Consistent Query Answering

Sławek Staworko [1]

University of Lille
INRIA Mostrare Project

DEIS 2010
November 9, 2010

[1] Some slides are due to [Cho07]

Overview

Motivation

Traditional Databases

Database instance D:

- a finite first-order structure
- represents the information about the world

Integrity constraints Σ

- first-order logic formulas
- express the properties/rules of the world

Consistent database

- Formula satisfaction in a first-order structure $D \models \Sigma$
- RDBMS ensures consistency

Example

Muppet

Name	Role	DoB
Kermit	Manager	14.03.1965
Miss Piggy	Diva	21.06.1976
T. Statler	Old Man	12.04.1946

Example

Muppet (CBS)

Name	Role	DoB
Kermit	Manager	14.03.1965
Miss Piggy	Diva	21.06.1976
T. Statler	Old Man	12.04.1946

Muppet (Vanity Fair)

Name	Role	DoB
Kermit	Manager	14.03.1965
Miss Piggy	Diva	01.04.1936
T. Statler	Old Man	18.06.1942

Muppet (Federated Database)

Name	Role	DoB
Kermit	Manager	14.03.1965
Miss Piggy	Diva	21.06.1976
Miss Piggy	Diva	01.04.1950
T. Statler	Old Man	12.04.1946
T. Statler	Old Man	18.06.1942

Inconsistency

Source of Inconsistency

- integration of independent data sources with overlapping data
- time lag of updates (eventual consistency)
- unenforced integrity constraints (denormalized DBs)

Eliminating inconsistency?

- not enough information, time, or money
- difficult, impossible or undesirable
- unnecessary: queries may be insensitive to inconsistency

Living with inconsistency?

- ignoring inconsistency
- modifying the schema
- exceptions to constraints
- redefining query answers

Ignorantia Beatitudo Est?

A (young) woman of taste doesn't look at the price!

Muppet

Name	Role	DoB
Kermit	Manager	14.03.1965
Miss Piggy	Diva	21.06.1976
Miss Piggy	Diva	01.04.1950
T. Statler	Old Man	12.04.1946
T. Statler	Old Man	18.06.1942

Who's eligible for senior discount?

$Q(x) = \exists y, z.\ \mathit{Muppet}(x, y, z) \wedge z \leq 9.11.1950$

Standard answer semantics is (in)consistency oblivious

{Miss Piggy, T. Statler}

Impact of Inconsistency on Queries

Traditional view

- query results defined irrespective of integrity constraints
- integrity constraints may be used to optimize the query

Our view

- inconsistency leads to uncertainty (possible worlds)
- integrity constraints guide the user when formulating her queries
- query results may depend on satisfaction of integrity constraints
- inconsistency may be eliminated (repairing) or tolerated (consistent query answering)

Basic Notions

Restoring Consistency: Two operations

$$R[A, B] \subseteq P[A, B]$$

r:

