An SQL query walks into a bar ...
Reasoning about Domain Semantics over Relations, Bags, Partial Relations, and Partial Bags

Sebastian Link

Department of Computer Science
The University of Auckland
New Zealand

This research is supported by the Marsden Fund Council from Government fundings, administered by the Royal Society of New Zealand.
Main Contributions

- Fundamentals for reasoning about data semantics in
  - bags (no null markers allowed, but duplicate rows),
  - partial relations (no duplicate rows, but null markers allowed), and
  - partial bags (both duplicate rows and null markers allowed)

- Consider expressive combined class of
  - uniqueness constraints
  - functional dependencies
  - multivalued dependencies
  - `NOT NULL` constraints

- Closes gap between theory over relations and SQL practice
  - Establish axiomatizations for each data structure considered
  - Establish almost-linear time algorithms to decide implication
  - Illustrate various application areas
Running Example - Informal

- suppliers deliver articles from a location at a certain cost:

```sql
CREATE TABLE SUPPLIES (
    Article CHAR[20],
    Supplier VARCHAR NOT NULL,
    Location VARCHAR NOT NULL,
    Cost CHAR[8]);
```

- DBMS explicitly enforces additional business rules:
  - for every article there is at most one supplier,
  - cost determined by article and location, and
  - location sets determined by supplier (independent of article & cost)

- are the following business rules already enforced implicitly?
  - combinations of article and cost values are unique
  - cost is determined by article
  - location sets determined by article (independent of supplier & cost)
Running Example - Formal

- db schema \( S = (S, nfs(S)) \):
  \[
  \text{SUPPLIES}=\{ \text{A(rticle), S(upplier), L(ocation), C(ost)} \}
  \]

- no-information null value \( \ni \) (distinguished value of each domain)
  \( \ni \rightarrow \) value may not exist, or value exists and is unknown [Zaniolo]

- null-free subschema NFS \( nfs(S) \subseteq S \) over \( S \) [Atzeni/Morfuni]:
  \[
  nfs(\text{SUPPLIES})=\{ \text{Supplier, Location} \}
  \]
  \( \ni \rightarrow \) db satisfies \( nfs(S) \) iff every row \( r \in db \) is \( nfs(S) \)-total
  \( \ni \rightarrow \) row \( r \) is \( nfs(S) \)-total iff \( \forall H \in nfs(S)(r(H) \neq \ni) \)

- data dependencies enforce business rules:
  \( \ni \rightarrow \) FDs: \( A \rightarrow S \) and \( AL \rightarrow C \)
  \( \ni \rightarrow \) MVD: \( S \rightarrow L \)
Data Dependencies in Presence of No-information Nulls

- database $db$ satisfies UC $u(X)$ over $S$:
  - if for all rows $r_1, r_2 \in db$ we have:
    
    if $r_1(X) = r_2(X)$ and $r_1, r_2$ are $X$-total, then $r_1 = r_2$

- database $db$ satisfies FD $X \rightarrow Y$ over $S$ [Lien, Atzeni/Morfuni]:
  - if for all rows $r_1, r_2 \in db$ we have:
    
    if $r_1(X) = r_2(X)$ and $r_1, r_2$ are $X$-total, then $r_1(Y) = r_2(Y)$

- database $db$ satisfies MVD $X \rightarrow Y$ over $S = (S, nfs(S))$ [Lien]:
  - if for all rows $r_1, r_2 \in db$ we have:
    
    if $r_1(X) = r_2(X)$ and $r_1, r_2$ are $X$-total, then there is some row $r$ in $db$ such that $r(XY) = r_1(XY)$ and $r(X(S - Y)) = r_2(X(S - Y))$
Data Structures

