Incomplete information, monotonicity and homomorphism preservation

Amélie Gheerbrant (Univ. Paris Diderot)  Leonid Libkin (Univ. of Edinburgh)  Cristina Sirangelo (LSV, ENS Cachan)
Incomplete information and query answering

- **Incomplete information in data**: missing / unknown / partially specified data
  - several possible “completions”
- Still one of the most poorly understood aspects of data management
- **Query answering**
  - over usual databases: model checking \( D \models Q \)
  - over incomplete databases: entailment \( R \models Q \) for all completions \( R \) of \( D \)

When can entailment be solved by (straightforward) model checking?

In a database perspective:

When can we answer queries correctly on incomplete databases by using classical query evaluation engines?
Model of incompleteness

**Employee**

<table>
<thead>
<tr>
<th></th>
<th>Smith</th>
<th>Brown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brown</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Manager**

<table>
<thead>
<tr>
<th></th>
<th>Smith</th>
<th>Brown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>$x_1$</td>
<td></td>
</tr>
<tr>
<td>$x_1$</td>
<td></td>
<td>Brown</td>
</tr>
<tr>
<td>Brown</td>
<td></td>
<td>$x_2$</td>
</tr>
</tbody>
</table>

- $\text{Const}$: a countably infinite set of constants
- $\text{Nulls}$: a countably infinite set of variables ranging over $\text{Const}$ (marked nulls)
- $\sigma$: a finite relational signature

Incomplete database over $\sigma$ (naïve table) [Imielinski, Lipski 84]:

a finite structure of signature $\sigma$ with domain $\subset \text{Const} \cup \text{Nulls}$

- variables model *unknown* data values
Model of incompleteness

Semantics of incompleteness:
\[ [D] = \text{a set of complete } \sigma\text{-structures} \]

- **Closed world assumption**
  \[ [D]_{CWA} = \{ R \text{ over } Const \mid R = v(D) \text{ for some valuation } v: \text{Nulls} \to \text{Const} \} \]

- **Open world assumption**
  \[ [D]_{OWA} = \{ R \text{ over } Const \mid R \supseteq v(D) \text{ for some valuation } v: \text{Nulls} \to \text{Const} \} \]

- **Weak Closed World assumption** [Reiter 77]
  \[ [D]_{WCWA} = \{ R \text{ over } Const \mid R \supseteq v(D), \text{dom}(R) = \text{dom}(v(D)) \text{ for } v: \text{Nulls} \to \text{Const} \} \]
Query answering over incomplete databases

For a Boolean query $Q$ and an incomplete database $D$

- Query answering semantics (entailment): testing whether $R \models Q$ for all $R \in \llbracket D \rrbracket$
  
  (*certain answers*, in database terminology)

- Usual query answering in db systems (model checking): testing whether $D \models Q$
  
  (*naive evaluation*)

- Model-checking solves entailment for $Q$:

  \[
  \forall D \quad D \models Q \iff \forall R \in \llbracket D \rrbracket \quad R \models Q
  \]

  (*naive evaluation works for $Q$*)

  - correct query answering semantics (entailment), classical query evaluation algorithms (model-checking)
  - clearly not always possible (*undecidable vs. PTIME for FO*)
A concrete example

“All employees are managers”  \( Q = \forall x ( \text{Employee}(x) \rightarrow \exists y \text{Manager}(x, y) ) \)

<table>
<thead>
<tr>
<th>Employee</th>
<th>Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>Smith</td>
</tr>
<tr>
<td>( x_1 )</td>
<td>( x_1 )</td>
</tr>
<tr>
<td>Brown</td>
<td>Brown</td>
</tr>
<tr>
<td>Brown</td>
<td>( x_2 )</td>
</tr>
</tbody>
</table>

\( D \models Q \)

\( \llbracket D \rrbracket_{\text{OWA}} \)

<table>
<thead>
<tr>
<th>Smith</th>
<th>White</th>
</tr>
</thead>
<tbody>
<tr>
<td>White</td>
<td>Brown</td>
</tr>
<tr>
<td>Brown</td>
<td>Black</td>
</tr>
</tbody>
</table>

\( R \not\models Q \)

- Under **OWA**  \( \exists R \in \llbracket D \rrbracket \text{ s.t } R \not\models Q: \) naïve evaluation does not work for \( Q \)
- Under **CWA**  \( \forall R \in \llbracket D \rrbracket \text{ s.t } R \not\models Q: \) naïve evaluation works for \( Q \) over \( D \)

What makes naïve evaluation work?
What makes naïve evaluation work?