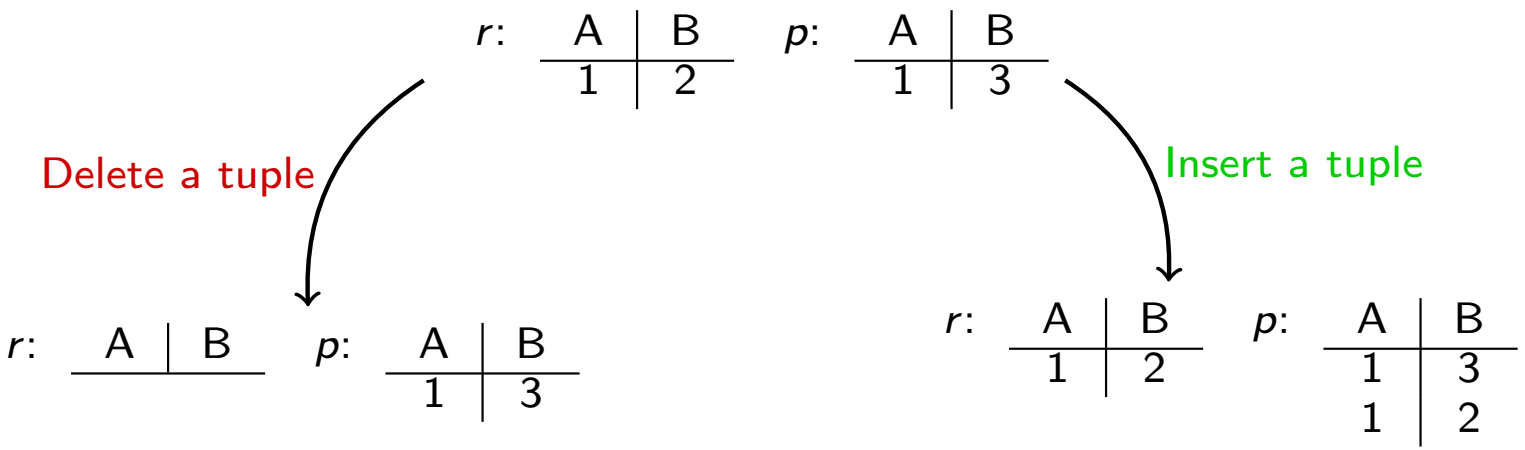
A	B
1	2

p:

A	B
1	3

Restoring Consistency: Two operations

$$R[A, B] \subseteq P[A, B]$$



Repair

A consistent instance obtained by performing a minimal set of operations.

Name	Role	DoB
Kermit	Manager	14.03.1965
Miss Piggy	Diva	21.06.1976
Miss Piggy	Diva	01.04.1950
T. Statler	Old Man	12.04.1946
T. Statler	Old Man	18.06.1942

r_1:

Name	Role	DoB
Kermit	Manager	14.03.1965
Miss Piggy	Diva	01.04.1950
T. Statler	Old Man	18.06.1942

r_2:

Name	Role	DoB
Kermit	Manager	14.03.1965
Miss Piggy	Diva	21.06.1976
T. Statler	Old Man	18.06.1942

r_3:

Name	Role	DoB
Kermit	Manager	14.03.1965
Miss Piggy	Diva	21.06.1976
T. Statler	Old Man	12.04.1946

r_4:

Name	Role	DoB
Kermit	Manager	14.03.1965
Miss Piggy	Diva	01.04.1950
T. Statler	Old Man	12.04.1946

Consistent Query Answers

Consistent Query Answer

Query answer present in every repair.

Who's eligible for senior discount?

$Q(x) = \exists y, z.\ \mathit{Muppet}(x, y, z) \wedge z \leq 9.11.1950$

Consistent Answers to $Q(x)$

- T. Statler is a consistent answer to $Q(x)$
- Miss Piggy is not a consistent answer to $Q(x)$ because of r_2 and r_3

CQA scientifically proven to make you feel much younger !

Naïve Data Cleansing

How about removing all conflicting data?

Name	Role	DoB
Kermit	Manager	14.03.1965
Miss Piggy	Diva	21.06.1976
Miss Piggy	Diva	01.04.1950
T. Statler	Old Man	12.04.1946
T. Statler	Old Man	18.06.1942

$\longrightarrow$ r^o:

Name	Role	DoB
Kermit	Manager	14.03.1965

$Q(x) = \exists y, z.\ Muppet(x, y, z) \wedge z \leq 9.11.1950$
The set of answers to $Q(x)$ in r_0 is empty

Radical approaches lead to information loss.

Computing Consistent Query Answers

Warning: Exponentially Many Repairs

A	B
1	0
1	1
⋮	
n	0
n	1

There are 2^n repairs of this instance w.r.t. the FD $A \rightarrow B$.

It is impractical to apply the definition of CQA directly.

