DATA QUALITY AWARE QUERYING*

Ognjen Savković
Free University of Bozen-Bolzano, Italy

* Supported by the project MAGIC, funded by the Province of Bozen-Bolzano

What is Data Quality?

- **Data** is of a **high quality** if it is **fit** for **intended uses**

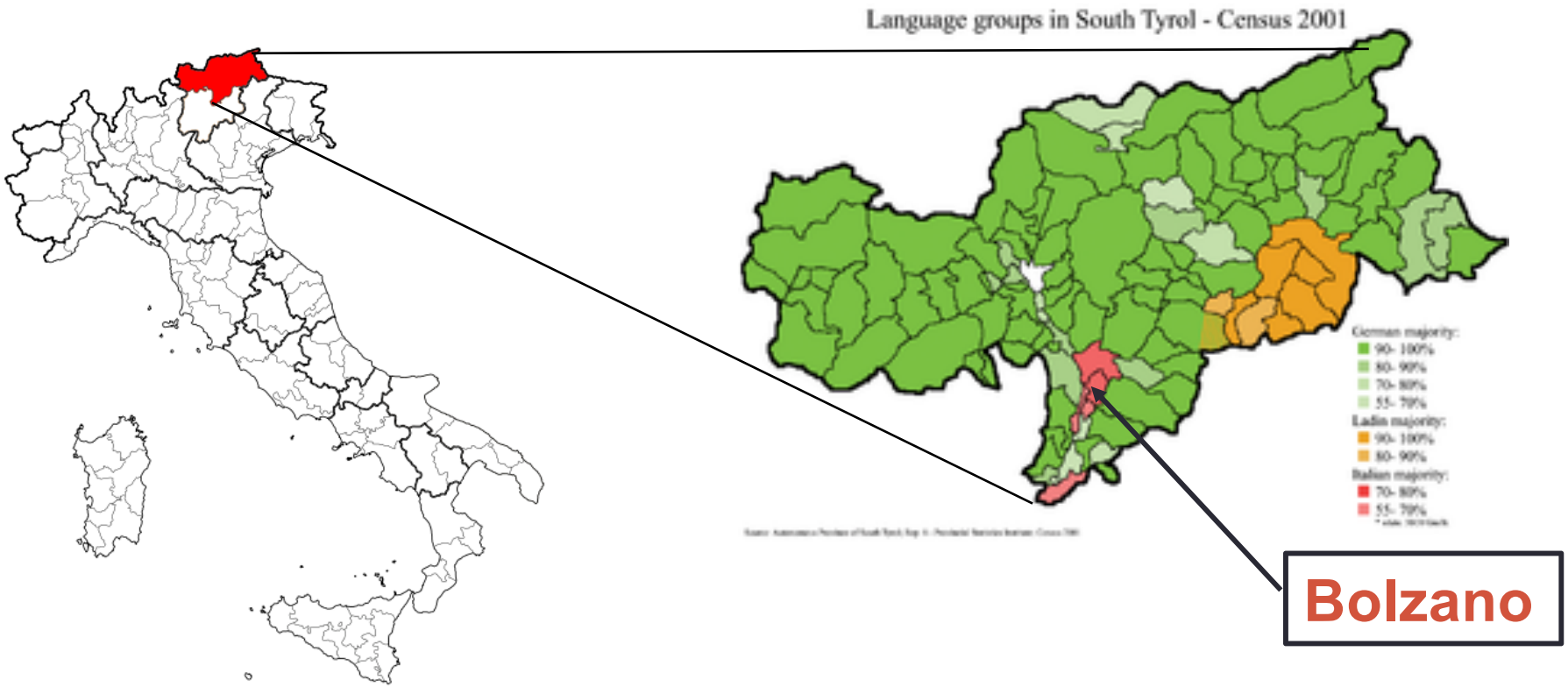
- Typically, **Data Quality** is considered as a problem of "**dirty data**". E.g.,
  - **Unclean data**, *student('John J.','Street*^#@0','+00399#)?:<')*, or
  - **Missing values**, *student('John J.',NULL,NULL)*

