

Contents


Page

Foreword.....	iv
Introduction.....	v
1 Scope.....	1
2 Normative references.....	1
3 Terms and definitions.....	1
4 Notation.....	1
4.1 XPath.....	1
4.2 Predicate Logic.....	1
4.3 Syntax.....	2
4.3.1 Namespace and Whitespace.....	2
4.3.2 active element.....	2
4.3.3 assert element.....	2
4.3.4 dir element.....	2
4.3.5 diagnostics element.....	3
4.3.6 diagnostic element.....	3
4.3.7 extends element.....	3
4.3.8 key element.....	3
4.3.9 let element.....	3
4.3.10 library element.....	3
4.3.11 name element.....	3
4.3.12 ns element.....	3
4.3.13 p element.....	3
4.3.14 param element.....	3
4.3.15 pattern element.....	3
4.3.16 phase element.....	3
4.3.17 report element.....	3
4.3.18 rule element.....	3
4.3.19 schema element.....	3
4.3.20 span element.....	4
4.3.21 title element.....	4
4.3.22 value-of element.....	4
4.4 Semantics.....	4
4.4.1 Validation Function.....	4
4.4.2 Simple Syntax.....	4
4.5 Schema Semantics.....	5
4.6 Query Language Binding.....	5
4.7 Order and side-effects.....	6
5 Conformance.....	6
5.1 Full Conformance.....	6
5.2 Simple Conformance.....	7
Annex A (normative) RELAX NG schema for Schematron.....	8
Annex B (normative) Schematron schema for Schematron.....	9
Annex C (normative) Schematron Query Language Binding for XSLT.....	10
Bibliography.....	11

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

 ISO/IEC 19757-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information Technology*, Subcommittee SC 34, Document Description and Processing Languages.

- *Part 0: Overview*
- *Part 1: Interoperability framework*
- *Part 2: Grammar-based validation — RELAX NG*
- *Part 3: Rule-based validation — Schematron*
- *Part 4: Selection of validation candidates*
- *Part 5: Datatypes*
- *Part 6: Path-based integrity constraints*
- *Part 7: Character repertoire validation*
- *Part 8: Declarative document manipulation*
- *Part 9: Datatype- and namespace-aware DTDs*

Introduction

The structure of this part of ISO/IEC 19757 is as follows. 4.3 describes the syntax of a Schematron schema. 4.4 describes the semantics of a correct Schematron schema; the semantics specify when a document is valid with respect to a Schematron schema. Finally, Clause 5 describes conformance requirements for Schematron validators.

Normative annexes provide RELAX NG and Schematron schemas for Schematron, and the default query language binding to XSLT.

This part of ISO/IEC 19757 is based on the Schematron^[1].





Document Schema Definition Languages (DSDL) — Part 3: Rule-based validation — Schematron

1 Scope

This part of ISO/IEC 19757 specifies Schematron, a schema language for XML.

Considered theoretically, a Schematron schema reduces to a non-chaining rule system whose terms are boolean functions invoking an external query language on the instance and other visible XML documents, with syntactic features to reduce specification size and to allow efficient implementation.

Considered as a document type, a Schematron schema contains natural language assertions concerning an instance, marked up with various elements and attributes for testing these assertions, and for simplifying and grouping the assertions.

Considered analytically, Schematron has two characteristic high-level abstractions: the pattern and the phase. These allow the representation of non-regular, non-sequential constraints that Part 2 cannot specify, and various dynamic or contingent constraints.

This part of ISO/IEC 19757 establishes requirements for Schematron schemas and specifies when an XML document matches the patterns specified by a Schematron schema.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 19757. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 19757 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

The following referenced documents are indispensable for the application of this part of ISO/IEC 19757. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

NOTE Each of the following documents has a unique identifier that is used to cite the document in the text. The unique identifier consists of the part of the reference up to the first comma.

NOTE The definitions of Part 1 and Part 2 also apply to this part of ISO/IEC 19757.

W3C XPath, *XPATH???*, W3C Recommendation, <http://www.w3.org/TR/???>

W3C XSLT, *XSLT??*, W3C Recommendation, <http://www.w3.org/TR/???>

3 Terms and definitions

4 Notation

4.1 XPath

This part of ISO/IEC 19757 uses W3C XPath to express the names of information items in the schema. In this part of ISO/IEC 19757 the prefix `sch` is bound to the Schematron namespace URI.

4.2 Predicate Logic

This part of ISO/IEC 19757 uses predicate logic to express the semantics of Schematron schema. The following functions are defined:

- \in is member of, an infix relation, used in the set-operation sense.

