XML – Merkblatt 5 – XPath

Einführung XPath ist ein XML-Standard um Daten im XML-Baum abzufragen oder innerhalb der zugehörigen Baumrepräsentation zu navigieren. Mit einem XPath können die Knoten ähnlich abgefragt werden, wie Dateien und Verzeichnisse mit einem Dateipfad, etwa `/usr/home/pape/*`. XPath wird in anderen XML-Standards wie XSLT, XPointer oder XQuery verwendet. Web-Entwicklungstools besitzen oft auch eine Suchmöglichkeit von HTML- oder XML-Dokumenten mit XPath.

XPath Datenmodell Jeder Knoten im XML-Baum eines XML-Dokuments hat einen Typ, eine möglichen Namen und einen Wert. Es gibt sieben verschiedene Typen: Wurzelknoten, Element, Attribut, Text, Verarbeitungsanweisung, Kommentar und Namensraum. Textknoten stellen die Blätter im XML-Baum dar. Analog zu Dateibäumen wird der Wurzelknoten mit `/` bezeichnet.

```
<?xml version="1.0"?>
<movies>
  <movie lang="en">
    <title>Die Hard</title>
    <rating>3</rating>
  </movie>
  <movie>
    <title>Terminator</title>
    <rating>2</rating>
  </movie>
</movies>
```

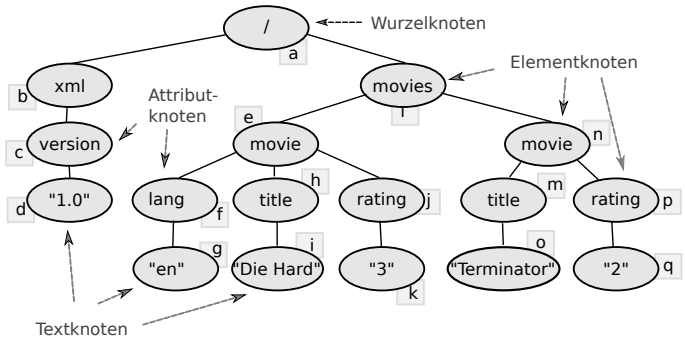

Abbildung 1: Links ein XML-Dokument. Rechts der zugehörige XML-Baum mit den zugehörigen Typen. Die Kleinbuchstaben sind lediglich als zusätzliche Information zur Identifikation und Ordnung der Knoten im XML-Baum angegeben.

Syntax und Semantik Ein XPath-Ausdruck beschreibt Pfade im XML-Baum mit Hilfe eines *Positionspfads* (location path). Ein Positionspfad besteht aus eine Folge ein oder mehrerer *Positionsschritt* (location steps), die durch `/` getrennt sind. Ein Positionsschritt besteht aus drei Teilen und hat die Form `axis::node-test[..]..[..]`:

1. Einer optionalen *Axis*, gefolgt von zwei Doppelpunkten.
2. Einem *Knotentest* (node test). Elemente werden durch ihren Namen und die Textknoten durch `text()` identifiziert.
3. Kein, ein oder beliebig viele Prädikate, die aus Booleschen-Ausdrücken bestehen.

Wir betrachten im folgenden nur den obligatorischen Knotentest.

`/movies/movie/title` ist ein *absoluter* Positionspfad. `title/text()` ist *relativer* Positionspfad.

Das Ergebnis eines Positionsschritts und seiner Einzelteile ist immer eine Menge von Knoten (node-set) *ohne* Duplikate. XPath-Ausdrücke wie ein Positionspfad werden immer in einem Kontext ausgewertet. Dieser wird durch übergeordnete Standards wie zum Beispiel XSLT definiert. Der *Kontext* besteht unter anderem aus

- einem Knoten im XML-Baum, dem *Kontextknoten* (context node) sowie
- einem Paar von positiven ganzen Zahlen, der *Kontextposition* (context position) und der *Kontextgröße* (context size).

Wenn man eine Knotenmenge nach der Reihenfolge der Knoten im XML-Dokument sortiert, dann ist die Kontextposition, die entsprechende Nummer – beginnend mit Eins – in dieser Sortierung. Die Kontextgrösse ist die Anzahl Elemente der Knotenmenge. Man sollte sich deswegen die Knotenmenge besser als eine geordnete Menge vorstellen.

Es sei `j` der Kontextknoten im XML-Baum 1 aus der Knotenmenge `{n, e, j}`, dann ist die geordnete Menge `{e, j, n}`. Die Kontextposition ist 2 und die Kontextgröße 3.

Bei einem absolutem Positionspfad beginnt die Auswertung mit dem Wurzelknoten, Kontextpositon 1 und Kontextgröße 1.

Die Positionsschritte werden von links ausgehend Schritt-für-Schritt auf den aktuellen Kontext angewendet. Das Resultat ist immer eine Knotenmenge. Der nachfolgende Positionsschritt wird auf *jeden Knoten* dieser Knotenmenge als Kontextknoten angewendet. Das Resultat ist die Vereinigungsmenge der einzelnen Ergebnisse.

Beispielsweise wird `/movies/movie/title/text()` in vier Einzelschritten ausgewertet:

1. `movies` identifiziert alle `movies`-Knoten unterhalb des Kontextknotens `a`. Das Ergebnis ist `{i}`.
2. Der Kontext wandert zu Knoten `i` und der Knotentest `movie` wird darauf angewendet. Der Ergebnis ist jeder `movie`-Elementknoten unterhalb `i`, also die Knotenmenge `{e, n}`.
3. `title` wird erst mit `e` als Kontextknoten (Ergebnis `{h}`) und dann mit `n` (Ergebnis `{m}`) angewendet. Das Resultat ist `{h, m}`.
4. `text()` identifiziert jeden Textknoten unterhalb des Kontextknotens. Das Ergebnis ist `{i, o}`.

