Functional Dependencies: Redundancy Analysis and Correcting Violations

Solmaz Kolahi

solmaz@cs.ubc.ca

Postdoctoral Research Fellow
Department of Computer Science
University of British Columbia

Joint work with Leonid Libkin and Laks Lakshmanan
Both relational and XML databases may store redundant data:

### Functional Dependency:

\( \text{title} \rightarrow \text{year} \)

<table>
<thead>
<tr>
<th>title</th>
<th>director</th>
<th>actor</th>
<th>year</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Departed</td>
<td>Scorsese</td>
<td>DiCaprio</td>
<td>2006</td>
</tr>
<tr>
<td>The Departed</td>
<td>Scorsese</td>
<td>Nicholson</td>
<td>2006</td>
</tr>
<tr>
<td>Shrek the Third</td>
<td>Miller</td>
<td>Myers</td>
<td>2007</td>
</tr>
<tr>
<td>Shrek the Third</td>
<td>Miller</td>
<td>Murphy</td>
<td>2007</td>
</tr>
<tr>
<td>Shrek the Third</td>
<td>Hui</td>
<td>Myers</td>
<td>2007</td>
</tr>
<tr>
<td>Shrek the Third</td>
<td>Hui</td>
<td>Murphy</td>
<td>2007</td>
</tr>
</tbody>
</table>

### Functional Dependency:

\( \text{@AreaCode} \rightarrow \text{@City} \)
Motivation

Normalization techniques try to remove redundancies:

- **BCNF** eliminates all redundancies.
  - only key dependencies are allowed.
  - cannot always be achieved without losing dependencies.

\[ R(A, B, C') \quad AB \rightarrow C \quad C \rightarrow B \]

- **3NF** eliminates some redundancies.
  - allows redundancy on prime attributes.
  - preserves dependencies.

- **XNF** eliminates all redundancies w.r.t. XML functional dependencies.
  - only XML keys are allowed: if \( X \rightarrow p.@l \), then \( X \rightarrow p \).
  - introduced by Arenas & Libkin in 2002.
Motivation

Traditional normalization theory
- characterizes a database as redundant or non-redundant.
- does not measure redundancy.
- cannot provide guidelines to reduce redundancy.

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

Our goal is
- to show that there is a spectrum of redundancy using an information-theoretic tool.
- to choose database designs with low redundancy.
- to handle databases with dependency violations.
Outline

- Motivation.
- Reducing redundancy in relational and XML data:
  - Measure of redundancy.
  - Redundancy analysis of normal forms and schemas.
- Correcting functional dependency violations.
- Conclusions.
- Future work.
**Measure of Information Content**

- 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**

- 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.

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

$$\text{Ric}_I(P|\Sigma) = 0.875$$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
Measure of Information Content

- 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.

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

| $\text{Ric}_I(P|\Sigma)$ | $A$ | $B$ | $C$ | $D$
|--------------------------|----|----|----|----
| 0.875                    | 1  | 2  | 3  | 4  |
| 0.781                    | 1  | 2  | 3  | 5  |
|                          | 1  | 2  | 3  | 6  |
Measure of Information Content

- 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.

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

| $\text{Ric}_I(P|\Sigma)$ | $A$ | $B$ | $C$ | $D$ |
|---------------------------|-----|-----|-----|-----|
| 0.875                     | 1   | 2   | 3   | 4   |
| 0.781                     | 1   | 2   | 3   | 5   |
| 0.711                     | 1   | 2   | 3   | 6   |
|                           | 1   | 2   | 3   | 7   |
Measure of Information Content

- 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.

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

| $\text{Ric}_I(P|\Sigma)$ | $A$ | $B$ | $C$ | $D$ |
|--------------------------|-----|-----|-----|-----|
| 0.875                    | 1   | 2   | 3   | 4   |
| 0.781                    | 1   | 2   | 3   | 5   |
| 0.711                    | 1   | 2   | 3   | 6   |
| 0.658                    | 1   | 2   | 3   | 7   |
| 0.658                    | 1   | 2   | 3   | 8   |
Measure of Information Content

- Used to measure the redundancy of a data value in a database instance with respect to a set of constraints.
- Intuitively, $R_{IC}(p|\Sigma)$ measures the relative information content of position $p$ in instance $I$ w.r.t. constraints $\Sigma$.
- Independent of data models and query languages.

\[
\Sigma = \{ A \rightarrow C \} \\
R_{IC}(P|\Sigma) \\
0.875 \\
0.781 \\
0.711 \\
0.658 \\
A \quad B \\
1 \quad 2 \quad 3 \quad 4 \\
\Sigma = \{ A \rightarrow C, \ B \rightarrow C \} \\
R_{IC}(P|\Sigma) \\
0.781 \\
0.629 \\
0.522 \\
0.446 \\
1 \quad 2 \quad 3 \quad 5 \\
1 \quad 2 \quad 3 \quad 6 \\
1 \quad 2 \quad 3 \quad 7 \\
1 \quad 2 \quad 3 \quad 8
\]
Measure of Information Content

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

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

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

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

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

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

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

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

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

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>7</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

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

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

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 \( a \in \{1, \ldots, k\} \).