Computing Consistent Query Answers

Query Rewriting

Given a query Q and a set of integrity constraints Σ, build a query Q^{Σ} such that

answers to Q^{Σ} *in* D = *consistent answers to* Q *in* D *w.r.t.* Σ

for every database D.

Representing all repairs

Given a database D and a set of integrity constraints Σ

1. build a compact representation of all repairs of D w.r.t. Σ
2. use it to compute the consistent answers

Logic programs

Given a database D, a set of integrity constraints Σ, and a query Q

1. build a logic program $P_{\Sigma,D}$ whose models represent repairs of D w.r.t. Σ
2. build a logic program P_Q expressing Q
3. use a LP system (Smodels, dlv) with cautious evaluation semantics to find answers present in all repairs.

Query Rewriting Example

Database Schema

$Muppet(Name, Role, DoB)$ with $Muppet : Name \rightarrow Role\ DoB$

Query

$\exists y, z.\ Muppet(x, y, z) \wedge z \leq 9.11.1950$

Integrity constraint $Muppet : Name \rightarrow Role\ DoB$

$\forall x, y, z, y', z'.\ \neg Muppet(x, y, z) \vee \neg Muppet(x, y', z') \vee (y = y' \wedge z = z')$

Rewritten query

$\exists y, z.\ Muppet(x, y, z) \wedge z \leq 9.11.1950 \wedge \nexists x', y'.\ Muppet(x, y', z') \wedge z' > 9.11.1950$

Milestones in Query Rewriting

- Arenas, Bertrossi, Chomicki [ABC99]
  - binary universal constraints (includes FDs and full INDs)
  - quantifier-free conjunctive queries
- Fuxman, Miler [FM07]
  - primary key dependencies
  - a class of conjunctive queries C_{forest}
    - no cycles (join graph is a forest)
    - no non-key or non-full joins
    - no repeated relation symbols
    - no built-ins
- Wijsen [Wij10]
  - primary key dependencies
  - a class of conjunctive queries C_{rooted}
    - semantic definition
    - syntactic (effective) characterization that is:
    - based on a notion of an attack graph
    - sound for conjunctive queries without self-join
    - complete for acyclic conjunctive queries without self-join

Rewriting SQL Queries

SQL query

```
SELECT Name FROM Muppet
WHERE DoB ≤ '9.11.1950'
```

SQL rewritten query

```
SELECT m1.Name FROM Muppet m1
WHERE m1.DoB ≤ '9.11.1950' AND NOT EXISTS
       (SELECT * FROM Muppet m2
       WHERE m2.Name = m1.Name AND m2.DoB > '9.11.1950')
```

Together, we shall CONQUER the universe !

(Fuxman, Fazli, Miller [FFM05])

- ConQuer: a system for computing CQAs
- conjunctive (C_{forest}) and aggregation SQL queries
- databases can be annotated with consistency indicators
- tested on TPC-H queries and medium-size databases

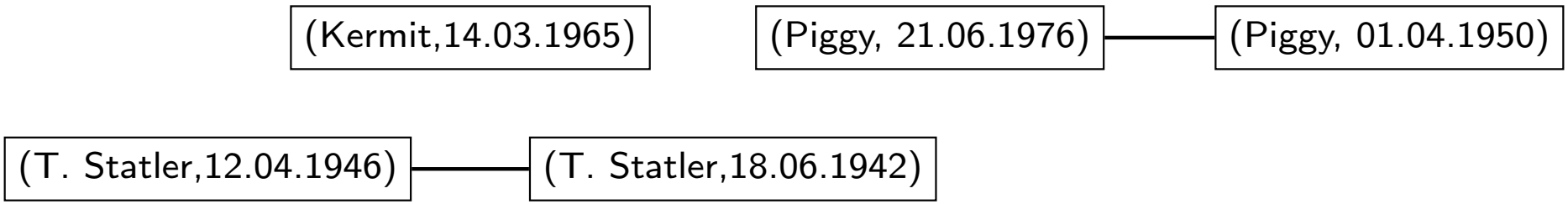
Conflict Hypergraph

Conflict Graph (Arenas et al. [ABC+03b])