What we already know:

Over incomplete relational databases (naïve tables), under the OWA, if \( Q \) is Boolean \( \text{FO} \) query:

\[
\text{Naïve evaluation works for } Q \ \uparrow
\]

\( Q \) is an \( \exists \text{Pos} \) query

\( \exists \text{Pos} : \ \exists, \wedge, \vee \) fragment of FO

(Unions of Conjunctive Queries in database terminology)

- The \( \uparrow \) direction \([\text{Imielinski, Lipski 84}]\)

- The \( \downarrow \) direction \([\text{Libkin 2011}] \) relies on Rossman’s homomorphism preservation theorem in the finite
Relating naïve evaluation and syntactic fragments

A unified framework for relating naïve evaluation and syntactic fragments for several possible semantics:

1. Naïve evaluation works for \( Q \) under \( [] \)
2. \( Q \) is “monotone” w.r.t. \( [] \)
3. \( Q \) is preserved under a class of homomorphisms
4. Preservation theorems
5. \( Q \) belongs to a syntactic fragment
Relating naïve evaluation and syntactic fragments

A unified framework for relating naïve evaluation and syntactic fragments for several possible semantics:

Naïve evaluation works for Q under $\square$

$Q$ is “monotone” w.r.t. $\square$

$Q$ is preserved under a class of homomorphisms

Preservation theorems:

- Usually proved over arbitrary structures (both finite and infinite)
- some fail in the finite
- the direction Syntax $\Rightarrow$ Preservation always holds in the finite as well

Preservation theorems (even over arbitrary structures) can give us relevant classes of queries where naïve evaluation works.
Naïve evaluation and syntactic fragments

Three well known semantics as instances of our framework

Naïve evaluation works for $Q$ under $[]$

$Q$ is “monotone” w.r.t. $[]$

$Q$ is preserved under a class of homomorphisms

Preservation theorems

$Q$ belongs to a syntactic fragment

Naïve evaluation works under:

OWA

Preservation under homomorphism

∃Pos

[Rossman]

WCWA

Preservation under onto homomorphism

Pos

[Lyndon]

CWA

Preservation under “strong onto” homomorphism

Pos+∀G
Naïve evaluation and syntactic fragments

The framework is much more general

Holds in a very general setting:
- arbitrary domain of database objects equipped with a semantic function
- do not even need to be relational databases
- very mild assumptions on the domain ("saturated subdomain")

Holds for a whole class of relational semantics:
- semantics described by relations on structures
- general notion of homomorphism based on these relations
Naïve evaluation and syntactic fragments

Beyond OWA, CWA and WCWA:

Naïve evaluation works for Q under $\exists Pos^{+} \forall G^{bool}$

Q is “monotone” w.r.t. $\exists Pos^{+} \forall G^{bool}$

Q is preserved under a class of homomorphisms

Preservation theorems

Q belongs to a syntactic fragment

Naïve evaluation works under:

Powerset semantics

Preservation under unions of strong onto homomorphisms

Minimal semantics

Preservation under unions of minimal homomorphisms

over cores
Reference

Details in our paper:

“When is Naive Evaluation Possible?” PODS 2013
by Amélie Gheerbrant, Leonid Libkin and Cristina Sirangelo
Conclusions and future work

- A general framework for relating naïve evaluation and syntactic fragments
  - applied to (generalizations of) existing relational semantics
- All results extend to non-boolean relational queries
- Extend to other data models
  - more complex form of relational incompleteness (e.g. conditional tables), incomplete trees, incomplete graphs
- Preservation theorems
  - new notions of preservation, candidate fragments, preservation theorems in the infinite?
  - do they hold in the finite?
- Extend to other languages: fixed-point, fragments of SO, etc.
- Naïve evaluation over restricted instances/ in the presence of constraints
Syntactic fragments

- **Pos**: FO without negation (but with \( \forall \))
  - Pos = FO queries preserved under onto homomorphisms over arbitrary structures (Lyndon positivity theorem)

- **Pos+\(\forall\)G**: Positive fragment with Universal Guards
  \[ \varphi ::= \top \mid \perp \mid R(\bar{x}) \mid x = y \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \exists x \varphi \mid \forall x \varphi \mid \left( \forall \bar{x} \ (G(\bar{x}) \rightarrow \varphi) \right) \]
  - with
  - \( G \): a relation or equality symbol
  - \( \bar{x} \): a tuple of distinct variables
  - preserved under strong onto homomorphisms, a good syntax
  - extends [Keisler ‘65] (complex syntactic restrictions, one binary relation only)
The most general setting: database domains

Database domain: a quadruple \( \langle D, C, [], \approx \rangle \)

<table>
<thead>
<tr>
<th>( D ) : a set</th>
<th>description</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>database objects (complete and incomplete)</td>
<td>all naïve relational instances over a fixed schema ( \sigma )</td>
<td></td>
</tr>
</tbody>
</table>

| \( C \) : a subset of \( D \) | complete database objects | all complete relational instances over \( \sigma \) |

| \( [] : D \to 2^C \) | semantics of incompleteness | \([\ ]_{\text{OWA}}, [\ ]_{\text{CWA}}, \text{etc.} \) |

| \( \approx \) : an equivalence relation on \( D \) | equivalence of objects (w.r.t. queries) | isomorphism of relational instances |
The most general setting: database domains