NOTE The more familiar term "is element of" is not used, to avoid confusion with XML elements. Where y is an element in a simplified schema, $x \in y$ is defined here as the `child:x` path from context of subject y as defined by W3C XPath. Where y is the instance being validated, $x \in y$ is defined here as all the subjects (information items) in the instance that can be accessed by the query language, as specified in the query language binding. Where y is the name of the active active-phase, in which case one of the following is true:

- $x \in y$ is defined here as the path `//sch:pattern` when y has the special value `#ALL`
 - $x \in y$ is defined here as the path `//sch:pattern[@id=/sch:schema/@default-phase]` when the y has the special value `#DEFAULT`
 - Otherwise $x \in y$ is defined here as the path `../sch:pattern[@id=//sch:phase[id="y"]/active/@pattern]` where y is a name.
-
- `position(r)` the XPath function `position()` of a rule r in its parent pattern
 - `match (r, s, d)` a function returning boolean provided by the query language binding: it returns true iff the subject s from the document d matches the context expression of rule r
 - `assert (a, s, d)` a function returning boolean provided by the query language binding: it returns true iff the assertion a is true when applied to the subject s from the document d

4.3 Syntax

4.3.1 Namespace and Whitespace

All elements shown in the grammar are qualified with the namespace URI:

`http://www.ascc.net/xml/schematron`

Any element can also have foreign attributes in addition to the attributes shown in the grammar. A foreign attribute is an attribute with a name whose namespace URI is neither the empty string nor the Schematron namespace URI. Any non-empty element may have foreign child elements in addition to the child elements shown in the grammar. A foreign element is an element with a name whose namespace URI is not the Schematron namespace URI. There are no constraints on the relative position of foreign child elements with respect to other child elements.

Any element can also have as children strings that consist entirely of whitespace characters, where a whitespace character is one of U+0020, U+009, U+00D or U+00A. There are no constraints on the relative position of whitespace string children with respect to child elements.

Leading and trailing whitespace is allowed for the value of any attribute, and shall be stripped.

4.3.2 active element

empty

4.3.3 assert element

empty

4.3.4 dir element

empty

4.3.5 diagnostics element

empty

4.3.6 diagnostic element

empty

4.3.7 extends element

empty

4.3.8 key element

empty

4.3.9 let element

empty

4.3.10 library element

empty

4.3.11 name element

empty

4.3.12 ns element

empty

4.3.13 p element

empty

4.3.14 param element

empty

4.3.15 pattern element

empty

4.3.16 phase element

empty

4.3.17 report element

empty

4.3.18 rule element

empty

4.3.19 schema element

empty

4.3.20 span element

empty

4.3.21 title element

empty

4.3.22 value-of element

empty



4.4 Semantics

4.4.1 Validation Function

A Schematron validator is a function returning "valid", "invalid" or "error". The function performs notationally performs two steps: transforming the schema into a simple syntax, then testing the instance against the simple syntax.

NOTE This part of ISO/IEC 19757 does not constrain other information provided by an implementation nor other uses of Schematron schemas. However, it is the intent of this part of ISO/IEC 19757 to support implementations to provide rich, specific diagnostics customized with values that assist in detecting and rectifying problems.

A Schematron validator is a function over the following:

- a query language binding
- a schema document
- an instance to be validated
- external instances, if the query language invokes them
- a phase name, or #ALL if all patterns shall be active patterns, or #DEFAULT if the phase attribute on the schema element shall be used
- a list of name-value pairs, if the schema uses external variables.

4.4.2 Simple Syntax

A Schematron validator shall perform the following transformation steps on the schema, resulting in a schema in the simple syntax:

- Resolve all libraries by element inclusion
- Resolve all abstract patterns by macro substitution
- Resolve all abstract rules in the schema by element inclusion
- Resolve all top-level parameters by text substitution
- Negate all sch:report elements into sch:assert elements
- Remove (ignore) all elements in foreign namespaces.

The resulting simple syntax is also a valid Schematron instance in the full syntax. The simple syntax differs from the complex syntax by not containing the following XPath:

- //sch:library
- //sch:pattern/@abstract="true"
- //sch:rule/@abstract="true"
- /sch:schema/sch:param
- //sch:rule/sch:report

4.5 Schema Semantics

This clause gives the semantics of a good schema that has been transformed into the simple syntax.

A good schema with no use of keys or variables satisfies the following predicate:

```

∃ ( instance, schema, active-phase ),
∀ ( subject, pattern, rule, assertion ) :
  subject ∈ instance,
  subject ∈ schema,
  pattern ∈ active-phase,
  rule ∈ pattern,
  assertion ∈ rule :
  match ( rule, subject, instance )
  ∧ ( ∀ ( previous-rule ) :
    previous-rule ∈ pattern,
    position ( previous-rule ) < position ( rule ) :
      ¬ ( match ( previous-rule, subject, instance ) ) )
  ⇒ assert ( assertion, subject, instance ) = true

```

NOTE In natural language, that is "There exists an instance, schema and active-phase combination where, for each subject, pattern, rule and assertion (the subject being a member of that instance, the pattern being a member of that schema, the pattern being a member of that active-phase, the rule being a member of that pattern, the assertion being a member of that rule), the following is true: if the subject in an instance matches the rule, and that subject has not been matched by a previous rule in the same pattern, then the particular assertion evaluates to true when applied to the particular subject and instance."

4.6 Query Language Binding

A query language binding shall provide the following:

- The general query language used. A name token which identifies the query language. The data model.
- The rule context query language. The rule context scope.
- The assertion test, a function which returns a data value coerceable into boolean.