- But there are also more **subtle** **Data Quality** aspects
  - **Data Completeness** (we cannot see what is missing)
  - **Data Stability** (data that we see is updating frequently)

QUERY COMPLETENESS REASONING UNDER CONSTRAINTS

joint work with Werner Nutt, Sergey Paramonov

Bolzano is in the Province of South Tyrol

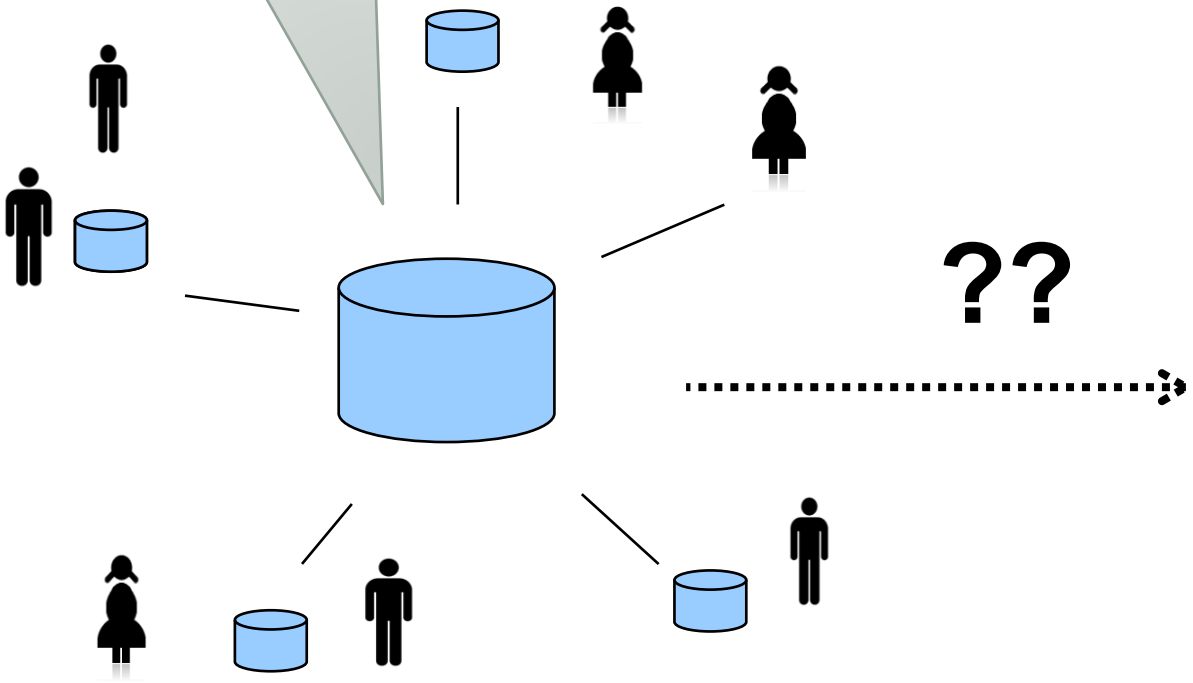


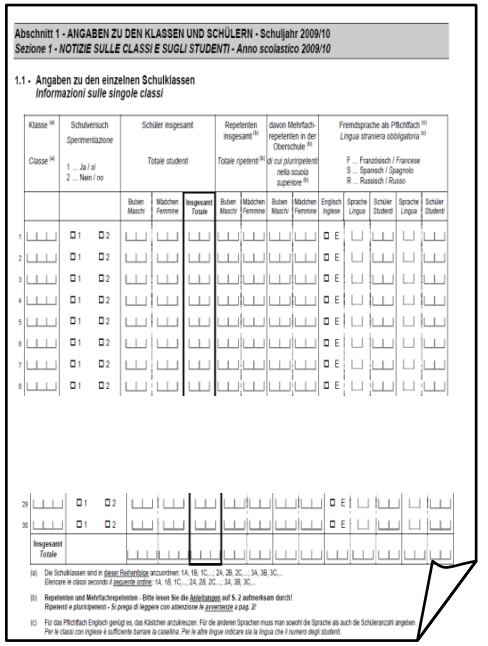
- Trilingual autonomous province in the North of Italy
  - Has its own school administration