Vertex tuple in the database

Edge two conflicting tuples

Repair is a maximal independent set



Extentions

- Conflict Hypergraph for denial constraints: hyperedges span on sets of tuples (Chomicki, Marcinkowski)[CM05]
- Extended Conflict Hypergraph for universal constraints: hyperedges may contain tuples to be added (S., Chomicki [SC10])

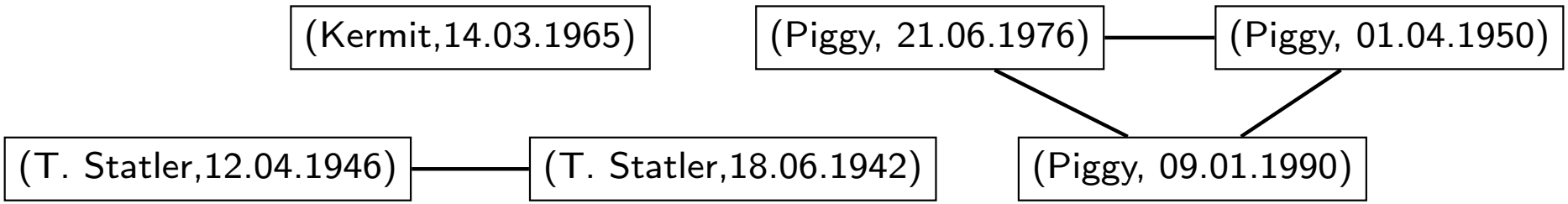
Conflict Hypergraph

Conflict Graph (Arenas et al. [ABC+03b])

