Motivation

Like relational data, XML documents can contain redundant information due to functional dependencies.

Functional Dependency: \( \text{@AreaCode} \rightarrow \text{@City} \)
Motivation

This redundancy is reflected in the relational storage of XML documents.

XML functional dependency
\(@\text{AreaCode} \rightarrow @\text{City}\)

Relational Storage

<table>
<thead>
<tr>
<th>AreaCode</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>416</td>
<td>Toronto</td>
</tr>
<tr>
<td>416</td>
<td>Toronto</td>
</tr>
<tr>
<td>416</td>
<td>Toronto</td>
</tr>
<tr>
<td>416</td>
<td>Toronto</td>
</tr>
</tbody>
</table>

regular functional dependency
\(\text{AreaCode} \rightarrow \text{City}\)
Motivation

This redundancy is reflected in the relational storage of XML documents.

XML Document

Relational Storage

<table>
<thead>
<tr>
<th>AreaCode</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>416</td>
<td>Toronto</td>
</tr>
<tr>
<td>416</td>
<td>Toronto</td>
</tr>
<tr>
<td>416</td>
<td>Toronto</td>
</tr>
<tr>
<td>416</td>
<td>Toronto</td>
</tr>
</tbody>
</table>

XML functional dependency

@AreaCode → @City

equality-generating dependency

∀ R₁(\(\bar{x}, a\)) ∧ R₂(\(\bar{y}, c\)) ∧ R₁(\(\bar{x}', a\)) ∧ R₂(\(\bar{y}', c'\)) → c = c'
Motivation

This redundancy is reflected in the relational storage of XML documents.

The more redundant the data, the more prone to update anomalies.
Motivation

Solution: normalizing data to eliminate redundancies.

Normal forms for relational data:

- **BCNF** eliminates all redundancies w.r.t. functional dependencies but may lose dependencies.
- **3NF** eliminates some redundancies but preserves dependencies.

Normal forms for XML documents:

- **XNF** eliminates redundancies w.r.t. XML functional dependencies.

If nontrivial FD $p_1, \ldots, p_n \rightarrow q @ l$ holds then $p_1, \ldots, p_n \rightarrow q$ should also hold.
Motivation

Many XML documents are stored and queried using relational database management systems.

Questions:

- How do XML constraints translate to the relational storage?
- What is the best XML design to have a redundancy-free relational storage?
- What is a good XML design to have a low-redundancy relational storage?

We use an information-theoretic technique to measure the redundancy of data.
Outline

• Overview of the information-theoretic measure.
• Storing XML in relations and constraint translation.
• Designing XML to achieve low redundancy in relational storage.
• Concluding remarks.
Measure of Information Content

- Proposed by Arenas & Libkin in PODS’03.
- Used to measure the redundancy of a data value in a database instance with respect to a set of constraints.
- Intuitively, $\text{RIC}_I(p|\Sigma)$ measures the relative information content of position $p$ in instance $I$ w.r.t. constraints $\Sigma$.
- Independent of data models and query languages.
Measure of Information Content

- Proposed by Arenas & Libkin in PODS’03.
- Used to measure the redundancy of a data value in a database instance with respect to a set of constraints.
- Intuitively, $R_{IC}^{I}(p|\Sigma)$ measures the relative information content of position $p$ in instance $I$ w.r.t. constraints $\Sigma$.
- Independent of data models and query languages.

\[
\Sigma = \{A \rightarrow C\}
\]

\[
R_{IC}^{I}(P|\Sigma) = \begin{bmatrix}
1 & 2 & 3 & 4 \\
1 & 2 & 3 & 5
\end{bmatrix}
\]

\[
R_{IC}^{I}(P|\Sigma) = 0.875
\]

\[
\Sigma = \{A \rightarrow C, \ B \rightarrow C\}
\]

\[
R_{IC}^{I}(P|\Sigma) = \begin{bmatrix}
1 & 2 & 3 & 4 \\
1 & 2 & 3 & 5
\end{bmatrix}
\]

\[
R_{IC}^{I}(P|\Sigma) = 0.781
\]
Measure of Information Content

- Proposed by Arenas & Libkin in PODS’03.
- Used to measure the redundancy of a data value in a database instance with respect to a set of constraints.
- Intuitively, $R_{IC_I}(p|\Sigma)$ measures the relative information content of position $p$ in instance $I$ w.r.t. constraints $\Sigma$.
- Independent of data models and query languages.

$$\Sigma = \{ A \rightarrow C \}$$

$$\Sigma = \{ A \rightarrow C, B \rightarrow C \}$$

<table>
<thead>
<tr>
<th></th>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{IC_I}(P</td>
<td>\Sigma)$</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0.875</td>
<td>0.781</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{IC_I}(P</td>
<td>\Sigma)$</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0.781</td>
<td>0.629</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Measure of Information Content

- Proposed by Arenas & Libkin in PODS’03.
- Used to measure the redundancy of a data value in a database instance with respect to a set of constraints.
- Intuitively, $\text{RIC}_I(p|\Sigma)$ measures the relative information content of position $p$ in instance $I$ w.r.t. constraints $\Sigma$.
- Independent of data models and query languages.