NOTE The following query language names are reserved and recommended:

- xslt
- exslt
- xslt2
- xpath

- xpath2
- xquery

A Schematron implementation which does not support the specification language shall fail with an error.

A schema language binding may provide the following:

- The name query language, a function which returns a data value coerceable into a string.
- The value-of query language, a function which returns a data value coerceable into a string.
- The key path language.
- The let value query language, a function which returns a data value.
- The variable delimiter convention, a lexical convention such as a delimiter by which the use of a variable in a query expression shall be recognized.
- The abstract pattern parameter convention, a lexical convention such as a delimiter by which the parameters of abstract patterns inside query expressions shall be recognized.

A Schematron implementation which does not support the specification language shall fail with an error.

4.7 Order and side-effects

The order in which elements are validated is implementation-dependent, without altering the validity of the instance.

The order in which patterns are used is implementation-dependent, without altering the validity of the instance

The order in which elaborated rule-contexts are matched is implementation-dependent, without altering the validity of the instance

The order in which assertions are tested is implementation-dependent, without altering the validity of the instance.

The only elements for which order is significant are the sch:rule and sch:let elements.

An sch:rule element acts as an if-then-else chain within each pattern. An implementation may make order non-significant by converting rules contexts to elaborated rule contexts. An elaborated rule context consists the negated union of all the lexically previous rule contexts in the same pattern intersected with the current rule context.

An sch:let element may use lexically previous variables within the same rule or global variables.

NOTE A wide variety of implementation strategies are therefore possible.

All queries shall act as pure functions. Queries shall not alter the instance in any way visible to other queries. This part of ISO/IEC 19757 does not specify any outcome augmentation of the instance being validated.

The only element which has a side-effect is key, which may provide extra index information for other queries.

5 Conformance

5.1 Full Conformance

A full-conformance Schematron validator shall be able to determine for any XML document whether it is a correct schema.

- A correct schema conforms to the constraints of the normative RELAX NG schema of this part of ISO/IEC 19757.
- A correct schema conforms to the constraints of the normative Schematron schema of this part of ISO/IEC 19757.
- A correct schema's attributes conform to the grammars specified by the query language binding in use.

A full-conformance Schematron validator shall be able to determine for any XML document and for any good schema whether the document is valid with respect to the schema.

NOTE It is not a requirement of this part of ISO/IEC 19757 that a full-conformance Schematron validator shall be able to determine whether the validation will terminate or whether the queries are feasible against some other schema for the instance. The ability to determine these depends on the query language used. Where the query language allows incorrectness to be established, implementations are encouraged to report this information as part of validation.

5.2 Simple Conformance

A simple-conformance Schematron validator shall be able to report for any XML document whether it may not be a valid Schematron schema.

- A valid schema conforms to the constraints of the normative RELAX NG schema of this part of ISO/IEC 19757.
- A valid schema conforms to the constraints of the normative Schematron schema of this part of ISO/IEC 19757.
- A valid schema's attributes conform to the grammars specified by the query language binding in use.

A simple-conformance Schematron validator shall be able to determine for any XML document and for any good schema whether the document is valid with respect to the schema.

NOTE It is not a requirement of this part of ISO/IEC 19757 that a simple-conformance Schematron validator shall be able to determine whether validation will terminate or whether the queries are feasible against some other schema for the instance. The ability to determine these depends on the query language used. Where the query language allows incorrectness to be established, implementations are encouraged to report this information as part of validation.

Annex A
(normative)

RELAX NG schema for Schematron

A correct Schematron schema shall be valid with respect to the following RELAX NG schema.

RELAX NG schema for Schematron goes here



Annex B (normative)

Schematron schema for Schematron

A correct Schematron schema shall be valid with respect to the following Schematron schema. This schema does not specify constraints which can be expressed by the RELAX NG schema for Schematron.

Schematron Schema for Schematron goes here



Annex C (normative)

Schematron Query Language Binding for XSLT

A Schematron schema with no language binding or a language binding with the value "xslt" shall use the following binding.

- The query language used is the extended version of W3C XPath specified in W3C XSLT. Consequently, the data model used is the data model of those specifications.
- The rule context is interpreted according to the production XXX of XSLT. The rule context may be elements, attributes, comments and processing instructions.
- The assertion test is interpreted according to production XXX of XSLT.
- The name query is interpreted according to production XXX of XSLT.
- The value-of query is interpreted according to production XXX of XSLT.
- The key path is interpreted according to production XXX of XSLT. A Schematron key is equivalent to an XSLT key.
- The let value is interpreted according to production XXX of XSLT.
- A Schematron let expression is treated as an XSLT variable. The XSLT \$ delimiter signifies the use of a variables in an assertion test, name query or value-of query.
- The notation for signifying abstract pattern is to prefix the token with the **????**. This is a character not found in URLs or XPath.

Bibliography

- [1] *Resource Description for Schematron (web page)*, Rick Jelliffe, Computing Centre, Academia Sinica, Taipei, <http://www.ascc.net/xml/schematron>



Summary of editorial comments:

