XPath Queries in the Real World

David Baelde    Anthony Lick    Sylvain Schmitz

PODS, July 2, 2019, Amsterdam
XPath
XPath

Decidable Fragments

Problem (Satisfiability)

\[
\text{input } \varphi \text{ an XPath query} \\
\text{question } \exists \text{data tree } t . t \models \varphi
\]
**XPath**

**Decidable Fragments**

Problem (Satisfiability)

- input $\varphi$ an XPath query
- question $\exists \text{data tree } t. t \models \varphi$
**XPath**

**Decidable Fragments**

Problem (Satisfiability)

\[
\text{input } \varphi \text{ an XPath query}
\]

\[
\text{question } \exists \text{data tree } t . t \models \varphi ?
\]

**Expressiveness/Complexity**

- NonMixing
  - TOWER-complete
  - Core 2.0
  - Core 1.0
  - EXP-complete
  - NP-complete

- Positive
  - open TOWER-hard

- Data
  - EMSO²
  - Vertical
  - Forward
  - decidable
  - ACKERMANN-hard

- Undecidable
  - TOWER-complete
XPath

XML Path Language (XPath) 3.1
W3C Recommendation 21 March 2017

This version:
https://www.w3.org/TR/2017/REC-xpath-31-20170321/

Latest version of XPath 3.1:
https://www.w3.org/TR/xpath-31/

Previous versions of XPath 3.1:
https://www.w3.org/TR/2017/PR-xpath-31-20170117/
https://www.w3.org/TR/2016/PR-xpath-31-20161213/
https://www.w3.org/TR/2015/CR-xpath-31-20151217/
https://www.w3.org/TR/2014/CR-xpath-31-20141218/
https://www.w3.org/TR/2014/WD-xpath-31-20141007/
https://www.w3.org/TR/2014/WD-xpath-31-20140424/

Most recent version of XPath 3.0:
https://www.w3.org/TR/xpath-3/

Most recent version of XPath:
https://www.w3.org/TR/xpath/

Most recent Recommendation of XPath:
https://www.w3.org/TR/2014/REC-xpath-30-20140408/

Editors:
Jonathan Robie, biblicalhumanities.org <jonathan.robie@biblicalhumanities.org>
Michael Dyck, Invited Expert <jmdyck@libbiblo.org>
Josh Spiegel, Oracle Corporation <josh.spiegel@oracle.com>

Please check the errata for any errors or issues reported since publication.

See also translations.

This document is also available in these non-normative formats: XML and Change markings relative to previous edition.

Copyright © 2017 W3C® (MIT, ERCIM, Keio, Beihang). W3C liability, trademark and document use rules apply.

Abstract
XPath 3.1 is an expression language that allows the processing of values conforming to the data model defined in [XML Path and Data Model (XDM) 3.1]. The name of the language derives from its most distinctive feature, the path expression, which provides a means of hierarchic addressing of the nodes in an XML tree. As well as modeling the tree structure of XML, the data model also includes atomic values, function items, and sequences. This version of XPath supports JSON as well as XML, adding maps and arrays to the data model and supporting them with new expressions in the language and new functions in [XQuery and XPath Functions and Operators 3.1]. These are the most important new features in XPath 3.1.

Decidable Fragments

Problem (Satisfiability)
input $\varphi$ an XPath query
question $\exists$ data tree $t$. $t \models \varphi$?

Expressiveness/Complexity

NonMixing

NP-complete

EXP-complete

open

NP

NP-completeness

TOWER-complete

TOWER-hard

EMSO^2

Vertical

Data

Forward

Core 1.0

Core 2.0

decidable

ACKERMANN-hard

decidable

undecidable

Data
Overview

Benchmark

- open source
- 21,141 XPath queries

Coverage of decidable XPath fragments

- "vanilla"
- simple extensions

Analysis
OVERVIEW

benchmark
  ▶ open source
  ▶ 21,141 XPath queries
coverage of decidable XPath fragments
  ▶ “vanilla”
  ▶ simple extensions
analysis
Overview Benchmark Coverage Analysis