Over \( \langle D, C, \lbrack \rceil \rceil, \approx \rangle \)

- **Boolean query**: \( Q : D \rightarrow \{true, false\} \)

  - **Q generic**: \( x \approx y \) implies \( Q(x) = Q(y) \)

  - **Q monotone w.r.t. \( \lbrack \rceil \rceil \)**: \( y \in \lbrack x \rceil \) implies \( Q(x) \Rightarrow Q(y) \)

- **Certain answers** for \( x \in D \):

  \[ cert(Q, x) = \land_{c \in \lbrack x \rceil} Q(c) \]

- **Naïve evaluation** works for \( Q \):

  \[ \text{For all } x \in D \quad Q(x) = cert(Q, x) \]
Naïve evaluation and monotonicity

Naïve evaluation works for $Q$ under $\llbracket \cdot \rrbracket$

$Q$ is “monotone” w.r.t. $\llbracket \cdot \rrbracket$

For all $x \in \mathcal{D}$ there exists $y \in \llbracket x \rrbracket$ such that $y \approx x$

Saturation property for $\langle \mathcal{D}, C, \llbracket \cdot \rrbracket, \approx \rangle$:

$Q$ belongs to a syntactic fragment

Preservation theorems

$Q$ is preserved under a class of homomorphisms

Proposition

Over a saturated database domain, if $Q$ is a generic Boolean query:

Naïve evaluation works for $Q$ iff $Q$ is monotone w.r.t. $\llbracket \cdot \rrbracket$
Homomorphisms

Homomorphism \( D \rightarrow D' \):

a mapping \( h: \text{dom}(D) \rightarrow \text{dom}(D') \) s.t.
\( h(D) \subseteq D' \)

Onto homomorphism \( D \rightarrow D' \):

a homomorphism \( h: D \rightarrow D' \) s.t.
\( h(\text{dom}(D)) = \text{dom}(D') \)

Strong onto homomorphism \( D \rightarrow D' \):

a homomorphism \( h: D \rightarrow D' \) s.t.
\( h(D) = D' \)
Homomorphisms

- **Union of strong onto homomorphisms** \( D \to D' : \bigcup_i h_i(D) = D' \)

- **D-minimal homomorphism** \( h \) on \( D \):
  - there exists no \( h' \), preserving all constants preserved by \( h \), s.t. \( h'(D) \subsetneq h(D) \)

- **Union of minimal homomorphisms** \( D \to D' : \bigcup_i h_i(D) = D' \)
  - with \( h_1 \ldots h_n \) \( D \)-minimal and preserving the same constants
• Extend (and generalize) an ordering-based semantics of Codd databases
• Naïve evaluation ↔ Monotonicity ↔ Preservation continues to hold
• Under the powerset CWA the needed notion is preservation under
  unions of strong onto homomorphisms (i.e. homomorphisms \( D \to \bigcup_{i=1}^{n} h_i(D) \))
• An FO fragment preserved under this relationship: \( \exists \text{Pos+} \forall G \text{bool} \)

Corollary
Naïve evaluation works for \( \exists \text{Pos+} \forall G \text{bool} \) Boolean queries under \( (\cdot)_{\text{CWA}} \)
Minimal semantics

- A special form of powerset semantics, finds its roots in [Minker ’82]
- Later modified and adopted as data exchange semantics (GCWA* [Hernich’11])
- We define it here for arbitrary incomplete instances:

A valuation $v$ on $D$ is D-minimal if there is no valuation $v'$ s.t. $v'(D) \subsetneq v(D)$

Minimal Powerset CWA

$D' \in \langle D \rangle_{\text{CWA}}^{\text{min}}$ iff

$\exists$ D-minimal valuations $v_1, ... v_n$

$D' = \bigcup_{i} v_i(D)$

- Under the minimal powerset CWA the saturation property does not hold
- Cores come to the rescue: naive evaluation recovered over cores
Non-Boolean queries

All results can be lifted to non-boolean relational queries.

- unified technique: reduction to the boolean case

<table>
<thead>
<tr>
<th>Semantics</th>
<th>Naïve evaluation works for</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWA</td>
<td>$\exists$Pos</td>
</tr>
<tr>
<td>WCWA</td>
<td>Pos</td>
</tr>
<tr>
<td>CWA</td>
<td>$\text{Pos+}\forall G$</td>
</tr>
<tr>
<td>Powerset CWA</td>
<td>$\exists\text{Pos+}\forall G^{\text{bool}}$</td>
</tr>
<tr>
<td>Min Powerset CWA</td>
<td>$\exists\text{Pos+}\forall G^{\text{bool}}$ iff $Q(D)=Q(\text{core}(D))$</td>
</tr>
</tbody>
</table>