Ex: School Data Management in Bolzano

Generally incomplete

Require complete data





- School Database is **decentralized**
- Some data is inserted **voluntarily**

Statistical Reports

Running Example: School Schema

pupil(name, ccode, sname)

e.g., pupil(john, a1, goethe)

class(ccode, sname, level, branch)

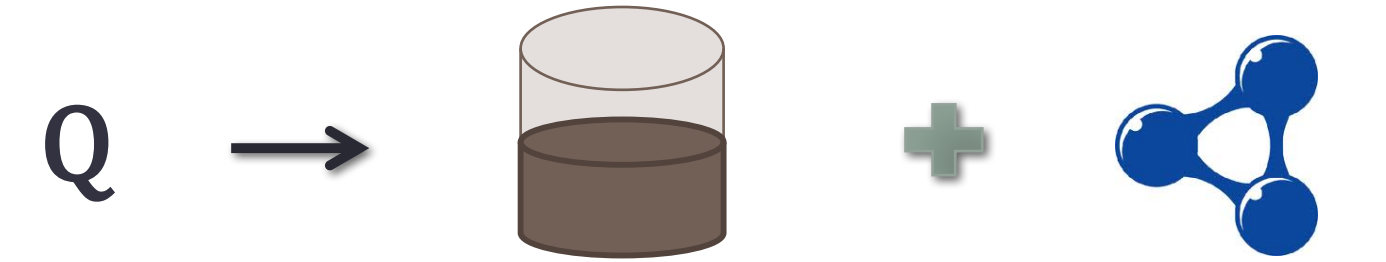
e.g., class(a1, gothe, 1, science)

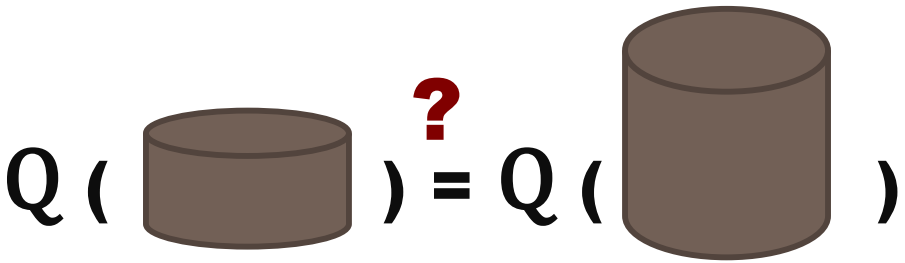
takes(name, activity)

e.g., takes(john, chess)

Query Completeness Reasoning under Constraints

- We want to reason about **Query Completeness** over
  - **Partially complete database** and
  - **Constraints** that hold over the database

