INTERNATIONAL STANDARD

ISO/IEC 19757-7

Second edition
2020-08

Information technology — Document Schema Definition Languages (DSDL) —

Part 7: Character Repertoire Description Language (CREPDL)

Technologies de l'information — Langages de définition de schéma de documents (DSDL) —

Partie 7: Langage de description de répertoire de caractères (CREPDL)

Reference number
ISO/IEC 19757-7:2020(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see http://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 34, *Document description and processing languages*.

This second edition cancels and replaces the first edition (ISO/IEC 19757-7:2009), which has been technically revised. It also incorporates the Technical Corrigendum ISO/IEC 19757-7:2009/Cor 1:2015.

The main changes compared to the previous edition are as follows:

— addition of validation of grapheme clusters such as 'n' followed by COMBINING GRAVE ACCENT (U+0300) and a CJK unified ideograph followed by a variation selector.

— addition of the Unicode Ideographic Variation Database as a registry.

A list of all parts in the ISO/IEC 19757 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

ISO/IEC 19757 (all parts) defines a set of Document Schema Definition Languages (DSDL) that can be used to specify one or more validation processes performed against Extensible Markup Language (XML) documents. A number of validation technologies are standardized in DSDL to complement those already available as standards or from industry.

The main objective of ISO/IEC 19757 (all parts) is to bring together different validation-related technologies to form a single extensible framework that allows technologies to work in series or in parallel to produce a single or a set of validation results. The extensibility of DSDL accommodates validation technologies not yet designed or specified.

This document provides a language for describing character repertoires. Descriptions in this language can be referenced from schemas. Furthermore, they can also be referenced from forms and stylesheets.

Descriptions of character repertoires doesn't need to be exact. Non-exact descriptions are made possible by kernels and hulls, which provide the lower and upper limits, respectively.

The structure of this document is as follows. Clause 5 provides an informal overview of CREPDL. Clause 6 specifies the syntax of CREPDL schemas. Clause 7 specifies the semantics of a correct CREPDL schema; the semantics specify when a code point or code point sequence is in a character repertoire described by a CREPDL schema. Clause 8 defines the behaviour of CREPDL processors. Finally, Annex A describes differences of conformant CREPDL processors; Annex B provides examples of CREPDL schemas.

Although the first edition was restricted to the validation of characters, this edition can also enable the validation of grapheme clusters such as 'n' followed by COMBINING GRAVE ACCENT (U+0300) and a CJK unified ideograph followed by a variation selector.

CREPDL schemas conformant to the first edition do not conform to this edition. In particular, this edition changes the namespace name for CREPDL schemas.

Information technology — Document Schema Definition Languages (DSDL) —

Part 7: Character Repertoire Description Language (CREPDL)

1 Scope

This document specifies a Character Repertoire Description Language (CREPDL). A CREPDL schema describes a character repertoire. A stream of UCS code points can be validated against a CREPDL schema.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 10646, *Information technology — Universal Multiple-Octet Coded Character Set (UCS)*

ISO/IEC 19757-2, *Information technology — Document Schema Definition Language (DSDL) — Part 2: Regular-grammar-based validation — RELAX NG*

ISO/IEC 19757-4, *Information technology — Document Schema Definition Languages (DSDL) — Part 4: Namespace-based Validation Dispatching Language (NVDL)*

W3C XML, *Extensible Markup Language (XML) 1.0 (Fourth Edition)*, W3C Recommendation, 16 August 2006, available at http://www.w3.org/TR/2006/REC-xml-20060816

W3C XML-Names, *Namespaces in XML (Second Edition)*, W3C Recommendation, 16 August 2006, available at http://www.w3.org/TR/2006/REC-xml-names-20060816

IETF RFC 3987, *Internationalized Resource Identifiers (IRIs), Internet Standards Track Specification*, January 2005, available at http://www.ietf.org/rfc/rfc3987.txt

Charsets I.A.N.A. *IANA CHARACTER SETS*, available at http://www.iana.org/assignments/character-sets

Unicode, *The Unicode Standard*, The Unicode Consortium, available at http://www.unicode.org/

CLDR, *Unicode Common Locale Data Repository*, The Unicode Consortium, available at http://www.unicode.org/cldr/

UAX29, *Unicode Standard Annex #29: Unicode Text Segmentation*, The Unicode Consortium, available at http:// unicode.org/reports/tr29/

UTS35, *Unicode Technical Standard #35: Unicode Locale Data Markup Language (LDML)*, The Unicode Consortium, available at https://www.unicode.org/reports/tr35/

UTS37, *Unicode Technical Standard #37: Unicode Ideographic Variation Database*, The Unicode Consortium, available at http://www.unicode.org/reports/tr37/

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1
CREPDL processor
computer program that validates a stream of code points not containing high- or low-surrogate code points against *CREPDL schemas* (3.2)

3.2
CREPDL schema
machine-readable description of a *repertoire* (3.8)

3.3
grapheme cluster
base character followed by zero or more continuing characters

Note 1 to entry: A grapheme cluster typically represents what the user thinks of as basic unit of a writing system for a language.