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

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

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 \( a \in \{1, \ldots, k\} \).

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

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

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

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

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

\[
P(2|X) = \frac{48}{48 + 6 \times 42} = 0.16
\]

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

Conditional entropy : 2.8057

Average over all possible \( X \): \( \text{Ric}_I^k = 2.4558 \)
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 \( X \subseteq \text{Pos}(I) - \{p\} \) compute probability distribution \( P(a|X) \) for every \( a \in \{1, \ldots, k\} \).

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

Conditional entropy : 2.8057
Average over all possible \( X \): \( \text{Ric}^k_I = 2.4558 \)

\[
\text{Ric}_I(p|\Sigma) = \lim_{k \to \infty} \frac{\text{Ric}_I^k(p | \Sigma)}{\log k} = 0.875
\]
A schema $S$ with constraints $\Sigma$ is well-designed if for every instance $I$ of $(S, \Sigma)$ and every position $p$ in $I$, $\text{RlC}_I(p|\Sigma) = 1$.

Known results (Arenas & Libkin, 2003):

- relational databases with FDs: $(S, \Sigma)$ is well-designed iff it is in BCNF.
- XML documents with FDs: $(S, \Sigma)$ is well-designed iff it is in XNF.
A schema $S$ with constraints $\Sigma$ is well-designed if for every instance $I$ of $(S, \Sigma)$ and every position $p$ in $I$ $\mathcal{R}_{i=1}^n(p|\Sigma) = 1$.

Known results (Arenas & Libkin, 2003):

- relational databases with FDs: $(S, \Sigma)$ is well-designed iff it is in BCNF.
- XML documents with FDs: $(S, \Sigma)$ is well-designed iff it is in XNF.

Well-designed databases cannot always be achieved:

- Performance issues.
- Dependency preservation.
A schema $S$ with constraints $\Sigma$ is well-designed if for every instance $I$ of $(S, \Sigma)$ and every position $p$ in $I$ $\text{RIC}_I(p|\Sigma) = 1$.

Known results (Arenas & Libkin, 2003):

- relational databases with FDs: $(S, \Sigma)$ is well-designed iff it is in BCNF.
- XML documents with FDs: $(S, \Sigma)$ is well-designed iff it is in XNF.

Well-designed databases cannot always be achieved:

- Performance issues.
- Dependency preservation.

**General design goal:** maximizing information content to the possible extent by enforcing some design conditions.
Guaranteed Information Content

Given a condition $C$, guaranteed information content (GIC) is the smallest information content found in instances of schemas satisfying $C$.

More formally,

- we look at the set of all possible values for information content

$$\text{POSS}_C(m) = \{ \text{RIC}_I(p | \Sigma) \mid I \text{ is an instance of } (R, \Sigma),$$

$R$ has $m$ attributes,

$(R, \Sigma)$ satisfies $C$, \}

- then $GIC_C(m)$, is the infimum of $\text{POSS}_C(m)$. 

Design goal: minimizing redundancy while preserving FDs.

For a normal form $\mathcal{NF}$, $\text{PRICE}(\mathcal{NF})$ is the minimum information content that $\mathcal{NF}$ loses to guarantee dependency preservation.

- if $c \in [0, 1]$ is the largest information content guaranteed for decompositions into $\mathcal{NF}$,

then price of dependency preservation, $\text{PRICE}(\mathcal{NF})$, is $1 - c$. 

\[
\text{PRICE}(\mathcal{NF}) = 1 - c.
\]
Price of Dependency Preservation

Theorem

\begin{itemize}
\item \text{PRICE}(3\text{NF}) = 1/2.
\item \text{PRICE}(\mathcal{NF}) \geq 1/2 \text{ for any dependency-preserving normal form } \mathcal{NF}.
\end{itemize}

To pay the smallest price for achieving dependency preservation, we should do a 3NF normalization.
Price of Dependency Preservation

Theorem

- \( \text{PRICE}(3\text{NF}) = \frac{1}{2} \).
- \( \text{PRICE}(\mathcal{NF}) \geq \frac{1}{2} \) for any dependency-preserving normal form \( \mathcal{NF} \).

To pay the smallest price for achieving dependency preservation, we should do a 3\text{NF} normalization.