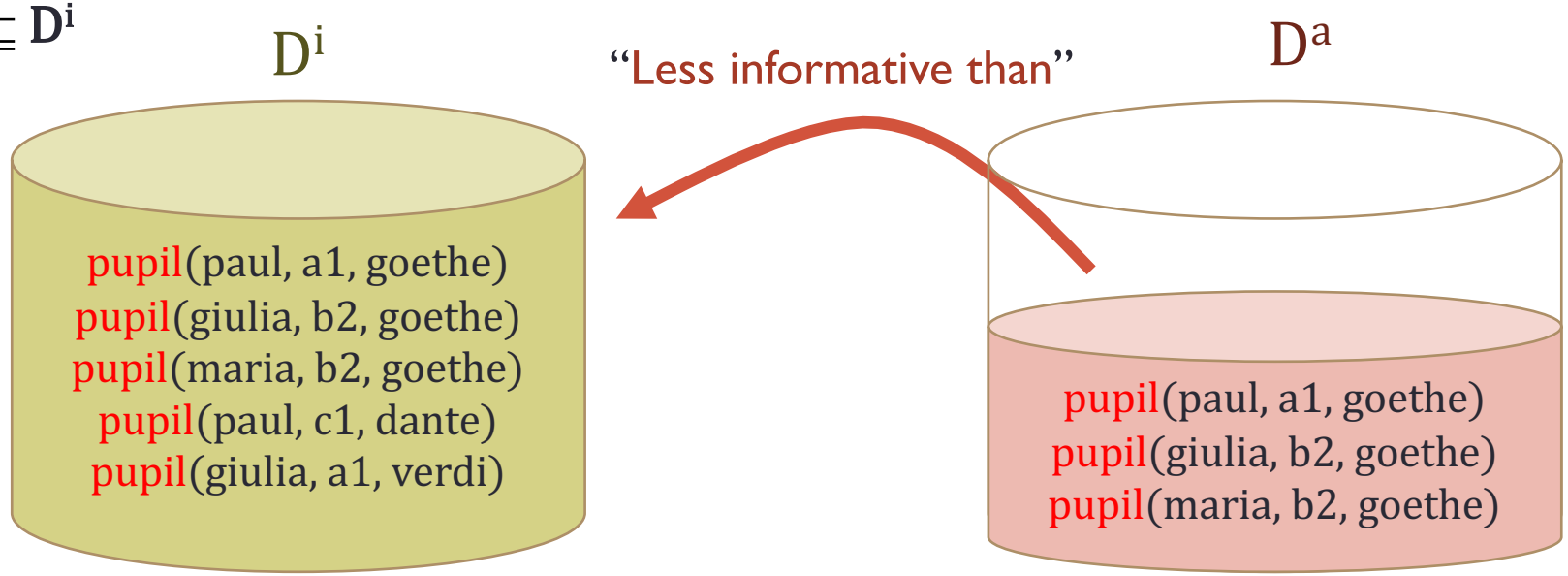




Completeness test

Formalization: Incomplete Database

- Incomplete database D is a pair $D = (D^i, D^a)$ [Motro 1989]
  - D^i - **ideal database** (complete facts that holds in the real world)
  - D^a - **available database** (actual database)
  - $D^a \subseteq D^i$



- Semantics

if $(D^i, D^a) \models \text{Compl}(Q)$ then $Q(D^i) = Q(D^a)$

Table Completeness Statements

A table completeness (TC) statement for a relation R is an expression

$$\mathrm{Compl}(\mathrm{R}(\mathrm{s}_1,\ldots,\mathrm{s}_n)\ ;\ \mathrm{G}) \qquad \text{[Halevy 96]}$$

consisting of

- an R-*atom* $\mathrm{R}(\mathrm{s}_1,\ldots,\mathrm{s}_n)$
- a condition G is conjunction of atoms

The TC-statement $\mathrm{C} = \mathrm{Compl}(\mathrm{R}(\mathrm{s}_1,\ldots,\mathrm{s}_n)\ ;\ \mathrm{G})$ can be seen as a rule

$$\mathrm{r_C} = \mathrm{R^i}(\mathrm{s}_1,\ldots,\mathrm{s}_n), \mathrm{G^i} \rightarrow \mathrm{R^a}(\mathrm{s}_1,\ldots,\mathrm{s}_n)$$

Semantics: $(\mathrm{D^i}, \mathrm{D^a}) \models \mathrm{C}$ iff $(\mathrm{D^i}, \mathrm{D^a}) \models \mathrm{r_C}$

TC-QC Reasoning Problem

- We want to decide (TC-QC entailment)

  "Does a set of table completeness (TC) statements C entail completeness of query Q (QC) ?"