[SOURCE: UAX 29]

3.4
hull
set of code points or code point sequences (excluding high- or low-surrogate code points) that are not guaranteed to be excluded from the *repertoire* (3.8)

3.5
kernel
set of code points or code point sequences (excluding high- or low-surrogate code points) that are guaranteed to be included by the *repertoire* (3.8)

3.6
mode
option to specify whether characters or *grapheme clusters* (3.3) are examined

Note 1 to entry: The first edition did not have modes. Thus, characters can be examined, but grapheme clusters cannot.

3.7
registry
collection of named *repertoires* (3.8)

3.8
repertoire
description of a set of code points or code point sequences excluding high- or low-surrogate code points

4 Notation

in(x, A): code point or code point sequence x is in the repertoire described by a CREPDL element A;

not-in(x, A): code point or code point sequence x is not in the repertoire described by a CREPDL element A;

unknown(x, A): it is unknown whether code point or code point sequence x is in the repertoire described by a CREPDL element A.

NOTE 1 This predicate-like notation captures the combination of three-valued logic and the interpretation of a formula for a given character or grapheme cluster. In other words, in(x, A) implies that the interpretation of A under x is truth in three-valued logic. Likewise, not-in(x, A) and unknown(x, A) imply the interpretations of A under x are false and unknown, respectively.

NOTE 2 This document is intended to ensure that exactly one of in(x, A), not-in(x, A), and unknown(x, A) holds.

5 Overview

5.1 Basic constructs and compound constructs

Basic constructs of CREPDL schemas are created from regular expressions or references to registries of repertoires. Compound constructs of CREPDL schemas are created by combining basic constructs by set operators such as union, intersection, and difference.

5.2 Characters and code points

Although the title of this document is "Character Repertoire Description Language", this document uses code points more often than characters. This is because CREPDL allows the use of unassigned code points, which are not characters. For example, U+1CBB is an unassigned code point, and is thus not a character. It is possible to create a CREPDL schema that allows this code point. A stream containing it is valid against such a CREPDL schema.

5.3 Grapheme clusters

CREPDL can enable the validation of grapheme clusters, which are sequences of code points. For example, a CREPDL schema can allow LATIN CAPITAL LETTER N (U+004E) or LATIN SMALL LETTER n (U+006E) followed by COMBINING GRAVE ACCENT (U+0300) while disallowing other characters followed by COMBINING GRAVE ACCENT (U+0300). Likewise, a CREPDL schema can indicate which variation selector can follow which CJK unified ideograph.

NOTE The first edition cannot enable the validation of sequences of code points. It was thus not possible to allow LATIN CAPITAL LETTER N (U+004E) or LATIN SMALL LETTER n (U+006E) followed by COMBINING GRAVE ACCENT (U+0300) without allowing other characters followed by COMBINING GRAVE ACCENT (U+0300).

5.4 Kernel and Hull

It is sometimes difficult to precisely specify a repertoire. As an example, consider collections in ISO/IEC 10646, which are numbered and named repertoires. Some collections are open: they contain assigned code points as well as unassigned code points, which can be assigned in the future.

Recall that some basic constructs of CREPDL schemas are created from regular expressions. Such basic constructs have pairs of regular expressions. One regular expression specifies what is guaranteed to be included, while the other specifies what is not guaranteed to be excluded. The former and latter are called kernel and hull, respectively. If a code point matches the kernel regular expression, the code point is definitely included in the repertoire. Even if it isn't, it it not guranteed be excluded from the repertoire if it matches the hull regular expression.

NOTE Kernel and hull are reproduced from W3C Note-charcol[3]. Some examples in Annex B are also reproduced from W3C Note-charcol[3].