**Overview**

- **Benchmark**
  - open source
  - 21,141 XPath queries

- **Coverage** of decidable XPath fragments
  - "vanilla"
  - simple extensions

**Analysis**
overview

benchmark

- open source
- 21,141 XPath queries

coverage of decidable XPath fragments

- “vanilla”
- simple extensions

analysis
OVERVIEW

benchmark

▶ open source

▶ 21,141 XPath queries

coverage of decidable XPath fragments

▶ “vanilla”

▶ simple extensions

analysis
BENCHMARK CONSTRUCTION
Benchmark Construction

Sources

XSL

XQuery
Benchmark Construction

XSL

Sources

XQuery

Core 2.0
Core 1.0
NonMixing
EMSO
Vertical
Forward
Downward
RelaxNG Schemas
Benchmark Construction

Sources

- XSL
- XQuery

xpparser

syntax trees

tuned W3C reference parser

RelaxNG Schemas
BENCHMARK CONSTRUCTION

Resources:
- XSL
- XQuery
- xpparser
- syntax trees

Source Language Processing:
1. XSL
2. XQuery
3. xpparser
4. Syntax Trees

Benchmark Construction Overview:
- Benchmark Coverage Analysis
- Benchmark Construction
- XSL
- XQuery
- xpparser
- Syntax Trees

Sample XQuery Code:
```xml
<xqx:flworExpr ... xqx:prefix="od-api">headwordLemmatron</xqx:functionName><xqx:arguments><xqx:varRef><xqx:name>result</xqx:name>
```

Source Language to Syntax Tree Transformation:
- RelaxNG Schemas
- Core 1.0
- Core 2.0
- Positive
- Vertical
- Forward
- Downward
- Data
**BENCHMARK CONSTRUCTION**

Sources

- XSL
- XQuery

xpparser -> syntax trees

NonMixing
Core 2.0
Core 1.0
Positive
EMSO
Vertical
Forward
Downward
Data
RelaxNG Schemas
**Benchmark Construction**

**Sources**
- **XSL**
- **XQuery**

**xpparser** → **syntax trees**

**RelaxNG Schemas**
**Benchmark Construction**

Sources

- XSL
- XQuery

xpparser

Syntax trees

RelaxNG Schemas

Positive

Non Mixing

Core 2.0

Core 1.0

EMSO²

Vertical

Forward

Downward

Data

rng

rng

rng

rng

rng

rng

rng

rng

rng

rng

rng

rng

rng

rng

rng

rng

rng

rng

rng

rng

rng

rng

rng

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)

(rng)
## Benchmark Composition

<table>
<thead>
<tr>
<th>Sources</th>
<th>Queries</th>
<th>Coverage</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>XPath 1.0</td>
<td>XPath 2.0</td>
<td>XPath 3.0</td>
<td></td>
</tr>
<tr>
<td>XSLT</td>
<td>14,675</td>
<td>98.4%</td>
<td>100.0%</td>
<td>100.0%</td>
<td></td>
</tr>
<tr>
<td>XQuery</td>
<td>6,466</td>
<td>76.1%</td>
<td>87.4%</td>
<td>99.8%</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>21,141</td>
<td>91.6%</td>
<td>96.1%</td>
<td>99.9%</td>
<td></td>
</tr>
</tbody>
</table>
**Overview**

**Benchmark**

**Coverage**

**Analysis**

---

**Benchmark: Functions**

---

**XPath and XQuery Functions and Operators 3.1**

W3C Recommendation 21 March 2017

This version:

https://www.w3.org/TR/2017/REC-xpath-functions-3.1-20170321/

Latest version of XPath and XQuery Functions and Operators 3.1:

https://www.w3.org/TR/xpath-functions-3.1/

Previous versions of XPath and XQuery Functions and Operators 3.1:

https://www.w3.org/TR/2017/PR-xpath-functions-3.1-20170117/
https://www.w3.org/TR/2016/CR-xpath-functions-3.1-20161213/
https://www.w3.org/TR/2014/CR-xpath-functions-3.1-20141218/
https://www.w3.org/TR/2014/WD-xpath-functions-3.1-20141007/
https://www.w3.org/TR/2014/WD-xpath-functions-3.1-20140424/

Most recent version of XPath and XQuery Functions and Operators 3:

https://www.w3.org/TR/xpath-functions-3/

Most recent version of XPath and XQuery Functions and Operators:

https://www.w3.org/TR/xpath-functions/

Most recent Recommendation of XPath and XQuery Functions and Operators:

https://www.w3.org/TR/2014/REC-xpath-functions-3-20140408/

Editor:

Michael Kay (XSLT WG), Saxonica (<http://www.saxonica.com/>)

Please check the errata for any errors or issues reported since publication.

