INTERNATIONAL STANDARD

ISO/IEC 8613-10

First edition
1991-06-01

AMENDMENT 3
1992-11-15

Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format –

Part 10:
Formal specifications

AMENDMENT 3: Formal specification of the character content architectures

Traitement de l'information – Bureautique – Architecture des documents de bureau (ODA) et format d'échange –

Partie 10: Spécifications formelles

AMENDEMENT 3: Spécifications formelles des architectures de contenus de caractères

Reference number
ISO/IEC 8613-10:1991/Amd. 3:1992 (E)

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Amendment 3 to International Standard ISO/IEC 8613-10:1991 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

ISO/IEC 8613 consists of the following parts, under the general title *Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format :*

- *Part 1: Introduction and general principles*
- *Part 2: Document structures*
- *Part 4: Document profile*
- *Part 5: Office Document Interchange Format (ODIF)*
- *Part 6: Character content architectures*
- *Part 7: Raster graphics content architectures*
- *Part 8: Geometric graphics content architectures*
- *Part 10: Formal specifications*

Annex C forms an integral part of ISO/IEC 8613-10.

International Organization for Standardization

Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Information processing – Text and office systems – Office Document Architecture (ODA) and interchange format –

Part 10:
Formal specifications

AMENDMENT 3: Formal specification of the character content architectures

Insert a new annex C as follows:

Annex C

(normative)

Formal specification of the character content architectures

C.1 Introduction

This annex gives a formal specification of the character content architectures as described in ISO 8613-6. This annex is composed of 6 clauses:

Clause C.1 provides a general introduction, including a list of all definitions which are given in C.2, C.3, C.4 and C.5.

Clause C.2 provides the interface to the document profile - by giving a formal specification of those attributes which may be included in the document profile attribute 'document application profile defaults'.

Clauses C.3 and C.4 provide the interface to the document architecture by giving a formal specification of character content portion presentation attributes; and any associated content portion attributes applicable to the character content portions.

Clause C.5 provides a description of the elements of the character content information. In particular, the following general rules on the syntactic and semantic usage of control functions are formally specified:

1. GCC-0 shall be followed by 2 or more graphic characters (see ISO 8613-6, clause 11.1.2).

   NOTE: ISO 8613-6, clause 11.1.2 states implicitly that 2 or more graphic characters can follow a GCC-0 but only the *2 graphic characters* immediately following the GCC-0 are concatenated (if they are not split by a line break which would introduce control function characters).

2. GCC-1 shall be followed by 2 or more graphic characters followed by GCC-2 (see ISO 8613-6, clause 11.1.2).

   NOTE: ISO 8613-6, clause 11.1.2 states implicitly that at least 2, but possibly more, graphic characters can follow a GCC-1 but only the *graphic characters* immediately following the GCC-1 and which are terminated by a GCC-2 are concatenated (if they are not split by a line break which would introduce control function characters).

3. LF is only permitted: (a) at the beginning of a character string, (b) after a CR and (c) after a LF (see ISO 8613-6, clause 11.1.4).

4. The number of PLUs shall be complemented exactly by PLDs before any LF is encountered and at any position in the string the following condition is applicable; $|\#\text{PLU} - \#\text{PLD}| \leq 1$ (see ISO 8613-6, clauses 11.1.5 and 11.1.6).

   NOTE: For rules 4. and 6., the # means *number of*. For rule 4. the $| x - y |$ represents the mathematical modulus.

5. SRS-1 shall be followed by one or more characters except CR LF and followed by SRS-0 (see ISO 8613-6, clause 11.1.11).

6. The #PLU = #PLD between SRS-1 and SRS-0 (see ISO 8613-6, clause 11.1.11).

7. The effects of VPB and VPR shall be cancelled (see 8. for details). within any SRS-1 and SRS-0 (see ISO 8613-6, clause 11.1.11).

8. The effects of all VPB-n_is and VPR-m_js shall be cancelled (i.e., $\sum_{i=1}^{p}$ VPB-n_i = $\sum_{j=1}^{q}$ VPR-m_j where $\{i, j, p, q, n_i, m_j \in \mathbb{N}\}$) before the control functions CR LF are encountered (see ISO 8613-6, clauses 11.1.15 and 11.1.16).

9. PTX-1 shall be followed by a character string which shall be followed by PTX-3 followed by a character string followed by PTX-0 (see ISO 8613-6, clause 11.3.3).

10. SOS is followed by a character string followed by ST (NOTE: Nesting is not forbidden). (See ISO 8613-6, clause 11.4.2).

Clause C.6 is an index to the terms (definitions, operators, attribute names) used in clauses C.2, C.3, C.4 and C.5.

Any time a clause number is specified in the semi-formal descriptions this refers to a clause number in ISO 8613-6.

What follows is the outline of the formula which specifies the character content architectures. The dots indicate formal text fragments which have been left out for the sake of readability. The full formula can be obtained by replacing each line (apart from the *and*) with the definition which is referenced by the superscript of the predicate symbol or operator symbol, respectively. The variables used in the definition of the predicate have to be replaced by those appearing in the outline (if they are different).

NOTE: A definition is a formula, hence it may never yield an undefined result, whatever value has been inserted for the variable.

... IsProfileDefaultableCharacterContentArchitectureAttribute$^{6.1}(att)$ ...
and ... IsProfileCharacterCodingSpecification$^{6.2}(v)$ ...
and ... IsProfileCharacterPresentationFeature$^{6.3}(v)$ ...
and ... SatisfiesCharacterContentArchitectureConstraints$^{6.4}(prof, doby)$ ...
and ... IsProcessableCharacterDescription$^{6.5}(cont)$ ...
and ... IsProcessableCharacterContent$^{6.6}(cont)$ ...
and ... IsFormattedCharacterDescription$^{6.7}(cont)$ ...
and ... IsFormattedCharacterContent$^{6.8}(cont)$ ...
and ... IsFormattedProcessableCharacterDescription$^{6.9}(cont)$ ...
and ... IsFormattedProcessableCharacterContent$^{6.10}(cont)$ ...
and ... IsCharacterContentPortionDescription$^{6.11}(cont)$ ...
and ... IsCharacterContentPresentationAttribute$^{6.12}(att)$ ...
and ... IsCharacterContentCodingAttribute$^{6.13}(att)$ ...
and ... IsCharacterContentPortionAttributeSet$^{6.14}(as)$ ...
and ... IsAlignmentValue$^{6.15}(v)$ ...
and ... IsCharacterFontsValue$^{6.16}(v)$ ...
and ... IsCharacterOrientationValue$^{6.17}(v)$ ...
and ... IsFormattingIndicatorValue$^{6.18}(v)$ ...
and ... IsCharacterPathValue$^{6.19}(v)$ ...
and ... IsCharacterContentTypeOfCodingValue$^{6.20}(v)$ ...
and ... IsCharacterSpacingValue$^{6.21}(v)$ ...
and ... IsFirstLineOffsetValue$^{6.22}(v)$ ...
and ... IsCharacterSetsValue$^{6.23}(v)$ ...
and ... IsGraphicCharacterSubrepertoireValue$^{6.24}(v)$ ...
and ... IsISO6937SubrepertoireId$^{6.25}(v)$ ...
and ... IsCharacterContentArchitectureClassValue$^{6.26}(v)$ ...
and ... IsGraphicRenditionValue$^{6.27}(v)$ ...
and ... IsWeightValue$^{6.28}(v)$ ...
and ... IsPostureValue$^{6.29}(v)$ ...
and ... IsUnderliningValue$^{6.30}(v)$ ...
and ... IsBlinkingValue$^{6.31}(v)$ ...
and ... IsImageInversionValue$^{6.32}(v)$ ...
and ... IsCrossingOutValue$^{6.33}(v)$ ...
and ... IsIndentationValue$^{6.34}(v)$ ...
and ... IsCharacterInitialOffsetValue$^{6.35}(v)$ ...
and ... IsKerningOffsetValue$^{6.36}(v)$ ...
and ... IsOrphanSizeValue$^{6.37}(v)$ ...
and ... IsCodeExtensionAnnouncersValue$^{6.38}(v)$ ...
and ... IsPairwiseKerningValue$^{6.39}(v)$ ...
and ... IsWidowSizeValue$^{6.40}(v)$ ...
and ... IsProportionalLineSpacingValue$^{6.41}(v)$ ...

and ... IsLineLayoutTableValue$^{6.42}(v)$ ...
and ... IsTabReferenceString$^{6.43}(v)$ ...
and ... IsLineSpacingValue$^{6.44}(v)$ ...
and ... IsCharacterLineProgressionValue$^{6.45}(v)$ ...
and ... IsItemizationValue$^{6.46}(v)$ ...
and ... ContainsFormattedElements$^{6.47}(v)$ ...
and ... ContainsProcessableElements$^{6.48}(v)$ ...
and ... ContainsFormattedProcessableElements$^{6.49}(v)$ ...
and ... IsValidFormattedContentSyntax$^{6.50}(v)$ ...
and ... IsValidProcessableContentSyntax$^{6.51}(v)$ ...
and ... IsValidFormattedProcessableContentSyntax$^{6.52}(v)$ ...
and ... IsFormattedContentElement$^{6.53}(v)$ ...
and ... IsProcessableContentElement$^{6.54}(v)$ ...
and ... IsFormattedProcessableContentElement$^{6.55}(v)$ ...
and ... IsCommonContentElement$^{6.56}(v)$ ...
and ... IsValidFormattedContentSemantics$^{6.57}(v)$ ...
and ... IsValidProcessableContentSemantics$^{6.58}(v)$ ...
and ... IsValidFormattedProcessableContentSemantics$^{6.59}(v)$ ...
and ... IsValidCommonContentSemantics$^{6.60}(v)$ ...
and ... IsValidLogicalControlSemantics$^{6.61}(v)$ ...
and ... IsValidDelimiterControlSemantics$^{6.62}(v)$ ...
and ... IsValidSharedControlSemantics$^{6.63}(v)$ ...
and ... IsValidGCC0$^{6.64}(v)$ ...
and ... IsValidGCC12$^{6.65}(v)$ ...
and ... IsValidGCC12End$^{6.66}(v)$ ...
and ... IsValidLF$^{6.67}(v)$ ...
and ... IsValidNextLF0$^{6.68}(v)$ ...
and ... IsValidNextLF1$^{6.69}(v)$ ...
and ... IsValidPLUD$^{6.70}(v)$ ...
and ... IsIncompletePLU$^{6.71}(v)$ ...
and ... IsIncompletePLD$^{6.72}(v)$ ...
and ... IsValidSRS$^{6.73}(v)$ ...
and ... IsValidSRSEnd$^{6.74}(v)$ ...
and ... IsValidSRSNest$^{6.75}(v, v_1)$ ...
and ... IsValidPLUDInSRS$^{6.76}(v)$ ...
and ... IsValidPLUDInSRSEnd$^{6.77}(v)$ ...
and ... IsIncompletePLUInSRS$^{6.78}(v)$ ...
and ... IsIncompletePLDInSRS$^{6.79}(v)$ ...
and ... IsValidVP$^{6.80}(v)$ ...
and ... IsValidVPSum$^{6.81}(v, n)$ ...
and ... IsValidPTX$^{6.82}(v)$ ...
and ... IsValidPTXNext$^{6.83}(v)$ ...
and ... IsValidPTXEnd$^{6.84}(v)$ ...
and ... IsValidSOS$^{6.85}(v)$ ...
and ... IsValidSOSEnd$^{6.86}(v)$ ...
and ... IsValidSOSNest$^{6.87}(v, v_1)$ ...
and ... IsValidVPInSRS$^{6.88}(v)$ ...
and ... IsValidVPSumInSRS$^{6.89}(v, n)$ ...
and ... IsValidVPSumInSRSEnd$^{6.90}(v)$ ...
and ... IsCodeExtensionControlFn$^{6.91}(v)$ ...
and ... IsISO2022String$^{6.92}(v)$ ...
and ... IsISO2022Character$^{6.93}(v)$ ...
and ... IsISO6429String$^{6.94}(v)$ ...
and ... IsISO6429Character$^{6.95}(v)$ ...
and ... IsSharedControlFn$^{6.96}(v)$ ...
and ... IsParameterlessSharedControlFn$^{6.97}(v)$ ...

