Simple Schemas for Unordered XML

Iovka Boneva    Radu Ciucanu    Sławek Staworko

University of Lille & INRIA, France

WebDB’13
June 23, 2013
Table of contents

1 Motivation

2 Multiplicity schemas

3 Computational properties

4 Expressiveness

5 Conclusions and future work
Unordered XML - the relative order of elements is not relevant
XML is used for *data-centric* applications (as a semi-structured database).
Simple Schemas for Unordered XML

Unordered XML - the relative order of elements is not relevant
XML is used for data-centric applications (as a semi-structured database).

Schema for XML
A schema for XML defines the content model for every node.
DTDs

Real-world DTD for DBLP repository

\[ \text{dblp} \rightarrow (\text{article} \mid \text{book})^* \]
\[ \text{article} \rightarrow (\text{title} \mid \text{author} \mid \text{year} \mid \text{editor})^* \]
\[ \text{book} \rightarrow (\text{title} \mid \text{author} \mid \text{year} \mid \text{editor})^* \]
DTDs

Real-world DTD for DBLP repository

\[
\begin{align*}
dblp & \rightarrow (\text{article} \mid \text{book})^* \\
\text{article} & \rightarrow (\text{title} \mid \text{author} \mid \text{year} \mid \text{editor})^* \\
\text{book} & \rightarrow (\text{title} \mid \text{author} \mid \text{year} \mid \text{editor})^*
\end{align*}
\]

Drawbacks

- Over-permissiveness
- Static analysis
Over-permissiveness

Real-world DTD for DBLP repository

\[
dblp \rightarrow (article \mid book)^*
\]
\[
article \rightarrow (title \mid author \mid year \mid editor)^*
\]
\[
book \rightarrow (title \mid author \mid year \mid editor)^*
\]

We do not want:

- an article with many title’s or without a title.
- an article with many year’s...
Over-permissiveness

Real-world DTD for DBLP repository

\[
\text{dblp} \rightarrow (\text{article} \mid \text{book})^*
\]

\[
\text{article} \rightarrow (\text{title} \mid \text{author} \mid \text{year} \mid \text{editor})^*
\]

\[
\text{book} \rightarrow (\text{title} \mid \text{author} \mid \text{year} \mid \text{editor})^*
\]

We do not want:

- an article with many title’s or without a title.
- an article with many year’s...

We propose a new schema formalism: DMS

\[
\text{dblp} \rightarrow \text{article}^+ \parallel \text{book}^+
\]

\[
\text{article} \rightarrow \text{title} \parallel \text{year} \parallel \text{author}^+
\]

\[
\text{book} \rightarrow \text{title} \parallel \text{year} \parallel (\text{author}^+ \mid \text{editor}^+)
\]
Query equivalence in the presence of schema

Consider DTD $D$:

- $\text{dblp} \rightarrow (\text{article} \mid \text{book})^*$
- $\text{article} \rightarrow (\text{title} \mid \text{author} \mid \text{year} \mid \text{editor})^*$
- $\text{book} \rightarrow (\text{title} \mid \text{author} \mid \text{year} \mid \text{editor})^*$

Twig query $q$

- $\text{dblp}$
- $\text{book}$
- $\text{title}$
- $\text{author}$
- $\text{year}$
- $\star$ “C. Papadimitriou” $\star$

Twig query $q'$

- $\text{dblp}$
- $\text{book}$
- $\text{author}$
- “C. Papadimitriou”
Query equivalence in the presence of schema

Consider DTD $D$:

\[
\begin{align*}
  dblp & \rightarrow (article \mid book)^* \\
  article & \rightarrow (title \mid author \mid year \mid editor)^* \\
  book & \rightarrow (title \mid author \mid year \mid editor)^*
\end{align*}
\]

Twig query $q$

Twig query $q'$

Twig query $q$ and $q'$ are not equivalent.
Query equivalence in the presence of schema

Consider DMS $S$:

$$
\begin{align*}
\text{dblp} & \rightarrow \text{article}^+ \parallel \text{book}^+ \\
\text{article} & \rightarrow \text{title} \parallel \text{year} \parallel \text{author}^+ \\
\text{book} & \rightarrow \text{title} \parallel \text{year} \parallel (\text{author}^+ \mid \text{editor}^+)
\end{align*}
$$

Twig query $q$

$$
\begin{align*}
\text{dblp} & \\
\text{book} & \\
\text{title} & \quad \text{author} & \quad \text{year} \\
\star & \quad \text{“C. Papadimitriou”} & \star
\end{align*}
$$