- consider databases \( db \) over \( S = (S, nfs(S)) \)
- partial bags (duplicate rows and null marker occurrences):
  \( \leadsto db \) can be any multiset of \( nfs(S) \)-total rows
- bags (duplicate rows, no null marker occurrences):
  \( \leadsto db \) can be any multiset of \( S \)-total rows
- partial relations (no subsumed rows, null marker occurrences):
  \( \leadsto db \) can be any subsumption-free relation of \( nfs(S) \)-total rows
  \( \leadsto \) i.e., there are no distinct \( r_1, r_2 \in db \) such that \( r_1 \) subsumes \( r_2 \)
  \( \leadsto r_1 \) subsumes \( r_2 \) iff for all \( H \in S: r_2(H) = \text{ni} \) or \( r_1(H) = r_2(H) \)
- relations (no duplicate rows, no null marker occurrences):
  \( \leadsto db \) can be any relation of \( S \)-total rows
- motivation for partial relations are lossless decompositions:
  \( \leadsto db_X[S] = db_X[XY] \bowtie db_X[X(S - Y)], \) if \( db \) satisfies \( X \rightarrow Y \)
  \( \leadsto db_X[S] = db_X[XY] \bowtie db_X[X(S - Y)] \) iff \( db \) satisfies \( X \rightarrow Y \)
The Implication Problem

- **input:**
  \[ S = (S, nfs(S)) \text{ and } \]
  dependency set \( \Sigma \cup \{ \varphi \} \) over \( S \)

- **output:**
  \[
  \begin{cases} 
  \text{yes, if } \Sigma \models nfs(S) \varphi \\
  \text{no , if } \Sigma \not\models nfs(S) \varphi 
  \end{cases}
  \]

- \( \Sigma \models nfs(S) \varphi \) holds, if
  \[
  \text{every database } db \text{ over } S \text{ that satisfies } \Sigma \text{ also satisfies } \varphi
  \]

- clearly, the underlying data structures matter!
  \[
  \text{in classical textbooks and literature only relations considered} \]
  \[
  \text{in database practice: partial bags are SQL instances in general} \]
  \[
  \text{if a primary key is defined, then we have partial relations}
  \]
Examples

- **impact of duplicates:**
  \[ S = (ASLC, SL), \Sigma = \{ A \rightarrow S; AL \rightarrow C; S \rightarrow L \} \]
  For partial relations: \( \Sigma \models_{SL} u(AL) \)
  In each case: \( \Sigma \models_{SL} A \rightarrow C \) and \( \Sigma \not\models_{SL} A \rightarrow L \)

<table>
<thead>
<tr>
<th>Article</th>
<th>Supplier</th>
<th>Location</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sugar</td>
<td>SweetAz Ltd</td>
<td>Reunion</td>
<td>10</td>
</tr>
<tr>
<td>Sugar</td>
<td>SweetAz Ltd</td>
<td>Reunion</td>
<td>10</td>
</tr>
</tbody>
</table>

- **impact of null marker occurrences (null-free subschema):**
  \[ S = (ASLC, ALC'), \Sigma = \{ A \rightarrow S; AL \rightarrow C; S \rightarrow L \} \]
  For partial relations:
  \( \Sigma \models_{ALC} u(AL) \), \( \Sigma \not\models_{ALC} A \rightarrow C \) and \( \Sigma \not\models_{ALC} A \rightarrow L \)

<table>
<thead>
<tr>
<th>Article</th>
<th>Supplier</th>
<th>Location</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sugar</td>
<td>ni</td>
<td>Reunion</td>
<td>10</td>
</tr>
<tr>
<td>Sugar</td>
<td>ni</td>
<td>Rio</td>
<td>5</td>
</tr>
</tbody>
</table>
Axiomatization $\mathcal{R}$ over Relation Schemata $(S, S)$

- $XY \rightarrow Y$ (reflexivity)
- $X \rightarrow Y$  
  $X \rightarrow S - Y$ (S-complementation)
- $X \rightarrow Y$  
  $X \rightarrow Y \rightarrow Z$ (MVD implication)
- $X \rightarrow Y$  
  $X \rightarrow Z - Y$ (MVD implication)

- $XU \rightarrow YV$ $V \subseteq U$ (FD augmentation)
- $X \rightarrow Y$  
  $X \rightarrow Z$ (transitivity)

