MAGIK: Managing Completeness of Data

Try out the demo at:

http://magik-demo.inf.unibz.it

Ognjen Savkovic
Free University of Bozen-Bolzano, Italy
savkovic@inf.unibz.it
joint work with **Sergey Paramonov, Mirza Paramita, and Werner Nutt**

Freie Universität Bozen
Libera Università di Bolzano
Free University of Bozen · Bolzano

Data Quality and Data Completeness

What is Data Quality?

- **Data is of a high quality if it is fit for intended uses**
- **Data quality** (**DQ**) has different **aspects**: **completeness, correctness, accuracy, etc.**
- **Little work** has been done on **data completeness**

What is Data Completeness?

- **A database is complete for a domain if it contains all facts that are true in the domain**
- In practice **no DB is complete**, but a **database can be sufficiently complete for a given query**
  e.g., *"IMDBb does not contain all movies but it contains all movies by Charlie Chaplin"*

Meta-information about completeness

- Completeness cannot be checked by inspecting the database
  - One cannot see what is missing
- We need information about the database completeness state – **meta-information**
- Often **meta-information about the data completeness is available**
- **Information** about partial completeness can come from:
  - **Business Processes** that manipulate the data
  - **Humans assertions** (e.g., school administration)
  - Origin of the data (**data provenance**)
  - **ETL processes** that integrate the data, etc.

MAGIK at Work: School Database

Scenario

The school administration provides **completeness statements** that describe which parts of the database are complete

School director creates a statistical report about the pupils in the school

Teacher at the school

System administrator maintains the database

MAGIK reasoner

School Database

pupil(name,level,code)	… a pupil belongs to a class of certain level and code
class(level,code,branch)	… every class belongs to some branch
learns(name,language)	… a pupil learns a language

Plain Reasoning

Statement 1: The school database is complete for all pupils.

Query: *Who are the pupils at the 1st level?*
Can I trust the query answer?

YES you can trust – Query is complete! Because statement 1 guarantees that the data request by the query is present in the database.

Statement 2: *Wait, we made a mistake. The database is **only** complete **for pupils** of class **'1a'**.*

Query: *Who are the pupils at the 1st level?*
Can I trust the query answer?

NO, you cannot trust the query answer. Because statement 2 only guarantees completeness for a specific part of the data asked by the query. Other parts might be incomplete.

Reasoning under Finite Domain Constraints (FDCs)

In our school every pupil can be either in **class 'a' or class 'b'**.
That is a finite domain constraint!

Statement 3: Now, *We are complete for all pupils at class **'1b'**.*

Query: *Who are the pupils at the 1st level?*
Can I trust the Query answer?

Well, we have all pupils in the class '1a'. Only pupils from class '1b' are missing. Please insert them!

YES you can trust – Query is complete! Because Statements 2 and 3 guarantees that all pupils from class '1a' and '1b' are there. Because class can be only 'a' or 'b' we have all pupils at level 1.

Reasoning under Foreign Keys (FKs)

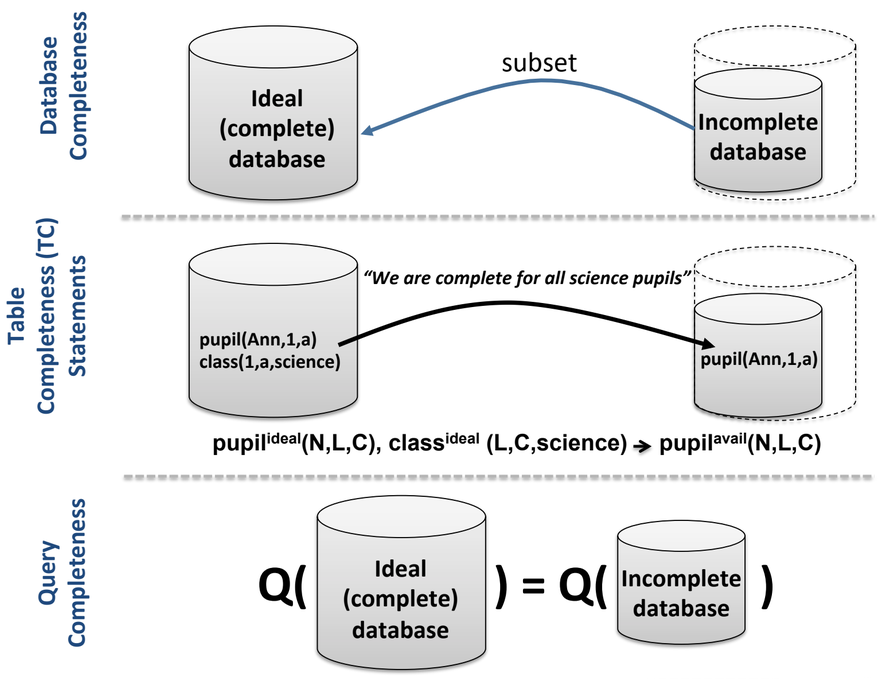
Wait, I designed the database!
I defined the FDC on table class, not on pupil, so every class[code] is in {a,b}.