Twig query $q'$

$$
\begin{align*}
\text{dblp} & \\
\text{book} & \\
\text{title} & \quad \text{author} & \text{year} \\
\star & \quad \text{“C. Papadimitriou”} & \star
\end{align*}
$$
Query equivalence in the presence of schema

Consider DMS $S$:

$\text{dblp} \rightarrow \text{article}^+ \parallel \text{book}^+$

$\text{article} \rightarrow \text{title} \parallel \text{year} \parallel \text{author}^+$

$\text{book} \rightarrow \text{title} \parallel \text{year} \parallel (\text{author}^+ \mid \text{editor}^+)$

Twig query $q$

Twig query $q'$

$\equiv_S$
We propose a new schema formalism: DMS

\[
\text{dblp} \rightarrow \text{article}^+ \parallel \text{book}^+ \\
\text{article} \rightarrow \text{title} \parallel \text{year} \parallel \text{author}^+ \\
\text{book} \rightarrow \text{title} \parallel \text{year} \parallel (\text{author}^+ \mid \text{editor}^+) 
\]

Questions

- What are its properties? (complexity, normal form, etc.)
- How much of the expressive power of DTDs does it capture?
Contributions

We propose new schema formalisms for unordered XML:

DMS  Disjunctive multiplicity schema
MS   Disjunction-free multiplicity schema

What are the computational properties of deciding problems involving unordered schemas?
  ▶ Static analysis: schema satisfiability, membership of a tree to the language of a schema, schema containment, twig query satisfiability, implication, and containment in the presence of schema.

How much of the expressive power of DTDs do they capture?
  ▶ w.r.t. real-life applications.
Related work

There are several approaches in the literature for unordered XML formalisms, they typically suffer from high computational complexity.

- Interpret a regular expression as its *closure*: consider all the permutations of words in the language.
  - Deciding string membership to regular expression closure is NP-complete [Kopczynski and To ’10].

- Consider that all the regular expressions are *order-invariant*: they define a language equal to its closure.
  - Deciding whether a regular expression is order-invariant is PSPACE-complete [Neven and Schwentick ’99].
Multiplicity schemas
We define languages of unordered words (no order on symbols).

\[ aab \equiv aba \equiv baa. \]

An unordered word is a multiset of symbols.

\[ \{a, a, b\}. \]
# Multiplicity expressions

## Multiplicities

<table>
<thead>
<tr>
<th>Multiplicities</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a^+$</td>
<td>$a, aa, \ldots$</td>
</tr>
<tr>
<td>$b^*$</td>
<td>$\varepsilon, b, bb, \ldots$</td>
</tr>
<tr>
<td>$c^?$</td>
<td>$\varepsilon, c$</td>
</tr>
<tr>
<td>$d^1$</td>
<td>$d$</td>
</tr>
<tr>
<td>$f^0$</td>
<td>$\varepsilon$</td>
</tr>
</tbody>
</table>
Multiplicity expressions

### Multiplicities

- $a^+$: $a, aa, \ldots$
- $b^*$: $\varepsilon, b, bb, \ldots$
- $c^?$: $\varepsilon, c$
- $d^1$: $d$
- $f^0$: $\varepsilon$

**A disjunction-free multiplicity expression** uses multiplicities and unordered concatenation ("∥").

$$E_1 = a^+ ∥ b^* ∥ c^? ∥ d$$

```
bd a ✓
ab c X
d d X
```
### Multiplicity expressions

#### Multiplicities

- $a^+ \ a, aa, \ldots$
- $b^* \ \varepsilon, b, bb, \ldots$
- $c^? \ \varepsilon, c$
- $d^1 \ d$
- $f^0 \ \varepsilon$

- A **disjunction-free multiplicity expression** uses multiplicities and unordered concatenation ("∥").

\[
E_1 = a^+ \parallel b^* \parallel c^? \parallel d
\]

- A **disjunctive multiplicity expression** additionally uses the disjunction ("|").

\[
E_2 = (a \mid b)^+ \parallel (c^? \mid d^*)
\]
Multiplicity schemas

**MS Disjunction-free multiplicity schema**

A set of disjunction-free multiplicity expressions

\[
\begin{align*}
  r & \rightarrow a^+ \| b^* \| c^? \\
  a & \rightarrow b^? \| c^* \\
  b & \rightarrow a^? \| d
\end{align*}
\]
**Multiplicity schemas**