- $XU \rightarrow YV$ $V \subseteq U$ (MVD augmentation)
- $X \rightarrow Y$  
  $Y \rightarrow Z$ (pseudo-transitivity)
- $X \rightarrow Y$  
  $Y \rightarrow Z$ (pseudo-transitivity)
Axiomatization $\mathfrak{B}$ over Bag Schema $(S, S)$

\[
\begin{align*}
\frac{XY \rightarrow Y}{(\text{reflexivity})} & \quad \frac{X \rightarrow Y}{XU \rightarrow YV}^{V \subseteq U} & \quad \frac{X \rightarrow Y}{X \rightarrow Z}^{(\text{transitivity})} \\
\frac{X \rightarrow Y}{X \rightarrow S - Y}^{(S\text{-complementation})} & \quad \frac{X \rightarrow Y}{XU \rightarrow YV}^{V \subseteq U} & \quad \frac{X \rightarrow Y}{X \rightarrow Z - Y}^{(\text{pseudo-transitivity})} \\
\frac{u(X)}{X \rightarrow Y}^{(\text{FD implication})} & \quad \frac{X \rightarrow Y}{u(Y)}^{u(X)} & \quad \frac{X \rightarrow Y}{u(X)}^{(\text{pullback})} \\
\frac{X \rightarrow Y}{X \rightarrow Y}^{(\text{MVD implication})} & \quad \frac{X \rightarrow Y}{Y \rightarrow Z} & \quad \frac{X \rightarrow Y}{X \rightarrow Z - Y}^{(\text{mixed pseudo-transitivity})}
\end{align*}
\]
Axiomatization $pR$ over Partial Relation Schema $(S, S')$

\[
\begin{align*}
X \rightarrow Y & \quad X \rightarrow YZ \\
(\text{reflexivity}) & \quad (\text{decomposition})
\end{align*}
\]

\[
\begin{align*}
X \rightarrow Y & \quad X \rightarrow Z \\
X \rightarrow S - Y & \quad X \rightarrow YZ \\
(\text{S-complementation}) & \quad (\text{MVD union})
\end{align*}
\]

\[
\begin{align*}
X \rightarrow W & \quad Y \rightarrow Z \\
X \rightarrow Y & \quad X \rightarrow Z - W \\
Y \subseteq X(W \cap S') & \quad Y \subseteq X(W \cap S') \\
(\text{null pseudo-transitivity}) & \quad (\text{null mixed pseudo-transitivity})
\end{align*}
\]
**Axiomatization $\mathcal{P}$ over Partial Bag Schema $(S, S')$**

- Reflexivity: $XY \rightarrow Y$
- Decomposition: $X \rightarrow Y Z$
- FD union: $X \rightarrow Y Z$
- MVD union: $X \rightarrow Y Z$
- Null pseudo-transitivity: $Y \subseteq X (W \cap S')$

- S-complementation: $X \rightarrow Y \rightarrow S - Y$
- MVD implication: $X \rightarrow W \rightarrow Y \rightarrow Z$

- FD implication: $u(X)$
- Null pullback: $u(Y)$
- Null mixed pseudo-transitivity: $Y \subseteq X (W \cap S')$
Efficiently Deciding the Implication Problem

$\varphi$ denotes either $u(X)$, $X \to Y$, or $X \to Y$ over partial bag schema $S = (S, S')$

The problem whether $\varphi$ is implied by a set $\Sigma$ of UCs, FDs and MVDs over $S$ can be decided in $O(||\Sigma|| + \min\{k_\Sigma[FM][XS'], \log \bar{p}_\Sigma[FM][XS']\} \times ||\Sigma[FM][XS']||)$ time.