<table>
<thead>
<tr>
<th>$\Sigma = {A \rightarrow C}$</th>
<th>$\Sigma = {A \rightarrow C, B \rightarrow C}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{RIC}_I(P</td>
<td>\Sigma)$</td>
</tr>
<tr>
<td>0.875</td>
<td>0.781</td>
</tr>
<tr>
<td>0.781</td>
<td>0.629</td>
</tr>
<tr>
<td>0.711</td>
<td>0.522</td>
</tr>
</tbody>
</table>
Measure of Information Content

- Proposed by Arenas & Libkin in PODS’03.
- Used to measure the redundancy of a data value in a database instance with respect to a set of constraints.
- Intuitively, $\text{RIC}_I(p|\Sigma)$ measures the relative information content of position $p$ in instance $I$ w.r.t. constraints $\Sigma$.
- Independent of data models and query languages.

<table>
<thead>
<tr>
<th>$\Sigma = { A \rightarrow C }$</th>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{RIC}_I(P</td>
<td>\Sigma)$</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>0.875</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.781</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.711</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.658</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\Sigma = { A \rightarrow C, \ B \rightarrow C }$</th>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{RIC}_I(P</td>
<td>\Sigma)$</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>0.781</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.629</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.522</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.446</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Measure of Information Content

\[ R(A, B, C) \quad \Sigma = \{ A \rightarrow B \} \]

\[
\begin{array}{ccc}
A & B & C \\
1 & 2 & 3 \\
1 & 2 & 4 \\
\end{array}
\]

Pick \( k \) such that \( \text{adom}(I) \subseteq \{1, \ldots, k\} \) \((k = 7)\).
For every \( X \subseteq Pos(I) - \{p\} \) compute probability distribution \( P(a|X) \) for every \( a \in \{1, \ldots, k\} \).
Measure of Information Content

$$R(A, B, C) \sum = \{A \rightarrow B\}$$

Pick $k$ such that $\text{adom}(I) \subseteq \{1, \ldots, k\}$ ($k = 7$).

For every $X \subseteq \text{Pos}(I) - \{p\}$ compute probability distribution $P(a|X)$ for every $a \in \{1, \ldots, k\}$.

$$P(2|X) =$$
Measure of Information Content

\[ R(A, B, C) \sum = \{A \rightarrow B\} \]

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Pick \( k \) such that \( \text{adom}(I) \subseteq \{1, \ldots, k\} \) (\( k = 7 \)).
For every \( X \subseteq \text{Pos}(I) - \{p\} \) compute probability distribution
\( P(a|X) \) for every \( a \in \{1, \ldots, k\} \).

\[ P(2|X) = \]
Measure of Information Content

\[ R(A, B, C) \quad \Sigma = \{ A \rightarrow B \} \]

\[
\begin{array}{ccc}
A & B & C \\
4 & 2 & 3 \\
1 & 2 & 7
\end{array}
\]

Pick \( k \) such that \( \text{adom}(I) \subseteq \{1, \ldots, k\} \) (\( k = 7 \)).
For every \( X \subseteq \text{Pos}(I) - \{p\} \) compute probability distribution \( P(a|X) \) for every \( a \in \{1, \ldots, k\} \).

\[ P(2|X) = \]
Measure of Information Content

\[ R(A, B, C) \sum = \{A \rightarrow B\} \]

\[
\begin{array}{ccc}
A & B & C \\
4 & 2 & 3 \\
1 & 2 & 7 \\
\end{array}
\]

Pick \( k \) such that \( \text{adom}(I) \subseteq \{1, \ldots, k\} \) (\( k = 7 \)).
For every \( X \subseteq Pos(I) - \{p\} \) compute probability distribution \( P(a | X) \) for every \( a \in \{1, \ldots, k\} \).

\[ P(2 | X) = 48/ \]
Measure of Information Content

\[ R(A, B, C) \sum = \{A \rightarrow B\} \]

Pick \( k \) such that \( \text{dom}(I) \subseteq \{1, \ldots, k\} \) \((k = 7)\).

