST PACK*;
LITERAL LIST*.

C141 REAL MOULD-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN; REAL PATTERN*;
COMPLEX PATTERN; FLOATING POINT MOULD;
STAGNANT MOULD*.

C142 FLOATING POINT MOULD-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN; REAL PATTERN*;
COMPLEX PATTERN.

C143 LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME,
LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE+-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN; STRING PATTERN*;
LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE*.

C144 RADIX MOULD-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
COLLECTION LIST PACK; PATTERN; BITS PATTERN*.

C145 ACTUAL SPECIFICATION LIST+, ACTUAL SPECIFICATION-
(NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;

MAY BE CONTAINED IN (C1): A, B - AB.

COLLECTION LIST PACK; PATTERN; GENERAL PATTERN;
ACTUAL SPECIFICATION PACK*;
ACTUAL SPECIFICATION LIST*.

```
MAY BEGIN WITH  (BW): AB - A.


  BW1 LETTER A TOKEN, LETTER B TOKEN, LETTER C TOKEN,
      LETTER D TOKEN, LETTER E TOKEN, LETTER F TOKEN,
      LETTER G TOKEN, LETTER I TOKEN, LETTER K TOKEN,
      LETTER L TOKEN, LETTER N TOKEN, LETTER P TOKEN,
      LETTER R TOKEN, LETTER S TOKEN, LETTER T TOKEN,
      LETTER X TOKEN, LETTER Y TOKEN, LETTER Z TOKEN,
      DIGIT TWO TOKEN, DIGIT FOUR TOKEN, DIGIT EIGHT TOKEN,
      DIGIT ONE TOKEN, DIGIT SIX TOKEN, PLUS TOKEN,
      MINUS TOKEN, POINT TOKEN, OPEN TOKEN, CLOSE TOKEN,
      BEGIN TOKEN, COMMA TOKEN, END TOKEN,
      STROP CASE START TOKEN, STROP CASE FINISH TOKEN,
      STROP CONDITION START TOKEN,
      STROP CONDITION FINISH TOKEN, INNER CLAUSE,
      INTEGRAL DENOTATION, CHARACTER DENOTATION,
      ROW OF CHARACTER DENOTATION-

  BW2 COLLECTION LIST-
    (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
        LETTER C TOKEN; LETTER D TOKEN; LETTER F TOKEN;
        LETTER G TOKEN; LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER S TOKEN;
        LETTER T TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; DIGIT TWO TOKEN; DIGIT FOUR TOKEN;
        DIGIT EIGHT TOKEN; DIGIT ONE TOKEN; PLUS TOKEN;
        MINUS TOKEN; POINT TOKEN; OPEN TOKEN; COMMA TOKEN*;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) COLLECTION*; PICTURE; INSERTION;
        REPLICATOR; COLLECTION LIST PACK; PATTERN; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION; SIGN MOULD;
        SIGN FRAME; INTEGRAL PATTERN; REAL PATTERN;
        BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
        STRING PATTERN; BITS PATTERN; GENERAL PATTERN;
        INCLUDED PATTERN; INTEGRAL MOULD;
        INTEGRAL CHOICE PATTERN; DIGIT FRAME; REAL MOULD;
        FLOATING POINT MOULD; STAGNANT MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
        RADIX MOULD; RADIX.

  BW3 COLLECTION-
    (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
        LETTER C TOKEN; LETTER D TOKEN; LETTER F TOKEN;
        LETTER G TOKEN; LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER S TOKEN;
        LETTER T TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; DIGIT TWO TOKEN; DIGIT FOUR TOKEN;
        DIGIT EIGHT TOKEN; DIGIT ONE TOKEN; PLUS TOKEN;
        MINUS TOKEN; POINT TOKEN; OPEN TOKEN;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) PICTURE*; INSERTION*; REPLICATOR*;
        COLLECTION LIST PACK*; PATTERN; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION; SIGN MOULD;
```

```
MAY BEGIN WITH  (BW): AB - A.

        SIGN FRAME; INTEGRAL PATTERN; REAL PATTERN;
        BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
        STRING PATTERN; BITS PATTERN; GENERAL PATTERN;
        INCLUDED PATTERN; INTEGRAL MOULD;
        INTEGRAL CHOICE PATTERN; DIGIT FRAME; REAL MOULD;
        FLOATING POINT MOULD; STAGNANT MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
        RADIX MOULD; RADIX.

  BW4 PICTURE-
    (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
        LETTER C TOKEN; LETTER D TOKEN; LETTER F TOKEN;
        LETTER G TOKEN; LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER S TOKEN;
        LETTER T TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; DIGIT TWO TOKEN; DIGIT FOUR TOKEN;
        DIGIT EIGHT TOKEN; DIGIT ONE TOKEN; PLUS TOKEN;
        MINUS TOKEN; POINT TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; PATTERN*;
        LITERAL; INSERT SEQUENCE; INSERT; ALIGNMENT;
        REPLICATION; DYNAMIC REPLICATION; STRING DENOTATION;
        SIGN MOULD; SIGN FRAME; INTEGRAL PATTERN;
        REAL PATTERN; BOOLEAN PATTERN; CHARACTER PATTERN;
        COMPLEX PATTERN; STRING PATTERN; BITS PATTERN;
        GENERAL PATTERN; INCLUDED PATTERN; INTEGRAL MOULD;
        INTEGRAL CHOICE PATTERN; DIGIT FRAME; REAL MOULD;
        FLOATING POINT MOULD; STAGNANT MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
        RADIX MOULD; RADIX.

  BW5 INSERTION-
    (TERMINALS:) LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER X TOKEN;
        LETTER Y TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) REPLICATOR; LITERAL*; INSERT SEQUENCE*;
        INSERT; ALIGNMENT; REPLICATION; DYNAMIC REPLICATION;
        STRING DENOTATION.

  BW6 REPLICATOR, REPLICATED LITERAL-
    (TERMINALS:) LETTER N TOKEN; INTEGRAL DENOTATION.
    (NONTERMINALS:) REPLICATION*; DYNAMIC REPLICATION.

  BW7 COLLECTION LIST PACK, LITERAL LIST PACK,
     BOOLEAN CHOICE MOULD, ACTUAL SPECIFICATION PACK-
    (TERMINALS:) OPEN TOKEN*.

  BW8 PATTERN-
    (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
        LETTER C TOKEN; LETTER D TOKEN; LETTER F TOKEN;
        LETTER G TOKEN; LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER S TOKEN;
        LETTER T TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; DIGIT TWO TOKEN; DIGIT FOUR TOKEN;
        DIGIT EIGHT TOKEN; DIGIT ONE TOKEN; PLUS TOKEN;
        MINUS TOKEN; POINT TOKEN; INTEGRAL DENOTATION;
```

```
MAY BEGIN WITH  (BW): AB - A.

        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION; SIGN MOULD;
        SIGN FRAME; INTEGRAL PATTERN*; REAL PATTERN*;
        BOOLEAN PATTERN*; CHARACTER PATTERN*;
        COMPLEX PATTERN*; STRING PATTERN*; BITS PATTERN*;
        GENERAL PATTERN*; INCLUDED PATTERN*; INTEGRAL MOULD;
        INTEGRAL CHOICE PATTERN; DIGIT FRAME; REAL MOULD;
        FLOATING POINT MOULD; STAGNANT MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
        RADIX MOULD; RADIX.

  BW9 LITERAL-
    (TERMINALS:) LETTER N TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) REPLICATOR*; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION*.

 BW10 INSERT SEQUENCE-
    (TERMINALS:) LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER X TOKEN;
        LETTER Y TOKEN; INTEGRAL DENOTATION.
    (NONTERMINALS:) REPLICATOR; INSERT*; ALIGNMENT;
        REPLICATION; DYNAMIC REPLICATION.

 BW11 INSERT-
    (TERMINALS:) LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER X TOKEN;
        LETTER Y TOKEN; INTEGRAL DENOTATION.
    (NONTERMINALS:) REPLICATOR*; ALIGNMENT*; REPLICATION;
        DYNAMIC REPLICATION.

 BW12 ALIGNMENT-
    (TERMINALS:) LETTER K TOKEN*; LETTER L TOKEN*;
        LETTER P TOKEN*; LETTER X TOKEN*; LETTER Y TOKEN*.

 BW13 REPLICATION-
    (TERMINALS:) LETTER N TOKEN; INTEGRAL DENOTATION*.
    (NONTERMINALS:) DYNAMIC REPLICATION*.

 BW14 DYNAMIC REPLICATION-
    (TERMINALS:) LETTER N TOKEN*.

 BW15 ENCLOSED CLAUSE-
    (TERMINALS:) OPEN TOKEN*; BEGIN TOKEN*;
        STROP CASE START TOKEN*; STROP CONDITION START TOKEN*.

 BW16 STRING DENOTATION-
    (TERMINALS:) CHARACTER DENOTATION*;
        ROW OF CHARACTER DENOTATION*.

 BW17 REPLICATED LITERAL SEQUENCE-
    (TERMINALS:) LETTER N TOKEN; INTEGRAL DENOTATION.
    (NONTERMINALS:) REPLICATION; DYNAMIC REPLICATION;
        REPLICATED LITERAL*.
```

```
MAY BEGIN WITH  (BW): AB - A.

 BW18 SIGN MOULD-
    (TERMINALS:) LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER X TOKEN;
        LETTER Y TOKEN; LETTER Z TOKEN*; PLUS TOKEN;
        MINUS TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR*; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION; SIGN FRAME*.

 BW19 SIGN FRAME-
    (TERMINALS:) PLUS TOKEN*; MINUS TOKEN*.

 BW20 INTEGRAL PATTERN-
    (TERMINALS:) LETTER C TOKEN; LETTER D TOKEN;
        LETTER K TOKEN; LETTER L TOKEN; LETTER N TOKEN;
        LETTER P TOKEN; LETTER S TOKEN; LETTER X TOKEN;
        LETTER Y TOKEN; LETTER Z TOKEN; PLUS TOKEN;
        MINUS TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION; SIGN MOULD*;
        SIGN FRAME; INTEGRAL MOULD*; INTEGRAL CHOICE PATTERN*;
        DIGIT FRAME.

 BW21 REAL PATTERN-
    (TERMINALS:) LETTER D TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; PLUS TOKEN; MINUS TOKEN; POINT TOKEN;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION; SIGN MOULD*;
        SIGN FRAME; INTEGRAL MOULD; DIGIT FRAME; REAL MOULD*;
        FLOATING POINT MOULD*; STAGNANT MOULD.

 BW22 BOOLEAN PATTERN-
    (TERMINALS:) LETTER B TOKEN*; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER X TOKEN; LETTER Y TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION.

 BW23 CHARACTER PATTERN,
     LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME-
    (TERMINALS:) LETTER A TOKEN*; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN*; LETTER X TOKEN; LETTER Y TOKEN;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR*; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
```

```
MAY BEGIN WITH  (BW): AB - A.

        DYNAMIC REPLICATION; STRING DENOTATION.

 BW24 COMPLEX PATTERN-
    (TERMINALS:) LETTER D TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; PLUS TOKEN; MINUS TOKEN; POINT TOKEN;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION; SIGN MOULD;
        SIGN FRAME; REAL PATTERN*; INTEGRAL MOULD;
        DIGIT FRAME; REAL MOULD; FLOATING POINT MOULD;
        STAGNANT MOULD.

 BW25 STRING PATTERN-
    (TERMINALS:) LETTER A TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN; LETTER T TOKEN*; LETTER X TOKEN;
        LETTER Y TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION*;
        DYNAMIC REPLICATION; STRING DENOTATION;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*.

 BW26 BITS PATTERN-
    (TERMINALS:) LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER X TOKEN;
        LETTER Y TOKEN; DIGIT TWO TOKEN; DIGIT FOUR TOKEN;
        DIGIT EIGHT TOKEN; DIGIT ONE TOKEN;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION; RADIX MOULD*;
        RADIX.

 BW27 GENERAL PATTERN-
    (TERMINALS:) LETTER G TOKEN*; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER X TOKEN; LETTER Y TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION.

