II. XML / 1. Unicode, URIs, and XML Syntax

1. Unicode

2. Uniform Resource Identifiers (URIs)

3. XML Syntax
Coded Character Sets

<table>
<thead>
<tr>
<th>name</th>
<th>codes</th>
<th>examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII code</td>
<td>0–127</td>
<td>64 ↦→ A</td>
</tr>
<tr>
<td>ISO-8859-1, ISO-LATIN-1</td>
<td>0–255</td>
<td>0–127 as ASCII, 196 ↦→ α</td>
</tr>
<tr>
<td>ISO-8859-7</td>
<td>0–255</td>
<td>0–127 as ASCII, 225 ↦→ α</td>
</tr>
<tr>
<td>Unicode</td>
<td>0–(2^{32} – 1)</td>
<td>0–255 as ISO-8859-1</td>
</tr>
</tbody>
</table>

Unicode is organized in 256 groups à 256 planes à 256 rows à 256 cells.

Plane 0 (codes 0–65535) is called **basis multilingual plane (BMP)**.

Non ISO-8859-1 characters are mapped to higher codes, e.g., 945 ↦→ α.
Assigned characters of the Unicode standard (v4.1, 03/2005) can be found at http://www.unicode.org/charts/.

Unicode also specifies character classes for each character, as

- letters (capital and small),
- digits,
- punctuation,
- control characters.
The Unicode Character Code Charts

Character Encoding Schemata

Character Encoding Schemata are trivial for 1-byte coded character sets.

Direct representations of Unicode:

**UCS-2:** direct representation of codes 0–65535 with 2 bytes.

**UCS-4:** direct representation of all codes with 4 bytes.
Drawbacks of direct representations:

- bytecode 0x00 occurs (that marks string endings in C), e.g., in UCS-4:

  \[ A \mapsto 65 \mapsto (0, 0, 0, 65) \]

- uniform blow-up of storage space, but most texts mostly use ASCII or ISO-8859-1.

- error-prone, as if one byte is lost, all following data will be decoded incorrectly.

Unicode Transformation Formats (UTF)

Unicode Transformation Formats (UTF) use a variable number of bytes for coding a character.

UTF-8:

- 0x00–0x7f (bit sequences 0....... ) code ASCII characters directly,
- 0xc0–0xfd (bit sequences 11.......) mark the start of a multi-byte character representation (and code its length and leading bits of its code),
- 0x80–0xbf (bit sequences 10....... ) code continuations of multi-byte character representations,
- 0xfe, 0xff (bit sequences 1111111.) are not used.

<table>
<thead>
<tr>
<th>bit sequence</th>
<th>bytes</th>
<th>free bits</th>
<th>character codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0........</td>
<td>1</td>
<td>7</td>
<td>0x00–</td>
</tr>
<tr>
<td>110.....</td>
<td>2</td>
<td>5 + 6 = 11</td>
<td>0x80–</td>
</tr>
<tr>
<td>1110....</td>
<td>3</td>
<td>4 + 2 \cdot 6 = 16</td>
<td>0x800–</td>
</tr>
<tr>
<td>11110....</td>
<td>4</td>
<td>3 + 3 \cdot 6 = 21</td>
<td>0x10000–</td>
</tr>
<tr>
<td>111110..</td>
<td>5</td>
<td>2 + 4 \cdot 6 = 26</td>
<td>0x200000– 0:</td>
</tr>
<tr>
<td>1111110.</td>
<td>6</td>
<td>1 + 5 \cdot 6 = 31</td>
<td>0x4000000– 0x:</td>
</tr>
</tbody>
</table>
II. XML / 1. Unicode, URIs, and XML Syntax

1. Unicode

2. Uniform Resource Identifiers (URIs)

3. XML Syntax

Uniform Resource Identifiers (URIs)

URIs are used to identify resources.

Example:

http://www.informatik.uni-freiburg.de/cgnm/lehre/xml-06s/index_en.html

URIs are defined in RFC 3986 (01/2005).
Hierarchical URIs

An URI is called **hierarchical** iff

\[
\langle \text{scheme-specific-part} \rangle := ( \ \langle \text{authority} \rangle [ \ \langle \text{path} \rangle ]
\]
\[
| \langle \text{path} \rangle \]
\[
\] \[
\langle \text{path} \rangle := ( / \langle \text{path-segment} \rangle )^+ \]

otherwise its called **opaque**.

The path-segments . and . . have special meaning: context path and parent path.