**MS** **Disjunction-free multiplicity schema**

A set of disjunction-free multiplicity expressions

\[
\begin{align*}
    r & \rightarrow a^+ \parallel b^* \parallel c^* \\
    a & \rightarrow b? \parallel c^* \\
    b & \rightarrow a? \parallel d
\end{align*}
\]

**DMS** **Disjunctive multiplicity schema**

A set of disjunctive multiplicity expressions

\[
\begin{align*}
    r & \rightarrow a^* \parallel (b \mid c)^+ \\
    a & \rightarrow b? \parallel c^* \\
    b & \rightarrow (c? \mid d?)
\end{align*}
\]
Computational properties
### Summary of complexity results

<table>
<thead>
<tr>
<th>Problem of interest</th>
<th>DTD</th>
<th>DMS</th>
<th>disj.-free DTD</th>
<th>MS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema satisfiability</td>
<td>PTIME</td>
<td>PTIME</td>
<td>PTIME</td>
<td>PTIME</td>
</tr>
<tr>
<td>Membership</td>
<td>PTIME</td>
<td>PTIME</td>
<td>PTIME</td>
<td>PTIME</td>
</tr>
<tr>
<td>Containment of two schemas</td>
<td>PSPACE-c&lt;sup&gt;a&lt;/sup&gt;</td>
<td>PTIME</td>
<td>coNP-h&lt;sup&gt;a&lt;/sup&gt;</td>
<td>PTIME</td>
</tr>
<tr>
<td>Query satisfiability</td>
<td>NP-c</td>
<td>NP-c</td>
<td>PTIME</td>
<td>PTIME</td>
</tr>
<tr>
<td>Query implication</td>
<td>EXPTIME-c</td>
<td>EXPTIME-c</td>
<td>PTIME</td>
<td>PTIME</td>
</tr>
<tr>
<td>Query containment</td>
<td>EXPTIME-c</td>
<td>EXPTIME-c</td>
<td>coNP-c</td>
<td>coNP-c</td>
</tr>
</tbody>
</table>

<sup>a</sup>PTIME when only 1-unambiguous regular expressions are allowed
## Summary of complexity results

<table>
<thead>
<tr>
<th>Problem of interest</th>
<th>DTD</th>
<th>DMS</th>
<th>disj.-free DTD</th>
<th>MS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema satisfiability</td>
<td>PTIME</td>
<td>PTIME</td>
<td>PTIME</td>
<td>PTIME</td>
</tr>
<tr>
<td>Membership</td>
<td>PTIME</td>
<td>PTIME</td>
<td>PTIME</td>
<td>PTIME</td>
</tr>
<tr>
<td>Containment of two schemas</td>
<td>PSPACE-a&lt;sup&gt;a&lt;/sup&gt;</td>
<td>PTIME</td>
<td>coNP-h-a&lt;sup&gt;a&lt;/sup&gt;</td>
<td>PTIME</td>
</tr>
<tr>
<td>Query satisfiability</td>
<td>NP-c</td>
<td>NP-c</td>
<td>PTIME</td>
<td>PTIME</td>
</tr>
<tr>
<td>Query implication</td>
<td>EXPTIME-c</td>
<td>EXPTIME-c</td>
<td>PTIME</td>
<td>PTIME</td>
</tr>
<tr>
<td>Query containment</td>
<td>EXPTIME-c</td>
<td>EXPTIME-c</td>
<td>coNP-c</td>
<td>coNP-c</td>
</tr>
</tbody>
</table>

<sup>a</sup>PTIME when only 1-unambiguous regular expressions are allowed
Normal form of the disjunctive multiplicity expressions

We illustrate the normal form on $E_0 = a^+ \parallel (b \mid c) \parallel d^?$:

$C_E$  *Conflicting pairs of siblings* - pairs of symbols which cannot be present in a word at the same time. $C_{E_0} = \{(b, c), (c, b)\}$

$N_E$  *Cardinality map* - all the numbers of occurrences for every symbol. $N_{E_0} = \{(b, 0), (b, 1), (c, 0), (c, 1), (d, 0), (d, 1), (a, 1), (a, 2), \ldots\}$

$P_E$  *Sets of required symbols* - at least one of them should be present in any word. $P_{E_0} = \{\{a\}, \{b, c\}, \ldots\}$

Usefulness of the normal form:

- Checking the containment of two schemas in polynomial time.
- Validating a XML document in streaming.
### Summary of complexity results