Note, there is also a foreign key from pupil[level,code] that REFERENCES class[level,code].
That is a foreign key constraint!

Query: *Who are the pupils at the 1st level?*
Can I trust the query answer?

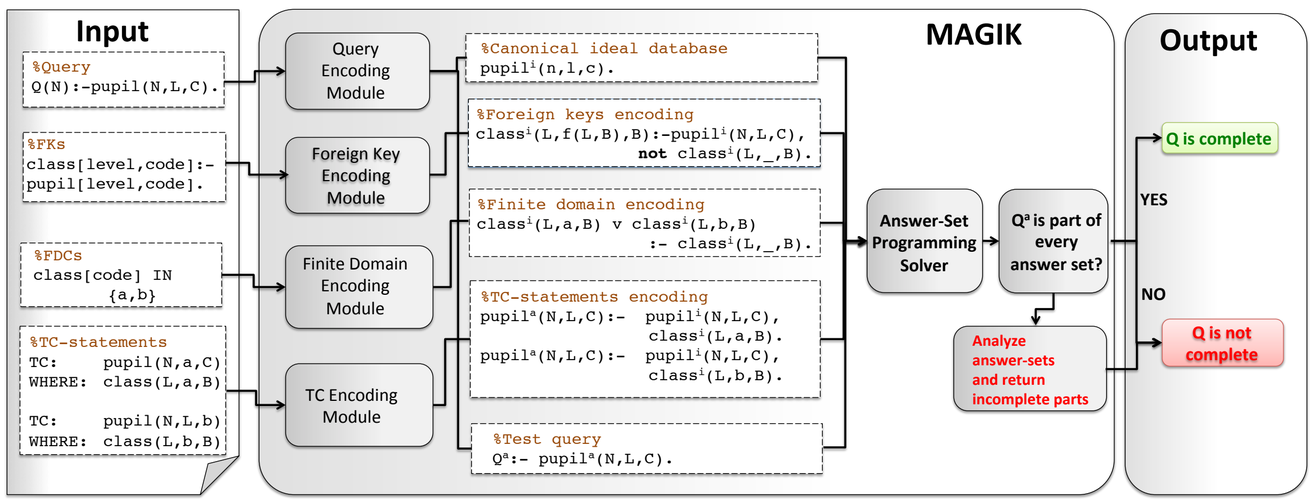
YES you can trust – Query is complete! Because the foreign key guarantees that for every pupil record there is a class record that according FDC must have class code either 'a' or 'b'. So the pupil record must have 'a' or 'b' for code as well. Then we are complete for '1a' and '1b' so we have all pupils at level 1.

Formalization of the Problem



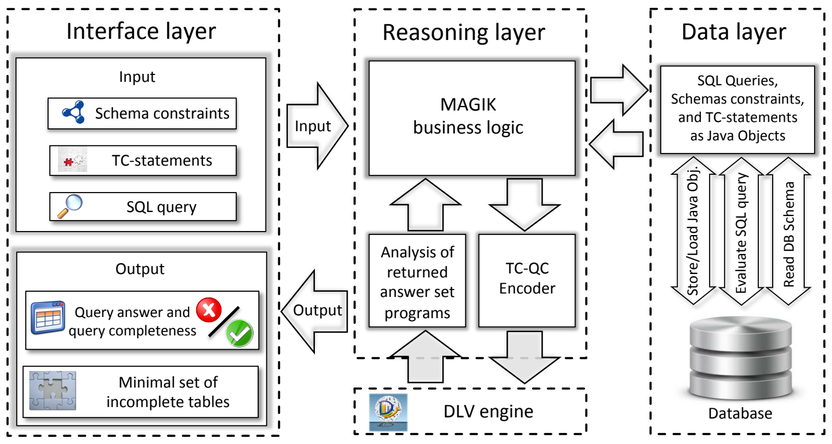
Implementation

Query is Complete iff Query is Complete wrt Canonical Ideal Database



Completeness of **Q** follows from a set of TCs, FKs and FDCs *iff* the fact **Q^a** is in every **answer-set** of the encoding answer-set program

System Architecture



System Features:

- **Web based** application written in **JAVA** with a GUI that allows one to:
  - **Create of a database schema** and **to extract a schema** from the database catalog
  - **Create/remove/edit TC-statements, FKs, FDs and Queries**
  - Pose **SQL select-project-joins queries**, possibly with **`DISTINCT`**, **`COUNT(*)`** and **`GROUP BY`**
- Encodes the completeness reasoning **into Logic Programming (Answer-Set Programming)**, which is executed by **DLV engine**

Summary

- **MAGIK** checks completeness of queries over incomplete databases given information about partially complete tables
- **MAGIK** reasons taking into account **schema constraints**: **foreign keys** and **finite domains constrains**
- **MAGIK explains** its answer and **suggests** which data to add to **make a database sufficiently complete** for a query