A hierarchical URI is called **server-based** iff

\[
\langle \text{authority} \rangle := [ \langle \text{userinfo} \rangle @ ] \langle \text{host} \rangle [ : \langle \text{port} \rangle ]
\]

otherwise it is called **registry-based**.
Fragment identifiers

Fragment identifiers are used to identify parts of the resource identified by an URI.

Example:

http://www.informatik.uni-freiburg.de/xml/books.html#R03

```html
<html>
<body>
<li><a name="EE04">Rainer Eckstein, Silke Eckstein: XML und Datenmodellierung</a>, 2004.</li>
<li><a name="R03">Erik T. Ray: Learning XML</a>, 2003.</li>
</body>
</html>
```

Figure 7: HTML document at http://www.informatik.uni-freiburg.de/xml/books.html.

Relative (hierarchical) URIs

A relative URI is defined as:

\[
\langle\text{relativeURI}\rangle ::= ( \text{ // } \langle\text{authority}\rangle [ \langle\text{path}\rangle ] \\
| \langle\text{path}\rangle \\
| \langle\text{relativePath}\rangle ) [ ? \langle\text{query}\rangle ]
\]

\[
\langle\text{relativePath}\rangle ::= \langle\text{path-segment}\rangle ( \text{ / } \langle\text{path-segment}\rangle )^*
\]

Figure 8: A Base URI is the context for resolving relative URIs [RFC 2396].
URI schemes are managed by Internet Assigned Numbers Authority (IANA).

<table>
<thead>
<tr>
<th>Scheme Name</th>
<th>Description</th>
<th>Reference</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ftp</td>
<td>File Transfer Protocol</td>
<td>RFC 1738</td>
<td>server-based</td>
</tr>
<tr>
<td>http</td>
<td>Hypertext Transfer Protocol</td>
<td>RFC 2616</td>
<td>server-based</td>
</tr>
<tr>
<td>mailto</td>
<td>Electronic mail address</td>
<td>RFC 2368</td>
<td>server-based</td>
</tr>
<tr>
<td>file</td>
<td>Host-specific file names</td>
<td>RFC 1738</td>
<td>server-based</td>
</tr>
<tr>
<td>pop</td>
<td>Post Office Protocol v3</td>
<td>RFC 2384</td>
<td>server-based</td>
</tr>
<tr>
<td>dav</td>
<td>dave</td>
<td>RFC 2518</td>
<td>server-based</td>
</tr>
<tr>
<td>tel</td>
<td>telephone</td>
<td>RFC 2806</td>
<td>opaque</td>
</tr>
<tr>
<td>https</td>
<td>Hypertext Transfer Protocol Secure</td>
<td>RFC 2818</td>
<td>server-based</td>
</tr>
<tr>
<td>urn</td>
<td>Uniform Resource Names</td>
<td>RFC 2141</td>
<td>opaque</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

50 URI schemes (as of 2006-04-27; http://www.iana.org/assignments/uri-schemes.html).

URI registrations are regulated by RFC 4396 (2/2006).

Example:

tel:+(49)-761-203-8164
Uniform Resource Names (URNs)

URNs are special kinds of URIs that

- **map other namespaces** into URN-space,

- are required to remain **globally unique and persistent**
  (even when the resource ceases to exist or becomes unavailable).

- have scheme `urn`.

\[
\langle URN \rangle := \text{urn:} \langle \text{namespace} \rangle : \langle \text{namespace-specific-part} \rangle
\]

Examples:

```
urn:isbn:0-395-36341-1
urn:newsml:reuters.com:20000206:IIMFFH05643_2000-02-06_17-54-01_L0615
```

A book or a news item (identified by an URN) may be retrieved from different locations (URLs).

### URN Namespaces

<table>
<thead>
<tr>
<th>URN Namespaces</th>
<th>Value</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>ietf</td>
<td>1</td>
<td>RFC 2648</td>
</tr>
<tr>
<td>pin</td>
<td>2</td>
<td>RFC 3043</td>
</tr>
<tr>
<td>issn</td>
<td>3</td>
<td>RFC 3044</td>
</tr>
<tr>
<td>oid</td>
<td>4</td>
<td>RFC 3061</td>
</tr>
<tr>
<td>newsml</td>
<td>5</td>
<td>RFC 3085</td>
</tr>
<tr>
<td>oasis</td>
<td>6</td>
<td>RFC 3121</td>
</tr>
<tr>
<td>xmlorg</td>
<td>7</td>
<td>RFC 3120</td>
</tr>
<tr>
<td>publicid</td>
<td>8</td>
<td>RFC 3151</td>
</tr>
<tr>
<td>isbn</td>
<td>9</td>
<td>RFC 3187</td>
</tr>
<tr>
<td>nbn</td>
<td>10</td>
<td>RFC 3188</td>
</tr>
<tr>
<td>web3d</td>
<td>11</td>
<td>RFC 3541</td>
</tr>
<tr>
<td>mpeg</td>
<td>12</td>
<td>RFC 3614</td>
</tr>
<tr>
<td>mace</td>
<td>13</td>
<td>RFC 3613</td>
</tr>
<tr>
<td>fipa</td>
<td>14</td>
<td>RFC 3616</td>
</tr>
<tr>
<td>swift</td>
<td>15</td>
<td>RFC 3615</td>
</tr>
</tbody>
</table>