For every \( X \subseteq \text{Pos}(I) - \{p\} \) compute probability distribution \( P(a|X) \) for every \( a \in \{1, \ldots, k\} \).

\[
\begin{align*}
P(2|X) &= 48/(48 + 6 \times 42) = 0.16 \\
P(a|X) &= 42/(48 + 6 \times 42) = 0.14 \text{ for every } a \neq 2
\end{align*}
\]
Measure of Information Content

\[ R(A, B, C) \sum = \{ A \rightarrow B \} \]

Pick \( k \) such that \( adom(I) \subseteq \{1, \ldots, k\} \) \((k = 7)\).
For every \( X \subseteq Pos(I) - \{p\} \) compute probability distribution \( P(a|X) \) for every \( a \in \{1, \ldots, k\} \).

\[
\begin{align*}
P(2|X) &= 48/(48 + 6 \times 42) = 0.16 \\
P(a|X) &= 42/(48 + 6 \times 42) = 0.14 \text{ for every } a \neq 2
\end{align*}
\]

Conditional entropy : 2.8057
Average over all possible \( X \): \( Ric^k_I = 2.4558 \)
Measure of Information Content

$R(A, B, C) \quad \Sigma = \{A \rightarrow B\}$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Pick $k$ such that $\text{adom}(I) \subseteq \{1, \ldots, k\}$ ($k = 7$).
For every $X \subseteq \text{Pos}(I) - \{p\}$ compute probability distribution $P(a|X)$ for every $a \in \{1, \ldots, k\}$.

$P(2|X) = \frac{48}{48 + 6 \times 42} = 0.16$
$P(a|X) = \frac{42}{48 + 6 \times 42} = 0.14$ for every $a \neq 2$

Conditional entropy : 2.8057
Average over all possible $X$ : $R_{IC}^k_I = 2.4558$

$R_{IC}(p|\Sigma) = \lim_{k \to \infty} \frac{R_{IC}^k(p|\Sigma)}{\log k} = 0.875$
Ideally, we want to achieve a well-designed database by having the maximum information content for the entire database.

If this is not achievable, we want to maximize the information content for all positions to the possible extent by enforcing some design conditions.

Given a condition $\mathcal{C}$, guaranteed information content ($\text{GIC}(\mathcal{C})$) is the largest number $g \in [0, 1]$ such that for all positions in all instances of all schemas satisfying $\mathcal{C}$, the information content is not smaller than $g$.

Schema satisfying condition $\mathcal{C}$

\[ (S, \Sigma) \]

\[ \Rightarrow \]

instances of $(S, \Sigma)$

\[ \text{RIC}_I(p|\Sigma) \geq g \]
Known results (Arenas&Libkin: PODS’03, Kolahi&Libkin: PODS’06):

- For relational schemas with functional dependencies:
  - BCNF is the only normal form that guarantees well-design databases: $GIC(BCNF) = 1$.
  - A good 3NF normalization guarantees a minimum of $1/2$ information content: $GIC(3NF^+) = 1/2$.

- For XML designs with functional dependencies:
  - XNF is the only normal form that guarantees well-designed XML documents: $GIC(XNF) = 1$.

How do we design XML to achieve high information content for the relational storage?
Storing XML in Relations

Inlining technique: Given a DTD, separate relations are created for the root and elements occurring under a Kleene star.

\[
\begin{align*}
\text{db} & \quad \ast \quad \text{student} \\
\ast & \quad \ast \\
\text{phone} & \quad \text{address} \\
\text{@number} & \quad \text{@streetNo} \quad \text{@city} \quad \text{@postalCode}
\end{align*}
\]

\[
\begin{align*}
\text{student}(stID, \text{name}, \text{conID}) \\
\text{address}(\text{addID}, \text{conID}, \text{postalCode}, \text{streetNo}, \text{city}) \\
\text{phone}(\text{phID}, \text{conID}, \text{number})
\end{align*}
\]
**Inlining technique:** Given a DTD, separate relations are created for the root and elements occurring under a Kleene star.