See also translations.

This document is also available in these non-normative formats: Specification in XML format using HTML5 vocabulary, XML function catalog, and HTML with change markings relative to version 3.0.

Copyright © 2017 W3C® (MIT, ERCIM, Keio, Beihang). W3C liability, trademark and document use rules apply.

---

**Abstract**

This document defines constructor functions, operators, and functions on the datatypes defined in [XML Schema Part 2: Datatypes Second Edition] and the datatypes defined in [XQuery and XPath Data Model (XDM) 3.1]. It also defines functions and operators on nodes and node sequences as defined in the [XQuery and XPath Data Model (XDM) 3.1]. These functions and operators are defined for use in [XML Path Language (XPath) 3.1] and [XQuery 3.1: An XML Query Language] and [XSL Transformations (XSLT) Version 3.0] and other related XML standards. The signatures and summaries of functions defined in this document are available at: [http://www.w3.org/2005/xpath-functions/](http://www.w3.org/2005/xpath-functions/).
### Benchmark: Functions

**XPath and XQuery Functions and Operators 3.1**

W3C Recommendation 21 March 2017

This version:
https://www.w3.org/TR/2017/REC-xpath-functions-3.1-20170321/

Latest version of XPath and XQuery Functions and Operators 3.1:
https://www.w3.org/TR/xpath-functions-31/

Previous versions of XPath and XQuery Functions and Operators 3.1:
https://www.w3.org/TR/2016/CR-xpath-functions-31-201612213/
https://www.w3.org/TR/2014/CR-xpath-functions-31-20141218/
https://www.w3.org/TR/2014/WD-xpath-functions-31-20141007/
https://www.w3.org/TR/2014/WD-xpath-functions-31-20140424/

Most recent version of XPath and XQuery Functions and Operators 3:
https://www.w3.org/TR/xpath-functions-3/

Most recent version of XPath and XQuery Functions and Operators:
https://www.w3.org/TR/xpath-functions/

Most recent Recommendation of XPath and XQuery Functions and Operators:
https://www.w3.org/TR/2014/REC-xpath-functions-30-20140408/

Editor:
Michael Kay (XSLT WG), Saxonica <http://www.saxonica.com/>

Please check the errata for any errors or issues reported since publication.

See also translations.

This document is also available in these non-normative formats: Specification in XML format using HTML5 vocabulary, XML function catalog, and HTML with change markings relative to version 3.0.

Copyright © 2017 W3C® (MIT, ERCIM, Keio, Beihang). W3C liability, trademark and document use rules apply.

---

**Abstract**

This document defines constructor functions, operators, and functions on the datatypes defined in [XML Schema Part 2: Datatypes Second Edition](https://www.w3.org/TR/2004/REC-xmlschema-2-20040204/) and the datatypes defined in [XQuery and XPath Data Model (XDM) 3.1](https://www.w3.org/TR/2016/CR-xquery11-datatypes-20161221/). It also defines functions and operators on nodes and node sequences as defined in the [XQuery and XPath Data Model (XDM) 3.1](https://www.w3.org/TR/2016/CR-xquery11-datatypes-20161221/). These functions and operators are defined for use in [XML Path Language (XPath) 3.1](https://www.w3.org/TR/2014/CR-xpath31-20141218/) and [XQuery 3.1: An XML Query Language](https://www.w3.org/TR/2014/CR-xquery11-20141218/) and [XSL Transformations (XSLT) Version 3.0](https://www.w3.org/TR/2014/CR-xslt31-20141104/) and other related XML standards. The signatures and summaries of functions defined in this document are available at: [http://www.w3.org/2005/xpath-functions/](http://www.w3.org/2005/xpath-functions/).