Attributknoten können mit `@attribut-name`, alle Elemente mit `*` und alle Attribute mit `@*` selektiert werden.

`/movies/movie/*` selektiert die Knoten `{h, j, m, p}`. `/movies/@*` ist die leere Menge. `/movies/*/@lang` ist `{f}`.

Mit `node()` oder kürzer `.` wird der Kontextknoten referenziert. Mit `..` der Vaterknoten des Kontextknotens. `/movies/movie/*/../node()` hat `{e, n}` zum Ergebnis.

Prädikate Ein Prädikat nutzt einen XPath-Ausdruck, um die selektierten Knoten des vorangehenden Location-steps weiter einzuschränken. Der Kontext für die Auswertung des Prädikats wandert zu jedem Knoten des vorangehenden Location-steps (oder Prädikats). DEr XPath `/movies/movie[rating/text() > 2]` hat alle `movie`-Knoten als Ergebnis deren `rating` eine Zahl enthält, die größer als 2 ist. Das Ergebnis dieses Positionsschritts ist `{e}`. Mit Literalen und den Positionspfad als Operanden werden mit arithmetischen, relationalen sowie booleschen Operatoren und Funktionen weiter Berechnungen durchgeführt.

Das Ergebnis einer Berechnung ist ein Objekt, mit einem der vier folgenden Typen:

- Knotenmenge, wie bei Positionspfaden.
- boolean (true or false).
- number (eine Gleitkommazahl).
- string (eine Folge von Unicodezeichen).

Für Knotenmengen existiert lediglich der Vereinigungsoperator `|`. Für number existieren die binären Operatoren `+`, `-`, `div`, `mod` und `-` als Vorzeichen. Für boolean existieren die binären Operatoren `and` und —`or`. Zu string gibt es keine Operatoren.

Objekte können mit den binären Vergleichsoperatoren `=`, `!=`, `>`, `>=`, `<` und `<=` verglichen werden. Das Ergebnis ist ein boolean-Objekt.

Zur Laufzeit können je nach Kontext automatisch Wert- und Typ-Umwandelungen stattfinden:

- Die leere Knotenmenge, leere Zeichenkette, 0 und NaN (not a number) wird je nach Kontext als false interpretiert, ansonsten als true.
- Strings wie '123' oder '54.5' werden zu Zahlen umgewandelt.
- `true` wird als 1 und `false` als 0 interpretiert.
- boolean, number können in string, etwa 65.5 zu '65.5', umgewandelt werden.
- Bei nicht leerer Knotenmenge wird in der Regel der Knoten mit der Kontextposition 1 zum geforderten Typ konvertiert.

Ausdrücke können mit `(` und `)` geklammert werden.

Der Kontext für die Typumwandlung wird durch die verwendeten Operatoren und Funktionsparameter bestimmt. Die Operanden einer Addition werden zum Beispiel immer in Zahlen umgewandelt.

`0 and ('1' = 1)` ergibt `false` und `'1' + 1` ergibt 2. `movie/rating + 1]` hat die Summe des erste Objekts der linken Knotenliste mit 1 als Resultat.

XPath Funktionen

Funktion / Beispiel	Beschreibung / Ergebnis
`not(boolean)`, `true()` und `false()`	Negation des Booleschen Werts Wahrheitswerte true und false
`/movies/movie[not(((rating mod 2)=0)=true()]`	`{j}`
`position()` `last()`	Position des Kontextknotens Kontextgröße

`/movies/movie[position() = last()]`
`/movies/movie[position() = 2]`
abgekürzt `/movies/movie[2]`
	`{n}`
	
	`{n}`

`count(node-set))` `sum(node-set)`	Anzahl Elemente der Knotenmenge Summe der Werte der Knotenmenge
`/movies/movie[rating <= (sum(../../rating) / count(../../rating)) ]`	Alle `movie`-Elemente mit unterdurchschnittlichem Rating
`string-length(string)` `contains(string, string)`	Anzahl Zeichen des Strings true, wenn 1. Parameter den 2. enthält
`/movies/movie[contains(title, 'Die')]`	erste movie-Element
`starts-with(string, string)`	true, wenn 1. Parameter mit 2. beginnt
`/movies/movie[starts-with(title, "Die")]`	Erste movie-Element
`substring-after(string1, string2)` `substring-before(string3, string4)`	Zeichenkette in string1 (string3), die hinter string2 (vor string4) vorkommt
`substring-after("Die Hard", " ")`	`"Hard"`

Ergänzende Literatur Zur Einführung, der Beschreibung der Axis-Komponenten, der genaueren Typumwandlung, Operatoren und Funktionen [1, Abschnitt 2.3] lesen. Weitere Details finden sich in der Spezifikation [2].

Literatur

[1] Serge Abiteboul, Ioana Manolescu, Philippe Rigaux, Marie-Christine Rousset, and Pierre Senellart. *Web Data Management*. Cambridge University Press, New York, NY, USA, 2011. http://webdam.inria.fr/Jorge/files/wdm.pdf.

[2] James Clark and Steve DeRose. XML Path Language (XPath) Version 1.0. Technical report, W3C, November 1999. https://www.w3.org/TR/xpath/.