25 URN namespaces (as of 2006-04-27; http://www.iana.org/assignments/urn-namespaces)
In URIs only some characters may be used literally in non-syntactic parts ("data"). All others have to be escaped using their code (in some character encoding):

\[
\begin{align*}
\langle \text{dataChars} \rangle & : = \langle \text{alphanum} \rangle \mid - \mid . \mid ! \mid \sim \mid * \mid ' \mid ( \mid ) \\
\langle \text{escapedChar} \rangle & : = \% \langle \text{hexDigit} \rangle \langle \text{hexDigit} \rangle
\end{align*}
\]

Codes have been interpreted as codes in different character encodings, depending on the URI scheme.

UTF-8 is recommended by RFC 2718 and already used by some schemes (e.g., urn, imap, pop).

Example:

http://www.informatik.uni-freiburg.de/login.jsp?name=Hans%20Meyer

Internationalized Resource Identifiers (IRIs)

IRIs allow more characters to be used literally (RFC 3987; 01/2005).

In IRIs only

- data characters that can be misinterpreted as syntactic characters and
- some bidirectional formatting characters

have to be escaped.

All other data characters are used literally (in some character encoding, e.g., UTF-8).

Example:

http://www.informatik.uni-freiburg.de/login.jsp?name=Hans%20Müller

Schemes still are restricted to US ASCII characters.
II. XML / 1. Unicode, URIs, and XML Syntax

1. Unicode

2. Uniform Resource Identifiers (URIs)

3. XML Syntax

W3C development process

W3C specifications are called **Recommendations**.

Stages of W3C recommendations:

<table>
<thead>
<tr>
<th>stage</th>
<th>completion date XML 1.0</th>
<th>completion date XML 1.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Draft</td>
<td>1996/11/14</td>
<td>2001/12/13</td>
</tr>
<tr>
<td>Last Call Working Draft</td>
<td>1997/11/17</td>
<td>2002/04/25</td>
</tr>
<tr>
<td>Candidate Recommendation</td>
<td></td>
<td>2002/10/15</td>
</tr>
<tr>
<td>Proposed Recommendation</td>
<td>1997/12/08</td>
<td>2003/11/05</td>
</tr>
<tr>
<td>Recommendation</td>
<td>1998/02/10</td>
<td>2004/04/15</td>
</tr>
<tr>
<td>Working Draft (2nd edition)</td>
<td>2000/08/14</td>
<td></td>
</tr>
<tr>
<td>Recommend</td>
<td>2000/10/06</td>
<td></td>
</tr>
<tr>
<td>Proposed Edited Recommendation</td>
<td>2003/10/30</td>
<td></td>
</tr>
<tr>
<td>Recommendation (3rd edition)</td>
<td>2004/02/04</td>
<td></td>
</tr>
</tbody>
</table>
Every XML document consists of a **prolog** and a single element, called **root element**.

\[
\text{⟨document⟩ := ⟨prolog⟩ ⟨element⟩ (⟨Comment⟩ | ⟨PI⟩ | ⟨S⟩ )*}
\]

\[
\text{⟨prolog⟩ := <?xml ⟨S⟩ version = "1.1"}
\]
\[
\text{(⟨S⟩ encoding = ⟨encoding⟩ )?}
\]
\[
\text{(⟨S⟩ standalone = ("yes" | "no"))?}
\]
\[
\text{⟨S⟩? ?>}
\]
\[
\text{(⟨Comment⟩ | ⟨PI⟩ | ⟨S⟩ )*}
\]
\[
\text{(⟨DoctypeDecl⟩ (⟨Comment⟩ | ⟨PI⟩ | ⟨S⟩ )* )?}
\]

In all productions

- matching " can be replaced by ‘.
- = may be surrounded by spaces (i.e., match ⟨S⟩?= ⟨S⟩?).

\[
\text{⟨S⟩ := (#x20 | #x9 | #xD | #xA)+}
\]

In XML 1.1 the version attribute is mandatory.