---

- **standard functions** *(dark violet)*
  - 57.23% of function calls
  - 76.32% in XSLT
  - 42.93% in XQuery

- **non-standard ones** *(light violet)*
  - typically user-defined
**Benchmark: Functions**

XPath and XQuery Functions and Operators 3.1

W3C Recommendation 21 March 2017

**Overview**

- **Benchmark Coverage Analysis**

**Benchmark: Functions**

- **Standard functions (dark violet)**
  - 57.23% of function calls
  - 76.32% in XSLT
  - 42.93% in XQuery

- **Non-standard ones (light violet)**
  - Typically user-defined

---

**Abstract**

This document defines constructor functions, operators, and functions on the datatypes defined in [XML Schema Part 2: Datatypes Second Edition] and the datatypes defined in [XQuery and XPath Data Model (XDM) 3.1]. It also defines functions and operators on nodes and node sequences as defined in the [XQuery and XPath Data Model (XDM) 3.1]. These functions and operators are defined for use in [XML Path Language (XPath) 3.1] and [XQuery 3.1: An XML Query Language] and [XSL Transformations (XSLT) Version 3.0] and other related XML standards. The signatures and summaries of functions defined in this document are available at: [http://www.w3.org/2005/xpath-functions/](http://www.w3.org/2005/xpath-functions/)
## Benchmark: Functions

<table>
<thead>
<tr>
<th>Sources</th>
<th>Queries</th>
<th>Coverage XPath 1.0</th>
<th>Coverage XPath 2.0</th>
<th>Coverage XPath 3.0</th>
<th>Coverage XPath 3.0 std</th>
</tr>
</thead>
<tbody>
<tr>
<td>XSLT</td>
<td>14,675</td>
<td>98.4%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>91.3%</td>
</tr>
<tr>
<td>XQuery</td>
<td>6,466</td>
<td>76.1%</td>
<td>87.4%</td>
<td>99.8%</td>
<td>46.7%</td>
</tr>
<tr>
<td>Total</td>
<td>21,141</td>
<td>91.6%</td>
<td>96.1%</td>
<td>99.9%</td>
<td>77.7%</td>
</tr>
</tbody>
</table>
**Basic Coverage**

- poor coverage

- except for NonMixing fragment
  
  MSO + positive data tests & data joins not mixing = and ≠
EXTENDING FRAGMENTS

A syntactic construct can be expressed through equivalent requests

encoded through equisatisfiable requests
EXTENDING FRAGMENTS

A syntactic construct can be

expressed through equivalent requests

coded through equisatisfiable requests
EXTENDING FRAGMENTS

A syntactic construct can be (polynomially) expressed through (polynomial time computable) equivalent requests encoded through (polynomial time computable) equisatisfiable requests defines a front-end
EXTENSIONS

/\pi\ root navigation, e.g.

//firstterm

\$x\ free variables, e.g.

\$module/merge

\pi\ \Delta\ d\ data tests against constants, e.g.

refmeta/refmiscinfo[@class = 'version']

\pi\ \Delta\ \pi\ positive data joins, e.g.

a/@href = preceding-sibling::li/a/@href

last()\ one-step positional positional predicates, e.g.

tocentry[position() = last()]
EXTENSIONS

\(/\pi\) root navigation, e.g. \\
\//firstterm

$\times$ \textbf{free variables}, e.g. \\
$\text{module/merge}$

$\pi \Delta d$ data tests against constants, e.g. \\
\text{refmeta/refmiscinfo[@class = 'version']}

$\pi \Delta \pi$ \textbf{positive data joins}, e.g. \\
a/@href = preceding-sibling::li/a/@href

\text{last()} \textbf{one-step positional positional predicates}, e.g. \\
tocentry[position() = last()]
EXTENSIONS

\[\pi\] root navigation, e.g.
   //firstterm

\$x\] free variables, e.g.
   $module/merge

\(\pi \triangle d\) data tests against constants, e.g.
   refmeta/refmmiscinfo[@class = 'version']

\(\pi \triangle \pi\) positive data joins, e.g.
   a/@href = preceding-sibling::li/a/@href