$$\forall\ (D^i, D^a)$$

$$(D^i, D^a) \models C \text{ iff } (D^i, D^a) \models \mathrm{Compl}(Q)$$

or shortly

$$C \models \mathrm{Compl}(Q)$$

T_C Operator

To $C = \mathrm{Compl}(R(\underline{s}) ; G)$ we associate the query

$$Q_C(\underline{s}) \text{ :- } R(\underline{s}) , G$$

and the transformation on db instances

$$T_C(D) := \{ R(\underline{t}) \mid \underline{t} \in Q_C(D) \}$$

For a set C of TC statements we define the transformation

$$T_C(D) := \cup_{C \in C} T_C(D)$$

T_C Operator: Properties

- For a given (ideal) database D, operator T_C computes $T_C(D)$ such that

- $(D, T_C(D))$ is the least incomplete databases that satisfies C

In other words,

1. $(D, T_C(D)) \models C$ and
2. $(D^i, D^a) \models C$ iff $T_C(D^i) \subseteq D^a$

TC-QC: Characterization

Let

- C set of TC statements
- $\mathrm{Q}(\underline{\mathbf{x}})$:- $\mathrm{A1},\ldots,\mathrm{An}$ conjunctive query
- $\mathrm{D_Q}$:= *frozen version* of $\mathrm{A1},\ldots,\mathrm{An}$ with θ

Theorem:

$$C \models \mathrm{Compl(Q)} \quad \text{iff} \quad \theta\underline{\mathbf{x}} \in \mathrm{Q}(\mathrm{T}_C(\mathrm{D_Q}))^*$$

* This is a NP problem

Example: Plain Reasoning

- "Who are the pupils in class a1 in Goethe school that play chess?"

  $Q_{a1g}(N)$:- pupil(N, a1, goethe), takes(N, chess)

- "We have all pupil from a1 Goethe school"

  C_{a1}: Compl(pupil(N, a1, goethe) ; true)

- "We have all activity takers in Goethe school"

  C_{activg}: Compl(takes(N, A) ; pupil(N, C, goethe))

- Is Q_{a1g} complete given the completeness assumptions?

Example: Plain Reasoning(2)

1. Assume Q_{a1g} returns n' over D^i

1. Then, $pupil(n', a1, goethe), takes(N, chess) \in D^i$

1. Now, according C_{a1g} follows $pupil(n', a1, goethe) \in D^a$
   and, according C_{activg} follows $takes(n', chess) \in D^a$

2. Therefore, $n' \in Q_{a1g}(D^a)$,
   $\rightarrow Q_{a1g}$ is complete!

What if have schema constraints?

Schema Constraints

- We consider 2 kinds of constraints

- Foreign Keys (FKs)
  - “For every pupil record exists corresponding class record”
  - $\mathbf{pupil[ccode, sname] \subseteq class[ccode,sname]}$

- Conditional Finite Doman Constraints (CFDCs)
  - “Science classes in Goethe school can be either a1 or b2”
  - $\mathbf{class[sname{=}Goethe, branch{=}science][ccode] = \{a1,b2\}}$

TC-QC under Constraints

- $\mathcal{K}$ is a set of PKs and FKs
- $\mathcal{F}$ is a set of CFDCs

- $C \models_{\mathcal{K},\mathcal{F}} \mathrm{Compl}(Q)$ (non-enforced FKs)

  $\forall (D^i, D^a)$: if $D^i \models \mathcal{F}$, $D^a \models \mathcal{F}$,and $D^i \models \mathcal{K}$
  then $(D^i, D^a) \models C$ iff $(D^i, D^a) \models \mathrm{Compl}(Q)$

- $C \models^e_{\mathcal{K},\mathcal{F}} \mathrm{Compl}(Q)$ (enforced FKs)

  $\forall (D^i, D^a)$: if $D^i \models \mathcal{F}$, $D^a \models \mathcal{F}$,and $D^i \models \mathcal{K}$ and $D^a \models \mathcal{K}$
  then $(D^i, D^a) \models C$ iff $(D^i, D^a) \models \mathrm{Compl}(Q)$

Example: Reasoning under non-enforced FKs

- "Who are the pupils in Goethe school?"