and ... IsParameterizedSharedControlFnName$^{6.98}(v)$ ...
and ... IsParameterizedSharedControlFn$^{6.99}(n, p)$ ...
and ... IsLayoutControlFn$^{6.100}(v)$ ...
and ... IsParameterlessLayoutControlFn$^{6.101}(v)$ ...
and ... IsParameterizedLayoutControlFnName$^{6.102}(v)$ ...
and ... IsParameterizedLayoutControlFn$^{6.103}(n, p)$ ...
and ... IsLogicalControlFn$^{6.104}(v)$ ...
and ... IsParameterlessLogicalControlFn$^{6.105}(v)$ ...
and ... IsParameterizedLogicalControlFnName$^{6.106}(v)$ ...
and ... IsParameterizedLogicalControlFn$^{6.107}(n, p)$ ...
and ... IsDelimiterControlFn$^{6.108}(v)$ ...
and ... IsSpaceControlFn$^{6.109}(v)$ ...
and ... IsEscapeControlFn$^{6.110}(v)$ ...
and ... IsCharacterContentInformationValue$^{6.111}(v)$ ...
and ... IsGraphicCharacterString$^{6.112}(v)$ ...
and ... IsGraphicCharacter$^{6.113}(v)$ ...

NOTE: Other predicates or operators which are used here, but are defined in clause 6, are not listed here.

C.2 Interface to the Document Profile

Semiformal Description 6.1

Predicate "is a profile defaultable character content architecture attribute" (clause 7)

A profile defaultable character content architecture attribute corresponds to the set of character presentation attributes.

NOTE: This predicate is used in annex B.

Definition 6.1

```
1   ∀ att
2   (₀ IsProfileDefaultableCharacterContentArchitectureAttribute(att) iff
3     IsCharacterContentPresentationAttribute^6.12(att)₀)
```

Semiformal Description 6.2

Predicate "is a profile character coding specification" (clause 8.1)

A character coding specification is a nomination where each element is a character content coding attribute.

NOTE: This predicate is used in annex B.

Definition 6.2

```
1   ∀ v
2   (₀ IsProfileCharacterCodingSpecification(v) iff
3     IsNom(v) and
4     ∀ b ∈̂ v • (IsCharacterContentCodingAttribute^6.13(C b))₀)
```

Semiformal Description 6.3

Predicate "is a profile character presentation feature" (clause 7)

A profile character presentation feature is a nomination where each element is a character content presentation attribute.

NOTE: This predicate is used in annex B.

Definition 6.3

```
1   ∀ v
2   (₀ IsProfileCharacterPresentationFeature(v) iff
3     IsNom(v) and
4     ∀ b ∈̂ v • (IsCharacterContentPresentationAttribute^6.12(C b))₀)
```

C.3 Interface to the Document Architecture

Semiformal Description 6.4

Predicate "satisfies character content architecture contraints" (clauses 4.8 and 15)

A document profile *prof* and a document body *doby* satisfy the constraints imposed by the character content architecture if the following holds for all constituents and associated content portions:

If the attribute 'content architecture class' of the constituent has the value '2 8 2 6 1' then the content portion contains processable character content (6), the attributes 'formatting indicator' and 'initial offset' are not specified for this content (7) and the value '2 8 2 6 1' is an element of the document profile attribute 'content architecture classes' (8).

If the attribute 'content architecture class' of the constituent has the value '2 8 2 6 0' then the content portion contains formatted character content (11), the attributes 'widow size', 'orphan size', 'proportional line spacing' and 'indentation' are not specified for this content (12,13) and the value '2 8 2 6 0' is an element of the document profile attribute 'content architecture classes' (14).

If the attribute 'content architecture class' has the value '2 8 2 6 2' then content portion contains formatted processable character content (17) and the value '2 8 2 6 2' is an element of the document profile attribute 'content architecture classes' (18).

NOTE: This predicate is used in clause 7 of this part of ISO 8613.

Definition 6.4

1 $\forall prof, doby$
2 $({}_{0}\text{SatisfiesCharacterContentArchitectureConstraints}(prof, doby)$ *iff*
3 $\forall cst, cont \in doby$
4 $({}_{1}({}_{2}({}_{3}(cont)\text{DescribesContPortOf}^{2.153}(cst)$ *and*
5 C ^cst • 'content architecture class' = '2 8 2 6 1'$_{3}$) *impl*
6 $({}_{4}\text{IsProcessableCharacterDescription}^{6.5}(cont)$ *and*
7 $\text{IsEmptyCol}^{1.4}(['\text{formatting indicator}'; '\text{initial offset}'] \cap \text{NAMS}^{1.18}(cst))$ *and*
8 '2 8 2 6 1' $\in$ C ^$prof$ • 'content architecture classes'$_{4}$)$_{2}$) *and*
9 $({}_{5}({}_{6}(cont)\text{DescribesContPortOf}^{2.153}(cst)$ *and*
10 C ^cst • 'content architecture class' = '2 8 2 6 0'$_{6}$) *impl*
11 $({}_{7}\text{IsFormattedCharacterDescription}^{6.7}(cont)$ *and*
12 $\text{IsEmptyCol}^{1.4}(['\text{widow size}'; '\text{orphan size}';$
13 'proportional line spacing'; 'indentation'] $\cap \text{NAMS}^{1.18}(cst))$ *and*
14 '2 8 2 6 0' $\in$ C ^$prof$ • 'content architecture classes'$_{7}$)$_{5}$) *and*
15 $({}_{8}({}_{9}(cont)\text{DescribesContPortOf}^{2.153}(cst)$ *and*
16 C ^cst • 'content architecture class' = '2 8 2 6 2'$_{9}$) *impl*
17 $({}_{10}\text{IsFormattedProcessableCharacterDescription}^{6.9}(cont)$ *and*
18 '2 8 2 6 2' $\in$ C ^$prof$ • 'content architecture classes'$_{10}$)$_{8}$)$_{1}$)$_{0}$)

Semiformal Description 6.5

Predicate "is processable character description" (clause 15)

Content is processable if it is a character content portion description and the character content is processable.

Definition 6.5

1 $\forall cont$
2 $({}_{0}\text{IsProcessableCharacterDescription}(cont)$ *iff*
3 $\text{IsCharacterContentPortionDescription}^{6.11}(cont)$ *and*
4 $\text{IsProcessableCharacterContent}^{6.6}(cont)_{0}$)

Semiformal Description 6.6

Predicate "is processable character content" (clause 15.2 and table 6)

A processable character content portion is a catenation containing a processable character string.

Definition 6.6

1 $\forall\, cont$
2 $({}_{0}$IsProcessableCharacterContent$(cont)$ *iff*
3 $\forall\, a \hat{\in} \hat{}cont$ •
4 (N a = 'content information' *impl* ContainsProcessableElements$^{6.48}$(C a))$_{0}$)

Semiformal Description 6.7

Predicate "is formatted character description" (clause 15)

Content is formatted if it is a character content portion description and the character content is formatted.

Definition 6.7

1 $\forall\, cont$
2 $({}_{0}$IsFormattedCharacterDescription$(cont)$ *iff*
3 IsCharacterContentPortionDescription$^{6.11}(cont)$ *impl*
4 IsFormattedCharacterContent$^{6.8}(cont)_{0}$)

Semiformal Description 6.8

Predicate "is formatted character content" (clause 15.1 and table 6)

A formatted character content portion is a catenation containing a formatted character string.

Definition 6.8

1 $(clause 15) \forall\, cont$
2 $({}_{0}$IsFormattedCharacterContent$(cont)$ *iff*
3 $\forall\, a \hat{\in} \hat{}cont$ •
4 (N a = 'content information' *impl* ContainsFormattedElements$^{6.47}$(C a))$_{0}$)

Semiformal Description 6.9

Predicate "is formatted processable character description" (clause 15)

Content is formatted processable if it is a character content portion description and the character content is formatted processable.

Definition 6.9

1 $\forall\, cont$
2 $({}_{0}$IsFormattedProcessableCharacterDescription$(cont)$ *iff*
3 IsCharacterContentPortionDescription$^{6.11}(cont)$ *impl*
4 IsFormattedProcessableCharacterContent$^{6.10}(cont)_{0}$)

Semiformal Description 6.10

Predicate "is formatted processable character content" (clause 15.3 and table 6)

A formatted processable character content is a catenation containing a formatted processable character string.

Definition 6.10

```
1   ∀ cont
2   (0 IsFormattedProcessableCharacterContent(cont) iff
3     ∀ a ∈ ^cont •
4     (N a = 'content information' impl ContainsFormattedProcessableElements^6.49(C a))0)
```

Semiformal Description 6.11

Predicate "is character content portion description" (clauses 8.1 and 8.3)

A character content portion description is a set of character content portion attributes.

NOTE: This predicate is used in clause 7 of this part of ISO 8613.

Definition 6.11

```
1   ∀ cont
2   (0 IsCharacterContentPortionDescription(cont) iff
3     IsCharacterContentPortionAttributeSet^6.14(cont)0)
```

Semiformal Description 6.12

Predicate "is character content presentation attribute" (clauses 4.3 and 7)

A character content architecture presentation attribute is one of the attributes 'alignment', 'character fonts', 'character orientation', 'character path', 'character spacing', 'code extension announcers', 'first line offset', 'formatting indicator', 'graphic character sets', 'graphic character subrepertoire', 'graphic rendition', 'indentation', 'initial offset', 'itemization', 'kerning offset', 'line layout table', 'line spacing', 'line progression', 'orphan size', 'pairwise kerning', 'proportional line spacing' or 'widow size' with corresponding value(s).

NOTE: This predicate is used in clause 7 of this part of ISO 8613.