Not all 3\text{NF} normalizations are equal:

- Special subclasses of 3\text{NF} exist (old research).
- Only one subclass (3\text{NF}+) achieves the smallest price.
- We compare normal forms based on guaranteed information content or highest redundancy they allow.
Comparing Normal Forms

**Theorem** For every $m > 2$:

\[
\begin{align*}
\text{GIC}_{\text{All}}(m) &= 2^{1-m} \\
\text{GIC}_{\text{3NF}}(m) &= 2^{2-m} \\
\text{GIC}_{\text{3NF}^+}(m) &= \frac{1}{2}
\end{align*}
\]

- 3NF is twice as good as doing nothing.
- 3NF$^+$ is exponentially better.
- similar results obtained if we compare normal forms based on guaranteed average information content.
Redundancy of an Arbitrary Schema

Normalizing into smaller relations is not always desirable.
- losing constraints.
- slowing down query answering.

Normalization decision can be made based on
- how much redundancy the schema allows; or
- where in the spectrum of redundancy the schema lies; or
- the lowest information content found in all instances of the schema.

Design goal: decomposing the schema with the highest potential for redundancy.
Redundancy of an Arbitrary Schema

**Theorem**  Given an arbitrary schema $R$ with FDs $\Sigma$, let

- $\Sigma_A = \{X | X \rightarrow A, X \text{ is minimal and non-key}\}$;
- $\#HS = \text{the number of hitting sets of } \Sigma_A$;
- $l = |\bigcup_{X \in \Sigma_A} X|$.

Then the smallest information content found in column $A$ of instances is

$$\text{GIC}_\Sigma^R(A) = \#HS \cdot 2^{-l}.$$
Redundancy of an Arbitrary Schema

Theorem  Given an arbitrary schema \( R \) with FDs \( \Sigma \), let

\[
\Sigma_A = \{ X | X \rightarrow A, \ X \text{ is minimal and non-key}\};
\]

\[
\#HS = \text{the number of hitting sets of } \Sigma_A;
\]

\[
l = |\bigcup_{X \in \Sigma_A} X|.
\]

Then the smallest information content found in column \( A \) of instances is

\[
\text{GIC}_\Sigma^R(A) = \#HS \cdot 2^{-l}.
\]

\[
R_1(A, B, C, D, E)
\]

\[
\Sigma_1 = \{ AB \rightarrow E, \ D \rightarrow E \}
\]

\[
R_2(A, B, C, D, E)
\]

\[
\Sigma_2 = \{ BC \rightarrow E, \ AC \rightarrow E, \ BD \rightarrow E \}
\]

Theorem  Given an arbitrary schema $R$ with FDs $\Sigma$, let

- $\Sigma_A = \{ X \mid X \rightarrow A, \ X \text{ is minimal and non-key} \}$;
- $\#HS = \text{ the number of hitting sets of } \Sigma_A$;
- $l = \left| \bigcup_{X \in \Sigma_A} X \right|$.

Then the smallest information content found in column $A$ of instances is

$$GIC^R_\Sigma(A) = \#HS \cdot 2^{-l}. $$

$R_1(A, B, C, D, E) \Sigma_1 = \{ AB \rightarrow E, \ D \rightarrow E \}$

$R_2(A, B, C, D, E) \Sigma_2 = \{ BC \rightarrow E, \ AC \rightarrow E, \ BD \rightarrow E \}$

$$GIC^{R_1}_{\Sigma_1}(E) = \frac{3}{8} \quad GIC^{R_2}_{\Sigma_2}(E) = \frac{1}{2}$$
Outline

- Motivation.
- Reducing redundancy in relational and XML data:
  - Measure of redundancy.
  - Redundancy analysis of normal forms and schemas.
- Correcting functional dependency violations.
- Conclusions.
- Future work.
Functional Dependency Violations

Large databases often tend to violate a set of FDs.
\[ \Sigma = \{ \text{cnt, arCode} \rightarrow \text{reg}, \ \text{cnt, reg} \rightarrow \text{prov} \} \]

An inconsistent database

<table>
<thead>
<tr>
<th>name</th>
<th>cnt</th>
<th>prov</th>
<th>reg</th>
<th>arCode</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_1 )</td>
<td>Smith</td>
<td>CAN</td>
<td>BC</td>
<td>Van</td>
<td>604</td>
</tr>
<tr>
<td>( t_2 )</td>
<td>Adams</td>
<td>CAN</td>
<td>BC</td>
<td>Van</td>
<td>604</td>
</tr>
<tr>
<td>( t_3 )</td>
<td>Simpson</td>
<td>CAN</td>
<td>BC</td>
<td>Man</td>
<td>604</td>
</tr>
<tr>
<td>( t_4 )</td>
<td>Rice</td>
<td>CAN</td>
<td>AB</td>
<td>Vic</td>
<td>604</td>
</tr>
</tbody>
</table>
Large databases often tend to violate a set of FDs.