 BW28 INCLUDED PATTERN-
    (TERMINALS:) LETTER F TOKEN*; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER X TOKEN; LETTER Y TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION.
```

```
MAY BEGIN WITH  (BW): AB - A.

 BW29 INTEGRAL MOULD-
    (TERMINALS:) LETTER D TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN*; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR*; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION; DIGIT FRAME*.

 BW30 INTEGRAL CHOICE PATTERN-
    (TERMINALS:) LETTER C TOKEN*; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER X TOKEN; LETTER Y TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENCTATION.

 BW31 DIGIT FRAME-
    (TERMINALS:) LETTER D TOKEN*; LETTER Z TOKEN*.

 BW32 LITERAL LIST-
    (TERMINALS:) LETTER N TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) REPLICATOR; LITERAL*; REPLICATION;
        DYNAMIC REPLICATION; STRING DENCTATION.

 BW33 REAL MOULD-
    (TERMINALS:) LETTER D TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN*; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; POINT TOKEN*; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION;
        INTEGRAL MOULD*; DIGIT FRAME.

 BW34 FLOATING POINT MOULD-
    (TERMINALS:) LETTER D TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; PLUS TOKEN; MINUS TOKEN; POINT TOKEN;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION; SIGN MOULD;
        SIGN FRAME; INTEGRAL MOULD; DIGIT FRAME; REAL MOULD;
        STAGNANT MOULD*.

 BW35 STAGNANT MOULD-
    (TERMINALS:) LETTER D TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; PLUS TOKEN; MINUS TOKEN; POINT TOKEN;
```

```
MAY BEGIN WITH  (BW): AB - A.

          INTEGRAL DENOTATION; CHARACTER DENOTATION;
          ROW OF CHARACTER DENOTATION.
      (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
          INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
          DYNAMIC REPLICATION; STRING DENOTATION; SIGN MOULD*;
          SIGN FRAME; INTEGRAL MOULD*; DIGIT FRAME; REAL MOULD*.

 BW36
       LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE-
      (TERMINALS:) LETTER A TOKEN; LETTER K TOKEN;
          LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
          LETTER S TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
          INTEGRAL DENOTATION; CHARACTER DENOTATION;
          ROW OF CHARACTER DENOTATION.
      (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
          INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
          DYNAMIC REPLICATION; STRING DENOTATION;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*.

 BW37 RADIX MOULD-
      (TERMINALS:) LETTER K TOKEN; LETTER L TOKEN;
          LETTER N TOKEN; LETTER P TOKEN; LETTER X TOKEN;
          LETTER Y TOKEN; DIGIT TWO TOKEN; DIGIT FOUR TOKEN;
          DIGIT EIGHT TOKEN; DIGIT ONE TOKEN;
          INTEGRAL DENOTATION; CHARACTER DENOTATION;
          ROW OF CHARACTER DENOTATION.
      (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
          INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
          DYNAMIC REPLICATION; STRING DENOTATION; RADIX*.

 BW38 RADIX-
      (TERMINALS:) DIGIT TWO TOKEN*; DIGIT FOUR TOKEN*;
          DIGIT EIGHT TOKEN*; DIGIT ONE TOKEN*.

 BW39 ACTUAL SPECIFICATION LIST-
      (TERMINALS:) MINUS TOKEN; INTEGRAL DENOTATION.
      (NONTERMINALS:) ACTUAL SPECIFICATION*.

 BW40 ACTUAL SPECIFICATION-
      (TERMINALS:) MINUS TOKEN*; INTEGRAL DENOTATION*.
```

```
MAY BE THE BEGIN OF  (B0): A - AB.


   B01 LETTER A TOKEN-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
         PATTERN; CHARACTER PATTERN*; STRING PATTERN;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE.

   B02 LETTER B TOKEN-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
         PATTERN; BOOLEAN PATTERN*.

   B03 LETTER C TOKEN-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
         PATTERN; INTEGRAL PATTERN; INTEGRAL CHOICE PATTERN*.

   B04 LETTER D TOKEN-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
         PATTERN; INTEGRAL PATTERN; REAL PATTERN;
         COMPLEX PATTERN; INTEGRAL MOULD; DIGIT FRAME*;
         REAL MOULD; FLOATING POINT MOULD; STAGNANT MOULD.

   B05 LETTER E TOKEN, LETTER I TOKEN, LETTER R TOKEN,
       DIGIT SIX TOKEN, CLOSE TOKEN, END TOKEN,
       STROP CASE FINISH TOKEN, STROP CONDITION FINISH TOKEN,
       INNER CLAUSE, COLLECTION LIST, ENCLOSED CLAUSE,
       REPLICATED LITERAL SEQUENCE, LITERAL LIST PACK,
       LITERAL LIST, BOOLEAN CHOICE MOULD,
       LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE,
       ACTUAL SPECIFICATION PACK, ACTUAL SPECIFICATION LIST-

   B06 LETTER F TOKEN-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
         PATTERN; INCLUDED PATTERN*.

   B07 LETTER G TOKEN-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
         PATTERN; GENERAL PATTERN*.

   B08 LETTER K TOKEN, LETTER L TOKEN, LETTER P TOKEN,
       LETTER X TOKEN, LETTER Y TOKEN-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
         INSERTION; PATTERN; INSERT SEQUENCE; INSERT;
         ALIGNMENT*; SIGN MOULD; INTEGRAL PATTERN;
         REAL PATTERN; BOOLEAN PATTERN; CHARACTER PATTERN;
         COMPLEX PATTERN; STRING PATTERN; BITS PATTERN;
         GENERAL PATTERN; INCLUDED PATTERN; INTEGRAL MOULD;
         INTEGRAL CHOICE PATTERN; REAL MOULD;
         FLOATING POINT MOULD; STAGNANT MOULD;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
         RADIX MOULD.

   B09 LETTER N TOKEN-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
         INSERTION; REPLICATOR; PATTERN; LITERAL;
         INSERT SEQUENCE; INSERT; REPLICATION;
```

```
MAY BE THE BEGIN OF  (BO): A - AB.

        DYNAMIC REPLICATION*; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; SIGN MOULD; INTEGRAL PATTERN;
        REAL PATTERN; BOOLEAN PATTERN; CHARACTER PATTERN;
        COMPLEX PATTERN; STRING PATTERN; BITS PATTERN;
        GENERAL PATTERN; INCLUDED PATTERN; INTEGRAL MOULD;
        INTEGRAL CHOICE PATTERN; LITERAL LIST; REAL MOULD;
        FLOATING POINT MOULD; STAGNANT MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
        RADIX MOULD.

 BO10 LETTER S TOKEN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; INTEGRAL PATTERN; REAL PATTERN;
        CHARACTER PATTERN*; COMPLEX PATTERN; STRING PATTERN;
        INTEGRAL MOULD*; REAL MOULD*; FLOATING POINT MOULD;
        STAGNANT MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE.

 BO11 LETTER T TOKEN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; STRING PATTERN*.

 BO12 LETTER Z TOKEN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; SIGN MOULD*; INTEGRAL PATTERN; REAL PATTERN;
        COMPLEX PATTERN; INTEGRAL MOULD; DIGIT FRAME*;
        REAL MOULD; FLOATING POINT MOULD; STAGNANT MOULD.

 BO13 DIGIT TWO TOKEN, DIGIT FOUR TOKEN, DIGIT EIGHT TOKEN,
     DIGIT ONE TOKEN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; BITS PATTERN; RADIX MOULD; RADIX*.

 BO14 PLUS TOKEN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; SIGN MOULD; SIGN FRAME*; INTEGRAL PATTERN;
        REAL PATTERN; COMPLEX PATTERN; FLOATING POINT MOULD;
        STAGNANT MOULD.

 BO15 MINUS TOKEN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; SIGN MOULD; SIGN FRAME*; INTEGRAL PATTERN;
        REAL PATTERN; COMPLEX PATTERN; FLOATING POINT MOULD;
        STAGNANT MOULD; ACTUAL SPECIFICATION LIST;
        ACTUAL SPECIFICATION*.

 BO16 POINT TOKEN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; REAL PATTERN; COMPLEX PATTERN; REAL MOULD*;
        FLOATING POINT MOULD; STAGNANT MOULD.

 BO17 OPEN TOKEN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION;
        COLLECTION LIST PACK*; ENCLOSED CLAUSE*;
        LITERAL LIST PACK*; BOOLEAN CHOICE MOULD*;
```

```
MAY BE THE BEGIN OF  (BO): A - AB.

        ACTUAL SPECIFICATION PACK*.

 BO18 BEGIN TOKEN, STROP CASE START TOKEN,
     STROP CONDITION START TOKEN-
    (NONTERMINALS:) ENCLOSED CLAUSE*.

 BO19 COMMA TOKEN, COLLECTION-
    (NONTERMINALS:) COLLECTION LIST*.

 BO20 INTEGRAL DENOTATION-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        INSERTION; REPLICATOR; PATTERN; LITERAL;
        INSERT SEQUENCE; INSERT; REPLICATION*;
        REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL;
        SIGN MOULD; INTEGRAL PATTERN; REAL PATTERN;
        BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
        STRING PATTERN; BITS PATTERN; GENERAL PATTERN;
        INCLUDED PATTERN; INTEGRAL MOULD;
        INTEGRAL CHOICE PATTERN; LITERAL LIST; REAL MOULD;
        FLOATING POINT MOULD; STAGNANT MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
        RADIX MOULD; ACTUAL SPECIFICATION LIST;
        ACTUAL SPECIFICATION*.

 BO21 CHARACTER DENOTATION, ROW OF CHARACTER DENOTATION-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        INSERTION; PATTERN; LITERAL; STRING DENOTATION*;
        SIGN MOULD; INTEGRAL PATTERN; REAL PATTERN;
        BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
        STRING PATTERN; BITS PATTERN; GENERAL PATTERN;
        INCLUDED PATTERN; INTEGRAL MOULD;
        INTEGRAL CHOICE PATTERN; LITERAL LIST; REAL MOULD;
        FLOATING POINT MOULD; STAGNANT MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
        RADIX MOULD.

 BO22 PICTURE, COLLECTION LIST PACK-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION*.

 BO23 INSERTION-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION*; PICTURE*;
        PATTERN; SIGN MOULD*; INTEGRAL PATTERN; REAL PATTERN;
        BOOLEAN PATTERN*; CHARACTER PATTERN*;
        COMPLEX PATTERN; STRING PATTERN*; BITS PATTERN;
        GENERAL PATTERN*; INCLUDED PATTERN*; INTEGRAL MOULD*;
        INTEGRAL CHOICE PATTERN*; REAL MOULD*;
        FLOATING POINT MOULD; STAGNANT MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
        RADIX MOULD*.

 BO24 REPLICATOR-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION*; PICTURE;
        INSERTION; PATTERN; LITERAL*; INSERT SEQUENCE;
        INSERT*; SIGN MOULD*; INTEGRAL PATTERN; REAL PATTERN;
```

```
MAY BE THE BEGIN OF  (BO): A - AB.

         BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
         STRING PATTERN; BITS PATTERN; GENERAL PATTERN;
         INCLUDED PATTERN; INTEGRAL MOULD*;
         INTEGRAL CHOICE PATTERN; LITERAL LIST; REAL MOULD;
         FLOATING POINT MOULD; STAGNANT MOULD;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
         RADIX MOULD.

 BO25 PATTERN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE*.

 BO26 LITERAL-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
         INSERTION*; PATTERN; SIGN MOULD; INTEGRAL PATTERN;
         REAL PATTERN; BOOLEAN PATTERN; CHARACTER PATTERN;
         COMPLEX PATTERN; STRING PATTERN; BITS PATTERN;
         GENERAL PATTERN; INCLUDED PATTERN; INTEGRAL MOULD;
         INTEGRAL CHOICE PATTERN; LITERAL LIST*; REAL MOULD;
         FLOATING POINT MOULD; STAGNANT MOULD;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
         RADIX MOULD.