Definition 6.12

```
1   ∀ att
2   (0 IsCharacterContentPresentationAttribute(att) iff
3     ∃ n, c
4     (1 att = [n : c] and
5       n ∈ ['alignment';       'character fonts';     'character orientation';
6             'character path'; 'character spacing';   'code extension announcers';
7             'first line offset'; 'formatting indicator';'graphic character sets';
8             'graphic rendition';'indentation';       'graphic character subrepertoire';
9             'initial offset';   'itemization';       'kerning offset';
10            'line layout table'; 'line progression'; 'line spacing';
11            'orphan size';      'pairwise kerning';  'proportional line spacing';
12            'widow size'] and
13      (2 n = 'alignment' impl
14            (3 IsPlaceholder^1.19(c) or IsAlignmentValue^6.15(c)3)2) and
```

15 $({}_{4}n$ = 'character fonts' *impl*
16 $({}_{5}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsCharacterFontsValue}^{6.16}(c)_{5})_{4})$ *and*
17 $({}_{6}n$ = 'character orientation' *impl*
18 $({}_{7}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsCharacterOrientationValue}^{6.17}(c)_{7})_{6})$ *and*
19 $({}_{8}n$ = 'character path' *impl*
20 $({}_{9}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsCharacterPathValue}^{6.19}(c)_{9})_{8})$ *and*
21 $({}_{10}n$ = 'character spacing' *impl*
22 $({}_{11}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsCharacterSpacingValue}^{6.21}(c)_{11})_{10})$ *and*
23 $({}_{12}n$ = 'code extension announcers' *impl*
24 $({}_{13}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsCodeExtensionAnnouncersValue}^{6.38}(c)_{13})_{12})$ *and*
25 $({}_{14}n$ = 'first line offset' *impl*
26 $({}_{15}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsFirstLineOffsetValue}^{6.22}(c)_{15})_{14})$ *and*
27 $({}_{16}n$ = 'formatting indicator' *impl*
28 $({}_{17}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsFormattingIndicatorValue}^{6.18}(c)_{17})_{16})$ *and*
29 $({}_{18}n$ = 'graphic character sets' *impl*
30 $({}_{19}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsCharacterSetsValue}^{6.23}(c)_{19})_{18})$ *and*
31 $({}_{20}n$ = 'graphic character subrepertoire' *impl*
32 $({}_{21}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsGraphicCharacterSubrepertoireValue}^{6.24}(c)_{21})_{20})$ *and*
33 $({}_{22}n$ = 'graphic rendition' *impl*
34 $({}_{23}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsGraphicRenditionValue}^{6.27}(c)_{23})_{22})$ *and*
35 $({}_{24}n$ = 'indentation' *impl*
36 $({}_{25}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsIndentationValue}^{6.34}(c)_{25})_{24})$ *and*
37 $({}_{26}n$ = 'initial offset' *impl*
38 $({}_{27}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsCharacterInitialOffsetValue}^{6.35}(c)_{27})_{26})$ *and*
39 $({}_{28}n$ = 'itemization' *impl*
40 $({}_{29}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsItemizationValue}^{6.46}(c)_{29})_{28})$ *and*
41 $({}_{30}n$ = 'kerning offset' *impl*
42 $({}_{31}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsKerningOffsetValue}^{6.36}(c)_{31})_{30})$ *and*
43 $({}_{32}n$ = 'line layout table' *impl*
44 $({}_{33}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsLineLayoutTableValue}^{6.42}(c)_{33})_{32})$ *and*
45 $({}_{34}n$ = 'line spacing' *impl*
46 $({}_{35}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsLineSpacingValue}^{6.44}(c)_{35})_{34})$ *and*
47 $({}_{36}n$ = 'line progression' *impl*
48 $({}_{37}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsCharacterLineProgressionValue}^{6.45}(c)_{37})_{36})$ *and*
49 $({}_{38}n$ = 'orphan size' *impl*
50 $({}_{39}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsOrphanSizeValue}^{6.37}(c)_{39})_{38})$ *and*
51 $({}_{40}n$ = 'pairwise kerning' *impl*
52 $({}_{41}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsPairwiseKerningValue}^{6.39}(c)_{41})_{40})$ *and*
53 $({}_{42}n$ = 'proportional line spacing' *impl*
54 $({}_{43}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsProportionalLineSpacingValue}^{6.41}(c)_{43})_{42})$ *and*
55 $({}_{44}n$ = 'widow size' *impl*
56 $({}_{45}\text{IsPlaceholder}^{1.19}(c)$ *or* $\text{IsWidowSizeValue}^{6.40}(c)_{45})_{44})_{1})_{0})$

Semiformal Description 6.13

Predicate "is a character content coding attribute" (clause 8.1)

A character content coding attribute consists of the attribute 'type of coding' with its corresponding value.

Definition 6.13

1 $\forall\, att$
2 $(_0$IsCharacterContentCodingAttribute(att) *iff*
3 $\exists\, n, c$
4 $(_1\, att = [n : c]$ *and*
5 $n =$ 'type of coding' *impl* IsCharacterContentTypeOfCodingValue$^{6.20}(c)_1)_0)$

C.4 Attributes of the Character Content Architecture

Semiformal Description 6.14

Predicate "is character content portion attribute set" (clause 8)

A character content architecture portion attribute set contains the listed attributes with corresponding attribute values.

Definition 6.14

1 $\forall\, as$
2 $(_0$IsCharacterContentPortionAttributeSet(as) *iff*
3 $(_1$IsNeNom$^{1.2}(as)$ *and*
4 $\forall\, a \in \breve{}as \bullet$
5 $(_2$N $a =$ 'type of coding' *impl*
6 $(_3$IsPlaceholder$^{1.19}$(C $a)_3)_2)$ *and*
7 $(_4$N $a =$ 'content information' *impl* IsCharacterContentInformationValue$^{6.111}$(C $a)_4)_1)_0)$

Semiformal Description 6.15

Predicate "is an alignment value" (clause 7.1.1)

The value of the attribute 'alignment' is 'start-aligned', 'end-aligned', 'centred' or 'justified'.

Definition 6.15

1 $\forall\, v$
2 $(_0$IsAlignmentValue(v) *iff*
3 $v \in [$'start-aligned'; 'end-aligned'; 'centred'; 'justified'$]_0)$

Semiformal Description 6.16

Predicate "is a character fonts value" (clause 7.1.2)

The value of the attribute 'character fonts' can contain up to ten pairs of parameters (3) named 'primary font', 'first alternative font', 'second alternative font', 'third alternative font', 'fourth alternative font', 'fifth alternative font', 'sixth alternative font', 'seventh alternative font', 'eighth alternative font' and 'ninth alternative font' (4-8), each with sub-parameters 'font size' and 'font identifier' (10), both having a positive integer value (11).

Definition 6.16

```
1   ∀ v
2   (0 IsCharacterFontsValue(v) iff
3     (1 IsNom(v) and   CARD (v) ≤ 10 and
4       NAMS1.18(v) ⊆ ['primary font';          'first alternative font';
5                      'second alternative font';'third alternative font';
6                      'fourth alternative font'; 'fifth alternative font';
7                      'sixth alternative font';  'seventh alternative font';
8                      'eighth alternative font'; 'ninth alternative font'] and
9       ∀ a ∈̂ v •
10      (2 IsNom(C a) and NAMS1.18(C a) = ['font size'; 'font identifier'] and
11      ∀ b ∈̂ (C a) • IsNat(C b)2)1)0)
```

Semiformal Description 6.17

Predicate "is a character orientation value" (clause 7.1.3)

The value of the attribute 'character orientation' is '0°', '90°', '180°' or '270°'.

Definition 6.17

```
1   ∀ v
2   (0 IsCharacterOrientationValue(v) iff
3     v ∈ ['0°';'90°';'180°';'270°']0)
```

Semiformal Description 6.18

Predicate "is a formatting indicator value" (clause 7.2.1)

The attribute 'formatting indicator' has the value 'yes' or 'no'.

Definition 6.18

```
1   ∀ v
2   (0 IsFormattingIndicatorValue(v) iff
3     v ∈ ['yes';'no']0)
```

Semiformal Description 6.19

Predicate "is a character path value" (clause 7.1.4)

The value of the attribute 'character path' is '0°', '90°', '180°' or '270°'.

Definition 6.19

```
1   ∀ v
2   (0 IsCharacterPathValue(v) iff
3     v ∈ ['0°';'90°';'180°';'270°']0)
```

Semiformal Description 6.20

Predicate "is a character content type of coding value" (clause 8.1)

The value of the attribute 'type of coding' is an ASN.1 object indentifier '2 8 3 6 0' .

Definition 6.20

1 $\forall v$
2 $({}_0$IsCharacterContentTypeOfCodingValue(v) *iff*
3 $v =$ '2 8 3 6 0'$_0)$

Semiformal Description 6.21

Predicate "is a character spacing value" (clause 7.1.5)

The value of the attribute 'character spacing' is any positive integer.

Definition 6.21

1 $\forall v$
2 $({}_0$IsCharacterSpacingValue(v) *iff*
3 IsNat$(v)_0)$

Semiformal Description 6.22

Predicate "is a first line offset value" (clause 7.1.7)

The value of the attribute 'first line offset' is any integer.

Definition 6.22

1 $\forall v$
2 $({}_0$IsFirstLineOffsetValue(v) *iff*
3 IsInt$(v)_0)$

Semiformal Description 6.23

Predicate "is a character sets value" (clause 7.1.8)

The value of the attribute 'graphic character sets' is a string of escape sequences (in accordance with ISO 2022 and ISO 2375) found in the *ISO International register of coded character sets to be used with escape sequences* (held by the *European Computer Manufacturers Association* (ECMA) – the registration authority). These escape sequences are used to designate one or more graphic character sets, and any locking shift functions needed for their invocation.

NOTE: This predicate is used in annex B.

Definition 6.23

1 $\forall v$
2 $({}_0$IsCharacterSetsValue(v) *iff*
3 IsISO2022String$^{6.92}(v)_0)$

Semiformal Description 6.24

Predicate "is a graphic character subrepertoire value" (clause 7.1.9)

The value of the attribute 'graphic character subrepertoire' is either 0 or an identifier of a subrepertoire.

Definition 6.24

1 $\forall v$
2 $(_0$IsGraphicCharacterSubrepertoireValue(v) *iff*
3 $v = 0$ *or* IsISO6937SubrepertoireId$^{6.25}(v)_0)$

Semiformal Description 6.25

Predicate "is a ISO 6937 subrepertoire identifier" (clause 7.1.9)

An ISO 6937 subrepertoire identifier is considered atomic for this definition.

Definition 6.25

1 $\forall v$
2 $(_0$IsISO6937SubrepertoireId(v) *iff*
3 IsAtom$(v)_0)$

Semiformal Description 6.26

Predicate "is a character content architecture class value" (clause 7.4.1)

A content architecture class attribute value is an ASN.1 object identifier '2 8 2 6 1', '2 8 2 6 2' or '2 8 2 6 0'.

Definition 6.26

1 $\forall v$
2 $(_0$IsCharacterContentArchitectureClassValue(v) *iff*
3 $v \in$ ['2 8 2 6 1';'2 8 2 6 2';'2 8 2 6 0']$_0)$

Semiformal Description 6.27

Predicate "is a graphic rendition value" (clauses 7.1.10 and 6.1)

The value of the attribute 'graphic rendition' can contain any sequence of integer values from those specified below (4-12).

Definition 6.27

1 $\forall v$
2 $(_0$IsGraphicRenditionValue(v) *iff*
3 IsNeCat$^{1.3}(v)$ *and*
4 $(_1 v = [\rightarrow 0 \rightarrow]$ *or*
5 $\forall x \hat{\in} \hat{} v \bullet$
6 $(_2$IsWeightValue$^{6.28}(x)$ *or*
7 IsPostureValue$^{6.29}(x)$ *or*
8 IsUnderliningValue$^{6.30}(x)$ *or*
9 IsBlinkingValue$^{6.31}(x)$ *or*
10 IsImageInversionValue$^{6.32}(x)$ *or*
11 IsCrossingOutValue$^{6.33}(x)$ *or*
12 $x \in$ [10;11;12;13;14;15;16;17;18;19;26;50]$_2)_1)_0)$

Semiformal Description 6.28

Predicate "is a weight value" (clauses 6.1 and 6.1.1)

A weight value is either the integer 2, which represents a rendition *feint* (decreased intensity), the integer 22, which represents a rendition *normal intensity* (neither *feint* nor *bold*), or the integer 1, which represents a rendition *bold* (increased intensity).

Definition 6.28

1 $\forall v$
2 $({}_{0}\mathrm{IsWeightValue}(v)$ *iff*
3 $v \in [2;22;1]_{0}$)

Semiformal Description 6.29

Predicate "is a posture value" (clauses 6.1 and 6.1.2)

A posture value is either the integer 23, which represents a rendition *not italicized*, or the integer 3, which represents a rendition *italicized*.

