Logic-based Approaches to Novel Applications of Databases

Jan Chomicki
Dept. CSE
University at Buffalo
State University of New York
`http://www.cse.buffalo.edu/~chomicki`

Why **logic**

- simple syntax and semantics, universal application
- encourages **declarative** thinking:
  - queries
  - integrity constraints
  - deduction
- many languages and tools:
  - classical logic
  - modal logics of time, knowledge, ...
  - logic programming
- system/application correctness easier to establish
- and *people take you seriously*

Jan Chomicki, Ron van der Meyden, Gunter Saake, editors. **Logics for Emerging Applications of Databases.** Springer (?), 2003:

1. XML: model, schemas, types, logics and queries.
2. Consistent query answers in inconsistent databases.
3. Logic-based methods for updating knowledge bases.
4. Intelligent agents: issues and logics.
5. A survey of logic-based approaches to workflow modeling.
6. Reasoning about transaction dependencies.
7. Logical data expiration.
8. Description logics for conceptual modeling of dynamic information.
9. Logics for authorization and security.
10. Logics for trust management.
11. Logical languages for data mining.

My own research projects

Past:

- **Deductive** databases.
- **Temporal** query languages and integrity constraints.
- **Constraint**-based database interoperability.

Present:

- **Consistent** query answers in **inconsistent** databases.
- **Preference** queries.
- **Policy** management and **conflict** resolution.
- **Spatiotemporal** data models and query languages.

Consistent query answers

Databases may be **inconsistent** by violating integrity constraints:

- integration of heterogeneous databases with overlapping information
- the consistency of the database will be restored by executing further transactions
- it is impossible/undesirable to repair the database to restore consistency.

How to distinguish between **reliable** and **unreliable** information in an inconsistent database?

Repairs and consistent query answers [PODS'99]

Repair:

- a database that satisfies the integrity constraints
- difference from the given database is minimal (the set of inserted/deleted tuples is minimal under set inclusion)

Typically, more than one repair of a given inconsistent database.

A tuple is a **consistent query answer** to a query Q in a database r if it is an element of the result of Q in **every repair** of r.

What queries and integrity constraints

Ultimately: **SQL2**.

Now:

- queries:
  - first-order queries (equivalently: relational algebra)
  - scalar aggregation queries
- constraints:
  - functional dependencies: *"every student has a single address"*
  - referential integrity: *"students can enroll only in the offered courses."*

Further restrictions to obtain tractability (i.e., PTIME).

Future work:

- aggregation and relational algebra
- specialized algorithms and approximation
- preferences and ranking
- spatial inconsistency: **disaster management** databases
- data cleaning
- XML

Work with Marcelo Arenas (Toronto), Leo Bertossi (Carleton), Jerzy Marcinkowski (Wroclaw, Poland) and Doug Flewelling (UB Geography), with contributions from Roger He (UB CSE), Vijay Raghavan and Jeremy Spinrad (Vanderbilt).

Preference queries

Preferences have been studied in AI, philosophical logic and decision theory ... **but not in databases!**

Querying with preferences:

- information extraction and filtering
- solving complex configuration problems.

Contributions (so far):

- a logical framework for specifying preferences
- its embedding into relational algebra/SQL.

Problems:

- preference specification, composition and propagation
- query processing and optimization

```
CREATE VIEW Meal(Dish,DishType,Wine,WineType) AS
  SELECT * FROM Wine, Dish;
```

Preference formula (on `Meal`):

$$\begin{aligned}(d, dt, w, wt) \quad \succ \quad (d', dt', w', wt') \quad \equiv \quad & d = d' \\ & \wedge (dt = dt' = '\text{fish}' \wedge wt = '\text{white}' \wedge wt' = '\text{red}' \\ & \vee dt = dt' = '\text{meat}' \wedge wt = '\text{red}' \wedge wt' = '\text{white}')\end{aligned}$$

Winnowing the view `Meal` with this preference formula returns the **optimal available meals**.

Policy management

Policies: collections of principles specifying the desired behavior of a system.

Applications:

- network management
- e-commerce
- security and authorization.

Problems:

- specification of policies
- policy semantics, verification and analysis
- **conflict resolution**, preferences

Work with Jorge Lobo, formerly: Lucent, now: Teltier Technologies.

Spatiotemporal databases

How to represent information about **continuous phenomena** in a database?

A big culture gap:

- database researchers: we will extend **our** spatial data models with a temporal dimension
- natural scientists: **our** data does not quite fit into **your** models.

A $1G spatiotemporal query beyond the reach of current software technology:

> **Find all the areas where spring came earlier this year than last.**

Joint incubation project with Geoff Henebry (Nebraska) and others.