Vertex tuple in the database

Edge two conflicting tuples

Repair is a maximal independent set



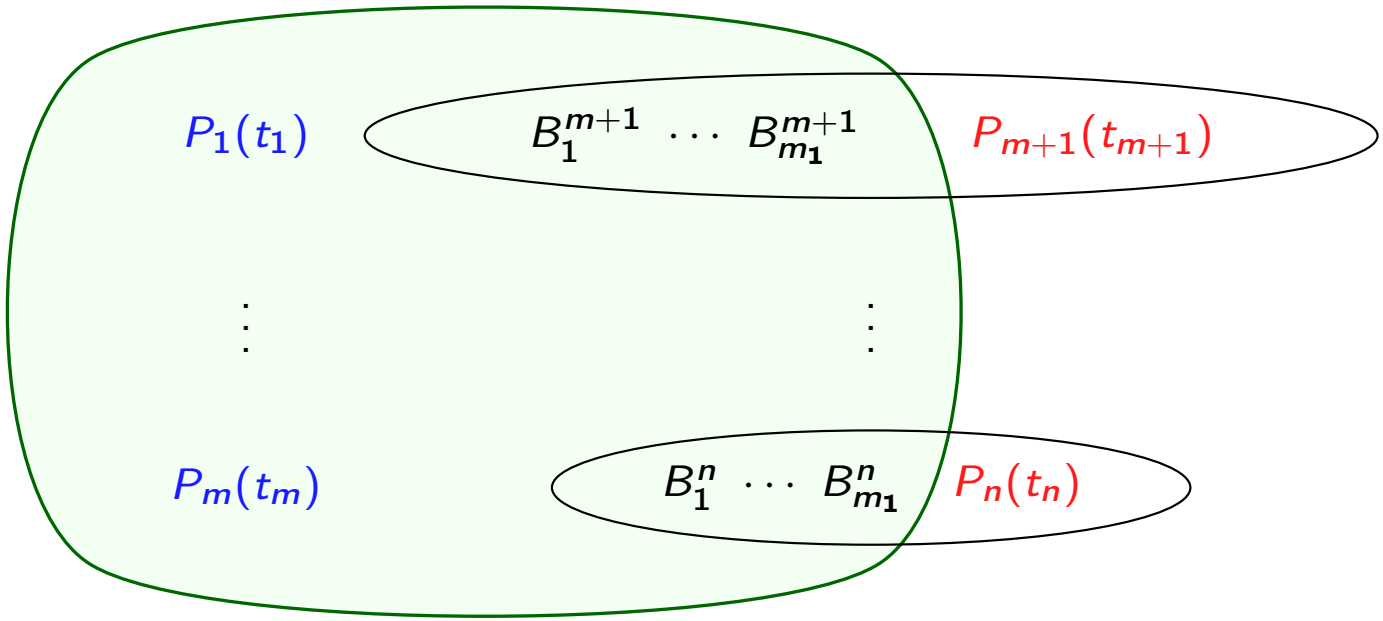
Extentions

- Conflict Hypergraph for denial constraints: hyperedges span on sets of tuples (Chomicki, Marcinkowski)[CM05]
- Extended Conflict Hypergraph for universal constraints: hyperedges may contain tuples to be added (S., Chomicki [SC10])

Algorithm HProver

Input: Φ a disjunction of ground literals, conflict hypergraph G of I w.r.t. Σ
Output: NO if Φ is false in some repair of D w.r.t. Σ?

1. $\neg\Phi = P_1(t_1) \wedge \cdots \wedge P_m(t_m) \wedge \neg P_{m+1}(t_{m+1}) \wedge \cdots \wedge \neg P_n(t_n)$
2. find a consistent set of facts S such that
   - S supports all positive facts i.e., $S \supseteq \{P_1(t_1), \ldots, P_m(t_m)\}$
   - S blocks all negative fact i.e., for every $A \in \{P_{m+1}(t_{m+1}), \ldots, P_n(t_n)\} \setminus D$ there is an edge $\{A, B_1, \ldots, B_m\}$ in G such that $S \supseteq \{B_1, \ldots, B_m\}$.



Computing CQA using Conflict Hypergraphs (cont.)

Quantifier-free CNF query Ψ

1. compute a superset A of consistent answers (with an envelope expression)
2. ground the query with a candidate tuple $t \in A$ and convert to CNF
$$\Psi(t) = \Phi_1 \wedge \ldots \wedge \Phi_k$$
3. if for some Φ_i HProver returns NO then discard t
4. otherwise, t is a consistent answer to the query

I'm a powerful beast too !

(Chomicki, Marcinkowski, S. [CMS04])

- Hippo: a system for computing CQAs in PTIME
- quantifier-free queries and denial constraints
- only edges of the conflict hypergraph hold in memory
- tested for medium-size synthetic databases

Logic Programs for computing CQAs

Logic Programs [ABC03a, GGZ03, CLR03]

- disjunction and classical negation
- checking whether an atom is in all answer sets is Π_2^p-complete
- `dlv`, `smodels`, ...

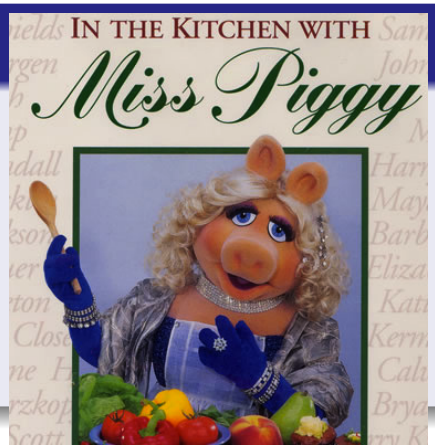
Scope

- arbitrary first-order queries and universal constraints
- approach unlikely to yield tractable cases

Guess what's in my MIX !