Definition 6.29

1 $\forall v$
2 $({}_{0}\mathrm{IsPostureValue}(v)$ *iff*
3 $v \in [23;3]_{0}$)

Semiformal Description 6.30

Predicate "is an underlining value" (clauses 6.1 and 6.1.3)

An underlining value is either the integer 24, which represents a rendition *not underlined*, the integer 4, which represents a rendition *underlined*, or the integer 21, which represents a rendition *doubly underlined*.

Definition 6.30

1 $\forall v$
2 $({}_{0}\mathrm{IsUnderliningValue}(v)$ *iff*
3 $v \in [24;4;21]_{0}$)

Semiformal Description 6.31

Predicate "is a blinking value" (clauses 6.1 and 6.1.4)

A blinking value is either the integer 25, which represents a rendition *steady* (not blinking), the integer 5, which represents a rendition *slowly blinking*, or the integer 6, which represents a rendition *rapidly blinking*.

Definition 6.31

1 $\forall v$
2 $({}_{0}\mathrm{IsBlinkingValue}(v)$ *iff*
3 $v \in [25;5;6]_{0}$)

Semiformal Description 6.32

Predicate "is an image inversion value" (clauses 6.1 and 6.1.5)

An image inversion value is either the integer 27, which represents a rendition *positive image*, or the integer 7, which represents a rendition *negative image*.

Definition 6.32

1 $\forall v$
2 $({}_0$IsImageInversionValue(v) *iff*
3 $v \in [27;7]_0)$

Semiformal Description 6.33

Predicate "is a crossing-out value" (clauses 6.1 and 6.1.6)

A crossing-out value is either the integer 29, which represents a rendition *not crossed-out*, or the integer 9, which represents a rendition *crossed-out*.

Definition 6.33

1 $\forall v$
2 $({}_0$IsCrossingOutValue(v) *iff*
3 $v \in [29;9]_0)$

Semiformal Description 6.34

Predicate "is an indentation value" (clause 7.3.1)

The value of the attribute 'indentation' is any non-negative integer.

Definition 6.34

1 $\forall v$
2 $({}_0$IsIndentationValue(v) *iff*
3 IsNnInt$^{1.7}(v)_0)$

Semiformal Description 6.35

Predicate "is an initial offset value" (clause 7.2.2)

The value of the attribute 'initial offset' consists of two parameters 'horizontal coordinate' and 'vertical coordinate' each having a non-negative integer value.

NOTE: Since the attribute 'initial offset' is also used within the raster graphics content architectures the qualifier "Character" has been added to the predicate name.

Definition 6.35

1 $\forall v$
2 $({}_0$IsCharacterInitialOffsetValue(v) *iff*
3 $({}_1$IsNeNom$^{1.2}(v)$ *and*
4 NAMS$^{1.18}(v)$ = ['horizontal coordinate';'vertical coordinate'] *and*
5 $\forall b \in \hat{}v$ •
6 $({}_2$N b = 'horizontal coordinate' *impl* IsNnInt$^{1.7}$(C b) *and*
7 N b = 'vertical coordinate' *impl* IsNnInt$^{1.7}$(C $b)_2)_1)_0)$

Semiformal Description 6.36

Predicate "is a kerning offset value" (clause 7.1.12)

The value of the attribute 'kerning offset' consists of two parameters 'start edge offset' and 'end edge offset' each having a non-negative integer value.

Definition 6.36

```
1   ∀ v
2   (0 IsKerningOffsetValue(v) iff
3     (1 IsNom(v) and
4       NAMS^1.18(v) = ['start edge offset'; 'end edge offset'] and
5       ∀ b ∈̂ ^v •
6       (2 N b = 'start edge offset' impl IsNnInt^1.7(C b) and
7         N b = 'end edge offset' impl IsNnInt^1.7(C b)2)1)0)
```

Semiformal Description 6.37

Predicate "is an orphan size value" (clause 7.3.2)

The value of the attribute 'orphan size' is any positive integer.

Definition 6.37

```
1   ∀ v
2   (0 IsOrphanSizeValue(v) iff
3     IsNat(v)0)
```

Semiformal Description 6.38

Predicate "is a code extension announcers value" (clause 7.1.6)

The value of the attribute 'code extension announcers' is a string of escape sequences in accordance with ISO 2022, used to announce code extension features.

Definition 6.38

```
1   ∀ v
2   (0 IsCodeExtensionAnnouncersValue(v) iff
3     IsISO2022String^6.92(v)0)
```

Semiformal Description 6.39

Predicate "is a pairwise kerning value" (clause 7.1.16)

The value of the attribute 'pairwise kerning' is either 'yes' or 'no'.

Definition 6.39

```
1   ∀ v
2   (0 IsPairwiseKerningValue(v) iff
3     v ∈ ['yes'; 'no']0)
```

Semiformal Description 6.40

Predicate "is a widow size value" (clause 7.3.4)

The value of the attribute 'widow size' is any positive integer.

Definition 6.40

```
1   ∀ v
2   (0 IsWidowSizeValue(v) iff
3     IsNat(v)0)
```

Semiformal Description 6.41

Predicate "is a proportional line spacing value" (clause 7.3.3)

The value of the attribute 'proportional line spacing' is either 'yes' or 'no'.

Definition 6.41

```
1   ∀ v
2   (0 IsProportionalLineSpacingValue(v) iff
3     v ∈ ['yes';'no']0)
```

Semiformal Description 6.42

Predicate "is a line layout table value" (clause 7.1.13)

The value of the attribute 'line layout table' can consists of four parameters: 'tab reference', 'tab position', 'alignment' and conditionally 'alignment string' each with corresponding values. In the case of 'alignment string' this parameter is only specified if the value 'aligned-around' is specified for the parameter 'alignment' (9-13). The value of 'alignment string' is represented as an atomic structure (13).

Definition 6.42

```
1   ∀ v
2   (0 IsLineLayoutTableValue(v) iff
3     (1 IsNom(v) and
4       NAMS^1.18(v) ⊇ ['tab reference';'tab position';'alignment']   and
5       NAMS^1.18(v) ⊆ ['tab reference';'tab position';'alignment';'alignment string'] and
6       ∀ b ∈̂ v •
7       (2 N b = 'tab reference' impl IsTabReferenceString^6.43(C b) and
8         N b = 'tab position' impl IsNnInt^1.7(C b) and
9         N b = 'alignment' impl
10        (3 C b ∈ ['start-aligned';'end-aligned';'centred';'aligned-around'] and
11          C b = 'aligned-around' impl
12        ('alignment string' ∈ NAMS^1.18(v)4)3) and
13        N b = 'alignment string' impl IsAtom(C b)2)1)0)
```

Semiformal Description 6.43

Predicate "is a tab reference string" (clause 7.1.13)

A tab reference string in this context consists of a concatenated sequence of integers whose length is greater than or equal to 1 and less than or equal to 4.

Definition 6.43

1 $\forall v$
2 $(_0$IsTabReferenceString(v) *iff*
3 IsNeCat$^{1.3}(v)$ *and*
4 (LENGTH$^{1.16}$ $(v) \geq 1$ *and* LENGTH$^{1.16}$ $(v) \leq 4)$ *and*
5 $\forall x \hat{\in} \hat{} v \bullet$ (IsInt(C x))$_0$)

Semiformal Description 6.44

Predicate "is a line spacing value" (clause 7.1.15)

The value of the attribute 'line spacing' is any positive integer.

Definition 6.44

1 $\forall v$
2 $(_0$IsLineSpacingValue(v) *iff*
3 IsNat$(v)_0$)

Semiformal Description 6.45

Predicate "is a line progression value" (clause 7.1.14)

The value of the attribute 'line progression' is '90°' or '270°'.

NOTE: Since the attribute 'line progression' is also used within the raster graphics content architectures the qualifier "Character" has been added to the predicate name.

Definition 6.45

1 $\forall v$
2 $(_0$IsCharacterLineProgressionValue(v) *iff*
3 $v \in$ ['90°'; '270°']$_0$)

Semiformal Description 6.46

Predicate "is an itemization value" (clause 7.1.11)

The value of the attribute 'itemization' consists of three parameters 'identifier alignment', 'identifier start offset' and 'identifier end offset'. The values of 'identifier start offset' and 'identifier end offset' are integers (9, 10). The value of 'identifier alignment' is an element of the set 'no itemization', 'start-aligned' and 'end-aligned' (8).

Definition 6.46

```
1   ∀ v
2   (0 IsItemizationValue(v) iff
3     (1 IsNom(v) and
4       NAMS^1.18(v) = ['identifier alignment'; 'identifier start offset';
5                       'identifier end offset']1) and
6       ∀ a ∈̂ ^v •
7       (2 N a = 'identifier alignment' impl
8         (3 C a ∈ ['no itemization';'start-aligned'; 'end-aligned']3) and
9         N a = 'identifier start offset' impl IsInt(C a) and
10        N a = 'identifier end offset' impl IsInt(C a)2)0)
```

C.5 Elements of the Character Content Information

Semiformal Description 6.47

Predicate "contains formatted elements" (clause 4)

A character string represents a valid formatted content if the string has a valid syntax (3) and valid semantics (4) for formatted content.

Definition 6.47

```
1   ∀ v
2   (0 ContainsFormattedElements(v) iff
3     IsValidFormattedContentSyntax^6.50(v) and
4     IsValidFormattedContentSemantics^6.57(v)0)
```

Semiformal Description 6.48

Predicate "contains processable elements" (clause 4)

A character string represents a valid processable content if the string has a valid syntax (3) and valid semantics (4) for processable content.

Definition 6.48

```
1   ∀ v
2   (0 ContainsProcessableElements(v) iff
3     IsValidProcessableContentSyntax^6.51(v) and
4     IsValidProcessableContentSemantics^6.58(v)0)
```

Semiformal Description 6.49

Predicate "contains formatted processable elements" (clause 4)

A character string represents a valid formatted processable content if the string has a valid syntax (3) and valid semantics (4) for formatted processable content.

Definition 6.49

1 $\forall v$
2 $({}_{0}$ ContainsFormattedProcessableElements(v) *iff*
3 IsValidFormattedProcessableContentSyntax$^{6.52}(v)$ *and*
4 IsValidFormattedProcessableContentSemantics$^{6.59}(v)_{0})$

Semiformal Description 6.50

Predicate "is a valid formatted content syntax" (clause 4)

A valid formatted character content syntax is an empty string (3) or a string in which the first element is a formatted content element (4) and the remainder of the string has a valid formatted content syntax (5).

Definition 6.50

1 $\forall v$
2 $({}_{0}$ IsValidFormattedContentSyntax(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $({}_{1}$ IsFormattedContentElement$^{6.53}$(HEAD$^{1.13}(v))$ *and*
5 IsValidFormattedContentSyntax$^{6.50}$(TAIL$^{1.15}(v))_{1})_{0})$

Semiformal Description 6.51

Predicate "is a valid processable content syntax" (clause 4)

A valid processable character content syntax is an empty string (3) or a string in which the first element is processable content element (4) and the remainder of the string has a valid processable content syntax (5).

Definition 6.51

1 $\forall v$
2 $({}_{0}$ IsValidProcessableContentSyntax(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $({}_{1}$ IsProcessableContentElement$^{6.54}$(HEAD$^{1.13}(v))$ *and*
5 IsValidProcessableContentSyntax$^{6.51}$(TAIL$^{1.15}(v))_{1})_{0})$

Semiformal Description 6.52

Predicate "is a valid formatted processable content syntax" (clause 4)

A valid formatted processable character content syntax is an empty string (3) or a string in which the first element is a formatted processable content element (4) and the remainder of the string has a valid formatted processable content syntax (5).

Definition 6.52

1 $\forall v$
2 $({}_{0}$IsValidFormattedProcessableContentSyntax(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $({}_{1}$IsFormattedProcessableContentElement$^{6.55}$(HEAD$^{1.13}(v)$) *and*
5 IsValidFormattedProcessableContentSyntax$^{6.52}$(TAIL$^{1.15}(v)){}_{1}){}_{0}$)

Semiformal Description 6.53

Predicate "is a formatted content element" (clause 4)

A formatted character content element is either a content element common to all character content architectures (3) or a layout control function (4).

Definition 6.53

1 $\forall v$
2 $({}_{0}$IsFormattedContentElement(v) *iff*
3 IsCommonContentElement$^{6.56}(v)$ *or*
4 IsLayoutControlFn$^{6.100}(v){}_{0}$)

Semiformal Description 6.54

Predicate "is a processable content element" (clause 4)

A processable character content element is either a content element common to all character content architectures (3) or a logical control function (4).

Definition 6.54

1 $\forall v$
2 $({}_{0}$IsProcessableContentElement(v) *iff*
3 IsCommonContentElement$^{6.56}(v)$ *or*
4 IsLogicalControlFn$^{6.104}(v){}_{0}$)

Semiformal Description 6.55

Predicate "is a formatted processable content element" (clause 4)

A formatted processable character content element is either a content element common to all character content architectures (3), a layout control function (4), a logical control function (5), or a delimiter control function (6).

Definition 6.55

1 $\forall v$
2 ($_0$IsFormattedProcessableContentElement(v) *iff*
3 IsCommonContentElement$^{6.56}$(v) *or*
4 IsLayoutControlFn$^{6.100}$(v) *or*
5 IsLogicalControlFn$^{6.104}$(v) *or*
6 IsDelimiterControlFn$^{6.108}$(v)$_0$)

Semiformal Description 6.56

Predicate "is common content element" (clause 4)

A content element common to all character content architectures is either a shared control function (3), a space control function (4), a code extension control function (5), an escape control function (6), or a graphic character string (7).

Definition 6.56

1 $\forall v$
2 ($_0$IsCommonContentElement(v) *iff*
3 IsSharedControlFn$^{6.96}$(v) *or*
4 IsSpaceControlFn$^{6.109}$(v) *or*
5 IsCodeExtensionControlFn$^{6.91}$(v) *or*
6 IsEscapeControlFn$^{6.110}$(v) *or*
7 IsGraphicCharacterString$^{6.112}$(v)$_0$)

Semiformal Description 6.57

Predicate "is a valid formatted content semantics" (clause 4)

A character content with valid formatted content semantics must have valid semantics for its common content elements (3).

Definition 6.57

1 $\forall v$
2 ($_0$IsValidFormattedContentSemantics(v) *iff*
3 IsValidCommonContentSemantics$^{6.60}$(v)$_0$)

Semiformal Description 6.58

Predicate "is a valid processable content semantics" (clause 4)

A character content with valid processable content semantics must have valid semantics for its common content elements (3) and its logical control elements (4).

Definition 6.58

1 $\forall v$
2 $(_0$IsValidProcessableContentSemantics(v) *iff*
3 IsValidCommonContentSemantics$^{6.60}(v)$ *and*
4 IsValidLogicalControlSemantics$^{6.61}(v)_0)$

Semiformal Description 6.59

Predicate "is a valid formatted processable content semantics" (clause 4)

A character content with valid formatted processable content semantics must have valid semantics for its common content elements (3), its logical control elements (4), and its delimiter control elements (5).

Definition 6.59

1 $\forall v$
2 $(_0$IsValidFormattedProcessableContentSemantics(v) *iff*
3 IsValidCommonContentSemantics$^{6.60}(v)$ *and*
4 IsValidLogicalControlSemantics$^{6.61}(v)$ *and*
5 IsValidDelimiterControlSemantics$^{6.62}(v)_0)$

Semiformal Description 6.60

Predicate "is a valid common content semantics" (clause 11.1)

A character content with valid common content semantics must have valid shared control semantics (3).

Definition 6.60

1 $\forall v$
2 $(_0$IsValidCommonContentSemantics(v) *iff*
3 IsValidSharedControlSemantics$^{6.63}(v)_0)$

Semiformal Description 6.61

Predicate "is a valid logical control semantics" (clause 11.3)

A character content with valid logical control semantics must have valid PTX function sequence semantics (3).

Definition 6.61

1 $\forall v$
2 $(_0$IsValidLogicalControlSemantics(v) *iff*
3 IsValidPTX$^{6.82}(v)_0)$

Semiformal Description 6.62

Predicate "is a valid delimiter control semantics" (clause 11.4)

A character content with valid logical control semantics must have valid SOS function sequence semantics (3).

Definition 6.62

1 $\forall v$
2 $(_0$IsValidDelimiterControlSemantics(v) *iff*
3 IsValidSOS$^{6.85}(v)_0)$

Semiformal Description 6.63

Predicate "is a valid shared control semantics" (clause 11.1)

A character content with valid shared control semantics must have valid semantics for the function sequences LF (3), GCC (4,5), PLU-PLD (6,8), SRS (7,8,10), VPB-VPR (9-10).

Definition 6.63

1 $\forall v$
2 $(_0$IsValidSharedControlSemantics(v) *iff*
3 IsValidLF$^{6.67}(v)$ *and*
4 IsValidGCC0$^{6.64}(v)$ *and*
5 IsValidGCC12$^{6.65}(v)$ *and*
6 IsValidPLUD$^{6.70}(v)$ *and*
7 IsValidSRS$^{6.73}(v)$ *and*
8 IsValidPLUDInSRS$^{6.76}(v)$ *and*
9 IsValidVP$^{6.80}(v)$ *and*
10 IsValidVPInSRS$^{6.88}(v)_0)$

Semiformal Description 6.64

Predicate "is a valid GCC0" (clause 11.1.2)

A character content string with valid GCC0 control function and parameter semantics is either: (a) an empty string (3), or (b) a string that begins with a GCC control function with the parameter 0 followed by two graphic characters a and b (5, 6) and for which the remainder of the content string has valid GCC0 semantics (7), or (c) a non-empty string that does not begin with a GCC control function with the parameter 0 (9, 10) and for which the remainder of the content string has valid GCC0 semantics (11).

Definition 6.64

1 $\forall v$
2 $(_0$IsValidGCC0(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $\exists t, a, b$
5 $(_1 v = [\rightarrow [\rightarrow$ 'GCC' $\rightarrow 0 \rightarrow] \rightarrow a \rightarrow b \rightarrow]//t$ *and*
6 IsGraphicCharacter$^{6.113}(a)$ *and* IsGraphicCharacter$^{6.113}(b)$ *and*
7 IsValidGCC0$^{6.64}(t)_1)$ *or*
8 $\exists t$
9 $(_2 v \neq [\rightarrow [\rightarrow$ 'GCC' $\rightarrow 0 \rightarrow] \rightarrow]//t$ *and*
10 IsNeCat$^{1.3}(v)$ *and*
11 IsValidGCC0$^{6.64}$(TAIL$^{1.15}(v))_2)_0)$

Semiformal Description 6.65

Predicate "is a valid GCC12" (clause 11.1.2)

A character content string with valid GCC12 control function semantics and parameter is either: (a) an empty string (3), or (b) a string that begins with a GCC control function with the parameter 1 (5) and for which the remainder of the content string has a length of at least 2 (7) and valid GCC12 semantics (6), or (c) a non-empty string that does not begin with a GCC control function with the parameter 1 or 2 (9, 10) and for which the remainder of the content string has valid GCC12 semantics (11).

Definition 6.65

1 $\forall v$
2 $(_0$IsValidGCC12(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $\exists t$
5 $(_1 v = [\rightarrow [\rightarrow$ 'GCC' $\rightarrow 1 \rightarrow] \rightarrow]//t$ *and*
6 IsValidGCC12End$^{6.66}(t)$ *and*
7 LENGTH$^{1.16}(t) \geq 2_1)$ *and*
8 $\exists t, n$
9 $(_2$ *not* $(v = [\rightarrow [\rightarrow$ 'GCC' $\rightarrow n \rightarrow] \rightarrow]//t$ *and* $n \in [1;2])$ *and*
10 IsNeCat$^{1.3}(v)$ *and*
11 IsValidGCC12$^{6.65}$(TAIL$^{1.15}(v))_2)_0)$

Semiformal Description 6.66

Predicate "is a valid GCC12 end" (clause 11.1.2)

A character string with valid GCC12 end semantics is (a) a string which begins with the control function GCC with the parameter 2 and followed by a string with valid GCC12 semantics (3–5) or (b) a non-empty character string which does not start with the control function GCC with the parameter 1, 2, or 3 and where the first character is a graphic character (7, 8) and where the remainder of the string has valid GCC12 end semantics (9).

Definition 6.66

1 $\forall v$
2 $(_0$IsValidGCC12End(v) *iff*
3 $\exists t$
4 $(_1 v = [\rightarrow [\rightarrow$ 'GCC' $\rightarrow 2 \rightarrow] \rightarrow]//t$ *and*
5 IsValidGCC12$^{6.65}(t)_1)$ *or*
6 $\exists t, n$
7 $(_2$ *not* $(v = [\rightarrow [\rightarrow$ 'GCC' $\rightarrow n \rightarrow] \rightarrow]//t$ *and* $n \in [0;1;2])$ *and*
8 IsNeCat$^{1.3}(v)$ *and* IsGraphicCharacter$^{6.113}$(HEAD$^{1.13}(v))$ *and*
9 IsValidGCC12End$^{6.66}$(TAIL$^{1.15}(v))_2)_0)$

Semiformal Description 6.67

Predicate "is a valid LF" (clause 11.1.4)

A character content string with valid LF function semantics is either: (a) an empty string (3), or (b) a string that begins with an LF function (5) and for which the remainder of the content string has valid LF semantics assuming a leading LF function (6), or (c) a non-empty string that does not begin with an LF function (8,9) and for which the remainder of the content string has valid LF semantics assuming that there is no leading LF function (10).

Definition 6.67

1 $\forall v$
2 $({}_0\text{IsValidLF}(v)$ *iff*
3 $\text{IsEmptyCat}^{1.6}(v)$ *or*
4 $\exists t$
5 $({}_1 v = [\rightarrow [\rightarrow \text{'LF'} \rightarrow] \rightarrow]//t$ *and*
6 $\text{IsValidNextLF1}^{6.69}(t)_1)$ *or*
7 $\exists t$
8 $({}_2 v \neq [\rightarrow [\rightarrow \text{'LF'} \rightarrow] \rightarrow]//t$ *and*
9 $\text{IsNeCat}^{1.3}(v)$ *and*
10 $\text{IsValidNextLF0}^{6.68}(\text{TAIL}^{1.15}(v))_2)_0)$

Semiformal Description 6.68

Predicate "is a valid next LF zero" (clause 11.1.4)

A character content string with valid LF function semantics assuming no leading LF is either: (a) an empty string (3), or (b) a string that begins with CR-LF functions (5) and for which the remainder of the content string has valid LF semantics assuming a leading LF function (6), or (c) a non-empty string that does not begin with CR-LF functions (8,9) and for which the remainder of the content string has valid LF semantics assuming that there are no leading LF functions (10).