\text{last()} one-step positional positional predicates, e.g.
   tocentry[position() = last()]
EXTENSIONS

$/\pi$ root navigation, e.g.
   //firstterm

$x$ free variables, e.g.
   $\$module/merge$

$\pi \triangle d$ data tests against constants, e.g.
   refmeta/refmiscinfo[@class = 'version']

$\pi \triangle \pi$ positive data joins, e.g.
   a/@href = preceding-sibling::li/a/@href

last() one-step positional positional predicates, e.g.
   tocentry[position() = last()]
EXTENSIONS

/\pi \quad \text{root navigation, e.g.}
   \quad //firstterm

\$x \quad \text{free variables, e.g.}
   \quad $\text{module/merge}$

\pi \triangle d \quad \text{data tests against constants, e.g.}
   \quad \text{refmeta/refmiscinfo[@class = 'version']}

\pi \triangle \pi \quad \text{positive data joins, e.g.}
   \quad \text{a/@href = preceding-sibling::li/a/@href}

\text{last()} \quad \text{one-step positional predicates, e.g.}
   \quad \text{tocentry[position() = last()]}
**EXPRESSIBLE EXTENSIONS**

Example *(expressible root navigation)*

```
//firstterm
ancestor-or-self::*[not(parent::*)]//firstterm
```
**Expressible Extensions**

Example (**expressible** root navigation)

```xml
//firstterm

ancestor-or-self::*[not(parent::*)]//firstterm
```
**Expressible Extensions**

Example (expressible root navigation)

```
//firstterm
↓
ancestor-or-self::*[not(parent::*)]//firstterm
```

Fact (¬expressible)

Root navigation cannot be expressed in DownwardXPath.
**Encodable Extensions**

Example (**encodable** free variables)

\[
\text{\$module/merge}
\]

\[\downarrow\text{extend label set } \Sigma \text{ to } \Sigma \times 2^{\{\text{free vars}\}}\]

//.[or \alpha \in \Sigma, \text{\$module} \in S(\alpha, S)]/self::*[or S(\text{merge}, S)]
ENCODABLE EXTENSIONS

Example (encodable free variables)

\$\text{module/merge}\$

\[\downarrow\] extend label set \(\Sigma\) to \(\Sigma \times 2^{\{\text{free vars}\}}\)

\[
//.[\text{or}_{a \in \Sigma}, \text{module} \in \text{S}(a,S)]/\text{self::*[or}_{S}(\text{merge},S)]
\]
**Encodable Extensions**

Example (encodable free variables)

\[ \text{module/merge} \]

\[ \text{extend label set } \Sigma \text{ to } \Sigma \times 2^{\text{free vars}} \]

//.[or \alpha \in \Sigma, \text{module} \in S(\alpha, S)]/self::*[or S(merge, S)]

Proposition (¬encodable, c.f. Figueira & Segoufin 2009)
Satisfiability in ForwardXPath extended with root navigation or free variables is undecidable.
**Encodable Extensions**

Example (encodable free variables)

\[\text{$\text{module/merge}$} \]

\[\downarrow\text{extend label set } \Sigma \text{ to } \Sigma \times 2^{\{\text{free vars}\}}\]

\[//. [\text{or } a \in \Sigma, \text{$\text{module}$} \in \text{$\text{S(a,S)}$}] / \text{self::*[\text{or}$\text{$\text{S(merge,S)}$}]}\]

Proposition (\(\neg\text{encodable}, \text{c.f. Figueira & Segoufin 2009}\))

Satisfiability in ForwardXPath extended with root navigation or free variables is undecidable.