If the version attribute is missing, version 1.0 is assumed.
Elements and Attributes

\[
\text{⟨element⟩} := \langle\text{emptyElementTag}\rangle \\
\hspace{1cm} | \langle\text{STag}\rangle \langle\text{content}\rangle \langle\text{ETag}\rangle
\]

\[
\langle\text{emptyElementTag}\rangle := <\langle\text{Name}\rangle (\langle\text{S}\rangle \langle\text{Name}\rangle = "\langle\text{AttValue}\rangle " )* \langle\text{S}\rangle ? \rangle
\]

\[
\langle\text{STag}\rangle := <\langle\text{Name}\rangle (\langle\text{S}\rangle \langle\text{Name}\rangle = "\langle\text{AttValue}\rangle " )* \langle\text{S}\rangle ? >
\]

\[
\langle\text{ETag}\rangle := </\langle\text{Name}\rangle \langle\text{S}\rangle ? >
\]

\(\langle\text{Name}\rangle\)s

- start with a unicode letter or _
  ( : is also allowed, but used for namespaces).
- may contain unicode letters, unicode digits, -, ., or ..

A wellformed document requires,

- that start and end tag of each element match,
- that for each tag the same attribute never occurs twice.

---

Not-wellformed Documents (1/2)

```xml
<?xml version="1.1"?>
<book>
  <author><fn>Rainer</fn><sn>Eckstein</sn></author>
  <author><fn>Silke</fn><sn>Eckstein</sn></author>
  <title>XML und Datenmodellierung</title>
  <year>2004</year>
</book>

<book>
  <author><fn>Erik T.</fn><sn>Ray</sn></author>
  <title>Learning XML</title>
  <year edition="2">2003</year>
</book>
```

Figure 11:
The contents of an element can be made up from 6 different things:

1. other elements,
2. Character data,
3. References,
4. CDATA sections,
5. Processing instructions, and
6. comments.

\[
\langle \text{content} \rangle := \langle \text{CharData} \rangle? \\
( ( \langle \text{element} \rangle | \langle \text{Reference} \rangle | \langle \text{CDSect} \rangle | \langle \text{PI} \rangle | \langle \text{Comment} \rangle ) \\
\langle \text{CharData} \rangle? )^* 
\]
Character data

<CharacterData> may contain any characters except

    <, &, or the sequence > ] ]

Attribute values may not contain

    • "", if delimited by ",
    • ’’, if delimited by ’’,

These characters can be expressed by references.

<?xml version="1.1"?>
<abstract>
    x^2 = y has no real solution for y < 0.
    But there are solutions for y = 0 & for y > 0.
</abstract>

Figure 14: Forbidden characters in character data.

<?xml version="1.1"?>
<abstract>
    x^2 = y has no real solution for y &lt; 0.
    But there are solutions for y = 0 &amp; for y &gt; 0.
</abstract>

Figure 15: Using references in character data.
There are five predefined entity references:

<table>
<thead>
<tr>
<th>Character</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td>&lt;</td>
</tr>
<tr>
<td>&gt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>&amp;</td>
<td>&amp;</td>
</tr>
<tr>
<td>'</td>
<td>'</td>
</tr>
<tr>
<td>&quot;</td>
<td>&quot;</td>
</tr>
</tbody>
</table>

All other entities known from HTML (as &uuml;) are not predefined in XML.

Custom entities can be defined in the document type declaration.

CDATA sections allow the literal usage of all characters (except the sequence ]]>).

```xml
(CDSect) := <![CDATA[ ⟨CDATA] ]]>]
```

CDATA sections are typically used for longer text containing < or &.

CDATA sections are flat, i.e., there is no possibility to structure them with elements (as < or & are interpreted literally).
Character data and CDATA sections

<?xml version="1.1"?>
<abstract>
  \[ x^2 = y \] has no real solution for \( y \lt 0. \)
  But there are solutions for \( y = 0 \) and for \( y \gt 0. \)
</abstract>

Figure 16: Using numeric character references.

<?xml version="1.1"?>
<abstract><![CDATA[
  \[ x^2 = y \] has no real solution for \( y \lt 0. \)
  But there are solutions for \( y = 0 \) and for \( y \gt 0. \)
]]></abstract>

Figure 17: Using a CDATA-section.