Definition 6.68

1 $\forall v$
2 $({}_0\text{IsValidNextLF0}(v)$ *iff*
3 $\text{IsEmptyCat}^{1.6}(v)$ *or*
4 $\exists t$
5 $({}_1 v = [\rightarrow [\rightarrow \text{'CR'} \rightarrow] \rightarrow [\rightarrow \text{'LF'} \rightarrow] \rightarrow]//t$ *and*
6 $\text{IsValidNextLF1}^{6.69}(t)_1)$ *or*
7 $\exists t$
8 $({}_2 v \neq [\rightarrow [\rightarrow \text{'CR'} \rightarrow] \rightarrow [\rightarrow \text{'LF'} \rightarrow] \rightarrow]//t$ *and*
9 $\text{IsNeCat}^{1.3}(v)$ *and*
10 $\text{IsValidNextLF0}^{6.68}(\text{TAIL}^{1.15}(v))_2)_0)$

Semiformal Description 6.69

Predicate "is a valid next LF one" (clause 11.1.4)

A character content string with valid LF function semantics assuming a leading LF is either: (a) an empty string (3), or (b) a string that begins with an LF function (5) and for which the remainder of the content string has valid LF semantics assuming a leading LF function (6), or (c) a string that begins with CR-LF functions (8) and for which the remainder of the content string has valid LF semantics assuming a leading LF function (9), or (d) a string that begins with a CR function with no subsequent LF function (11) and for which the remainder of the content string has valid LF semantics assuming that there are no leading LF functions (12), or (e) a non-empty string that does not begin with CR or LF functions (14,15) and for which the remainder of the content string has valid LF semantics assuming that there are no leading LF functions (16).

Definition 6.69

1 $\forall v$
2 $(_0$ IsValidNextLF1(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $\exists t$
5 $(_1 v = [\rightarrow [\rightarrow$ 'LF' $\rightarrow] \rightarrow]//t$ *and*
6 IsValidNextLF1$^{6.69}(t)_1$) *or*
7 $\exists t$
8 $(_2 v = [\rightarrow [\rightarrow$ 'CR' $\rightarrow] \rightarrow [\rightarrow$ 'LF' $\rightarrow] \rightarrow]//t$ *and*
9 IsValidNextLF1$^{6.69}(t)_2$) *or*
10 $\exists t, x$
11 $(_3 v = [\rightarrow [\rightarrow$ 'CR' $\rightarrow] \rightarrow x \rightarrow]//t$ *and* $x \neq [\rightarrow$ 'LF' $\rightarrow]$ *and*
12 IsValidNextLF0$^{6.68}([\rightarrow x \rightarrow]//t)_3$) *or*
13 $\exists t, x$
14 $(_4$ *not* $(v = [\rightarrow [\rightarrow x \rightarrow] \rightarrow]//t$ *and* $x \in$ ['CR','LF']) *and*
15 IsNeCat$^{1.3}(v)$ *and*
16 IsValidNextLF0$^{6.68}$(TAIL$^{1.15}(v))_4)_0$)

Semiformal Description 6.70

Predicate "is a valid PLUD" (clauses 11.1.5 and 11.1.6)

A character content string with valid PLU-PLD function sequence semantics is either: (a) an empty string (3), or (b) a string that begins with a PLU function (5) and for which the remainder of the content string has valid incomplete PLU-PLD semantics (6), or (c) a string that begins with a PLD function (8) and for which the remainder of the content string has valid incomplete PLD-PLU semantics (9), or (d) a non-empty string that does not begin with PLU or PLD functions (11,12) and for which the remainder of the content string has valid PLU-PLD function sequence semantics (13).

Definition 6.70

1 $\forall v$
2 $(_0$IsValidPLUD(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $\exists t$
5 $(_1 v = [\rightarrow [\rightarrow$ 'PLU' $\rightarrow] \rightarrow]//t$ *and*
6 IsIncompletePLU$^{6.71}(t)_1)$ *or*
7 $\exists t$
8 $(_2 v = [\rightarrow [\rightarrow$ 'PLD' $\rightarrow] \rightarrow]//t$ *and*
9 IsIncompletePLD$^{6.72}(t)_2)$ *or*
10 $\exists t, x$
11 $(_3$ *not* $(v = [\rightarrow [\rightarrow x \rightarrow] \rightarrow]//t$ *and* $x \in$ ['PLU'; 'PLD'$]$) *and*
12 IsNeCat$^{1.3}(v)$ *and*
13 IsValidPLUD$^{6.70}$(TAIL$^{1.15}(v))_3)_0)$

Semiformal Description 6.71

Predicate "is incomplete PLU" (clause 11.1.6)

A character content string with valid incomplete PLU-PLD semantics is either: (a) a string that begins with a PLD function (4) and for which the remainder of the content string has valid PLU-PLD function sequence semantics (5), or (b) a non-empty string that does not begin with a PLU, PLD or LF function (7-8) and for which the remainder of the content string has valid incomplete PLU-PLD semantics (9).

Definition 6.71

1 $\forall v$
2 $(_0$IsIncompletePLU(v) *iff*
3 $\exists t$
4 $(_1 v = [\rightarrow [\rightarrow$ 'PLD' $\rightarrow] \rightarrow]//t$ *and*
5 IsValidPLUD$^{6.70}(t)_1)$ *or*
6 $\exists t, x$
7 $(_2$ *not* $(v = [\rightarrow [\rightarrow x \rightarrow] \rightarrow]//t$ *and* $x \in$ ['PLU'; 'PLD'; 'LF'$]$) *and*
8 IsNeCat$^{1.3}(v)$ *and*
9 IsIncompletePLU$^{6.71}$(TAIL$^{1.15}(v))_2)_0)$

Semiformal Description 6.72

Predicate "is incomplete PLD" (clause 11.1.5)

A character content string with valid incomplete PLD-PLU semantics is either: (a) a string that begins with a PLU function (4) and for which the remainder of the content string has valid PLU-PLD function sequence semantics (5), or (b) a non-empty string that does not begin with a PLU, PLD or LF function (7-8) and for which the remainder of the content string has valid incomplete PLD-PLU semantics (9).

Definition 6.72

$$
\begin{array}{ll}
1 & \forall v \\
2 & ({}_0\,\mathrm{IsIncompletePLD}(v)\ \underline{\textit{iff}} \\
3 & \quad \exists t \\
4 & \quad ({}_1\, v = [\rightarrow [\rightarrow \texttt{'PLU'} \rightarrow] \rightarrow]//t\ \underline{\textit{and}} \\
5 & \quad\ \mathrm{IsValidPLUD}^{6.70}(t)_1)\ \underline{\textit{or}} \\
6 & \quad \exists t, x \\
7 & \quad ({}_2\,\underline{\textit{not}}\ (v = [\rightarrow [\rightarrow x \rightarrow] \rightarrow]//t\ \underline{\textit{and}}\ x \in [\texttt{'PLU'}; \texttt{'PLD'}; \texttt{'LF'}])\ \underline{\textit{and}} \\
8 & \qquad\quad \mathrm{IsNeCat}^{1.3}(v)\ \underline{\textit{and}} \\
9 & \qquad\quad \mathrm{IsIncompletePLD}^{6.72}(\mathrm{TAIL}^{1.15}(v))_2)_0)
\end{array}
$$

Semiformal Description 6.73

Predicate "is a valid SRS" (clause 11.1.11)

A character content string with valid SRS function sequence semantics is either: (a) an empty string (3), or (b) a string that begins with an SRS-1 function (5) and for which the remainder of the content string has valid ending SRS semantics (6), or (c) a non-empty string that does not begin with an SRS-1 or SRS-0 function (8) and for which the remainder of the content string has valid SRS function sequence semantics (10).

Definition 6.73

$$
\begin{array}{ll}
1 & \forall v \\
2 & ({}_0\,\mathrm{IsValidSRS}(v)\ \underline{\textit{iff}} \\
3 & \quad \mathrm{IsEmptyCat}^{1.6}(v)\ \underline{\textit{or}} \\
4 & \quad \exists t \\
5 & \quad ({}_1\, v = [\rightarrow [\rightarrow \texttt{'SRS'} \rightarrow 1 \rightarrow] \rightarrow]//t\ \underline{\textit{and}} \\
6 & \quad\ \mathrm{IsValidSRSEnd}^{6.74}(t)_1)\ \underline{\textit{or}} \\
7 & \quad \exists t, n \\
8 & \quad ({}_2\,\underline{\textit{not}}\ (v = [\rightarrow [\rightarrow \texttt{'SRS'} \rightarrow n \rightarrow] \rightarrow]//t\ \underline{\textit{and}}\ n \in [0; 1])\ \underline{\textit{and}} \\
9 & \qquad\quad \mathrm{IsNeCat}^{1.3}(v)\ \underline{\textit{and}} \\
10 & \qquad\quad \mathrm{IsValidSRS}^{6.73}(\mathrm{TAIL}^{1.15}(v))_2)_0)
\end{array}
$$

Semiformal Description 6.74

Predicate "is a valid SRS end" (clause 11.1.11)

A character content string with valid ending SRS semantics is either: (a) a string that begins with an SRS-0 function which terminates the sequence (4) and for which the remainder of the content string has valid SRS function sequence semantics (5), or (b) a string that begins with an SRS-1 function, which indicates the start of a nested SRS function sequence (7) and for which the remainder of the content string has two parts, the first of which has valid nested SRS function sequence semantics (8) and the remainder of which has valid ending SRS semantics (9), or (c) a non-empty string that does not begin with an SRS-1, SRS-0, or CR LF function pair (10-13) and for which the remainder of the content string has valid ending SRS semantics (14).

Definition 6.74

1 $\forall v$
2 $({}_{0}\text{IsValidSRSEnd}(v)$ *iff*
3 $\exists t$
4 $({}_{1}\, v = [\rightarrow [\rightarrow \text{'SRS'} \rightarrow 0 \rightarrow] \rightarrow]//t$ *and*
5 $\text{IsValidSRS}^{6.73}(t)_{1})$ *or*
6 $\exists s, t$
7 $({}_{2}\, v = [\rightarrow [\rightarrow \text{'SRS'} \rightarrow 1 \rightarrow] \rightarrow]//s$ *and*
8 $\text{IsValidSRSNest}^{6.75}(s, t)$ *and*
9 $\text{IsValidSRSEnd}^{6.74}(t)_{2})$ *or*
10 $\exists t, n$
11 $({}_{3}\, v \neq [\rightarrow [\rightarrow \text{'CR'} \rightarrow] \rightarrow [\rightarrow \text{'LF'} \rightarrow] \rightarrow]//t$ *and*
12 $v \neq [\rightarrow [\rightarrow \text{'SRS'} \rightarrow n \rightarrow] \rightarrow]//t$ *and* $n \in [0; 1]$ *and*
13 $\text{IsNeCat}^{1.3}(v)$ *and*
14 $\text{IsValidSRSEnd}^{6.74}(\text{TAIL}^{1.15}(v))_{3})_{0})$

Semiformal Description 6.75

Predicate "is a valid SRS nest" (clause 11.1.11)

A character content string and substring with valid nested SRS function sequence semantics are either: (a) a string that begins with an SRS-0 function which terminates the nested sequence and for which the substring remains (4), or (b) a string that begins with an SRS-1 function, which indicates the start of a nested SRS function sequence (6) and for which the remainder of the content string has two parts, the first of which has valid nested SRS function sequence semantics (7) and the remainder is the substring (8), or (c) a non-empty string that does not begin with an SRS-1, SRS-0, or CR LF function pair (10-12) and for which the remainder of the content string has valid nested SRS function sequence semantics with the substring as remainder (13).