  $Q_g(N)$:- pupil(N, C, goethe)

- "We have all pupil from a1 and b2 Goethe school"

  C_{a1}: Compl(pupil(N, a1, goethe) ; class(a1, goethe, L, B))

  C_{b2}: Compl(pupil(N, b2, goethe) ; class(b2, goethe, L, B))

- Can we conclude completness for Q_{gs}?

Example: Reasoning under non-enforced FKs (2)

- Assume constraints
  fk_1: $pupil[ccode, sname] \subseteq class[ccode,sname]$
  $cfdc_1$: $class[sname=goethe][ccode] = \{a1,b2\}$

1. $n' \in Q_g(D^i)$, then $pupil(n', c', goethe) \in D^i$

1. From fk_1, $class(c', goethe, l_{c',goethe}, b_{c',goethe}) \in D^i$

2. From $cfdc_1$ follows that $c'=a1$ or $c'=b2$ in $class$-record

3. In either case, acc. C_{a1} and C_{b2},
   follows $pupil(n', c', goethe) \in D^a$

1. Therefore, $n' \in Q_g(D^a)$, $\rightarrow$ Q_g is complete!

TC-QC under CSTRs: Characterization

- $\mathcal{C}$ and $\mathrm{Q}(\underline{\mathbf{x}})$:- $\mathrm{A1},\ldots,\mathrm{An}$ as before
- $\mathcal{K}$ set of PKs and acyclic FKs
  - $\mathcal{Chase}_{\mathcal{K}}$ oblivious chase
- $\mathcal{F}$ set of CFDCs
  - Γ set of all "maximal relevant" instantiations of $\mathcal{Chase}_{\mathcal{K}}(\mathrm{D_Q})$ according $\mathcal{F}$

Theorem: The following are equivalent

a) $\mathcal{C} \models_{\mathcal{K},\mathcal{F}} \mathrm{Compl(Q)}$

b) $\gamma\theta\underline{\mathbf{x}}' \in \mathrm{Q}(\gamma(\mathrm{T}_C(\gamma(\mathcal{Chase}_{\mathcal{K}}(\mathrm{D_Q}))))) \text{ for all } \gamma \in \Gamma$

* b) is a Π^{P}_2 problem

Example: Reasoning under enforced FKs

- "Who are the science pupils in Goethe school?"

  $Q_{gs}(N)$:- pupil(N, C, goethe), class(C, goethe,L, science)

- "We have all pupil from a1 and b2 Goethe school"

  C_{a1}: Compl(pupil(N, a1, goethe) ; class(a1, goethe, L, B))

  C_{b2}: Compl(pupil(N, b2, goethe) ; class(b2, goethe, L, B))

- And assume constraints

  fk_1: pupil[ccode, sname] ⊆ class[ccode,sname]

  $cfdc_1$: class[sname=goethe][ccode] = {a1,b2}

- Can we conclude completness for Q_{gs}?

Example: Reasoning under enforced FKs (2)

1. $n' \in Q_{gs}(D^i)$, then
   $pupil(n', c', goethe), class(c',goethe,l',science) \in D^i$

1. From $cfdc_1$ follows that $c'=a1$ or $c'=b2$ in $class$-record

2. In either case, acc. C_{a1} and C_{b2},
   follows $pupil(n', c', goethe) \in D^a$

3. From fk_1, $class(c', goethe, l_{c',goethe}, b_{c',goethe}) \in D^a$ (no!)
   Since $D^a \subseteq D^i$ it must be $class(c',goethe,l',science) \in D^a$

4. Therefore, $n' \in Q_{gs}(D^a)$, $\rightarrow$ Q_{gs} is complete!

Implementing Completeness Reasoning

- How can one implement completeness reasoning?
- Reminder: TC-QC ranges from NP to Π^{P}_2
- **SMT** (SAT modulo theories) solvers?
- **ASP**?
  - **Receipt**: Follow the **characterization** theorems

Encoding TC-QC into ASP

1. "Freeze" $Q_{a1g}(N)$ over ideal db

   pupil_i(n', a1, goethe). takes_i(n', chess).