- **student**
  - @name
  - contact
    - *
      - phone
        - @number
        - @streetNo
        - @city
        - @postalCode
      - address
        - @name
        - *
          - @streetNo
          - @number
          - @city
          - @postalCode

- **student**\((stID, name, conID)\)
- **address**\((addID, conID, postalCode, streetNo, city)\)
- **phone**\((phID, conID, number)\)

\[address[conID]\subseteq_{FK} student[conID]\]
\[phone[conID]\subseteq_{FK} student[conID]\]
Storing XML in Relations

**Inlining technique:** Given a DTD, separate relations are created for the root and elements occurring under a Kleene star.

\[
\text{student}(stID, \text{name}, conID) \\
\text{address}(addID, conID, postalCode, streetNo, city) \\
\text{phone}(phID, conID, number)
\]

\[
\text{address}[\text{conID}] \subseteq_{\text{FK}} \text{student}[\text{conID}] \\
\text{phone}[\text{conID}] \subseteq_{\text{FK}} \text{student}[\text{conID}]
\]

\[
\text{student}, \text{@postalCode} \rightarrow \text{address} \\
\forall \text{ student}(s, n, c) \land \text{address}(a, c, pc, st, apt, ct) \land \text{student}(s, n, c) \land \text{address}(a', c, pc, st', apt', ct') \rightarrow a = a'
\]
## Redundancy-Free Design

<table>
<thead>
<tr>
<th>XML Design</th>
<th>Relational Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>• DTD</td>
<td>• Relational schema</td>
</tr>
<tr>
<td>• XML functional dependencies</td>
<td>• Keys, foreign keys</td>
</tr>
<tr>
<td></td>
<td>• EGDs</td>
</tr>
</tbody>
</table>
Redundancy-Free Design

### XML Design
- DTD
- XML functional dependencies

### Relational Storage
- Relational schema
- Keys, foreign keys
- EGDs

\[
RIC_1(P|\Sigma) = ?
\]
Redundancy-Free Design

XML Design

• DTD
• XML functional dependencies

Relational Storage

• Relational schema
• Keys, foreign keys
• EGDs

Theorem: XML design in XNF $\Leftrightarrow$ Max information content for positions in the relational storage.
Non-XNF Designs

If there are non-XNF functional dependencies,
- we can do XNF normalization, but
- normalizing is not always efficient.

With no restriction on FDs, we can have high redundancy in the relational storage.
- For any $\varepsilon > 0$, the information content can get as close as $\varepsilon$ to zero.

Can we restrict XML functional dependencies to guarantee a reasonable information content?
- like the restriction that 3NF enforces to guarantee 1/2 information content.
Relative vs Absolute Functional Dependencies

Relative functional dependency: holds within each *student* element.

\[ \text{student}, \; @\text{postalCode} \rightarrow \text{address} \]

Absolute functional dependency: holds globally.

\[ @\text{postalCode} \rightarrow @\text{city} \]

We are interested in FDs relative to an element that occurs under a Kleene star in the DTD.
Another Good Design

**Theorem:** If all XML functional dependencies are either XNF or relative, then the information content for all data values in the relational storage is not less than $1/2$. 
Another Good Design

**Theorem:** If all XML functional dependencies are either XNF or relative, then the information content for all data values in the relational storage is not less than $1/2$.

\[ \text{only XNF functional dependencies} \quad \iff \quad RIC_I(p|\Sigma) = 1 \]

\[ \text{XNF or relative functional dependencies} \quad \Rightarrow \quad RIC_I(p|\Sigma) \geq 1/2 \]
Another Relational Storage for XML

We can treat XML documents as edge-labeled graphs and shred them into relations:

\[ \text{Edge}(\text{source}, \text{target}, \text{label}) \quad \text{or} \quad \text{Blabel}(\text{source}, \text{target}) \]
\[ \text{Value}(\text{vid}, \text{val}) \quad \text{or} \quad \text{Value}(\text{vid}, \text{val}) \]
Another Relational Storage for XML

We can treat XML documents as edge-labeled graphs and shred them into relations:

- $Edge(source, target, label)$ or $B_{label}(source, target)$
- $Value(vid, val)$ or $Value(vid, val)$

Our results extend:

- only XNF functional dependencies $\iff RIC_I(p|\Sigma) = 1$
- XNF or relative functional dependencies $\Rightarrow RIC_I(p|\Sigma) \geq 1/2$

But enforcing XML functional dependencies requires arbitrarily many more joins.
Conclusions

Design tips to have a good relational storage:

- try to have an XNF design to ensure a redundancy-free relational storage.
- organize XML elements so that there is no absolute FD to ensure a low-redundancy relational storage.

Comparing inlining and edge relational representations:

- they are equivalent in terms of redundancy.
- edge can be significantly worse when enforcing constraints.

Future work:

- is it always possible to avoid absolute FDs?