 BO27 INSERT SEQUENCE-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
         INSERTION*; PATTERN; SIGN MOULD; INTEGRAL PATTERN;
         REAL PATTERN; BOOLEAN PATTERN; CHARACTER PATTERN;
         COMPLEX PATTERN; STRING PATTERN; BITS PATTERN;
         GENERAL PATTERN; INCLUDED PATTERN; INTEGRAL MOULD;
         INTEGRAL CHOICE PATTERN; REAL MOULD;
         FLOATING POINT MOULD; STAGNANT MOULD;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
         RADIX MOULD.

 BO28 INSERT-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
         INSERTION; PATTERN; INSERT SEQUENCE*; SIGN MOULD;
         INTEGRAL PATTERN; REAL PATTERN; BOOLEAN PATTERN;
         CHARACTER PATTERN; COMPLEX PATTERN; STRING PATTERN;
         BITS PATTERN; GENERAL PATTERN; INCLUDED PATTERN;
         INTEGRAL MOULD; INTEGRAL CHOICE PATTERN; REAL MOULD;
         FLOATING POINT MOULD; STAGNANT MOULD;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
         RADIX MOULD.

 BO29 ALIGNMENT-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
         INSERTION; PATTERN; INSERT SEQUENCE; INSERT*;
         SIGN MOULD; INTEGRAL PATTERN; REAL PATTERN;
         BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
         STRING PATTERN; BITS PATTERN; GENERAL PATTERN;
         INCLUDED PATTERN; INTEGRAL MOULD;
         INTEGRAL CHOICE PATTERN; REAL MOULD;
         FLOATING POINT MOULD; STAGNANT MOULD;
```

```
MAY BE THE BEGIN OF  (BO): A - AB.

         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
         RADIX MOULD.

 BO30 REPLICATION-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
         INSERTION; REPLICATOR*; PATTERN; LITERAL;
         INSERT SEQUENCE; INSERT; REPLICATED LITERAL SEQUENCE;
         REPLICATED LITERAL*; SIGN MOULD; INTEGRAL PATTERN;
         REAL PATTERN; BOOLEAN PATTERN; CHARACTER PATTERN;
         COMPLEX PATTERN; STRING PATTERN*; BITS PATTERN;
         GENERAL PATTERN; INCLUDED PATTERN; INTEGRAL MOULD;
         INTEGRAL CHOICE PATTERN; LITERAL LIST; REAL MOULD;
         FLOATING POINT MOULD; STAGNANT MOULD;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
         RADIX MOULD.

 BO31 DYNAMIC REPLICATION-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
         INSERTION; REPLICATOR; PATTERN; LITERAL;
         INSERT SEQUENCE; INSERT; REPLICATION*;
         REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL;
         SIGN MOULD; INTEGRAL PATTERN; REAL PATTERN;
         BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
         STRING PATTERN; BITS PATTERN; GENERAL PATTERN;
         INCLUDED PATTERN; INTEGRAL MOULD;
         INTEGRAL CHOICE PATTERN; LITERAL LIST; REAL MOULD;
         FLOATING POINT MOULD; STAGNANT MOULD;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
         RADIX MOULD.

 BO32 STRING DENOTATION-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
         INSERTION; PATTERN; LITERAL*; SIGN MOULD;
         INTEGRAL PATTERN; REAL PATTERN; BOOLEAN PATTERN;
         CHARACTER PATTERN; COMPLEX PATTERN; STRING PATTERN;
         BITS PATTERN; GENERAL PATTERN; INCLUDED PATTERN;
         INTEGRAL MOULD; INTEGRAL CHOICE PATTERN;
         LITERAL LIST; REAL MOULD; FLOATING POINT MOULD;
         STAGNANT MOULD;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
         LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
         RADIX MOULD.

 BO33 REPLICATED LITERAL-
    (NONTERMINALS:) REPLICATED LITERAL SEQUENCE*.

 BO34 SIGN MOULD-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
         PATTERN; INTEGRAL PATTERN*; REAL PATTERN*;
         COMPLEX PATTERN; FLOATING POINT MOULD;
         STAGNANT MOULD*.

 BO35 SIGN FRAME-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
```

```
MAY BE THE BEGIN OF  (BO): A - AB.

        PATTERN; SIGN MOULD*; INTEGRAL PATTERN; REAL PATTERN;
        COMPLEX PATTERN; FLOATING POINT MOULD;
        STAGNANT MOULD.

 B036 INTEGRAL PATTERN, BOOLEAN PATTERN, CHARACTER PATTERN,
     COMPLEX PATTERN, STRING PATTERN, BITS PATTERN,
     GENERAL PATTERN, INCLUDED PATTERN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN*.

 B037 REAL PATTERN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN*; COMPLEX PATTERN*.

 B038 INTEGRAL MOULD-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; INTEGRAL PATTERN*; REAL PATTERN;
        COMPLEX PATTERN; REAL MOULD*; FLOATING POINT MOULD;
        STAGNANT MOULD*.

 B039 INTEGRAL CHOICE PATTERN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; INTEGRAL PATTERN*.

 B040 DIGIT FRAME-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; INTEGRAL PATTERN; REAL PATTERN;
        COMPLEX PATTERN; INTEGRAL MOULD*; REAL MOULD;
        FLOATING POINT MOULD; STAGNANT MOULD.

 B041 REAL MOULD-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; REAL PATTERN*; COMPLEX PATTERN;
        FLOATING POINT MOULD; STAGNANT MOULD*.

 B042 FLOATING POINT MOULD-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; REAL PATTERN*; COMPLEX PATTERN.

 B043 STAGNANT MOULD-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; REAL PATTERN; COMPLEX PATTERN;
        FLOATING POINT MOULD*.

 B044 LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; STRING PATTERN*;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE*.

 B045 RADIX MOULD-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; BITS PATTERN*.

 B046 RADIX-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; BITS PATTERN; RADIX MOULD*.
```

```
MAY BE THE BEGIN OF  (BO): A - AB.

 BO47 ACTUAL SPECIFICATION-
    (NONTERMINALS:) ACTUAL SPECIFICATION LIST*.
```

```
MAY END WITH  (EW): AB - B.


  EW1 LETTER A TOKEN, LETTER B TOKEN, LETTER C TOKEN,
      LETTER D TOKEN, LETTER E TOKEN, LETTER F TOKEN,
      LETTER G TOKEN, LETTER I TOKEN, LETTER K TOKEN,
      LETTER L TOKEN, LETTER N TOKEN, LETTER P TOKEN,
      LETTER R TOKEN, LETTER S TOKEN, LETTER T TOKEN,
      LETTER X TOKEN, LETTER Y TOKEN, LETTER Z TOKEN,
      DIGIT TWO TOKEN, DIGIT FOUR TOKEN, DIGIT EIGHT TOKEN,
      DIGIT ONE TOKEN, DIGIT SIX TOKEN, PLUS TOKEN,
      MINUS TOKEN, POINT TOKEN, OPEN TOKEN, CLOSE TOKEN,
      BEGIN TOKEN, COMMA TOKEN, END TOKEN,
      STROP CASE START TOKEN, STROP CASE FINISH TOKEN,
      STROP CONDITION START TOKEN,
      STROP CONDITION FINISH TOKEN, INNER CLAUSE,
      INTEGRAL DENOTATION, CHARACTER DENOTATION,
      ROW OF CHARACTER DENOTATION-

  EW2 COLLECTION LIST+-
    (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
        LETTER D TOKEN; LETTER G TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER P TOKEN; LETTER T TOKEN;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
        POINT TOKEN; CLOSE TOKEN; COMMA TOKEN*; END TOKEN;
        STROP CASE FINISH TOKEN;
        STROP CONDITION FINISH TOKEN; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) COLLECTION LIST*; COLLECTION*; PICTURE;
        INSERTION; COLLECTION LIST PACK; PATTERN; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; INTEGRAL PATTERN; REAL PATTERN;
        BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
        STRING PATTERN; BITS PATTERN; GENERAL PATTERN;
        INCLUDED PATTERN; INTEGRAL MOULD;
        INTEGRAL CHOICE PATTERN; DIGIT FRAME;
        LITERAL LIST PACK; REAL MOULD; FLOATING POINT MOULD;
        BOOLEAN CHOICE MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
        ACTUAL SPECIFICATION PACK.

  EW3 COLLECTION-
    (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
        LETTER D TOKEN; LETTER G TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER P TOKEN; LETTER T TOKEN;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
        POINT TOKEN; CLOSE TOKEN; END TOKEN;
        STROP CASE FINISH TOKEN;
        STROP CONDITION FINISH TOKEN; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) PICTURE*; INSERTION*;
        COLLECTION LIST PACK*; PATTERN; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; INTEGRAL PATTERN; REAL PATTERN;
        BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
```

```
MAY END WITH  (EW): AB - B.

        STRING PATTERN; BITS PATTERN; GENERAL PATTERN;
        INCLUDED PATTERN; INTEGRAL MOULD;
        INTEGRAL CHOICE PATTERN; DIGIT FRAME;
        LITERAL LIST PACK; REAL MOULD; FLOATING POINT MOULD;
        BOOLEAN CHOICE MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
        ACTUAL SPECIFICATION PACK.

  EW4 PICTURE-
    (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
        LETTER D TOKEN; LETTER G TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER P TOKEN; LETTER T TOKEN;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
        POINT TOKEN; CLOSE TOKEN; END TOKEN;
        STROP CASE FINISH TOKEN;
        STROP CONDITION FINISH TOKEN; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; PATTERN*; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; INTEGRAL PATTERN; REAL PATTERN;
        BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
        STRING PATTERN; BITS PATTERN; GENERAL PATTERN;
        INCLUDED PATTERN; INTEGRAL MOULD;
        INTEGRAL CHOICE PATTERN; DIGIT FRAME;
        LITERAL LIST PACK; REAL MOULD; FLOATING POINT MOULD;
        BOOLEAN CHOICE MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME:
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
        ACTUAL SPECIFICATION PACK.

  EW5 INSERTION, INSERT SEQUENCE+-
    (TERMINALS:) LETTER K TOKEN; LETTER L TOKEN;
        LETTER P TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) LITERAL*; INSERT SEQUENCE*; INSERT*;
        ALIGNMENT; STRING DENOTATION;
        REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL.

  EW6 REPLICATOR-
    (TERMINALS:) CLOSE TOKEN; END TOKEN;
        STROP CASE FINISH TOKEN;
        STROP CONDITION FINISH TOKEN; INTEGRAL DENOTATION.
    (NONTERMINALS:) REPLICATION*; DYNAMIC REPLICATION;
        ENCLOSED CLAUSE.

  EW7 COLLECTION LIST PACK, LITERAL LIST PACK,
     BOOLEAN CHOICE MOULD, ACTUAL SPECIFICATION PACK-
    (TERMINALS:) CLOSE TOKEN*.

  EW8 PATTERN-
    (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
        LETTER D TOKEN; LETTER G TOKEN; LETTER T TOKEN;
        LETTER Z TOKEN; POINT TOKEN; CLOSE TOKEN; END TOKEN;
        STROP CASE FINISH TOKEN;
        STROP CONDITION FINISH TOKEN.
```

```
MAY END WITH  (EW): AB - B.

    (NONTERMINALS:) ENCLOSED CLAUSE; INTEGRAL PATTERN*;
        REAL PATTERN*; BOOLEAN PATTERN*; CHARACTER PATTERN*;
        COMPLEX PATTERN*; STRING PATTERN*; BITS PATTERN*;
        GENERAL PATTERN*; INCLUDED PATTERN*; INTEGRAL MOULD;
        INTEGRAL CHOICE PATTERN; DIGIT FRAME;
        LITERAL LIST PACK; REAL MOULD; FLOATING POINT MOULD;
        BOOLEAN CHOICE MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
        ACTUAL SPECIFICATION PACK.

  EW9 LITERAL, REPLICATED LITERAL SEQUENCE+-
    (TERMINALS:) CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) STRING DENOTATION*;
        REPLICATED LITERAL SEQUENCE*; REPLICATED LITERAL*.

  EW10 INSERT-
    (TERMINALS:) LETTER K TOKEN; LETTER L TOKEN;
        LETTER P TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) LITERAL*; ALIGNMENT*; STRING DENOTATION;
        REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL.

  EW11 ALIGNMENT-
    (TERMINALS:) LETTER K TOKEN*; LETTER L TOKEN*;
        LETTER P TOKEN*; LETTER X TOKEN*; LETTER Y TOKEN*.

  EW12 REPLICATION-
    (TERMINALS:) CLOSE TOKEN; END TOKEN;
        STROP CASE FINISH TOKEN;
        STROP CONDITION FINISH TOKEN; INTEGRAL DENOTATION*.
    (NONTERMINALS:) DYNAMIC REPLICATION*; ENCLOSED CLAUSE.

  EW13 DYNAMIC REPLICATION, INCLUDED PATTERN-
    (TERMINALS:) CLOSE TOKEN; END TOKEN;
        STROP CASE FINISH TOKEN;
        STROP CONDITION FINISH TOKEN.
    (NONTERMINALS:) ENCLOSED CLAUSE*.

  EW14 ENCLOSED CLAUSE-
    (TERMINALS:) CLOSE TOKEN*; END TOKEN*;
        STROP CASE FINISH TOKEN*;
        STROP CONDITION FINISH TOKEN*.

  EW15 STRING DENOTATION-
    (TERMINALS:) CHARACTER DENOTATION*;
        ROW OF CHARACTER DENOTATION*.

  EW16 REPLICATED LITERAL-
    (TERMINALS:) CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) STRING DENOTATION*.

  EW17 SIGN MOULD-
    (TERMINALS:) PLUS TOKEN; MINUS TOKEN.
    (NONTERMINALS:) SIGN FRAME*.
```

```
MAY END WITH  (EW): AB - B.

 EW18 SIGN FRAME-
    (TERMINALS:) PLUS TOKEN*; MINUS TOKEN*.

 EW19 INTEGRAL PATTERN-
    (TERMINALS:) LETTER D TOKEN; LETTER Z TOKEN; CLOSE TOKEN.
    (NONTERMINALS:) INTEGRAL MOULD*; INTEGRAL CHOICE PATTERN*;
        DIGIT FRAME; LITERAL LIST PACK.

 EW20 REAL PATTERN-
    (TERMINALS:) LETTER D TOKEN; LETTER Z TOKEN; POINT TOKEN.
    (NONTERMINALS:) INTEGRAL MOULD; DIGIT FRAME; REAL MOULD*;
        FLOATING POINT MOULD*.

 EW21 BOOLEAN PATTERN-
    (TERMINALS:) LETTER B TOKEN*; CLOSE TOKEN.
    (NONTERMINALS:) BOOLEAN CHOICE MOULD*.

 EW22 CHARACTER PATTERN,
     LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME-
    (TERMINALS:) LETTER A TOKEN*.

 EW23 COMPLEX PATTERN-
    (TERMINALS:) LETTER D TOKEN; LETTER Z TOKEN; POINT TOKEN.
    (NONTERMINALS:) REAL PATTERN*; INTEGRAL MOULD;
        DIGIT FRAME; REAL MOULD; FLOATING POINT MOULU.

 EW24 STRING PATTERN-
    (TERMINALS:) LETTER A TOKEN*; LETTER T TOKEN*.
    (NONTERMINALS:)
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE*.

 EW25 BITS PATTERN, INTEGRAL MOULD+, FLOATING POINT MOULD-
    (TERMINALS:) LETTER D TOKEN; LETTER Z TOKEN.
    (NONTERMINALS:) INTEGRAL MOULD*; DIGIT FRAME*.

 EW26 GENERAL PATTERN-
    (TERMINALS:) LETTER G TOKEN*; CLOSE TOKEN.
    (NONTERMINALS:) ACTUAL SPECIFICATION PACK*.

 EW27 INTEGRAL CHOICE PATTERN-
    (TERMINALS:) CLOSE TOKEN.
    (NONTERMINALS:) LITERAL LIST PACK*.

 EW28 DIGIT FRAME-
    (TERMINALS:) LETTER D TOKEN*; LETTER Z TOKEN*.

 EW29 LITERAL LIST+-
    (TERMINALS:) CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) LITERAL*; STRING DENOTATION;
        REPLICATED LITERAL SEQUFNCE; REPLICATED LITERAL;
        LITERAL LIST*.

 EW30 REAL MOULD-
    (TERMINALS:) LETTER D TOKEN; LETTER Z TOKEN; POINT TOKEN*.
```

```
MAY END WITH  (EW): AB - B.

    (NONTERMINALS:) INTEGRAL MOULD*; DIGIT FRAME.

 EW31 STAGNANT MOULD-
    (TERMINALS:) LETTER D TOKEN; LETTER Z TOKEN; POINT TOKEN.
    (NONTERMINALS:) INTEGRAL MOULD*; DIGIT FRAME; REAL MOULD*.

 EW32
     LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE+-
    (TERMINALS:) LETTER A TOKEN.
    (NONTERMINALS:)
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE*.

 EW33 RADIX MOULD-
    (TERMINALS:) LETTER R TOKEN*.

 EW34 RADIX-
    (TERMINALS:) DIGIT TWO TOKEN*; DIGIT FOUR TOKEN*;
        DIGIT EIGHT TOKEN*; DIGIT SIX TOKEN*.

 EW35 ACTUAL SPECIFICATION LIST+-
    (TERMINALS:) INTEGRAL DENOTATION.
    (NONTERMINALS:) ACTUAL SPECIFICATION LIST*;
        ACTUAL SPECIFICATION*.

 EW36 ACTUAL SPECIFICATION-
    (TERMINALS:) INTEGRAL DENOTATION*.
```

```
MAY BE THE END OF  (EO): B - AB.


  EO1 LETTER A TOKEN-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
          PATTERN; CHARACTER PATTERN*; STRING PATTERN*;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE.

  EO2 LETTER B TOKEN, BOOLEAN CHOICE MOULD-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
          PATTERN; BOOLEAN PATTERN*.

  EO3 LETTER C TOKEN, LETTER E TOKEN, LETTER F TOKEN,
      LETTER I TOKEN, LETTER N TOKEN, LETTER S TOKEN,
      DIGIT ONE TOKEN, OPEN TOKEN, BEGIN TOKEN,
      STROP CASE START TOKEN, STROP CONDITION START TOKEN,
      INNER CLAUSE, REPLICATOR, SIGN MOULD, STAGNANT MOULD,
      RADIX MOULD, RADIX-

  EO4 LETTER D TOKEN, LETTER Z TOKEN-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
          PATTERN; INTEGRAL PATTERN; REAL PATTERN;
          COMPLEX PATTERN; BITS PATTERN; INTEGRAL MOULD;
          DIGIT FRAME*; REAL MOULD; FLOATING POINT MOULD;
          STAGNANT MOULD.

  EO5 LETTER G TOKEN, ACTUAL SPECIFICATION PACK-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
          PATTERN; GENERAL PATTERN*.

  EO6 LETTER K TOKEN, LETTER L TOKEN, LETTER P TOKEN,
      LETTER X TOKEN, LETTER Y TOKEN-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
          INSERTION; INSERT SEQUENCE; INSERT; ALIGNMENT*.

  EO7 LETTER R TOKEN-
     (NONTERMINALS:) RADIX MOULD*.

  EO8 LETTER T TOKEN-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
          PATTERN; STRING PATTERN*.

  EO9 DIGIT TWO TOKEN, DIGIT FOUR TOKEN, DIGIT EIGHT TOKEN,
      DIGIT SIX TOKEN-
     (NONTERMINALS:) RADIX*.

 EO10 PLUS TOKEN, MINUS TOKEN-
     (NONTERMINALS:) SIGN MOULD; SIGN FRAME*.

 EO11 POINT TOKEN-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
          PATTERN; REAL PATTERN; COMPLEX PATTERN; REAL MOULD*;
          STAGNANT MOULD.

 EO12 CLOSE TOKEN-
     (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
          REPLICATOR; COLLECTION LIST PACK*; PATTERN;
```

```
MAY BE THE END OF  (EO): B - AB.

        REPLICATION; DYNAMIC REPLICATION; ENCLOSED CLAUSE*;
        INTEGRAL PATTERN; BOOLEAN PATTERN; GENERAL PATTERN;
        INCLUDED PATTERN; INTEGRAL CHOICE PATTERN;
        LITERAL LIST PACK*; BOOLEAN CHOICE MOULD*;
        ACTUAL SPECIFICATION PACK*.

 E013 COMMA TOKEN, COLLECTION LIST+, COLLECTION-
    (NONTERMINALS:) COLLECTION LIST*.

 E014 END TOKEN, STROP CASE FINISH TOKEN,
     STROP CONDITION FINISH TOKEN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        REPLICATOR; PATTERN; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE*;
        INCLUDED PATTERN.

 E015 INTEGRAL DENOTATION-
    (NONTERMINALS:) REPLICATOR; REPLICATION*;
        ACTUAL SPECIFICATION LIST; ACTUAL SPECIFICATION*.

 E016 CHARACTER DENOTATION, ROW OF CHARACTER DENOTATION-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        INSERTION; LITERAL; INSERT SEQUENCE; INSERT;
        STRING DENOTATION*; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; LITERAL LIST.

 E017 PICTURE, COLLECTION LIST PACK-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION*.

 E018 INSERTION, PATTERN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION*; PICTURE*.

 E019 LITERAL-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        INSERTION*; INSERT SEQUENCE; INSERT*; LITERAL LIST*.

 E020 INSERT SEQUENCE+, INSERT-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        INSERTION*; INSERT SEQUENCE*.

 E021 ALIGNMENT-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        INSERTION; INSERT SEQUENCE; INSERT*.

 E022 REPLICATION-
    (NONTERMINALS:) REPLICATOR*.

 E023 DYNAMIC REPLICATION-
    (NONTERMINALS:) REPLICATOR; REPLICATION*.

 E024 ENCLOSED CLAUSE-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        REPLICATOR; PATTERN; REPLICATION;
        DYNAMIC REPLICATION*; INCLUDED PATTERN*.

 E025 STRING DENOTATION-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
```

```
MAY BE THE END OF  (EO): B - AB.

        INSERTION; LITERAL*; INSERT SEQUENCE; INSERT;
        REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL*;
        LITERAL LIST.

 E026 REPLICATED LITERAL SEQUENCE+, REPLICATED LITERAL-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        INSERTION; LITERAL*; INSERT SEQUENCE; INSERT;
        REPLICATED LITERAL SEQUENCE*; LITERAL LIST.

 E027 SIGN FRAME-
    (NONTERMINALS:) SIGN MOULD*.

 E028 INTEGRAL PATTERN, BOOLEAN PATTERN, CHARACTER PATTERN,
     COMPLEX PATTERN, STRING PATTERN, BITS PATTERN,
     GENERAL PATTERN, INCLUDED PATTERN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN*.

 E029 REAL PATTERN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN*; COMPLEX PATTERN*.

 E030 INTEGRAL MOULD+, DIGIT FRAME-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; INTEGRAL PATTERN*; REAL PATTERN;
        COMPLEX PATTERN; BITS PATTERN*; INTEGRAL MOULD*;
        REAL MOULD*; FLOATING POINT MOULD*; STAGNANT MOULD*.

 E031 INTEGRAL CHOICE PATTERN-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; INTEGRAL PATTERN*.

 E032 LITERAL LIST PACK-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; INTEGRAL PATTERN; INTEGRAL CHOICE PATTERN*.

 E033 LITERAL LIST+-
    (NONTERMINALS:) LITERAL LIST*.

 E034 REAL MOULD-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; REAL PATTERN*; COMPLEX PATTERN;
        STAGNANT MOULD*.

 E035 FLOATING POINT MOULD-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; REAL PATTERN*; COMPLEX PATTERN.

 E036 LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME,
     LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE+-
    (NONTERMINALS:) COLLECTION LIST; COLLECTION; PICTURE;
        PATTERN; STRING PATTERN*;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE*.

 E037 ACTUAL SPECIFICATION LIST+, ACTUAL SPECIFICATION-
    (NONTERMINALS:) ACTUAL SPECIFICATION LIST*.
```

MAY BE THE END OF (EO): B - AB.

```
MAY FOLLOW  (MF): B - A.


  MF1 LETTER A TOKEN+,
      LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME+-
    (TERMINALS:) LETTER A TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        CLOSE TOKEN*; COMMA TOKEN*; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE*.

  MF2 LETTER B TOKEN-
    (TERMINALS:) LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER X TOKEN;
        LETTER Y TOKEN; OPEN TOKEN; CLOSE TOKEN*;
        COMMA TOKEN*; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION;
        BOOLEAN CHOICE MOULD*.

  MF3 LETTER C TOKEN-
    (TERMINALS:) OPEN TOKEN.
    (NONTERMINALS:) LITERAL LIST PACK*.

  MF4 LETTER D TOKEN+, DIGIT FRAME+-
    (TERMINALS:) LETTER D TOKEN; LETTER E TOKEN*;
        LETTER I TOKEN*; LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER S TOKEN*;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
        POINT TOKEN*; CLOSE TOKEN*; COMMA TOKEN*;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION;
        INTEGRAL MOULD*; DIGIT FRAME.

  MF5 LETTER E TOKEN-
    (TERMINALS:) LETTER D TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; PLUS TOKEN; MINUS TOKEN;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION; SIGN MOULD*;
        SIGN FRAME; INTEGRAL MOULD*; DIGIT FRAME.

  MF6 LETTER F TOKEN, LETTER N TOKEN-
    (TERMINALS:) OPEN TOKEN; BEGIN TOKEN;
```

```
MAY FOLLOW  (MF): B - A.

        STROP CASE START TOKEN; STROP CONDITION START TOKEN.
    (NONTERMINALS:) ENCLOSED CLAUSE*.

  MF7 LETTER G TOKEN-
    (TERMINALS:) LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER X TOKEN;
        LETTER Y TOKEN; OPEN TOKEN; CLOSE TOKEN*;
        COMMA TOKEN*; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION;
        ACTUAL SPECIFICATION PACK*.

  MF8 LETTER I TOKEN-
    (TERMINALS:) LETTER D TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; PLUS TOKEN; MINUS TOKEN; POINT TOKEN;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION; SIGN MOULD;
        SIGN FRAME; REAL PATTERN*; INTEGRAL MOULD;
        DIGIT FRAME; REAL MOULD; FLOATING POINT MOULD;
        STAGNANT MOULD.

  MF9 LETTER K TOKEN+, LETTER L TOKEN+, LETTER P TOKEN+,
     LETTER X TOKEN+, LETTER Y TOKEN+, ALIGNMENT+-
    (TERMINALS:) LETTER A TOKEN*; LETTER B TOKEN*;
        LETTER C TOKEN*; LETTER D TOKEN; LETTER E TOKEN*;
        LETTER F TOKEN*; LETTER G TOKEN*; LETTER I TOKEN*;
        LETTER K TOKEN; LETTER L TOKEN; LETTER N TOKEN;
        LETTER P TOKEN; LETTER S TOKEN*; LETTER T TOKEN*;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN*;
        DIGIT TWO TOKEN; DIGIT FOUR TOKEN; DIGIT EIGHT TOKEN;
        DIGIT ONE TOKEN; PLUS TOKEN; MINUS TOKEN;
        POINT TOKEN*; OPEN TOKEN; CLOSE TOKEN*; COMMA TOKEN*;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) REPLICATOR*; COLLECTION LIST PACK*;
        LITERAL*; INSERT SEQUENCE*; INSERT; ALIGNMENT;
        REPLICATION*; DYNAMIC REPLICATION; STRING DENOTATION;
        SIGN FRAME*; DIGIT FRAME*; RADIX*.

 MF10 LETTER R TOKEN, RADIX MOULD-
    (TERMINALS:) LETTER D TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION;
        INTEGRAL MOULD*; DIGIT FRAME.
```

```
MAY FOLLOW  (MF): B - A.

 MF11 LETTER S TOKEN-
    (TERMINALS:) LETTER A TOKEN*; LETTER D TOKEN;
        LETTER E TOKEN*; LETTER I TOKEN*; LETTER Z TOKEN;
        POINT TOKEN*.
    (NONTERMINALS:) DIGIT FRAME*.

 MF12 LETTER T TOKEN, COLLECTION LIST PACK, PATTERN,
     INTEGRAL PATTERN, BOOLEAN PATTERN, CHARACTER PATTERN,
     COMPLEX PATTERN, STRING PATTERN, BITS PATTERN,
     GENERAL PATTERN, INCLUDED PATTERN,
     INTEGRAL CHOICE PATTERN, LITERAL LIST PACK,
     BOOLEAN CHOICE MOULD,
     LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE,
     ACTUAL SPECIFICATION PACK-
    (TERMINALS:) LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER X TOKEN;
        LETTER Y TOKEN; CLOSE TOKEN*; COMMA TOKEN*;
        INTEGRAL DENOTATION; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION.

 MF13 LETTER Z TOKEN+-
    (TERMINALS:) LETTER D TOKEN; LETTER E TOKEN*;
        LETTER I TOKEN*; LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER S TOKEN*;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
        PLUS TOKEN; MINUS TOKEN; POINT TOKEN*; CLOSE TOKEN*;
        COMMA TOKEN*; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION; SIGN FRAME*;
        INTEGRAL MOULD*; DIGIT FRAME.

 MF14 DIGIT TWO TOKEN, DIGIT FOUR TOKEN, DIGIT EIGHT TOKEN,
     DIGIT SIX TOKEN, RADIX-
    (TERMINALS:) LETTER R TOKEN*.

 MF15 DIGIT ONE TOKEN-
    (TERMINALS:) DIGIT SIX TOKEN*.

 MF16 PLUS TOKEN, MINUS TOKEN, SIGN MOULD, SIGN FRAME-
    (TERMINALS:) LETTER D TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; POINT TOKEN; INTEGRAL DENOTATION*;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION;
        INTEGRAL MOULD*; DIGIT FRAME; REAL MOULD*.

 MF17 POINT TOKEN-
    (TERMINALS:) LETTER D TOKEN; LETTER E TOKEN*;
        LETTER I TOKEN*; LETTER K TOKEN; LETTER L TOKEN;
```

```
MAY FOLLOW  (MF): B - A.

        LETTER N TOKEN; LETTER P TOKEN; LETTER S TOKEN*;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
        CLOSE TOKEN*; COMMA TOKEN*; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION;
        INTEGRAL MOULD*; DIGIT FRAME.

 MF18 OPEN TOKEN+-
    (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
        LETTER C TOKEN; LETTER D TOKEN; LETTER F TOKEN;
        LETTER G TOKEN; LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER S TOKEN;
        LETTER T TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; DIGIT TWO TOKEN; DIGIT FOUR TOKEN;
        DIGIT EIGHT TOKEN; DIGIT ONE TOKEN; PLUS TOKEN;
        MINUS TOKEN; POINT TOKEN; OPEN TOKEN; CLOSE TOKEN*;
        COMMA TOKEN; INNER CLAUSE*; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) COLLECTION LIST*; COLLECTION; PICTURE;
        INSERTION; REPLICATOR; COLLECTION LIST PACK; PATTERN;
        LITERAL*; INSERT SEQUENCE; INSERT; ALIGNMENT;
        REPLICATION; DYNAMIC REPLICATION; STRING DENOTATION;
        SIGN MOULD; SIGN FRAME; INTEGRAL PATTERN;
        REAL PATTERN; BOOLEAN PATTERN; CHARACTER PATTERN;
        COMPLEX PATTERN; STRING PATTERN; BITS PATTERN;
        GENERAL PATTERN; INCLUDED PATTERN; INTEGRAL MOULD;
        INTEGRAL CHOICE PATTERN; DIGIT FRAME; LITERAL LIST*;
        REAL MOULD; FLOATING POINT MOULD; STAGNANT MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
        RADIX MOULD; RADIX; ACTUAL SPECIFICATION LIST*;
        ACTUAL SPECIFICATION.

 MF19 CLOSE TOKEN+, END TOKEN, STROP CASE FINISH TOKEN,
     STROP CONDITION FINISH TOKEN, ENCLOSED CLAUSE-
    (TERMINALS:) LETTER A TOKEN*; LETTER D TOKEN;
        LETTER K TOKEN; LETTER L TOKEN; LETTER N TOKEN;
        LETTER P TOKEN; LETTER S TOKEN*; LETTER X TOKEN;
        LETTER Y TOKEN; LETTER Z TOKEN*; OPEN TOKEN;
        CLOSE TOKEN*; COMMA TOKEN*; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR;
        COLLECTION LIST PACK*; LITERAL; INSERT SEQUENCE;
        INSERT; ALIGNMENT*; REPLICATION; DYNAMIC REPLICATION;
        STRING DENOTATION*; DIGIT FRAME*.

 MF20 BEGIN TOKEN, STROP CASE START TOKEN,
     STROP CONDITION START TOKEN-
    (TERMINALS:) INNER CLAUSE*.

 MF21 COMMA TOKEN+-
    (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
        LETTER C TOKEN; LETTER D TOKEN; LETTER F TOKEN;
        LETTER G TOKEN; LETTER K TOKEN; LETTER L TOKEN;
        LETTER N TOKEN; LETTER P TOKEN; LETTER S TOKEN;
        LETTER T TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
```

```
MAY FOLLOW  (MF): B - A.

        LETTER Z TOKEN; DIGIT TWO TOKEN; DIGIT FOUR TOKEN;
        DIGIT EIGHT TOKEN; DIGIT ONE TOKEN; PLUS TOKEN;
        MINUS TOKEN; POINT TOKEN; OPEN TOKEN; CLOSE TOKEN*;
        COMMA TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) COLLECTION LIST*; COLLECTION; PICTURE;
        INSERTION; REPLICATOR; COLLECTION LIST PACK; PATTERN;
        LITERAL*; INSERT SEQUENCE; INSERT; ALIGNMENT;
        REPLICATION; DYNAMIC REPLICATION; STRING DENOTATION;
        SIGN MOULD; SIGN FRAME; INTEGRAL PATTERN;
        REAL PATTERN; BOOLEAN PATTERN; CHARACTER PATTERN;
        COMPLEX PATTERN; STRING PATTERN; BITS PATTERN;
        GENERAL PATTERN; INCLUDED PATTERN; INTEGRAL MOULD;
        INTEGRAL CHOICE PATTERN; DIGIT FRAME; LITERAL LIST*;
        REAL MOULD; FLOATING POINT MOULD; STAGNANT MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
        RADIX MOULD; RADIX; ACTUAL SPECIFICATION LIST*;
        ACTUAL SPECIFICATION.

 MF22 INNER CLAUSE-
    (TERMINALS:) CLOSE TOKEN*; END TOKEN*;
        STROP CASE FINISH TOKEN*;
        STROP CONDITION FINISH TOKEN*.

 MF23 INTEGRAL DENOTATION-
    (TERMINALS:) LETTER A TOKEN*; LETTER D TOKEN;
        LETTER K TOKEN; LETTER L TOKEN; LETTER P TOKEN;
        LETTER S TOKEN*; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN*; OPEN TOKEN; CLOSE TOKEN*;
        COMMA TOKEN*; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) COLLECTION LIST PACK*; ALIGNMENT*;
        STRING DENOTATION*; DIGIT FRAME*.

 MF24 CHARACTER DENOTATION, ROW OF CHARACTER DENOTATION,
     STRING DENOTATION, REPLICATED LITERAL*-
    (TERMINALS:) LETTER A TOKEN*; LETTER B TOKEN*;
        LETTER C TOKEN*; LETTER D TOKEN; LETTER E TOKEN*;
        LETTER F TOKEN*; LETTER G TOKEN*; LETTER I TOKEN*;
        LETTER K TOKEN; LETTER L TOKEN; LETTER N TOKEN;
        LETTER P TOKEN; LETTER S TOKEN*; LETTER T TOKEN*;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN*;
        DIGIT TWO TOKEN; DIGIT FOUR TOKEN; DIGIT EIGHT TOKEN;
        DIGIT ONE TOKEN; PLUS TOKEN; MINUS TOKEN;
        POINT TOKEN*; OPEN TOKEN; CLOSE TOKEN*; COMMA TOKEN*;
        INTEGRAL DENOTATION.