2. Represent TCs as rules from ideal to available db

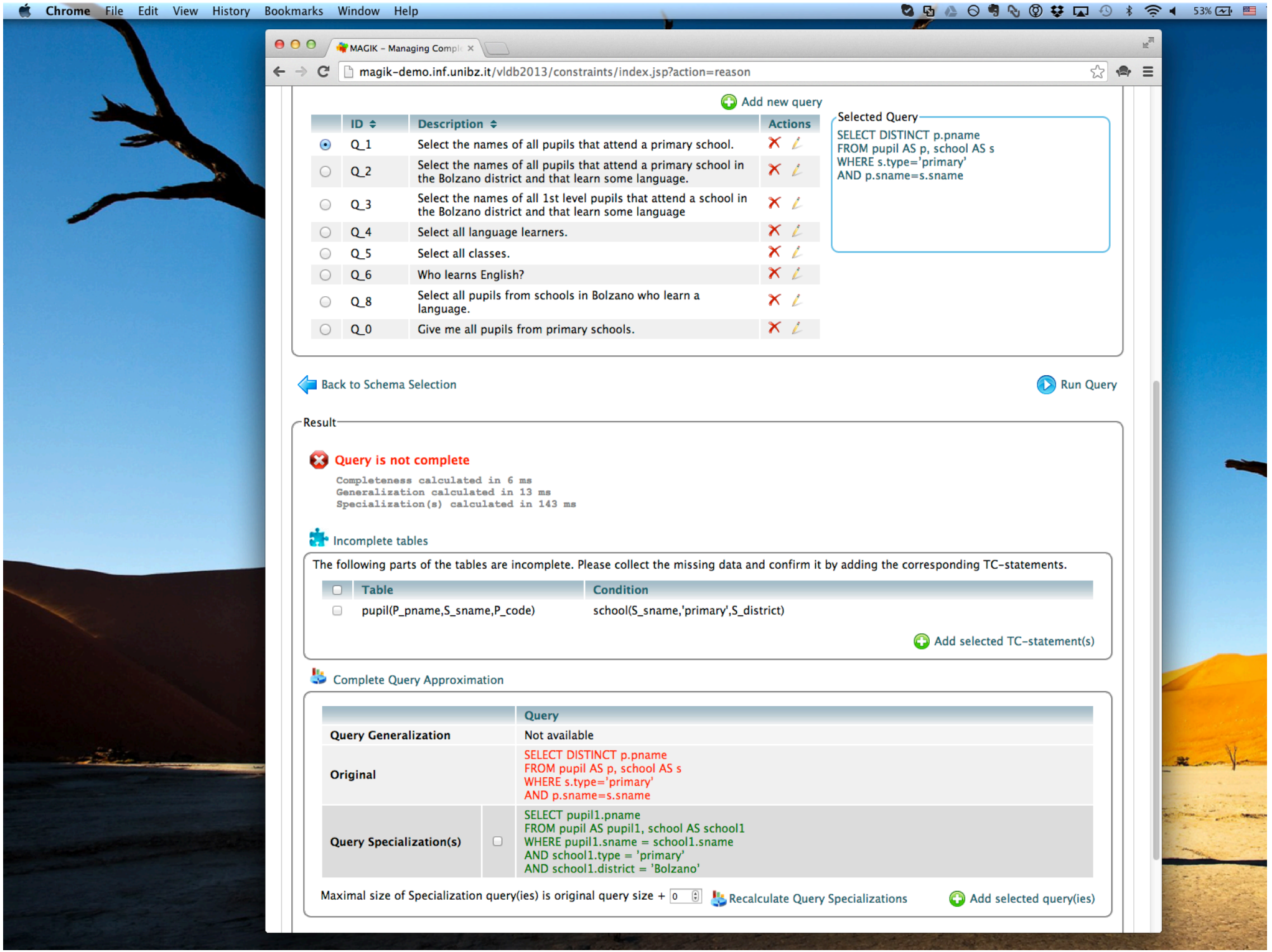
   pupil_a(N, a1, goethe) :- pupil_i(N, a1, goethe)

   takes_a(N, A) :- takes_i(N,A), pupil_i(N, C, goethe)