INFOMIX (Eiter et al. [EFGL03, EFGL08])

- combines CQA with data integration (GAV)
- uses `dlv` for repair computations
- optimization techniques: localization, factorization
- tested on small-to-medium-size legacy databases



Summary of Complexity Results

What's so (coNP-)hard about it?

$$\varphi = (x_1 \vee \neg x_2 \vee x_4) \wedge (x_2 \vee \neg x_4 \vee x_3) \wedge (\neg x_3 \vee x_4 \vee \neg x_1)$$

Reduction

R :

A	B	$A \to B$
1	0	$x_1 = false$
1	1	$x_1 = true$
⋮		⋮
5	0	$x_5 = false$
5	1	$x_5 = true$

Falsifying valuations for clauses

P :

A_1	B_1	A_2	B_2	A_3	B_3
1	0	2	1	4	0
2	0	4	1	3	0
3	1	4	0	1	1

- repairs correspond to all valuations of variables
- we want all valuations to fail to satisfy φ i.e. there always should be one clause whose none of literals isn't satisfied.

$$Q = \exists x_1, y_1, x_2, y_2, x_3, y_3.\ P(x_1, y_1, x_2, y_2, y_3) \wedge R(x_1, y_1) \wedge R(x_2, y_2) \wedge R(x_3, y_3)$$

Claim

True is the consistent answer to Q iff $\varphi \notin 3SAT$

Constraint classes

Universal constraints

$\forall.\ A_1 \wedge \cdots \wedge A_n \Rightarrow B_1 \vee \cdots \vee B_m$

Tuple-generating dependencies

$\forall.\ A_1 \wedge \cdots \wedge A_n \Rightarrow \exists.\ B$

Denial constraints

$\forall.\ \neg(A_1 \wedge \cdots \wedge A_n)$

Functional dependencies

$X \rightarrow Y$

- key dependency: $Y = U$

Inclusion dependencies

$R[X] \subseteq S[Y]$

- a foreign key constraint: key Y

Example

$\forall.\ Par(x, y) \Rightarrow Ma(x, y) \vee Fa(x, y)$

Example full TGD

$\forall.\ Ma(x, y) \wedge Ma(x, z) \Rightarrow Sib(y, z)$

Example

$\forall.\ \neg(M(n, s, m) \wedge M(m, t, w) \wedge s > t)$

Example primary-key dependency

Name $\rightarrow$ Address Salary

Example foreign key constraint

$M[Manager] \subseteq M[Name]$

Data complexity of CQA

- PTIME for $\{\sigma, \times, \setminus\}$-queries and binary universal constraints (FD + full IND) [ABC99]
- PTIME for $\{\sigma, \times, \setminus, \cup\}$-queries and denial constraints [CM05]
- PTIME for $\{\pi, \sigma\}$-queries and primary keys [CM05]
- coNP-complete for $\{\pi, \sigma, \times\}$-queries and primary keys, and $\{\pi, \sigma\}$-queries and FDs [CM05]
- undecidable for arbitrary functional and inclusion dependencies [CLR03]
- Π_p^2-complete for arbitrary sets of functional and inclusion dependencies (repairs obtained by deletions only) [CM05]
- PTIME for $\{\pi, \sigma, \times\}$-queries in C_{forest} and primary keys [FM07]
- PTIME for quantifier-free queries and acyclic full TGDs, join dependencies, and denial constraints [SC10]
- Π_2^p-complete for universal constraints [SC10]

Repair Checking

Problem statement

Fixed: Σ the set of integrity constraints

Input: Two databases instances D and D'

Question: Is D' a repair of D w.r.t. Σ?

Motivation

- Close connections with data-cleaning (the model checking problem for repairs)
- In some cases repair checking is log-space reducible to CQA [CM05].
- Negative results highlight limitations of integrity enforcement mechanisms.