    (NONTERMINALS:) REPLICATOR*; COLLECTION LIST PACK*;
        INSERT SEQUENCE*; INSERT; ALIGNMENT; REPLICATION*;
        DYNAMIC REPLICATION; REPLICATED LITERAL SEQUENCE*;
        REPLICATED LITERAL; SIGN FRAME*; DIGIT FRAME*; RADIX*.

 MF25 COLLECTION LIST, LITERAL LIST,
     ACTUAL SPECIFICATION LIST-
    (TERMINALS:) CLOSE TOKEN*.

 MF26 COLLECTION, PICTURE, ACTUAL SPECIFICATION-
    (TERMINALS:) CLOSE TOKEN*; COMMA TOKEN*.
```

```
MAY FOLLOW  (MF): B - A.

 MF27 INSERTION, INSERT SEQUENCE-
    (TERMINALS:) LETTER A TOKEN*; LETTER B TOKEN*;
        LETTER C TOKEN*; LETTER D TOKEN; LETTER E TOKEN*;
        LETTER F TOKEN*; LETTER G TOKEN*; LETTER I TOKEN*;
        LETTER N TOKEN; LETTER S TOKEN*; LETTER T TOKEN*;
        LETTER Z TOKEN*; DIGIT TWO TOKEN; DIGIT FOUR TOKEN;
        DIGIT EIGHT TOKEN; DIGIT ONE TOKEN; PLUS TOKEN;
        MINUS TOKEN; POINT TOKEN*; OPEN TOKEN; CLOSE TOKEN*;
        COMMA TOKEN*; INTEGRAL DENOTATION.
    (NONTERMINALS:) REPLICATOR*; COLLECTION LIST PACK*;
        REPLICATION*; DYNAMIC REPLICATION; SIGN FRAME*;
        DIGIT FRAME*; RADIX*.

 MF28 REPLICATOR, REPLICATION, DYNAMIC REPLICATION-
    (TERMINALS:) LETTER A TOKEN*; LETTER D TOKEN;
        LETTER K TOKEN; LETTER L TOKEN; LETTER P TOKEN;
        LETTER S TOKEN*; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN*; OPEN TOKEN; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) COLLECTION LIST PACK*; ALIGNMENT*;
        STRING DENOTATION*; DIGIT FRAME*.

 MF29 LITERAL, INSERT+, REPLICATED LITERAL SEQUENCE-
    (TERMINALS:) LETTER A TOKEN*; LETTER B TOKEN*;
        LETTER C TOKEN*; LETTER D TOKEN; LETTER E TOKEN*;
        LETTER F TOKEN*; LETTER G TOKEN*; LETTER I TOKEN*;
        LETTER K TOKEN; LETTER L TOKEN; LETTER N TOKEN;
        LETTER P TOKEN; LETTER S TOKEN*; LETTER T TOKEN*;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN*;
        DIGIT TWO TOKEN; DIGIT FOUR TOKEN; DIGIT EIGHT TOKEN;
        DIGIT ONE TOKEN; PLUS TOKEN; MINUS TOKEN;
        POINT TOKEN*; OPEN TOKEN; CLOSE TOKEN*; COMMA TOKEN*;
        INTEGRAL DENOTATION.
    (NONTERMINALS:) REPLICATOR*; COLLECTION LIST PACK*;
        INSERT SEQUENCE*; INSERT; ALIGNMENT; REPLICATION*;
        DYNAMIC REPLICATION; SIGN FRAME*; DIGIT FRAME*;
        RADIX*,

 MF30 REAL PATTERN, FLOATING POINT MOULD-
    (TERMINALS:) LETTER I TOKEN*; LETTER K TOKEN;
        LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
        LETTER S TOKEN*; LETTER X TOKEN; LETTER Y TOKEN;
        CLOSE TOKEN*; COMMA TOKEN*; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; STRING DENOTATION.

 MF31 INTEGRAL MOULD-
    (TERMINALS:) LETTER E TOKEN*; LETTER I TOKEN*;
        LETTER K TOKEN; LETTER L TOKEN; LETTER N TOKEN;
        LETTER P TOKEN; LETTER S TOKEN*; LETTER X TOKEN;
        LETTER Y TOKEN; POINT TOKEN*; CLOSE TOKEN*;
        COMMA TOKEN*; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
```

```
MAY FOLLOW  (MF): B - A.

          INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
          DYNAMIC REPLICATION; STRING DENOTATION.

 MF32 REAL MOULD-
     (TERMINALS:) LETTER E TOKEN*; LETTER I TOKEN*;
          LETTER K TOKEN; LETTER L TOKEN; LETTER N TOKEN;
          LETTER P TOKEN; LETTER S TOKEN*; LETTER X TOKEN;
          LETTER Y TOKEN; CLOSE TOKEN*; COMMA TOKEN*;
          INTEGRAL DENOTATION; CHARACTER DENOTATION;
          ROW OF CHARACTER DENOTATION.
     (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
          INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
          DYNAMIC REPLICATION; STRING DENOTATION.

 MF33 STAGNANT MOULD-
     (TERMINALS:) LETTER E TOKEN*; LETTER K TOKEN;
          LETTER L TOKEN; LETTER N TOKEN; LETTER P TOKEN;
          LETTER S TOKEN*; LETTER X TOKEN; LETTER Y TOKEN;
          INTEGRAL DENOTATION; CHARACTER DENOTATION;
          ROW OF CHARACTER DENOTATION.
     (NONTERMINALS:) INSERTION*; REPLICATOR; LITERAL;
          INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
          DYNAMIC REPLICATION; STRING DENOTATION.
```

```
MAY PRECEDE  (MP): A - B.


  MP1 LETTER A TOKEN+-
    (TERMINALS:) LETTER A TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER P TOKEN; LETTER S TOKEN*;
        LETTER X TOKEN; LETTER Y TOKEN; OPEN TOKEN*;
        CLOSE TOKEN; COMMA TOKEN*; END TOKEN;
        STROP CASE FINISH TOKEN;
        STROP CONDITION FINISH TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR*; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION*;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*.

  MP2 LETTER B TOKEN, LETTER C TOKEN, LETTER F TOKEN,
     LETTER G TOKEN, LETTER T TOKEN, DIGIT TWO TOKEN,
     DIGIT FOUR TOKEN, DIGIT EIGHT TOKEN, DIGIT ONE TOKEN,
     RADIX-
    (TERMINALS:) LETTER K TOKEN; LETTER L TOKEN;
        LETTER P TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        OPEN TOKEN*; COMMA TOKEN*; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; LITERAL; INSERT SEQUENCE;
        INSERT; ALIGNMENT; STRING DENOTATION;
        REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL.

  MP3 LETTER D TOKEN+, LETTER Z TOKEN+, DIGIT FRAME+-
    (TERMINALS:) LETTER D TOKEN; LETTER E TOKEN*;
        LETTER I TOKEN*; LETTER K TOKEN; LETTER L TOKEN;
        LETTER P TOKEN; LETTER R TOKEN; LETTER S TOKEN*;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
        PLUS TOKEN; MINUS TOKEN; POINT TOKEN*; OPEN TOKEN*;
        CLOSE TOKEN; COMMA TOKEN*; END TOKEN;
        STROP CASE FINISH TOKEN;
        STROP CONDITION FINISH TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR*; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; SIGN MOULD*; SIGN FRAME;
        DIGIT FRAME*; RADIX MOULD*.

  MP4 LETTER E TOKEN-
    (TERMINALS:) LETTER D TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER P TOKEN; LETTER S TOKEN*;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
        POINT TOKEN; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; LITERAL; INSERT SEQUENCE;
        INSERT; ALIGNMENT; STRING DENOTATION;
        REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL;
        INTEGRAL MOULD; DIGIT FRAME; REAL MOULD;
        STAGNANT MOULD*.
```

```
MAY PRECEDE  (MP): A - B.


  MP5 LETTER I TOKEN-
    (TERMINALS:) LETTER D TOKEN; LETTER K TOKEN;
        LETTER L TOKEN; LETTER P TOKEN; LETTER S TOKEN*;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
        POINT TOKEN; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; LITERAL; INSERT SEQUENCE;
        INSERT; ALIGNMENT; STRING DENOTATION;
        REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL;
        REAL PATTERN*; INTEGRAL MOULD; DIGIT FRAME;
        REAL MOULD; FLOATING POINT MOULD.

  MP6 LETTER K TOKEN+, LETTER L TOKEN+, LETTER P TOKEN+,
     LETTER X TOKEN+, LETTER Y TOKEN+, ALIGNMENT+-
    (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
        LETTER D TOKEN; LETTER E TOKEN*; LETTER G TOKEN;
        LETTER I TOKEN*; LETTER K TOKEN; LETTER L TOKEN;
        LETTER P TOKEN; LETTER R TOKEN; LETTER T TOKEN;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
        PLUS TOKEN; MINUS TOKEN; POINT TOKEN*; OPEN TOKEN*;
        CLOSE TOKEN; COMMA TOKEN*; END TOKEN;
        STROP CASE FINISH TOKEN;
        STROP CONDITION FINISH TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) REPLICATOR*; COLLECTION LIST PACK*;
        PATTERN*; LITERAL*; INSERT*; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; SIGN MOULD*; SIGN FRAME;
        INTEGRAL PATTERN; REAL PATTERN*; BOOLEAN PATTERN;
        CHARACTER PATTERN; COMPLEX PATTERN; STRING PATTERN;
        BITS PATTERN; GENERAL PATTERN; INCLUDED PATTERN;
        INTEGRAL MOULD*; INTEGRAL CHOICE PATTERN;
        DIGIT FRAME*; LITERAL LIST PACK; REAL MOULD;
        FLOATING POINT MOULD; STAGNANT MOULD*;
        BOOLEAN CHOICE MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
        RADIX MOULD*; ACTUAL SPECIFICATION PACK.

  MP7 LETTER N TOKEN, INTEGRAL DENOTATION, REPLICATOR,
     REPLICATION, DYNAMIC REPLICATION-
    (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
        LETTER D TOKEN; LETTER E TOKEN*; LETTER G TOKEN;
        LETTER I TOKEN*; LETTER K TOKEN; LETTER L TOKEN;
        LETTER P TOKEN; LETTER R TOKEN; LETTER T TOKEN;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
        PLUS TOKEN; MINUS TOKEN*; POINT TOKEN*; OPEN TOKEN*;
        CLOSE TOKEN; COMMA TOKEN*; END TOKEN;
        STROP CASE FINISH TOKEN;
        STROP CONDITION FINISH TOKEN; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; COLLECTION LIST PACK*;
        PATTERN*; LITERAL*; INSERT SEQUENCE; INSERT*;
        ALIGNMENT*; ENCLOSED CLAUSE; STRING DENOTATION*;
        REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL*;
```

```
MAY PRECEDE  (MP): A - B.

        SIGN MOULD*; SIGN FRAME; INTEGRAL PATTERN;
        REAL PATTERN*; BOOLEAN PATTERN; CHARACTER PATTERN;
        COMPLEX PATTERN; STRING PATTERN; BITS PATTERN;
        GENERAL PATTERN; INCLUDED PATTERN; INTEGRAL MOULD*;
        INTEGRAL CHOICE PATTERN; DIGIT FRAME*;
        LITERAL LIST PACK; REAL MOULD; FLOATING POINT MOULD;
        STAGNANT MOULD*; BOOLEAN CHOICE MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
        RADIX MOULD*; ACTUAL SPECIFICATION PACK.

  MP8 LETTER R TOKEN-
    (TERMINALS:) DIGIT TWO TOKEN; DIGIT FOUR TOKEN;
        DIGIT EIGHT TOKEN; DIGIT SIX TOKEN.
    (NONTERMINALS:) RADIX*.

  MP9 LETTER S TOKEN-
    (TERMINALS:) LETTER A TOKEN; LETTER D TOKEN;
        LETTER E TOKEN*; LETTER I TOKEN*; LETTER K TOKEN;
        LETTER L TOKEN; LETTER P TOKEN; LETTER R TOKEN;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
        PLUS TOKEN; MINUS TOKEN; POINT TOKEN*; OPEN TOKEN*;
        CLOSE TOKEN; COMMA TOKEN*; END TOKEN;
        STROP CASE FINISH TOKEN;
        STROP CONDITION FINISH TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR*; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION*;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; SIGN MOULD*; SIGN FRAME;
        REAL PATTERN*; INTEGRAL MOULD*; DIGIT FRAME*;
        REAL MOULD; FLOATING POINT MOULD; STAGNANT MOULD*;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
        RADIX MOULD*.