Definition 6.75

1 $\forall v, v_1$
2 $({}_{0}\text{IsValidSRSNest}(v, v_1)$ *iff*
3 $\exists t$
4 $({}_{1}\, v = [\rightarrow [\rightarrow \text{'SRS'} \rightarrow 0 \rightarrow] \rightarrow]//t$ *and* $t = v_{1\,1})$ *or*
5 $\exists s, t$
6 $({}_{2}\, v = [\rightarrow [\rightarrow \text{'SRS'} \rightarrow 1 \rightarrow] \rightarrow]//s$ *and*
7 $\text{IsValidSRSNest}^{6.75}(s, t)$ *and*
8 $\text{IsValidSRSNest}^{6.75}(t, v_1)_{2})$ *or*
9 $\exists t, n$
10 $({}_{3}\, v \neq [\rightarrow [\rightarrow \text{'CR'} \rightarrow] \rightarrow [\rightarrow \text{'LF'} \rightarrow] \rightarrow]//t$ *and*
11 $v \neq [\rightarrow [\rightarrow \text{'SRS'} \rightarrow n \rightarrow] \rightarrow]//t$ *and* $n \in [0; 1]$ *and*
12 $\text{IsNeCat}^{1.3}(v)$ *and*
13 $\text{IsValidSRSNest}^{6.75}(\text{TAIL}^{1.15}(v), v_1)_{3})_{0})$

Semiformal Description 6.76

Predicate "is a valid PLUD in SRS" (clause 11.1.11)

A character content string with valid PLU-PLD in SRS function sequence semantics is either: (a) an empty string (3), or (b) a string that begins with an SRS-1 function (5) and for which the remainder of the content string has PLU-PLD in ending SRS semantics (6), or (c) a non-empty string that does not begin with an SRS-1 function (8) and for which the remainder of the content string has valid PLU-PLD in SRS function sequence semantics (9).

Definition 6.76

1 $\forall v$
2 $({}_0$IsValidPLUDInSRS(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $\exists t$
5 $({}_1 v = [\rightarrow [\rightarrow$ 'SRS' $\rightarrow 1 \rightarrow] \rightarrow] // t$ *and*
6 IsValidPLUDInSRSEnd$^{6.77}(t)_1)$ *or*
7 $\exists t$
8 $({}_2 v \neq [\rightarrow [\rightarrow$ 'SRS' $\rightarrow 1 \rightarrow] \rightarrow] // t$ *and* IsNeCat$^{1.3}(v)$ *and*
9 IsValidPLUDInSRS$^{6.76}$(TAIL$^{1.15}(v))_2)_0)$

Semiformal Description 6.77

Predicate "is a valid PLUD in SRS end" (clause 11.1.11)

A character content string with valid PLU-PLD in ending SRS semantics is either: (a) a string that begins with an SRS-0 function (4) and for which the remainder of the content string has valid PLU-PLD in SRS function sequeunce semantics (5), or (b) a string that begins with an SRS-1 function (7) and for which the remainder of the string has valid PLU-PLD in ending SRS semantics (8), or (c) a string that begins with a PLU function (10) and for which the remainder of the content string has valid incomplete PLU-PLD in SRS semantics (11), or (d) a string that begins with a PLD function (13) and for which the remainder of the content string has valid incomplete PLD-PLU in SRS semantics (14), or (e) a non-empty string that does not begin with PLU, PLD, SRS-0, or SRS-1 functions (16-18) and for which the remainder of the content string has valid PLU-PLD in ending SRS semantics (19).

Definition 6.77

1 $\forall v$
2 $({}_0$IsValidPLUDInSRSEnd(v) *iff*
3 $\exists t$
4 $({}_1 v = [\rightarrow [\rightarrow$ 'SRS' $\rightarrow 0 \rightarrow] \rightarrow] // t$ *and*
5 IsValidPLUDInSRS$^{6.76}(t)_1)$ *or*
6 $\exists t$
7 $({}_2 v = [\rightarrow [\rightarrow$ 'SRS' $\rightarrow 1 \rightarrow] \rightarrow] // t$ *and*
8 IsValidPLUDInSRSEnd$^{6.77}(t)_2)$ *or*
9 $\exists t$
10 $({}_3 v = [\rightarrow [\rightarrow$ 'PLU' $\rightarrow] \rightarrow] // t$ *and*
11 IsIncompletePLUInSRS$^{6.78}(t)_3)$ *or*
12 $\exists t$
13 $({}_4 v = [\rightarrow [\rightarrow$ 'PLD' $\rightarrow] \rightarrow] // t$ *and*
14 IsIncompletePLDInSRS$^{6.79}(t)_4)$ *or*
15 $\exists t, x, n$
16 $({}_5 v \neq [\rightarrow [\rightarrow x \rightarrow] \rightarrow] // t$ *and* $x \in$ ['PLU'; 'PLD'] *and*
17 $v \neq [\rightarrow [\rightarrow$ 'SRS' $\rightarrow n \rightarrow] \rightarrow] // t$ *and* $n \in [0; 1]$ *and*
18 IsNeCat$^{1.3}(v)$ *and*
19 IsValidPLUDInSRSEnd$^{6.77}$(TAIL$^{1.15}(v))_5)_0)$

Semiformal Description 6.78

Predicate "is incomplete PLU in SRS" (clause 11.1.11)

A character content string with valid incomplete PLU-PLD in SRS semantics is either: (a) a string that begins with a PLD function (4) and for which the remainder of the content string has valid PLU-PLD in ending SRS semantics (5), or (b) a non-empty string that does not begin with PLD or SRS-0 (7,8) and for which the remainder of the content string has valid incomplete PLU-PLD in SRS semantics (9).

Definition 6.78

1 $\forall v$
2 $({}_0\text{IsIncompletePLUInSRS}(v)$ *iff*
3 $\exists t$
4 $({}_1 v = [\rightarrow [\rightarrow \text{'PLD'} \rightarrow] \rightarrow]//t$ *and*
5 $\text{IsValidPLUDInSRSEnd}^{6.77}(t)_1)$ *or*
6 $\exists t$
7 $({}_2 v \neq [\rightarrow [\rightarrow \text{'SRS'} \rightarrow 0 \rightarrow] \rightarrow]//t$ *and*
8 $v \neq [\rightarrow [\rightarrow \text{'PLD'} \rightarrow] \rightarrow]//t$ *and* $\text{IsNeCat}^{1.3}(v)$ *and*
9 $\text{IsIncompletePLUInSRS}^{6.78}(\text{TAIL}^{1.15}(v))_2)_0)$

Semiformal Description 6.79

Predicate "is incomplete PLD in SRS" (clause 11.1.11)

A character content string with valid incomplete PLD-PLU in SRS semantics is either: (a) a string that begins with a PLU function (4) and for which the remainder of the content string has valid PLU-PLD in ending SRS semantics (5), or (b) a non-empty string that does not begin with PLU or SRS-0 (7,8) and for which the remainder of the content string has valid incomplete PLD-PLU in SRS semantics (9).

Definition 6.79

1 $\forall v$
2 $({}_0\text{IsIncompletePLDInSRS}(v)$ *iff*
3 $\exists t$
4 $({}_1 v = [\rightarrow [\rightarrow \text{'PLU'} \rightarrow] \rightarrow]//t$ *and*
5 $\text{IsValidPLUDInSRSEnd}^{6.77}(t)_1)$ *or*
6 $\exists t$
7 $({}_2 v \neq [\rightarrow [\rightarrow \text{'SRS'} \rightarrow 0 \rightarrow] \rightarrow]//t$ *and*
8 $v \neq [\rightarrow [\rightarrow \text{'PLU'} \rightarrow] \rightarrow]//t$ *and* $\text{IsNeCat}^{1.3}(v)$ *and*
9 $\text{IsIncompletePLDInSRS}^{6.79}(\text{TAIL}^{1.15}(v))_2)_0)$

Semiformal Description 6.80

Predicate "is a valid VP" (clauses 11.1.15 and 11.1.16)

A character content string with valid VPB-VPR function sequence semantics is a string in which the sum of the VP changes is zero.

Definition 6.80

1 $\forall v$
2 $({}_0\text{IsValidVP}(v)$ *iff*
3 $\text{IsValidVPSum}^{6.81}(v, 0)_0)$

Semiformal Description 6.81

Predicate "is a valid VP sum" (clause 11.1.15)

n is a valid sum of VP changes for a character content string v if v and n satisfy: (a) v is an empty string and $n = 0$ (3, 4); or (b) v begins with VPB-i, where i is a natural number and $n + i$ is a valid sum of VP changes for the remainder of v (5–7); or (c) v begins with VPR-i, where i is a natural number and $n - i$ is a valid sum of VP changes for the remainder of v (8–10); or (d) v begins with the control functions CR-LF and $n = 0$ and zero is a valid sum of VP changes for the remainder of v (11-13); or (e) v is non-empty and does not begin with VPB, VPR or CR-LF, and n is a valid sum of VP changes for the remainder of v (14–18).

Definition 6.81

```
1   ∀v, n
2   (0 IsValidVPSum(v, n) iff
3     (1 IsInt(n) and
4       (2 IsEmptyCat^1.6(v) and n = 0 2) or
5       ∃t, i
6       (3 v = [ → [ → 'VPB' → i → ] → ]//t and IsNat(i) and
7         IsValidVPSum^6.81(t, n + i) 3) or
8       ∃t, i
9       (4 v = [ → [ → 'VPR' → i → ] → ]//t and IsNat(i) and
10        IsValidVPSum^6.81(t, n − i) 4) or
11      ∃t
12      (5 v = [ → [ → 'CR' → ] → [ → 'LF' → ] → ]//t and n = 0 and
13        IsValidVPSum^6.81(t, 0) 5) or
14      ∃t, i, x
15      (6 v ≠ [ → [ → x → i → ] → ]//t and x ∈ ['VPB', 'VPR'] and
16        v ≠ [ → [ → 'CR' → ] → [ → 'LF' → ] → ]//t and
17        IsNat(i) and IsNeCat^1.3(v) and
18        IsValidVPSum^6.81(TAIL^1.15(v), n) 6) 1) 0)
```

Semiformal Description 6.82

Predicate "is a valid PTX" (clause 11.3.3)

A character content string with valid PTX function sequence semantics is either: (a) an empty string (3), or (b) a string that begins with a PTX-1 function (5) and for which the remainder of the content string has valid next stage PTX semantics (6), or (c) a non-empty string that does not begin with PTX-0, PTX-1 or PTX-3 (8,9) and for which the remainder of the content string has valid PTX function sequence semantics (10).

Definition 6.82

```
1   ∀v
2   (0 IsValidPTX(v) iff
3     IsEmptyCat^1.6(v) or
4     ∃t
5     (1 v = [ → [ → 'PTX' → 1 → ] → ]//t and
6       IsValidPTXNext^6.83(t) 1) or
7     ∃t, n
8     (2 not (v = [ → [ → 'PTX' → n → ] → ]//t and n ∈ [0; 1; 3]) and
9           IsNeCat^1.3(v) and
10          IsValidPTX^6.82(TAIL^1.15(v)) 2) 0)
```

Semiformal Description 6.83

Predicate "is valid PTX next" (clause 11.3.3)

A character content string with valid next stage PTX semantics is either: (a) a string that begins with a PTX-3 function (4) and for which the remainder of the content string has valid ending PTX semantics (5), or (b) a non-empty string that does not begin with PTX-0, PTX-1 or PTX-3 (7,8) and for which the remainder of the content string has valid next stage PTX semantics (9).

Definition 6.83

1 $\forall v$
2 $({}_0\text{IsValidPTXNext}(v)$ *iff*
3 $\exists t$
4 $({}_1 v = [\rightarrow [\rightarrow \text{'PTX'} \rightarrow 3 \rightarrow] \rightarrow]//t$ *and*
5 $\text{IsValidPTXEnd}^{6.84}(t)_1)$ *or*
6 $\exists t, n$
7 $({}_2$ *not* $(v = [\rightarrow [\rightarrow \text{'PTX'} \rightarrow n \rightarrow] \rightarrow]//t$ *and* $n \in [0; 1; 3])$ *and*
8 $\text{IsNeCat}^{1.3}(v)$ *and*
9 $\text{IsValidPTXNext}^{6.83}(\text{TAIL}^{1.15}(v))_2)_0)$

Semiformal Description 6.84

Predicate "is a valid PTX end" (clause 11.3.3)

A character content string with valid ending PTX semantics is either: (a) a string that begins with a PTX-0 function (4) and for which the remainder of the content string has valid ending PTX semantics (5), or (b) a non-empty string that does not begin with PTX-0, PTX-1 or PTX-3 (7,8) and for which the remainder of the content string has valid ending PTX semantics (9).