6 Syntax

6.1 General

A CREPDL schema shall be an XML document (which shall be as specified in W3C XML and shall further conform to W3C XML-Names) valid against the NVDL (ISO/IEC 19757-4) script in 6.3, which in turn relies on the RELAX NG (ISO/IEC 19757-2) schema in 6.2. The elements allowed in the RELAX NG schema

are of the namespace `http://purl.oclc.org/dsdl/crepdl/ns/structure/2.0`. Further constraints on the character content of the `char`, `kernel` or `hull` elements are shown in 6.4.

NOTE 1 W3C XML specifies that characters in XML documents are either U+0009 (CHARACTER TABULATION), U+000A (LINE FEED), U+000D (CARRIAGE RETURN), or a character in the ranges from U+0020 to U+D7FF, U+E000 to U+FFFD, or U+10000 to U+10FFFF. Since CREPDL schemas are represented by XML documents, other characters cannot directly occur in CREPDL schemas.

NOTE 2 The first edition used a different namespace name.

6.2 RELAX NG schema

```
# The following permission notice and disclaimer shall be included in
# all copies of this schema ("the Schema"), and derivations of
# the Schema:
#
# Permission is hereby granted, free of charge in perpetuity, to any
# person obtaining a copy of the Schema, to use, copy, modify, merge and
# distribute free of charge, copies of the Schema for the purposes of
# developing, implementing, installing and using software based on the
# Schema, and to permit persons to whom the Schema is furnished to do
# so, subject to the following conditions:
#
# THE SCHEMA IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
# THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
# OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
# ARISING FROM, OUT OF OR IN CONNECTION WITH THE SCHEMA OR THE USE OR
# OTHER DEALINGS IN THE SCHEMA.
#
# In addition, any modified copy of the Schema shall include the following
# notice:
#
# THIS SCHEMA HAS BEEN MODIFIED FROM THE SCHEMA DEFINED IN ISO/IEC 19757-7,
# AND SHOULD NOT BE INTERPRETED AS COMPLYING WITH THAT STANDARD.

default namespace = "http://purl.oclc.org/dsdl/crepdl/ns/structure/2.0"

start = coll
coll =
  union | intersection | difference | ref | repertoire | char
union = element union { commonAtts, coll+ }
intersection = element intersection { commonAtts, coll+ }
difference = element difference { commonAtts, coll+ }
ref =
  element ref {
    commonAtts,
    attribute href { xsd:anyURI }
}
repertoire =
  element repertoire {
    commonAtts,
    attribute registry { text },
    attribute version { text }?,
    (attribute name { text } | attribute number {xsd:int})
}
char =
   element char {
     commonAtts,
     (text
      | element kernel { commonAtts, text }
     | element hull { commonAtts, text }
     | (element kernel { commonAtts, text },
        element hull { commonAtts, text }))
   }
commonAtts =
   attribute minUcsVersion { text }?,
   attribute maxUcsVersion { text }?,
```



```
   attribute mode { "character" | "graphemeCluster" }?
#
# Note that xml:id is allowed, since any foreign attribute is
# allowed by the NVDL script.
```