 MP10 DIGIT SIX TOKEN-
    (TERMINALS:) DIGIT ONE TOKEN*.

 MP11 PLUS TOKEN, MINUS TOKEN, SIGN FRAME-
    (TERMINALS:) LETTER E TOKEN*; LETTER I TOKEN*;
        LETTER K TOKEN; LETTER L TOKEN; LETTER P TOKEN;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN*;
        OPEN TOKEN*; COMMA TOKEN*; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; LITERAL; INSERT SEQUENCE;
        INSERT; ALIGNMENT; STRING DENOTATION;
        REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL.

 MP12 POINT TOKEN-
    (TERMINALS:) LETTER D TOKEN; LETTER I TOKEN*;
        LETTER K TOKEN; LETTER L TOKEN; LETTER P TOKEN;
        LETTER S TOKEN*; LETTER X TOKEN; LETTER Y TOKEN;
        LETTER Z TOKEN; PLUS TOKEN; MINUS TOKEN; OPEN TOKEN*;
        COMMA TOKEN*; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; LITERAL; INSERT SEQUENCE;
```

```
MAY PRECEDE  (MP): A - B.

          INSERT; ALIGNMENT; STRING DENOTATION;
          REPLICATED LITERAL SEQUENCE; REPLICATED LITERAL;
          SIGN MOULD*; SIGN FRAME; INTEGRAL MOULD*;
          DIGIT FRAME.

 MP13 OPEN TOKEN+-
     (TERMINALS:) LETTER B TOKEN*; LETTER C TOKEN*;
          LETTER F TOKEN*; LETTER G TOKEN*; LETTER K TOKEN;
          LETTER L TOKEN; LETTER N TOKEN*; LETTER P TOKEN;
          LETTER X TOKEN; LETTER Y TOKEN; OPEN TOKEN*;
          CLOSE TOKEN; COMMA TOKEN*; END TOKEN;
          STROP CASE FINISH TOKEN;
          STROP CONDITION FINISH TOKEN; INTEGRAL DENOTATION;
          CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
     (NONTERMINALS:) INSERTION*; REPLICATOR*; LITERAL;
          INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
          DYNAMIC REPLICATION; ENCLOSED CLAUSE;
          STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
          REPLICATED LITERAL.

 MP14 CLOSE TOKEN+-
     (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
          LETTER D TOKEN; LETTER G TOKEN; LETTER K TOKEN;
          LETTER L TOKEN; LETTER P TOKEN; LETTER T TOKEN;
          LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
          POINT TOKEN; OPEN TOKEN*; CLOSE TOKEN; COMMA TOKEN;
          END TOKEN; STROP CASE FINISH TOKEN;
          STROP CONDITION FINISH TOKEN; INNER CLAUSE*;
          INTEGRAL DENOTATION; CHARACTER DENOTATION;
          ROW OF CHARACTER DENOTATION.
     (NONTERMINALS:) COLLECTION LIST*; COLLECTION; PICTURE;
          INSERTION; COLLECTION LIST PACK; PATTERN; LITERAL*;
          INSERT SEQUENCE; INSERT; ALIGNMENT; ENCLOSED CLAUSE;
          STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
          REPLICATED LITERAL; INTEGRAL PATTERN; REAL PATTERN;
          BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
          STRING PATTERN; BITS PATTERN; GENERAL PATTERN;
          INCLUDED PATTERN; INTEGRAL MOULD;
          INTEGRAL CHOICE PATTERN; DIGIT FRAME;
          LITERAL LIST PACK; LITERAL LIST*; REAL MOULD;
          FLOATING POINT MOULD; BOOLEAN CHOICE MOULD;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
          ACTUAL SPECIFICATION PACK; ACTUAL SPECIFICATION LIST*;
          ACTUAL SPECIFICATION.

 MP15 BEGIN TOKEN, STROP CASE START TOKEN,
      STROP CONDITION START TOKEN, ENCLOSED CLAUSE-
     (TERMINALS:) LETTER F TOKEN*; LETTER N TOKEN*.

 MP16 COMMA TOKEN+-
     (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
          LETTER D TOKEN; LETTER G TOKEN; LETTER K TOKEN;
          LETTER L TOKEN; LETTER P TOKEN; LETTER T TOKEN;
          LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
          POINT TOKEN; OPEN TOKEN*; CLOSE TOKEN; COMMA TOKEN*;
          END TOKEN; STROP CASE FINISH TOKEN;
```

```
MAY PRECEDE  (MP): A - B.

          STROP CONDITION FINISH TOKEN; INTEGRAL DENOTATION;
          CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
     (NONTERMINALS:) COLLECTION*; PICTURE; INSERTION;
          COLLECTION LIST PACK; PATTERN; LITERAL*;
          INSERT SEQUENCE; INSERT; ALIGNMENT; ENCLOSED CLAUSE;
          STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
          REPLICATED LITERAL; INTEGRAL PATTERN; REAL PATTERN;
          BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
          STRING PATTERN; BITS PATTERN; GENERAL PATTERN;
          INCLUDED PATTERN; INTEGRAL MOULD;
          INTEGRAL CHOICE PATTERN; DIGIT FRAME;
          LITERAL LIST PACK; REAL MOULD; FLOATING POINT MOULD;
          BOOLEAN CHOICE MOULD;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
          ACTUAL SPECIFICATION PACK; ACTUAL SPECIFICATION*.

 MP17 END TOKEN, STROP CASE FINISH TOKEN,
      STROP CONDITION FINISH TOKEN-
     (TERMINALS:) INNER CLAUSE*.

 MP18 INNER CLAUSE-
     (TERMINALS:) OPEN TOKEN*; BEGIN TOKEN*;
          STROP CASE START TOKEN*; STROP CONDITION START TOKEN*.

 MP19 CHARACTER DENOTATION, ROW OF CHARACTER DENOTATION,
      STRING DENOTATION-
     (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
          LETTER D TOKEN; LETTER E TOKEN*; LETTER G TOKEN;
          LETTER I TOKEN*; LETTER K TOKEN; LETTER L TOKEN;
          LETTER P TOKEN; LETTER R TOKEN; LETTER T TOKEN;
          LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
          PLUS TOKEN; MINUS TOKEN; POINT TOKEN*; OPEN TOKEN*;
          CLOSE TOKEN; COMMA TOKEN*; END TOKEN;
          STROP CASE FINISH TOKEN;
          STROP CONDITION FINISH TOKEN; INTEGRAL DENOTATION.
     (NONTERMINALS:) REPLICATOR*; COLLECTION LIST PACK*;
          PATTERN*; ALIGNMENT*; REPLICATION*;
          DYNAMIC REPLICATION; ENCLOSED CLAUSE; SIGN MOULD*;
          SIGN FRAME; INTEGRAL PATTERN; REAL PATTERN*;
          BOOLEAN PATTERN; CHARACTER PATTERN; COMPLEX PATTERN;
          STRING PATTERN; BITS PATTERN; GENERAL PATTERN;
          INCLUDED PATTERN; INTEGRAL MOULD*;
          INTEGRAL CHOICE PATTERN; DIGIT FRAME*;
          LITERAL LIST PACK; REAL MOULD; FLOATING POINT MOULD;
          STAGNANT MOULD*; BOOLEAN CHOICE MOULD;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
          LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
          RADIX MOULD*; ACTUAL SPECIFICATION PACK.

 MP20 COLLECTION LIST, COLLECTION, PICTURE, PATTERN,
      INTEGRAL PATTERN, BOOLEAN PATTERN, CHARACTER PATTERN,
      COMPLEX PATTERN, STRING PATTERN, BITS PATTERN,
      GENERAL PATTERN, INCLUDED PATTERN,
      INTEGRAL CHOICE PATTERN, LITERAL LIST, RADIX MOULD,
      ACTUAL SPECIFICATION LIST, ACTUAL SPECIFICATION-
     (TERMINALS:) OPEN TOKEN*; COMMA TOKEN*.
```

```
MAY PRECEDE  (MP): A - B.


 MP21 INSERTION-
    (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
        LETTER D TOKEN; LETTER E TOKEN*; LETTER G TOKEN;
        LETTER I TOKEN*; LETTER R TOKEN; LETTER T TOKEN;
        LETTER Z TOKEN; PLUS TOKEN; MINUS TOKEN; POINT TOKEN*;
        OPEN TOKEN*; CLOSE TOKEN; COMMA TOKEN*; END TOKEN;
        STROP CASE FINISH TOKEN;
        STROP CONDITION FINISH TOKEN.
    (NONTERMINALS:) COLLECTION LIST PACK*; PATTERN*;
        ENCLOSED CLAUSE; SIGN MOULD*; SIGN FRAME;
        INTEGRAL PATTERN; REAL PATTERN*; BOOLEAN PATTERN;
        CHARACTER PATTERN; COMPLEX PATTERN; STRING PATTERN;
        BITS PATTERN; GENERAL PATTERN; INCLUDED PATTERN;
        INTEGRAL MOULD*; INTEGRAL CHOICE PATTERN;
        DIGIT FRAME*; LITERAL LIST PACK; REAL MOULD;
        FLOATING POINT MOULD; STAGNANT MOULD*;
        BOOLEAN CHOICE MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
        RADIX MOULD*; ACTUAL SPECIFICATION PACK.

 MP22 COLLECTION LIST PACK-
    (TERMINALS:) LETTER K TOKEN; LETTER L TOKEN;
        LETTER P TOKEN; LETTER X TOKEN; LETTER Y TOKEN;
        OPEN TOKEN*; CLOSE TOKEN; COMMA TOKEN*; END TOKEN;
        STROP CASE FINISH TOKEN;
        STROP CONDITION FINISH TOKEN; INTEGRAL DENOTATION;
        CHARACTER DENOTATION; ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) INSERTION*; REPLICATOR*; LITERAL;
        INSERT SEQUENCE; INSERT; ALIGNMENT; REPLICATION;
        DYNAMIC REPLICATION; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL.

 MP23 LITERAL-
    (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
        LETTER D TOKEN; LETTER E TOKEN*; LETTER G TOKEN;
        LETTER I TOKEN*; LETTER K TOKEN; LETTER L TOKEN;
        LETTER P TOKEN; LETTER R TOKEN; LETTER T TOKEN;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
        PLUS TOKEN; MINUS TOKEN; POINT TOKEN*; OPEN TOKEN*;
        CLOSE TOKEN; COMMA TOKEN*; END TOKEN;
        STROP CASE FINISH TOKEN;
        STROP CONDITION FINISH TOKEN.
    (NONTERMINALS:) COLLECTION LIST PACK*; PATTERN*;
        ALIGNMENT*; ENCLOSED CLAUSE; SIGN MOULD*; SIGN FRAME;
        INTEGRAL PATTERN; REAL PATTERN*; BOOLEAN PATTERN;
        CHARACTER PATTERN; COMPLEX PATTERN; STRING PATTERN;
        BITS PATTERN; GENERAL PATTERN; INCLUDED PATTERN;
        INTEGRAL MOULD*; INTEGRAL CHOICE PATTERN;
        DIGIT FRAME*; LITERAL LIST PACK; REAL MOULD;
        FLOATING POINT MOULD; STAGNANT MOULD*;
        BOOLEAN CHOICE MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
        RADIX MOULD*; ACTUAL SPECIFICATION PACK.
```

```
MAY PRECEDE  (MP): A - B.