Proposition (\(\neg\text{poly. enc.}, \text{c.f. Figueira & Segoufin 2009}\))

Satisfiability in DownwardXPath extended with 
\([\text{position()}=\text{last()}]\) and 
\([\text{position()}!=\text{last()}]\) is \text{ACKERMANN-hard}. 
EXTENSIONS

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Core 1.0</th>
<th>Core 2.0</th>
<th>Downward</th>
<th>Vertical</th>
<th>Forward</th>
<th>EMSO²</th>
<th>NonMixing</th>
</tr>
</thead>
<tbody>
<tr>
<td>/π</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$x$</td>
<td>nat.</td>
<td>nat.</td>
<td>nat.</td>
<td>nat.</td>
<td>nat.</td>
<td>nat.</td>
<td>nat.†</td>
<td>nat.†</td>
</tr>
<tr>
<td>$π \triangle d$</td>
<td>nat.</td>
<td>nat.</td>
<td>nat.</td>
<td>nat.</td>
<td>nat.</td>
<td>nat.†</td>
<td>nat.†</td>
<td>nat.†</td>
</tr>
<tr>
<td>last()</td>
<td>nat.</td>
<td>nat.</td>
<td>nat.</td>
<td>nat.</td>
<td>nat.</td>
<td>nat.†</td>
<td>nat.†</td>
<td>nat.†</td>
</tr>
</tbody>
</table>

* support limited by available axes
† support limited to non-mixing

support limited by available axes
† support limited to non-mixing
# Extensions

<table>
<thead>
<tr>
<th>Positive</th>
<th>Core 1.0</th>
<th>Core 2.0</th>
<th>Downward</th>
<th>Vertical</th>
<th>Forward</th>
<th>EMSO²</th>
<th>NonMixing</th>
</tr>
</thead>
<tbody>
<tr>
<td>/π</td>
<td>expr.</td>
<td>expr.</td>
<td>expr.</td>
<td>expr.</td>
<td>expr.</td>
<td>expr.</td>
<td>expr.</td>
</tr>
<tr>
<td>$x$</td>
<td>nat.</td>
<td>nat.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\pi \triangle d$</td>
<td>nat.</td>
<td>nat.</td>
<td>nat.</td>
<td>nat.</td>
<td>nat. $\dagger$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\pi \triangle \pi$</td>
<td>nat.</td>
<td>nat.*</td>
<td>nat.*</td>
<td>nat.*</td>
<td>nat.* $\dagger$</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>last()</code></td>
<td>expr.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>expr.</td>
</tr>
</tbody>
</table>

* support limited by available axes  
$\dagger$ support limited to non-mixing
# Extensions

<table>
<thead>
<tr>
<th>Positive</th>
<th>Core 1.0</th>
<th>Core 2.0</th>
<th>Downward</th>
<th>Vertical</th>
<th>Forward</th>
<th>EMSO$^2$</th>
<th>NonMixing</th>
</tr>
</thead>
<tbody>
<tr>
<td>/π</td>
<td>expr.</td>
<td>expr.</td>
<td>expr.</td>
<td>expr.</td>
<td>expr.</td>
<td>expr.</td>
<td>expr.</td>
</tr>
<tr>
<td>$x$</td>
<td>enc.</td>
<td>enc.</td>
<td>enc.</td>
<td>enc.</td>
<td>enc.</td>
<td>nat.</td>
<td>nat.</td>
</tr>
<tr>
<td>$\pi \triangle d$</td>
<td>nat.</td>
<td>enc.</td>
<td>enc.</td>
<td>enc.</td>
<td>enc.</td>
<td>enc.</td>
<td>enc.</td>
</tr>
<tr>
<td>$\pi \triangle \pi$</td>
<td>nat.</td>
<td>enc.</td>
<td>enc.</td>
<td>nat.$^*$</td>
<td>nat.$^*$</td>
<td>nat.$^*$</td>
<td>enc.</td>
</tr>
<tr>
<td>last()</td>
<td>enc.</td>
<td>expr.</td>
<td>enc.$^*$</td>
<td>enc.$^*$</td>
<td>expr.</td>
<td>expr.</td>
<td>expr.</td>
</tr>
</tbody>
</table>

* support limited by available axes  
$^\dagger$ support limited to non-mixing
## Extensions