<table>
<thead>
<tr>
<th>Problem of interest</th>
<th>DTD</th>
<th>DMS</th>
<th>disj.-free DTD</th>
<th>MS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema satisfiability</td>
<td>PTIME</td>
<td>PTIME</td>
<td>PTIME</td>
<td>PTIME</td>
</tr>
<tr>
<td>Membership</td>
<td>PTIME</td>
<td>PTIME</td>
<td>PTIME</td>
<td>PTIME</td>
</tr>
<tr>
<td>Containment of two schemas</td>
<td>PSPACE-c&lt;sup&gt;a&lt;/sup&gt;</td>
<td>PTIME</td>
<td>coNP-h&lt;sup&gt;a&lt;/sup&gt;</td>
<td>PTIME</td>
</tr>
<tr>
<td>Query satisfiability</td>
<td>NP-c</td>
<td>NP-c</td>
<td>PTIME</td>
<td>PTIME</td>
</tr>
<tr>
<td>Query implication</td>
<td>EXPTIME-c</td>
<td>EXPTIME-c</td>
<td>PTIME</td>
<td>PTIME</td>
</tr>
<tr>
<td>Query containment</td>
<td>EXPTIME-c</td>
<td>EXPTIME-c</td>
<td>coNP-c</td>
<td>coNP-c</td>
</tr>
</tbody>
</table>

<sup>a</sup>PTIME when only 1-unambiguous regular expressions are allowed
Dependency graphs for MS

\[ r \rightarrow a^+ \parallel b^* \]
\[ b \rightarrow a \]
\[ a \rightarrow c \]
\[ c \rightarrow d? \]
Query satisfiability (MS)

**Input:** a schema $S$ and a query $q$.

**Output:** is there a tree satisfying $S$ and $q$ at the same time?

The problem reduces to testing embedding of $q$ in $G_S$.

\[
S
r \rightarrow a^+ \parallel b^* \\
b \rightarrow a \\
a \rightarrow c \\
c \rightarrow d
\]

\[
G_S
r
\\downarrow
b \leftarrow a \\
\downarrow
c \\
\downarrow
d
\]

\[
q
r
\\downarrow
b \leftarrow a \\
\downarrow
b \\
\downarrow
c \\
\downarrow
d
\]

\[
t \in L(S)
\]

\[
r \leftarrow r \\
/ \\
\downarrow \\
a \\
/ \\
\downarrow \\
c \\
/ \\
\downarrow \\
d
\]
Query implication (MS)

**Input:** a schema $S$ and a query $q$.
**Output:** is $q$ satisfied by all the trees satisfying $S$?

The problem reduces to testing embedding of $q$ in $G^u_S$. 

I. Boneva, R. Ciucanu, S. Staworko
Simple Schemas for Unordered XML
WebDB’13 22 / 28
Expressiveness
Expressiveness

- A regular expression is *captured* by DME if there exists a DME such that they define the same language modulo the ordering.

- Examples:
  - $a.b?(c^* \mid d)^*$ is captured by $a \parallel b^? \parallel c^* \parallel d^*$ (ME).
  - $(a \mid b).(c^* \mid d)^?$ is captured by $(a \mid b) \parallel (c^* \mid d^?)$ (DME).
  - $(a \mid b).(a \mid b \mid c)$ and $(a.b)^*$ cannot be captured.

Observations

1. Regular expressions are strictly more expressive than DME.
2. CHAREs [Bex et al. ’10] are strictly less expressive than DME.

- A DTD is *captured* by DMS if all its regular expressions are captured by DME.
Expressiveness

- **XMark:**
  - all the regular expressions captured by DME,
  - 76/77 regular expressions captured by ME.

- **“University of Amsterdam XML Web Collection”**
  - 84% unique regular expressions captured by DME,
  - 74.6% unique regular expressions captured by ME,
  - 55.5% DTDs captured by DMS,
  - 45.8% DTDs captured by MS.
Conclusions and future work
Conclusions

- Schema formalisms for unordered XML:
  - Disjunction-free multiplicity schema (MS).
  - Disjunctive multiplicity schema (DMS).

- Static analysis:
  - No increase in computational complexity w.r.t. their DTD counterparts.

- Expressiveness:
  - Compelling covering percentages w.r.t. XMark and a real-life collection.
Future work

- Extend expressibility and maintain low complexity profile:
  - Generalize multiplicities to arbitrary intervals i.e., $a^{[n,m]}$,  
  - Hybrid schemas.

- Learning (D)MS from document examples.

- Query optimization in the presence of (D)MS.