 MP24 INSERT SEQUENCE, INSERT+-
    (TERMINALS:) LETTER A TOKEN; LETTER B TOKEN;
        LETTER D TOKEN; LETTER E TOKEN*; LETTER G TOKEN;
        LETTER I TOKEN*; LETTER K TOKEN; LETTER L TOKEN;
        LETTER P TOKEN; LETTER R TOKEN; LETTER T TOKEN;
        LETTER X TOKEN; LETTER Y TOKEN; LETTER Z TOKEN;
        PLUS TOKEN; MINUS TOKEN; POINT TOKEN*; OPEN TOKEN*;
        CLOSE TOKEN; COMMA TOKEN*; END TOKEN;
        STROP CASE FINISH TOKEN;
        STROP CONDITION FINISH TOKEN; CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) COLLECTION LIST PACK*; PATTERN*; LITERAL*;
        INSERT*; ALIGNMENT; ENCLOSED CLAUSE;
        STRING DENOTATION; REPLICATED LITERAL SEQUENCE;
        REPLICATED LITERAL; SIGN MOULD*; SIGN FRAME;
        INTEGRAL PATTERN; REAL PATTERN*; BOOLEAN PATTERN;
        CHARACTER PATTERN; COMPLEX PATTERN; STRING PATTERN;
        BITS PATTERN; GENERAL PATTERN; INCLUDED PATTERN;
        INTEGRAL MOULD*; INTEGRAL CHOICE PATTERN;
        DIGIT FRAME*; LITERAL LIST PACK; REAL MOULD;
        FLOATING POINT MOULD; STAGNANT MOULD*;
        BOOLEAN CHOICE MOULD;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*;
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE;
        RADIX MOULD*; ACTUAL SPECIFICATION PACK.

 MP25 REPLICATED LITERAL SEQUENCE, REPLICATED LITERAL+-
    (TERMINALS:) CHARACTER DENOTATION;
        ROW OF CHARACTER DENOTATION.
    (NONTERMINALS:) STRING DENOTATION*; REPLICATED LITERAL*.

 MP26 SIGN MOULD-
    (TERMINALS:) LETTER E TOKEN*; LETTER I TOKEN*;
        OPEN TOKEN*; COMMA TOKEN*.

 MP27 REAL PATTERN, FLOATING POINT MOULD, STAGNANT MOULD-
    (TERMINALS:) LETTER I TOKEN*; OPEN TOKEN*; COMMA TOKEN*.

 MP28 INTEGRAL MOULD-
    (TERMINALS:) LETTER D TOKEN; LETTER E TOKEN*;
        LETTER I TOKEN*; LETTER R TOKEN; LETTER Z TOKEN;
        PLUS TOKEN; MINUS TOKEN; POINT TOKEN*; OPEN TOKEN*;
        COMMA TOKEN*.
    (NONTERMINALS:) SIGN MOULD*; SIGN FRAME; DIGIT FRAME*;
        RADIX MOULD*.

 MP29 LITERAL LIST PACK-
    (TERMINALS:) LETTER C TOKEN*.

 MP30 REAL MOULD-
    (TERMINALS:) LETTER I TOKEN*; PLUS TOKEN; MINUS TOKEN;
        OPEN TOKEN*; COMMA TOKEN*.
    (NONTERMINALS:) SIGN MOULD*; SIGN FRAME.

 MP31 BOOLEAN CHOICE MOULD-
    (TERMINALS:) LETTER B TOKEN*.
```

```
MAY PRECEDE  (MP): A - B.

 MP32 LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*-
    (TERMINALS:) LETTER A TOKEN; OPEN TOKEN*; COMMA TOKEN*.
    (NONTERMINALS:)
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*.

 MP33
     LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE-
    (TERMINALS:) LETTER A TOKEN.
    (NONTERMINALS:)
        LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME*.

 MP34 ACTUAL SPECIFICATION PACK-
    (TERMINALS:) LETTER G TOKEN*.
```

INDEX

ACTUAL SPECIFICATION MC40,CI45,BW40,BO47,EW36,EO37,MF26,MP20
ACTUAL SPECIFICATION LIST MC39,CI45,BW39,BO5,EW35,EO37,MF25,MP20
ACTUAL SPECIFICATION PACK MC38,CI7,BW7,BO5,EW7,EO5,MF12,MP34
ALIGNMENT MC9,CI30,BW12,BO29,EW11,EO21,MF9,MP6
BEGIN TOKEN MC1,CI20,BW1,BO18,EW1,EO3,MF20,MP15
BITS PATTERN MC24,CI37,BW26,BO36,EW25,EO28,MF12,MP20
BOOLEAN CHOICE MOULD MC34,CI2,BW7,BO5,EW7,EO2,MF12.MP31
BOOLEAN PATTERN MC20,CI37,BW22,BO36,EW21,EO28,MF12,MP20
CHARACTER DENOTATION MC1,CI23,BW1,BO21,EW1,EO16,MF24,MP19
CHARACTER PATTERN MC21,CI37,BW23,BO36,EW22,EO28,MF12,MP20
CLOSE TOKEN MC1,CI19,BW1,BO5,EW1,EO12,MF19,MP14
COLLECTION LIST PACK MC2,CI24,BW7,BO22,EW7,EO17,MF12,MP22
COLLECTION MC2,CI24,BW3,BO19,EW3,EO13,MF26,MP20
COLLECTION LIST MC2,CI24,BW2,BO5,EW2,EO13,MF25,MP20
COMMA TOKEN MC1,CI21,BW1,BO19,EW1,EO13,MF21,MP16
COMPLEX PATTERN MC22,CI37,BW24,BO36,EW23,EO28,MF12,MP20
DIGIT EIGHT TOKEN MC1,CI15,BW1,BO13,EW1,EO9,MF14,MP2
DIGIT FOUR TOKEN MC1,CI15,BW1,BO13,EW1,EO9,MF14,MP2
DIGIT FRAME MC29,CI38,BW31,BO40,EW28,EO30,MF4,MP3
DIGIT ONE TOKEN MC1,CI15,BW1,BO13,EW1,EO3,MF15,MP2
DIGIT SIX TOKEN MC1,CI15,BW1,BO5,EW1,EO9,MF14,MP10
DIGIT TWO TOKEN MC1,CI15,BW1,BO13,EW1,EO9,MF14,MP2
DYNAMIC REPLICATION MC11,CI32,BW14,BO31,EW13,EO23,MF28,MP7
ENCLOSED CLAUSE MC12,CI10,BW15,BO5,EW14,EO24,MF19,MP15
END TOKEN MC1,CI20,BW1,BO5,EW1,EO14,MF19,MP17
FLOATING POINT MOULD MC32,CI42,BW34,BO42,EW25,EO35,MF30,MP27
GENERAL PATTERN MC25,CI37,BW27,BO36,EW26,EO28,MF12,MP20
INCLUDED PATTERN MC26,CI37,BW28,BO36,EW13,EO28,MF12,MP20
INNER CLAUSE MC1,CI20,BW1,BO5,EW1,EO3,MF22,MP18
INSERT MC8,CI29,BW11,BO28,EW10,EO20,MF29,MP24
INSERTION MC4,CI25,BW5,BO23,EW5,EO18,MF27,MP21
INSERT SEQUENCE MC4,CI29,BW10,BO27,EW5,EO20,MF27,MP24
INTEGRAL CHOICE PATTERN MC28,CI39,BW30,BO39,EW27,EO31,MF12,MP20
INTEGRAL DENOTATION MC1,CI22,BW1,BO20,EW1,EO15,MF23,MP7
INTEGRAL MOULD MC27,CI38,BW29,BO38,EW25,EO30,MF31,MP28
INTEGRAL PATTERN MC18,CI37,BW20,BO36,EW19,EO28,MF12,MP20
LETTER A TOKEN MC1,CI1,BW1,BO1,EW1,EO1,MF1,MP1
LETTER B TOKEN MC1,CI2,BW1,BO2,EW1,EO2,MF2,MP2
LETTER C TOKEN MC1,CI3,BW1,BO3,EW1,EO3,MF3,MP2
LETTER D TOKEN MC1,CI4,BW1,BO4,EW1,EO4,MF4,MP3
LETTER E TOKEN MC1,CI5,BW1,BO5,EW1,EO3,MF5,MP4
LETTER F TOKEN MC1,CI6,BW1,BO6,EW1,EO3,MF6,MP2
LETTER G TOKEN MC1,CI7,BW1,BO7,EW1,EO5,MF7,MP2
LETTER I TOKEN MC1,CI8,BW1,BO5,EW1,EO3,MF8,MP5
LETTER K TOKEN MC1,CI9,BW1,BO8,EW1,EO6,MF9,MP6
LETTER L TOKEN MC1,CI9,BW1,BO8,EW1,EO6,MF9,MP6
LETTER N TOKEN MC1,CI10,BW1,BO9,EW1,EO3,MF6,MP7
LETTER P TOKEN MC1,CI9,BW1,BO8,EW1,EO6,MF9,MP6
LETTER R TOKEN MC1,CI11,BW1,BO5,EW1,EO7,MF10,MP8
LETTER S TOKEN MC1,CI12,BW1,BO10,EW1,EO3,MF11,MP9
LETTER T TOKEN MC1,CI13,BW1,BO11,EW1,EO8,MF12,MP2
LETTER X TOKEN MC1,CI9,BW1,BO8,EW1,EO6,MF9,MP6
LETTER Y TOKEN MC1,CI9,BW1,BO8,EW1,EO6,MF9,MP6
LETTER Z TOKEN MC1,CI14,BW1,BO12,EW1,EO4,MF13,MP3
LITERAL MC7,CI28,BW9,BO26,EW9,EO19,MF29,MP23
LITERAL LIST MC30,CI40,BW32,BO5,EW29,EO33,MF25,MP20
LITERAL LIST PACK MC30,CI3,BW7,BO5,EW7,EO32,MF12,MP29
LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME

```
                          MC21,CI43,BW23,BO44,EW22,EO36,MF1,MP32
LOOSE REPLICATABLE SUPPRESSIBLE CHARACTER FRAME SEQUENCE
                          MC35,CI43,BW36,BO5,EW32,EO36,MF12,MP33
MINUS TOKEN               MC1,CI17,BW1,BO15,EW1,EO10,MF16,MP11
OPEN TOKEN                MC1,CI19,BW1,BO17,EW1,EO3,MF18,MP13
PATTERN                   MC6,CI27,BW8,BO25,EW8,EO18,MF12,MP20
PICTURE                   MC3,CI24,BW4,BO22,EW4,EO17,MF26,MP20
PLUS TOKEN                MC1,CI16,BW1,BO14,EW1,EO10,MF16,MP11
POINT TOKEN               MC1,CI18,BW1,BO16,EW1,EO11,MF17,MP12
RADIX                     MC37,CI11,BW38,BO46,EW34,EO3,MF14,MP2
RADIX MOULD               MC36,CI44,BW37,BO45,EW33,EO3,MF10,MP20
REAL MOULD                MC31,CI41,BW33,BO41,EW30,EO34,MF32,MP30
REAL PATTERN              MC19,CI8,BW21,BO37,EW20,EO29,MF30,MP27
REPLICATED LITERAL        MC15,CI34,BW6,BO33,EW16,EO26,MF24,MP25
REPLICATED LITERAL SEQUENCE
                          MC14,CI34,BW17,BO5,EW9,EO26,MF29,MP25
REPLICATION               MC10,CI31,BW13,BO30,EW12,EO22,MF28,MP7
REPLICATOR                MC5,CI26,BW6,BO24,EW6,EO3,MF28,MP7
ROW OF CHARACTER DENOTATION
                          MC1,CI23,BW1,BO21,EW1,EO16,MF24,MP19
SIGN FRAME                MC17,CI36,BW19,BO35,EW18,EO27,MF16,MP11
SIGN MOULD                MC16,CI35,BW18,BO34,EW17,EO3,MF16,MP26
STAGNANT MOULD            MC33,CI5,BW35,BO43,EW31,EO3,MF33,MP27
STRING DENOTATION         MC13,CI33,BW16,BO32,EW15,EO25,MF24,MP19
STRING PATTERN            MC23,CI37,BW25,BO36,EW24,EO28,MF12,MP20
STROP CASE FINISH TOKEN   MC1,CI20,BW1,BO5,EW1,EO14,MF19,MP17
STROP CASE START TOKEN    MC1,CI20,BW1,BO18,EW1,EO3,MF20,MP15
STROP CONDITION FINISH TOKEN
                          MC1,CI20,BW1,BO5,EW1,EO14,MF19,MP17
STROP CONDITION START TOKEN
                          MC1,CI20,BW1,BO18,EW1,EO3,MF20,MP15
```