Definition 6.84

1 $\forall v$
2 $({}_0\text{IsValidPTXEnd}(v)$ *iff*
3 $\exists t$
4 $({}_1 v = [\rightarrow [\rightarrow \text{'PTX'} \rightarrow 0 \rightarrow] \rightarrow]//t$ *and*
5 $\text{IsValidPTX}^{6.82}(t)_1)$ *or*
6 $\exists t, n$
7 $({}_2$ *not* $(v = [\rightarrow [\rightarrow \text{'PTX'} \rightarrow n \rightarrow] \rightarrow]//t$ *and* $n \in [0; 1; 3])$ *and*
8 $\text{IsNeCat}^{1.3}(v)$ *and*
9 $\text{IsValidPTXEnd}^{6.84}(\text{TAIL}^{1.15}(v))_2)_0)$

Semiformal Description 6.85

Predicate "is a valid SOS" (clause 11.4.1)

A character content string with valid SOS function sequence semantics is either: (a) an empty string (3), or (b) a string that begins with an SOS function (5) and for which the remainder of the content string has valid ending SOS semantics (6), or (c) a non-empty string that does not begin with an SOS function (8) and for which the remainder of the content string has valid SOS function sequence semantics (9).

Definition 6.85

1 $\forall v$
2 $(_0$IsValidSOS(v) *iff*
3 IsEmptyCat$^{1.6}(v)$ *or*
4 $\exists t$
5 $(_1 v = [\rightarrow [\rightarrow$ 'SOS' $\rightarrow] \rightarrow]//t$ *and*
6 IsValidSOSEnd$^{6.86}(t)_1)$ *or*
7 $\exists t$
8 $(_2 v \neq [\rightarrow [\rightarrow$ 'SOS' $\rightarrow] \rightarrow]//t$ *and* IsNeCat$^{1.3}(v)$ *and*
9 IsValidSOS$^{6.85}$(TAIL$^{1.15}(v))_2)_0)$

Semiformal Description 6.86

Predicate "is a valid SOS end" (clause 11.4.1)

A character content string with valid ending SOS semantics is either: (a) a string that begins with an ST function which terminates the sequence (4) and for which the remainder of the content string has valid SOS function sequence semantics (5), or (b) a string that begins with an SOS function, which indicates the start of a nested SOS function sequence and for which the remainder of the content string has two parts, the first of which has valid nested SOS function sequence semantics (8) and the remainder of which ending SOS semantics (9), or (c) a non-empty string that does not begin with an SOS or ST function (11,12) and for which the remainder of the content string has valid ending SOS semantics (13).

Definition 6.86

1 $\forall v$
2 $(_0$IsValidSOSEnd(v) *iff*
3 $\exists t$
4 $(_1 v = [\rightarrow [\rightarrow$ 'ST' $\rightarrow] \rightarrow]//t$ *and*
5 IsValidSOS$^{6.85}(t)_1)$ *or*
6 $\exists s,t$
7 $(_2 v = [\rightarrow [\rightarrow$ 'SOS' $\rightarrow] \rightarrow]//s$ *and*
8 IsValidSOSNest$^{6.87}(s,t)$ *and*
9 IsValidSOSEnd$^{6.86}(t)_2)$ *or*
10 $\exists t,a$
11 $(_3$ *not* $(v = [\rightarrow [\rightarrow a \rightarrow] \rightarrow]//t$ *and* $a \in$ ['SOS';'ST']) *and*
12 IsNeCat$^{1.3}(v)$ *and*
13 IsValidSOSEnd$^{6.86}$(TAIL$^{1.15}(v))_3)_0)$

Semiformal Description 6.87

Predicate "is a valid SOS nest" (clause 11.4.1)

A character content string and substring with valid nested SOS function sequence semantics are either: (a) a string that begins with an ST function which terminates the nested sequence and for which the substring remains (4), or (b) a string that begins with an SOS function, which indicates the start of a nested SOS function sequence (6) and for which the remainder of the content string has has two parts, the first of which has valid nested SOS function sequence semantics and the remainder is the substring (7,8), or (c) a non-empty string that does not begin with an SOS or ST function (10,11) and for which the remainder of the content string has valid nested SOS function sequence semantics with the substring as remainder (12).

Definition 6.87

```
1   ∀ v, v1
2   (0 IsValidSOSNest(v, v1) iff
3     ∃ t
4     (1 v = [ → [ → 'ST' → ] → ]//t and t = v1 1) or
5     ∃ s, t
6     (2 v = [ → [ → 'SOS' → ] → ]//s and
7       IsValidSOSNest^6.87(s, t) and
8       IsValidSOSNest^6.87(t, v1)2) or
9     ∃ t, a
10    (3 not (v = [ → [ → a → ] → ]//t and a ∈ ['SOS';'ST']) and
11          IsNeCat^1.3(v) and
12          IsValidSOSNest^6.87(TAIL^1.15(v), v1)3)0)
```

Semiformal Description 6.88

Predicate "is a valid VP in SRS" (clause 11.1.11)

A character content string with valid VPB-VPR in SRS function sequence semantics is a string in which the sum of the VP changes in an SRS sequence is zero.

Definition 6.88

```
1   ∀ v
2   (0 IsValidVPInSRS(v) iff
3     IsValidVPSumInSRS^6.89(v, 0)0)
```

Semiformal Description 6.89

Predicate "is a valid VP sum in SRS" (clause 11.1.11)

n is a valid sum of VP changes in SRS for a character content string v if v and n satisfy: (a) v is an empty string and $n = 0$ (3, 4); or (b) v begins with SRS-1 and n is a valid sum of VP changes in SRS end for the remainder of v (5–7); or (b) v is a non-empty string which does not begin with SRS-1 and n is a valid sum of VP changes in SRS for the remainder of v (8–10);

Definition 6.89

1 $\forall v, n$
2 $(_0$IsValidVPSumInSRS(v, n) *iff*
3 $\quad(_1$IsInt(n) *and*
4 $\quad\quad(_2$IsEmptyCat$^{1.6}(v)$ *and* $n = 0_2)$ *or*
5 $\quad\quad\exists t$
6 $\quad\quad(_3 v = [\rightarrow [\rightarrow$ 'SRS' $\rightarrow 1 \rightarrow] \rightarrow]//t$ *and*
7 $\quad\quad$IsValidVPSumInSRSEnd$^{6.90}(t, n)_3)$ *or*
8 $\quad\quad\exists t$
9 $\quad\quad(_4 v \neq [\rightarrow [\rightarrow$ 'SRS' $\rightarrow 1 \rightarrow] \rightarrow]//t$ *and* IsNeCat$^{1.3}(v)$ *and*
10 $\quad\quad$IsValidVPSumInSRS$^{6.89}$(TAIL$^{1.15}(v), n)_4)_1)_0)$

Semiformal Description 6.90

Predicate "is a valid VP sum in SRS end" (clause 11.1.11)

n is a valid sum of VP changes in SRS end for a character content string v if v and n satisfy: or (a) v begins with VPB-i, where i is a natural number and $n+i$ is a valid sum of VP changes in SRS end for the remainder of v (4–6); or (b) v begins with VPR-i, where i is a natural number and $n - i$ is a valid sum of VP changes in SRS end for the remainder of v (7–9); or (c) v begins with SRS-0 and $n = 0$ and zero is a valid sum of VP changes in SRS end for the remainder of v (10-12); or (d) v is non-empty and does not begin with VPB, VPR or SRS with a positive parameter, and n is a valid sum of VP changes in SRS end for the remainder of v (13–16).

Definition 6.90

1 $\forall v, n$
2 $(_0$IsValidVPSumInSRSEnd(v, n) *iff*
3 $\quad(_1$IsInt(n) *and*
4 $\quad\quad\exists t, i$
5 $\quad\quad(_2 v = [\rightarrow [\rightarrow$ 'VPB' $\rightarrow i \rightarrow] \rightarrow]//t$ *and* IsNat(i) *and*
6 $\quad\quad$IsValidVPSumInSRSEnd$^{6.90}(t, n + i)_2)$ *or*
7 $\quad\quad\exists t, i$
8 $\quad\quad(_3 v = [\rightarrow [\rightarrow$ 'VPR' $\rightarrow i \rightarrow] \rightarrow]//t$ *and* IsNat(i) *and*
9 $\quad\quad$IsValidVPSumInSRSEnd$^{6.90}(t, n - i)_3)$ *or*
10 $\quad\quad\exists t$
11 $\quad\quad(_4 v = [\rightarrow [\rightarrow$ 'SRS' $\rightarrow 0 \rightarrow] \rightarrow]//t$ *and* $n = 0$ *and*
12 $\quad\quad$IsValidVPSumInSRSEnd$^{6.90}(t, 0)_4)$ *or*
13 $\quad\quad\exists t, i, x$
14 $\quad\quad(_5$ *not* $(v = [\rightarrow [\rightarrow x \rightarrow i \rightarrow] \rightarrow]//t$ *and* $x \in$ ['VPB'; 'VPR'; 'SRS']) *and*
15 $\quad\quad\quad$IsNat(i) *and* IsNeCat$^{1.3}(v)$ *and*
16 $\quad\quad\quad$IsValidVPSumInSRSEnd$^{6.90}$(TAIL$^{1.15}(v), n)_5)_1)_0)$

Semiformal Description 6.91

Predicate "is code extension control function element" (clause 11)

An element is a code extension control function if it is a code extension control function for the designation and invocation of graphic character sets and control functions.

Definition 6.91

1 $\forall v$
2 $({}_{o}\text{IsCodeExtensionControlFn}(v)$ *iff*
3 $\text{IsISO2022String}^{6.92}(v)$ *or* $\text{IsISO6429String}^{6.94}(v)_{o})$

Semiformal Description 6.92

Predicate "is ISO 2022 string" (clause 11)

An ISO 2022 string is a catenation whose elements are ISO 2022 code extension control function characters which are used for the designation and invocation of graphic character sets.

Definition 6.92

1 $\forall v$
2 $({}_{o}\text{IsISO2022String}(v)$ *iff*
3 $\text{IsNeCat}^{1.3}(v)$ *and* $\forall a \hat{\in} \hat{}v \bullet (\text{IsISO2022Character}^{6.93}(\text{C } a))_{o})$

Semiformal Description 6.93

Predicate "is ISO 2022 character" (clause 11)

An ISO 2022 character is an element of ISO 2022 and is thus considered an atomic construct in the formal specification.

Definition 6.93

1 $\forall v$
2 $({}_{o}\text{IsISO2022Character}(v)$ *iff*
3 $\text{IsAtom}(v)_{o})$

Semiformal Description 6.94

Predicate "is ISO 6429 string" (clause 11)

An ISO 6429 string is a catenation whose elements are ISO 6429 code extension control function characters which are used for the designation and invocation of graphic character sets.

Definition 6.94

1 $\forall v$
2 $({}_{o}\text{IsISO6429String}(v)$ *iff*
3 $\text{IsNeCat}^{1.3}(v)$ *and* $\forall a \hat{\in} \hat{}v \bullet (\text{IsISO6429Character}^{6.95}(\text{C } a))_{o})$