Data complexity of Repair Checking

- PTIME for denial constraints [CM05]
- PTIME for FDs and acyclic INDs (deletion only) [CM05]
- coNP-complete for arbitrary FDs and INDs (deletion only) [CM05]
- PTIME for denial constrains and full TGDs [SC10]
- PTIME for weekly acyclic LAV dependencies [AK09]
- PTIME for semi-LAV dependencies [GO10]
- coNP for universal constraints [SC10]

Alternative Semantics

CQA: a lively topic

Tuple-based repairs

- asymmetric treatment of insertion and deletion:
  - repairs by minimal deletions only (Ch., Marcinkowski [CM05]): data possibly incorrect but complete
  - repairs by minimal deletions and arbitrary insertions (Calì, Lembo, Rosati [CLR03]): data possibly incorrect and incomplete
- minimal cardinality changes (Lopatenko, Bertossi [LB07]), (Afrati, Kolaitis [AK09])
- preferred repairs ([SCM06],[CGZ09], [MAA04], [GSTZ04], [GL04])
- null values (Bravo, Bertossi [BB06])

Attribute-based repairs

- ground and non-ground repairs (Wijsen [Wij05])
- project-join repairs (Wijsen [Wij06])
- repairs minimizing Euclidean distance (Bertossi et al. [BBFL08])
- repairs of minimum cost (Bohannon et al. [BFFR05])

CQA: a lively topic

Probabilistic framework for "dirty" databases (Andritsos, Fuxman, Miller [AFM06])

- potential duplicates identified and grouped into clusters
- worlds $\approx$ repairs: one tuple from each cluster
- world probability: product of tuple probabilities
- clean answers: in the query result in some (supporting) world
- clean answer probability: sum of the probabilities of supporting worlds
  - consistent answer: clean answer with probability 1

XML (S., Chomicki, Filiot [SC06, SFC08])

- tree edit distance for minimality
- schema: DTD (regular expressions) and tree automata
- XPath queries.

For more, see surveys

- Chomicki, ICDT'07 [Cho07]
- Bertossi, SIGMO Record [Ber06]


M. Arenas, L. Bertossi, and J. Chomicki.
Consistent query answers in inconsistent databases.
In *ACM Symposium on Principles of Database Systems (PODS)*, pages 68–79, 1999.


M. Arenas, L. Bertossi, and J. Chomicki.
Answer sets for consistent query answering in inconsistent databases.
Theory and Practice of Logic Programming, 3(4-5):393–424, 2003.


M. Arenas, L. Bertossi, J. Chomicki, X. He, V. Raghavan, and J. Spinrad.
Scalar aggregation in inconsistent databases.
Theoretical Computer Science (TCS), 296(3):405–434, 2003.


P. Andritsos, A. Fuxman, and R. J. Miller.
Clean answers over dirty databases: A probabilistic approach.
In *International Conference on Data Engineering (ICDE)*, page 30, 2006.


F. Afrati and P. Kolaitis.
Repair checking in inconsistent databases: Algorithms and complexity.
In *International Conference on Database Theory (ICDT)*. ACM, March 2009.


L. Bravo and L. E. Bertoss.
Semantically correct query answers in the presence of null values.
In *EDBT Workshops (IIDB)*, pages 336–357, 2006.


L. Bertossi, L. Bravo, E. Franconi, and A. Lopatenko.

The complexity and approximation of fixing numerical attributes in databases under integrity constraints.
Inf. Syst., 33(4-5):407–434, 2008.


L. Bertossi.
Consistent query answering in databases.
SIGMOD Record, 35(2):68–76, June 2006.


P. Bohannon, M. Flaster, W. Fan, and R. Rastogi.
A cost-based model and effective heuristic for repairing constraints by value modification.
In *ACM SIGMOD International Conference on Management of Data*, pages 143–154, 2005.


L. Caroprese, S. Greco, and E. Zumpano.
Active integrity constraints for database consistency maintenance.
IEEE Transactions on Knowledge and Data Engineering, 21(7):1042–1058, 2009.