<?xml version="1.1"?>
<book abstract="Discusses meaning of "wellformed"">
  <author>John Doe</author>
  <title>About wellformedness</title>
</book>

Figure 18: Literal usage of attribute delimiter.

<?xml version="1.1"?>
<book abstract='Discusses meaning of "wellformed"'>
  <author>John Doe</author>
  <title>About wellformedness</title>
</book>

Figure 19: Using different attribute delimiters.

<?xml version="1.1"?>
<book abstract="Discusses meaning of &quot;wellformed&quot;">
  <author>John Doe</author>
  <title>About wellformedness</title>
</book>

Figure 20: Using references in attribute values.
Comments can occur in the prolog and in the contents of elements.

Comments are not allowed to contain the character sequence `--`. 

\[
\langle \text{Comment} \rangle := <! -- (\text{Char})^* -->
\]

1. `<?xml version="1.1"?>`
2. `<!-- list is not complete yet ! -->`
3. `<books>`
4. `<!-- yet to be ordered -->`
5. `<book>`
6. `<author><fn>Rainer</fn><sn>Eckstein</sn></author>`
7. `<author><fn>Silke</fn><sn>Eckstein</sn></author>`
8. `<title>XML und Datenmodellierung</title>`
9. `<year> <!-- look up year of publication --></year>`
10. `</book>`
11. `</books>`

Figure 21: Comments in the prolog and in the contents of elements.
Processing instructions (PIs) allow documents to contain instructions for applications.

\[
\langle \text{PI} \rangle := \langle? \langle\text{Name}\rangle (\langle\text{S}\rangle \langle\text{Char}\rangle^* )? \rangle ?
\]

The name of a PI must be different from \text{xml}.
Character encoding schemata are specified by the name they are registered with at IANA (http://www.iana.org/assignments/character-sets), e.g.,

US-ASCII
ISO-8859-1
ISO-10646-UCS-2 or csUnicode (UCS2)
ISO-10646-UCS-4 or csUCS4 (UCS4)
UTF-8
UTF-16
...

If no encoding is specified in the XML declaration, UTF-8 is assumed.

<?xml version="1.1" encoding="ISO-8859-1" ?>
<page>
  Grüß Gott !
</page>

Figure 25: XML document coded in ISO-8859-1.
There are two predefined attributes,

- **xml:lang**

  and

- **xml:space**, that can be used with any element.

**xml:lang** specifies the language of the character contents of elements and attributes with (RFC 3066)

- an ISO language code
  
  (http://www.loc.gov/standards/iso639-2/langcodes.html)

  or

- an IANA language code
  
  (http://www.iana.org/assignments/language-tags).

---

**Language Attribute**

Example ISO and IANA language codes:

<table>
<thead>
<tr>
<th>language code</th>
<th>meaning</th>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td>de</td>
<td>ISO</td>
<td>German</td>
</tr>
<tr>
<td>de-CH</td>
<td>ISO</td>
<td>German, Swiss variant</td>
</tr>
<tr>
<td>de-DE</td>
<td>ISO</td>
<td>German, German variant</td>
</tr>
<tr>
<td>en</td>
<td>ISO</td>
<td>English</td>
</tr>
<tr>
<td>en-US</td>
<td>ISO</td>
<td>US English</td>
</tr>
<tr>
<td>en-GB</td>
<td>ISO</td>
<td>Britain English</td>
</tr>
<tr>
<td>tlh</td>
<td>ISO</td>
<td>Klingon</td>
</tr>
<tr>
<td>de-1901</td>
<td>IANA</td>
<td>German, traditional orthography</td>
</tr>
<tr>
<td>de-1996</td>
<td>IANA</td>
<td>German, orthography of 1996</td>
</tr>
</tbody>
</table>
<?xml version="1.1"?>
<page>
<p xml:lang="de">Guten <s>Morgen</s>!</p>
<p xml:lang="en">Good <s>morning</s>!</p>
<table>
<tr><td>USD</td><td>0</td><td>1</td><td>...</td></tr>
<tr><td>EUR</td><td>0</td><td>0.839818</td><td>...</td></tr>
</table>
</page>

Figure 26: Language attribute.

Exercises and tutorials

- There will be a bi-weekly sheet with two to three exercises handed out each second Thursday in the tutorial. 1st sheet will be handed out this Thur. 4.5.

- Solutions to the exercises can be submitted until every next Thursday before the lecture 1st sheet is due Thur. 11.5. 11am

- Mode of corrections is still to be decided on until next Tuesday.

- Tutorials each other second Thursday 11-13 instead of the lecture 1st tutorial at Thur. 4.5.