\[
\Sigma = \{ \text{cnt, arCode} \rightarrow \text{reg}, \ \text{cnt, reg} \rightarrow \text{prov} \} 
\]

### An inconsistent database

<table>
<thead>
<tr>
<th>name</th>
<th>cnt</th>
<th>prov</th>
<th>reg</th>
<th>arCode</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>CAN</td>
<td>BC</td>
<td>Van</td>
<td>604</td>
<td>123 4567</td>
</tr>
<tr>
<td>Adams</td>
<td>CAN</td>
<td>BC</td>
<td>Van</td>
<td>604</td>
<td>765 4321</td>
</tr>
<tr>
<td>Simpson</td>
<td>CAN</td>
<td>BC</td>
<td>Man</td>
<td>604</td>
<td>345 6789</td>
</tr>
<tr>
<td>Rice</td>
<td>CAN</td>
<td>AB</td>
<td>Vic</td>
<td>604</td>
<td>987 6543</td>
</tr>
</tbody>
</table>

### A minimal repair

<table>
<thead>
<tr>
<th>name</th>
<th>cnt</th>
<th>prov</th>
<th>reg</th>
<th>arCode</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>CAN</td>
<td>BC</td>
<td>Van</td>
<td>604</td>
<td>123 4567</td>
</tr>
<tr>
<td>Adams</td>
<td>CAN</td>
<td>BC</td>
<td>Van</td>
<td>604</td>
<td>765 4321</td>
</tr>
<tr>
<td>Simpson</td>
<td>CAN</td>
<td>BC</td>
<td>Van</td>
<td>604</td>
<td>345 6789</td>
</tr>
<tr>
<td>Rice</td>
<td>CAN</td>
<td>AB</td>
<td>Vic</td>
<td>604</td>
<td>987 6543</td>
</tr>
</tbody>
</table>
Handling Inconsistent Databases

Integrity constraints $\Sigma$ (FDs, keys, etc.).
Inconsistent database $D$: does not satisfy $\Sigma$.
We can produce a repair $R$ by inserting/deleting tuples or modifying values in $D$.

$$\Delta(D, R) = \text{number of modifications}$$
Handling Inconsistent Databases

Integrity constraints $\Sigma$ (FDs, keys, etc.).

Inconsistent database $D$: does not satisfy $\Sigma$.

We can produce a repair $R$ by inserting/deleting tuples or modifying values in $D$.

$$\Delta(D, R) = \text{number of modifications}$$

Handling inconsistency:

- Consistent query answering:
  
  certain answer for query $Q = \bigcap \{Q(R) \mid R \text{ is a minimal repair for } D\}$

- Producing an optimum repair $R_{opt}$ with minimum $\Delta$.

Both approaches are intractable in general.
Handling Inconsistent Databases

Integrity constraints $\Sigma$ (FDs, keys, etc.).
Inconsistent database $D$: does not satisfy $\Sigma$.
We can produce a repair $R$ by inserting/deleting tuples or modifying values in $D$.

$$\Delta(D, R) = \text{number of modifications}$$

Handling inconsistency:

- Consistent query answering:
  certain answer for query $Q = \bigcap \{ Q(R) \mid R \text{ is a minimal repair for } D \}$
- Producing an optimum repair $R_{\text{opt}}$ with minimum $\Delta$.

Both approaches are intractable in general.

Our approach: producing an approximate solution $R_{\text{app}}$ for optimum repair.

$$\Delta(D, R_{\text{app}}) \leq \alpha \cdot \Delta(D, R_{\text{opt}})$$
Approximating Optimum Repair

**Theorem.** Finding an optimum solution for FD violations is NP-hard.

**Theorem.** Finding a constant-factor approximation for all FD violations is NP-hard.

**Theorem.** For every fixed set of FDs, there is a polynomial-time algorithm that approximates optimum repair within a factor of $\alpha$, where $\alpha$ depends on FDs.

$$
\Sigma = \{A \rightarrow C, B \rightarrow C, CD \rightarrow E\}
$$
Conclusions

- We analyze schemas and normal forms based on worst cases of redundancy.
- There is a spectrum of information content (redundancy) for schemas.
- Producing optimum repair for FD violations is hard.
- We introduced an approximation framework.
Future Work

- Comparing quality of schemas with low / high information content in practice.

- Defining normalization concepts for XML such as:
  - dependency preserving decomposition.

- Finding an equivalent of 3NF for XML as a normal form
  - that guarantees an information content of $\frac{1}{2}$.
  - to which every XML document is decomposable.

- Extending the repair algorithm for other integrity constraints.