6.3 NVDL script

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- The following permission notice and disclaimer shall be included in all copies of
this schema ("the Schema"), and derivations of the Schema:  Permission is hereby granted,
free of charge in perpetuity, to any person obtaining a copy of the Schema, to use, copy,
modify, merge and distribute free of charge, copies of the Schema for the purposes of
developing, implementing, installing and using software based on the Schema, and to permit
persons to whom the Schema is furnished to do so, subject to the following conditions:
THE SCHEMA IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SCHEMA OR THE USE OR OTHER DEALINGS IN THE
SCHEMA.
 In addition, any modified copy of the Schema shall include the following notice:
THIS SCHEMA HAS BEEN MODIFIED FROM THE SCHEMA DEFINED IN ISO/IEC 19757-7, AND SHOULD NOT
BE INTERPRETED AS COMPLYING WITH THAT STANDARD.
-->
<rules xmlns="http://purl.oclc.org/dsdl/nvdl/ns/structure/1.0">
  <namespace ns="http://purl.oclc.org/dsdl/crepdl/ns/structure/2.0">
    <validate schema="crepdl.rnc"
       schemaType="application/relax-ng-compact-syntax">
            <mode>
               <anyNamespace match="elements">
                   <allow/>
               </anyNamespace>
               <namespace ns="" match="attributes">
                   <attach/>
               </namespace>
               <anyNamespace match="attributes">
                   <allow/>
               </anyNamespace>
            </mode>
        </validate>
    </namespace>
</rules>
```

This NVDL script allows foreign elements and attributes everywhere.

6.4 Regular Expressions

The character content of a `char`, `kernel` or `hull` element shall be a Unicode set as specified in UTS35, 5.3.3 (Unicode Sets) of Part 1 (Core).

NOTE A Unicode set is guaranteed to be a regular expression as specified in UTS18[8].

7 Semantics

7.1 General

This clause shall specify which character repertoire is represented by a CREPDL element. Specifically, given a code point (which shall be as specified in ISO/IEC 10646) or code point sequence *x*, this clause shall specify when x is in the repertoire, when x is not in the repertoire, and when it is unknown whether x is in the repertoire.

7.2 char

First, the semantics of Unicode sets occurring in `kernel` and `hull` elements shall be as specified in UTS35.

The semantics of `char` shall be defined below.

— Case 1: the `char` element has neither `kernel` nor `hull` as a child element.

  It is assumed that this element has a `kernel` element, the content of which is identical to the character content of this `char` element, and also has a `hull` element, the content of which is identical to the character content of this `char` element. The rest shall be the same as in Case 4.

— Case 2: the `char` element has a `kernel` element but does not have a `hull` element.

  — in(*x*, `<char> ... </char>`) when *x* matches the regular expression specified as the content of the `kernel` element.

  — not-in(*x*, `<char> ... </char>`) never holds.

  — unknown(*x*, `<char> ... </char>`) when *x* does not match the regular expression specified as the content of the `kernel` element.

— Case 3: the `char` element has a `hull` element but does not have a `kernel` element.

  — in(*x*, `<char> ... </char>`) never holds.

  — not-in(*x*, `<char> ... </char>`) when *x* does not match the regular expression specified as the content of the `hull` element.

  — unknown(*x*, `<char> ... </char>`) when *x* matches the regular expression specified as the content of the `hull` element.

— Case 4: the `char` element has a `hull` element and a `kernel` element.

  — in(*x*, `<char> ... </char>`) when *x* matches the regular expression specified as the content of the `kernel` element.

  — not-in(*x*, `<char> ... </char>`) when *x* does not match the regular expression specified as the content of the `kernel` element and *x* does not match the regular expression specified as the content of the `hull` element.

  — unknown(*x*, `<char> ... </char>`) when *x* does not match the regular expression specified as the content of the `kernel` element and *x* matches the regular expression specified as the content of the `hull` element.

    NOTE 1 It is possible but not a good practice to specify a hull that disallows some code point or code point sequence in the corresponding kernel. Note that the condition that a code point or code point sequence is in a repertoire does not mention the hull.

Since the semantics of regular expressions depend on the version of the Unicode standard, the author of a CREPDL schema may specify the intended versions by specifying the `minUcsVersion` and `maxUcsVersion` attributes.

EXAMPLE `<char minUcsVersion="4.0" maxUcsVersion="4.0">\p{Nd}</char>` represents the set of characters of the category "Nd" in Unicode Version 4.0.

NOTE 2 It is not guaranteed that every version between these two attribute values specify the same properties for every character. However, the CREPDL schema author is assumed to accept the discrepancies.

If the CREPDL processor cannot use some version between these two attribute values, it should report an error and may stop normal processing.

When a `char` element does not explicitly specify the `minUcsVersion` attribute, the nearest ancestor element having this attribute is searched. If it is found, its attribute value is used. If not found, there is

no lower bound on Unicode versions. When a `char` element does not explicitly specify the `maxUcsVersion` attribute, the nearest ancestor element having this attribute is searched. If it is found, its attribute value is used. If it is not found, there is no upper bound on Unicode versions.

7.3 union

First, define the semantics of union elements `<union>`*A B*`</union>`, which contain two child elements *A* and *B*. A code point or code point sequence shall be in the union repertoire described by this element if and only if it is in the one described by *A* or the one described by *B*. It shall not be in the union repertoire if and only if it is in neither the one described by *A* nor the one described by *B*.

— in(*x*, `<union>` *A B*`</union>`) when in(*x*, *A*) or in(*x*, *B*).

— not-in(*x*, `<union>` *A B*`</union>`) when not-in(*x*, *A*) and not-in(*x*, *B*).

— unknown(*x*, `<union>` *A B*`</union>`), otherwise.

When a `union` element has one and only one child element, the semantics shall be the same as that of the child element. When a `union` element has more than two child elements, the semantics shall be the same as that of `<union>`*A B*`</union>` where *A* is the first child and *B* is a `union` element containing the other child elements.

7.4 intersection

First, define the semantics of intersection elements `<intersection>`*A B*`</intersection>`, which contain two child elements *A* and *B*. A code point or code point sequence shall be in the repertoire described by this intersection element if and only if it is in the one described by *A* and it is in the one described by *B*. It shall not be in this intersection repertoire if and only if it is not in the one described by *A* or it is not in the one described by *B*.