<table>
<thead>
<tr>
<th>Positive</th>
<th>Core 1.0</th>
<th>Core 2.0</th>
<th>Downward</th>
<th>Vertical</th>
<th>Forward</th>
<th>EMSO²</th>
<th>NonMixing</th>
</tr>
</thead>
<tbody>
<tr>
<td>$/\pi$</td>
<td>expr.</td>
<td>expr.</td>
<td>expr.</td>
<td>$\neg$expr.</td>
<td>expr.</td>
<td>$\neg$enc.</td>
<td>expr.</td>
</tr>
<tr>
<td>$$x$</td>
<td>enc.</td>
<td>enc.</td>
<td>enc.</td>
<td>$\neg$expr.</td>
<td>enc.</td>
<td>$\neg$enc.</td>
<td>nat.</td>
</tr>
<tr>
<td>$\pi \triangle d$</td>
<td>nat.</td>
<td>enc.</td>
<td>enc.</td>
<td>enc.</td>
<td>enc.</td>
<td>enc.</td>
<td>enc.</td>
</tr>
<tr>
<td>$\pi \triangle \pi$</td>
<td>nat.</td>
<td>enc.</td>
<td>enc.</td>
<td>nat.*</td>
<td>nat.*</td>
<td>nat.*</td>
<td>enc.</td>
</tr>
<tr>
<td>last()</td>
<td>enc.</td>
<td>expr.</td>
<td>$\neg$p.enc.</td>
<td>enc.*</td>
<td>enc.*</td>
<td>expr.</td>
<td></td>
</tr>
</tbody>
</table>

* support limited by available axes  † support limited to non-mixing
Extended Coverage

combined 60.86%

Core 2.0 75.03% of XSLT and 28.08% of XQuery
COMPARISON

Interactive interface

http://www.lsv.fr/~schmitz/xpparser
Function Support

Coverage with respect to XPath 3.0 std.

combined 78.33%

Core 2.0 82.14% of XSLT and 60.00% of XQuery

Extra support through SMT?

▶ interval encoding of trees for Positive fragment
▶ support for linear arithmetic and string functions like concat(), contains(), string-length(), etc.
▶ 62.75% coverage, 84.77% of XSLT wrt. XPath 3.0 std
**FUNCTION SUPPORT**

Coverage with respect to XPath 3.0 std.

combined 78.33%

Core 2.0 82.14% of XSLT and 60.00% of XQuery

Extra support through SMT?

- interval encoding of trees for Positive fragment
- support for linear arithmetic and string functions like `concat()`, `contains()`, `string-length()`, etc.
- 62.75% coverage, 84.77% of XSLT wrt. XPath 3.0 std
**Function Support**

Extra support through SMT?

- interval encoding of trees for Positive fragment
- support for linear arithmetic and string functions like `concat()`, `contains()`, `string-length()`, etc.
- 62.75% coverage, 84.77% of XSLT wrt. XPath 3.0 std

![Percentage of XSLT and XQuery queries](chart.png)
Conclusions

benchmark of 21,141 XPath queries

spectability

▶ importance of front-end
▶ XSLT support vs. XQuery support
▶ challenge: function support
▶ future? SMT techniques
**Conclusions**

- **Benchmark** of 21,141 XPath queries

- **Satisfiability**
  - Importance of front-end
  - XSLT support vs. XQuery support
  - Challenge: function support
  - Future? SMT techniques
Conclusions

Benchmark of 21,141 XPath queries

- Importance of front-end
- XSLT support vs. XQuery support
- Challenge: function support
- Future? SMT techniques
Conclusions

Benchmark of 21,141 XPath queries

- Importance of front-end
- XSLT support vs. XQuery support
- Challenge: function support
- Future? SMT techniques
**Benchmark: Size Distribution**

![Graph showing the size distribution benchmark](image)
## Synthetic Benchmarks

<table>
<thead>
<tr>
<th>Sources</th>
<th>Queries</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>XPath 1.0</td>
</tr>
<tr>
<td>XPathMark-FT</td>
<td>64</td>
<td>100.0%</td>
</tr>
<tr>
<td>XPathMark-PT</td>
<td>38</td>
<td>100.0%</td>
</tr>
<tr>
<td>XMark</td>
<td>66</td>
<td>92.4%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>168</strong></td>
<td><strong>97.0%</strong></td>
</tr>
</tbody>
</table>

Percentage of queries:

- Positive
- Core 1.0
- Core 2.0
- Downward
- Vertical
- Forward
- EMSO
- NonMixing

- **+extras**
- **extended**
- **baseline**