$\Sigma[FM] = \{X \to S \mid u(X) \in \Sigma\} \cup \{X \to Y \mid X \to Y \in \Sigma\} \cup \{X \to Y \mid X \to Y \in \Sigma\}$

$||\varphi||$: total number of attributes occurring in $\varphi$

$||\Sigma||$: sum of $||\sigma||$ over all $\sigma \in \Sigma$

$k_\Sigma$: the number of MVDs in $\Sigma$

$\bar{p}_\Sigma$: the number of sets in $DepB_\Sigma(X)$ that have non-empty intersection with $X$

$\Sigma[XS']$: the set of elements in $\Sigma$ where the left-hand side is a subset of $XS'$

for bag schemata:

the bound becomes $O(||\Sigma|| + \min\{k_\Sigma[FM], \log \bar{p}_\Sigma[FM]\} \times ||\Sigma[FM]||)$ time
**Application 1:**
Decompositions for Efficient Processing of Updates

\[ (R = ASLC, R_S), \Sigma = \{A \rightarrow S; AL \rightarrow C; S \rightarrow L\} : \]

- decompose with \( S \rightarrow L \) first, then decompose \( ASC \) with \( A \rightarrow S \):
  \[ \rightarrow R_1 = SL; R_2 = AS \text{ and } R_3 = AC \]

- properties of decomposition:
  \[ R_S = ASL: \text{ both lossless and faithful} \]
  \[ \Sigma \models_{ASL} A \rightarrow C \]

  \[ R_S = AS: \text{ lossless, not faithful} \]
  \[ \Sigma \not\models_{AS} A \rightarrow C, \ AL \rightarrow C \text{ lost} \]

  \[ R_S = SL: \text{ not lossless, faithful} \]
  \[ \Sigma \models_{SL} A \rightarrow C \]

  \[ R_S = S: \text{ neither lossless nor faithful} \]
Application 2:
Rewriting for Efficient Processing of Queries

\[(R = ASLC, R_s = SL), \Sigma = \{A \rightarrow S; AL \rightarrow C; S \rightarrow L\}\):

- retrieve locations and costs associated with same articles:

\[
\begin{align*}
\text{SELECT } R.L, R'.C \\
\text{FROM } R, R \text{ AS } R' \\
\text{WHERE } R.A = R'.A
\end{align*}
\]

- since \(\Sigma \models_{SL} A \rightarrow C\) we can rewrite into

\[
\begin{align*}
\text{SELECT } R.L, R.C \text{ FROM } R
\end{align*}
\]
**Application 3: Inference Attacks**

- attacker may *infer* secrets without violating access control policies
- \( R = ASLC, \text{nfs}(R) = SL, \Sigma = \{ A \rightarrow S; AL \rightarrow C; S \rightarrow L \} : \)
- some users prohibited access to
  \[
  \Psi = (\exists X_S)R(\text{Sugar, } X_S, \text{Reunion, } 10)
  \]

- attacker asks the following queries without violating access policy \( \Psi : \)
  \[
  \Phi_1 = (\exists X_S)(\exists X_L)R(\text{Sugar, } X_S, X_L, 10)
  \]
  \[
  \Phi_2 = (\exists X_S)(\exists X_C)R(\text{Sugar, } X_S, \text{Reunion, } X_C)
  \]

- attacker exploits fact that \( \Sigma \models_{SL} A \rightarrow L \) holds:
  - applying \( A \rightarrow L \) to \( \Phi_1 \) and \( \Phi_2 \) reveal \( \Phi \)
  - if \( S \notin \text{nfs}(R) \), attacker cannot draw that conclusion

- understanding entailment assists in preventing inference attacks
Conclusion

- SQL tables permit occurrences of duplicate and partial information by specification of uniqueness and NOT NULL constraints.
- Combination of these restricts database instances to relations, partial relations, bags, or partial bags.
- Established dedicated tools for reasoning about uniqueness and NOT NULL constraints combined with functional and multivalued dependencies.
- Close the gap between theory of relations and SQL practice, provide insight not currently found in literature and textbooks, and have proven applications in schema design and data processing.
Research Agenda

- investigate other null markers, at least:
  - $\rightarrow$ not applicable
  - $\rightarrow$ does not exist
  - $\leftrightarrow$ exists, but unknown (theory valid under possible world semantics)

- study normalization and its justification

- automate semantic query optimization with constraints

- investigate the properties of *Armstrong* tables

- investigate implications for database security