— in(*x*, `<intersection>` *A B*`</intersection>`) when in(*x*, *A*) and in(*x*, *B*).

— not-in(*x*, `<intersection>` *A B*`</intersection>`) when not-in(*x*, *A*) or not-in(*x*, *B*)

— unknown(*x*, `<intersection>` *A B*`</intersection>`), otherwise.

When an `intersection` element has one and only one child element, the semantics shall be the same as that of the child element. When an `intersection` element has more than two child elements, the semantics shall be the same as that of `<intersection>`*A B*`</intersection>` where *A* is the first child and *B* is an `intersection` element containing the other child elements.

7.5 difference

First, define the semantics of difference elements `<difference>`*A B*`</difference>`, which contain two child elements *A* and *B*. A code point or code point sequence shall be in the repertoire described by this difference element if and only if it is in the one described by *A* and it is not in the one described by *B*. It shall not be in this difference repertoire if and only if either it is not in the one described by *A* or it is in the one described by *B*.

— in(*x*, `<difference>` *A B*`</difference>`) when in(*x*, *A*) and not-in(*x*, *B*)

— not-in(*x*, `<difference>` *A B*`</difference>`) when not-in(*x*, *A*) or in(*x*, *B*)

— unknown(*x*, `<difference>` *A B*`</difference>`), otherwise.

When a `difference` element has one and only one child element, the semantics shall be the same as that of the child element. When a `difference` element has more than two child elements, the semantics shall be the same as that of `<difference>`*A B*`</difference>` where *A* is the first child and *B* is a `union` element containing the other child elements.

7.6 ref

Define the semantics of `<ref href="`*iri*`"/>`, where *iri* is an IRI which shall be as specified in IETF RFC 3987. First, a CREPDL schema *S* shall be obtained by dereferencing *iri*. When dereferencing *iri* is not successful (e.g., network errors), the CREPDL processor should report an error, and it may stop normal processing or it may continue normal processing by assuming that "unknown" holds. When dereferencing is successful but recursive dereferencing results in an infinite loop, the schema shall be incorrect. Otherwise, the semantics shall be defined below:

— in(*x*, `<ref href="` *iri*`"/>`) when in(*x*, *S*).

— not-in(*x*, `<ref href="` *iri*`"/>`) when not-in(*x*, *S*).

— unknown(*x*, `<ref href="` *iri*`"/>`) when unknown(*x*, *S*).

7.7 repertoire

The `repertoire` element shall reference a repertoire in some registry. The `registry` attribute shall specify a registry. The `name` and `number` attributes shall specify a repertoire by name and number, respectively.

— When the value of the `registry` attribute is `"10646"`, a collection specified in ISO/IEC 10646 shall be referenced.

— When the value of the `registry` attribute is `"CLDR"`, a locale in the CLDR registry shall be referenced. When the `version` attribute is present, its value (`"32.0.1"`, for example) shall specify the version of the CLDR registry. The `name` attribute shall specify the name of the locale. The `number` attribute shall be ignored. The examplar characters of the specified locale shall be in the repertoire. The other characters shall not be in the repertoire.

— When the value of the `registry` attribute is `"IANA"`, a charset in the IANA registry of charsets (IANA Charsets) shall be referenced. The `name` attribute shall specify a name or alias, while the `number` attribute shall specify an MIBenum.

— When the value of the `registry` attribute is `"IVD"`, an Ideographic Variation Collection in the Unicode Ideographic Variation Database (UTS37) shall be referenced. When the `version` attribute is present, its value (`"2016-08-15"`, for example) shall specify the version of the Unicode Ideographic Variation Database. The `name` attribute shall specify the identifier of the Ideographic Variation Collection. The `number` attribute is ignored. Note that the first edition did not define the semantics of references to the Unicode Idegraphic Varaition Database.

— Otherwise, this document does not define the semantics.

The CREPDL processor is not required to recognise repertoires specified by `repertoire` elements. However, when the CREPDL processor does not recognise the specified repertoire, it should report an error. It may continue normal processing by assuming that "unknown" holds or it may stop normal processing.

NOTE Even when the repertoire specified by a `repertoire` element is recognised, different CREPDL processors can report different results, because the registry changes from time to time, also because the registry can be imprecise, and also because the semantics of regular expressions depend on the version of the Unicode standard.

8 Validation

When a CREPDL schema is incorrect, the CREPDL processor should report errors and halt.

The input stream to the CREPDL processor shall contain code points. It may contain non-BMP code points but shall not contain high- or low-surrogate code points.

The CREPDL processor shall examine the `mode` attribute of the root element of the CREPDL schema.

If the value of the attribute is `"character"` or the attribute is absent, the CREPDL processor shall validate each code point in the input stream against the given CREPDL schema. If the result is either "not-in" or "unknown" for some code point, the CREPDL processor should report the result.

If the value of the attribute is `"graphemeCluster"`, it shall repeatedly extract a grapheme cluster by applying the algorithm which shall be as specified in UAX29 and validate each grapheme cluster against the given CREPDL schema. If the result is either "not-in" or "unknown" for some grapheme cluster, the CREPDL processor should report the result.

Annex A
(informative)

Differences of conformant processors

Different conformant CREPDL processors can report different results only in the cases shown below:

— Case 1: Dereferencing IRIs can fail. However, the semantics of CREPDL is defined so that such failures make conformant CREPDL processors err on the safe side. In other words, such failures do not lead to "in" when "not-in" or "unknown" would have been reported, and do not lead to "not-in" when "in" or "unknown" would have been reported.

— Case 2: The semantics of regular expressions depends on the Unicode version, since different Unicode versions can assign different Unicode properties to characters. Different conformant CREPDL processors can thus behave differently. For example, one can report "in", while another, "not-in".

— Case 3: A repertoire specified by a `repertoire` element can be unrecognised by the CREPDL processor. Moreover, even when the repertoire is recognised, different CREPDL processors can have different interpretations of the repertoire.

— Case 4: Before CREPDL processors receive CREPDL schemas as well as characters or strings, character normalization (UAX #15[7]) can be applied. Such character normalization can cause two conformant CREPDL processors to behave differently.

Annex B
(informative)

Example CREPDL schemas

B.1 ISO/IEC 8859-6

The repertoire of ISO/IEC 8859-6[1] is described by the following CREPDL schema.