J. Chomicki.
Consistent query answering: Five easy pieces.
In *International Conference on Database Theory (ICDT)*, pages 1–17, 2007.


A Cali, D. Lembo, and R. Rosati.
On the decidability and complexity of query answering over inconsistent and incomplete databases.

In *ACM Symposium on Principles of Database Systems (PODS)*, pages 260–271, 2003.

J. Chomicki and J. Marcinkowski.
Minimal-change integrity maintenance using tuple deletions.
Information and Computation, 197(1-2):90–121, February 2005.

J. Chomicki, J. Marcinkowski, and S. Staworko.
Computing consistent query answers using conflict hypergraphs.
In *International Conference on Information and Knowledge Management (CIKM)*, pages 417–426. ACM Press, November 2004.

T. Eiter, M. Fink, G. Greco, and D. Lembo.
Efficient evaluation of logic programs for querying data integration systems.
In *International Conference on Logic Programming (ICLP)*, pages 163–177, 2003.

T. Eiter, M. Fink, G. Greco, and D. Lembo.
Repair localization for query answering from inconsistent databases.
ACM Transactions on Database Systems (TODS), 33(2), 2008.

A. Fuxman, E. Fazli, and R. J. Miller.
Conquer: Efficient management of inconsistent databases.
In *ACM SIGMOD International Conference on Management of Data*, pages 155–166, 2005.

A. Fuxman and R. J. Miller.
First-order query rewriting for inconsistent databases.

Journal of Computer and System Sciences, 73(4):610–635, 2007.

G. Greco, S. Greco, and E. Zumpano.
A logical framework for querying and repairing inconsistent databases.
IEEE Transactions on Knowledge and Data Engineering, 15(6):1389–1408, 2003.

G. Greco and D. Lembo.
Data integration with preferences among sources.
In *International Conference on Conceptual Modeling (ER)*, pages 231–244. Springer, November 2004.

G. Grahne and A. Onet.
Data correspondence, exchange and repair.
In *International Conference on Database Theory (ICDT)*, pages 219–230, 2010.

S. Greco, C. Sirangelo, I. Trubitsyna, and E. Zumpano.
Feasibility conditions and preference criteria in quering and repairing inconsistent databases.
In *International Conference on Database and Expert Systems Applications (DEXA)*, pages 44–55, 2004.

A. Lopatenko and L. Bertossi.
Complexity of consistent query answering in databases under cardinality-based and incremental repair semantics.
In *International Conference on Database Theory (ICDT)*, pages 179–193, 2007.

A. Motro, P. Anokhin, and A. C. Acar.
Utility-based resolution of data inconsistencies.
In *International Workshop on Information Quality in Information Systems (IQIS)*, pages 35–43. ACM, 2004.

S. Staworko and J. Chomicki.
Validity-sensitive querying of XML databases.
In *EDBT Workshops (dataX)*, pages 164–177. Springer, 2006.

S. Staworko and J. Chomicki.
Consistent query answers in the presence of universal constraints.
Information Systems, 35(1):1–22, 2010.

S. Staworko, J. Chomicki, and J. Marcinkowski.
Preference-driven querying of inconsistent relational databases.
In *EDBT Workshops (IIDB)*, pages 318–335. Springer, 2006.

S. Staworko, E. Filiot, and J. Chomicki.
Querying regular sets of XML documents.
In *International Workshop on Logic in Databases (LiD)*, 2008.

J. Wijsen.
Database repairing using updates.
ACM Transactions on Database Systems (TODS), 30(3):722–768, 2005.

J. Wijsen.

Project-join-repair: An approach to consistent query answering under functional dependencies.
In *Flexible Query Answering Systems (FQAS)*, pages 1–12, 2006.

J. Wijsen.
On the first-order expressibility of computing certain answers to conjunctive queries over uncertain databases.
In *ACM Symposium on Principles of Database Systems (PODS)*, pages 179–190, 2010.