```
<union xmlns="http://purl.oclc.org/dsdl/crepdl/ns/structure/2.0">
  <char>\p{IsBasicLatin}</char>
  <char
>[ &#xA4;&#xAD;&#x60C;&#x61B;&#x61F;&#x621;-&#x63A;&#x640;-&#x652;]</char>
</union>
```

NOTE `\p{IsBasicLatin}` is a block escape as defined in UTS35.

An alternative schema is shown below.

```
<union xmlns="http://purl.oclc.org/dsdl/crepdl/ns/structure/2.0">
  <char>\p{IsBasicLatin}</char>
  <char> </char>
  <char>&#xA4;</char>
  <char>&#xAD;</char>
  <char>&#x60C;</char>
  <char>&#x61B;</char>
  <char>&#x61F;</char>
  <char>[&#x621;-&#x63A;]</char>
  <char>[&#x640;-&#x652;]</char>
</union>
```

Yet another alternative is to rely on the IANA registry.

```
<repertoire xmlns="http://purl.oclc.org/dsdl/crepdl/ns/structure/2.0"
  registry="IANA" name="ISO_8859-6:1987" />
```

B.2 ISO/IEC 8859-15

The repertoire of ISO/IEC 8859-15[2] is described by the following CREPDL schema.

```
<union xmlns="http://purl.oclc.org/dsdl/crepdl/ns/structure/2.0">
  <char>\p{IsBasicLatin}</char>
  <char>[ -&#xA3;]</char>
  <char>&#xA5;</char>
  <char>&#xA7;</char>
  <char>[&#xA9;-&#xB3;]</char>
  <char>[&#xB5;-&#xB7;]</char>
  <char>[&#xB9;-&#xBB;]</char>
  <char>[&#xBF;-&#xFF;]</char>
  <char>[&#x152;-&#x153;]</char>
  <char>[&#x160;-&#x161;]</char>
  <char>&#x178;</char>
  <char>[&#x17D;-&#x17E;]</char>
  <char>&#x20AC;</char>
</union>
```

An alternative is to rely on the IANA registry. The value of the attribute `"number"` is 111, which is the MIBenum of the charset ISO/IEC 8859-15.

```
<repertoire xmlns="http://purl.oclc.org/dsdl/crepdl/ns/structure/2.0"
  registry="IANA" number="111" />
```