1. Test completeness with "test predicate" on available db

   q_test :- pupil_a(N, a1, goethe), takes_i(N, chess)

   $C \models$ Compl(Q) iff q_test is **entailed** by the program


Chrome File Edit View History Bookmarks Window Help
MAGIK – Managing Compl
magik-demo.inf.unibz.it/vldb2013/constraints/index.jsp?action=reason
Add new query
ID
Description
Actions
Q_1
Select the names of all pupils that attend a primary school.
Q_2
Select the names of all pupils that attend a primary school in the Bolzano district and that learn some language.
Q_3
Select the names of all 1st level pupils that attend a school in the Bolzano district and that learn some language
Q_4
Select all language learners.
Q_5
Select all classes.
Q_6
Who learns English?
Q_8
Select all pupils from schools in Bolzano who learn a language.
Q_0
Give me all pupils from primary schools.
Selected Query
SELECT DISTINCT p.pname
FROM pupil AS p, school AS s
WHERE s.type='primary'
AND p.sname=s.sname
Back to Schema Selection
Run Query
Result
Query is not complete
Completeness calculated in 6 ms
Generalization calculated in 13 ms
Specialization(s) calculated in 143 ms
Incomplete tables
The following parts of the tables are incomplete. Please collect the missing data and confirm it by adding the corresponding TC-statements.
Table
Condition
pupil(P_pname,S_sname,P_code)
school(S_sname,'primary',S_district)
Add selected TC-statement(s)
Complete Query Approximation
Query
Query Generalization
Not available
Original
SELECT DISTINCT p.pname
FROM pupil AS p, school AS s
WHERE s.type='primary'
AND p.sname=s.sname
Query Specialization(s)
SELECT pupil1.pname
FROM pupil AS pupil1, school AS school1
WHERE pupil1.sname = school1.sname
AND school1.type = 'primary'
AND school1.district = 'Bolzano'
Maximal size of Specialization query(ies) is original query size + 0
Recalculate Query Specializations
Add selected query(ies)

QUERY STABILITY IN BUSINESS PROCESSES

joint work with Werner Nutt

Ex: Student Registration

Faculty of Economics	Student places 2012 / 2013	Enrollments 2011 / 2012	Enrollments 2012 / 2013	%
Bachelor in Economics and Management	[illegible]	[illegible]	[illegible]	20.78%
Bachelor in Tourism, Sport and Event Management	[illegible]	[illegible]	[illegible]	74.51%
Bachelor in Economics and Social Sciences	[illegible]	[illegible]	[illegible]	17.24%
Master in Entrepreneurship and Innovation	[illegible]	[illegible]	[illegible]	114.29%
Master in Economics and Management of the public sector	[illegible]	[illegible]	[illegible]	22.73%
Sum	[illegible]	[illegible]	[illegible]	**41.45%**

Faculty of Computer Science	Student Places 2012 / 2013	Enrollments 2011 / 2012	Enrollments 2012 / 2013	%
Bachelor in Computer Science and Engineering	[illegible]	[illegible]	[illegible]	30.43%
Master of Science in Computer Science	[illegible]	[illegible]	[illegible]	-44.44%
PhD in Computer Science	[illegible]	[illegible]	[illegible]	#VALUE!
Sum	[illegible]	[illegible]	[illegible]	-25.37%

Data Stability

- **Data Completeness** is a rather strong requirement
  - Some information will be **always incomplete** or **incomplete "for a short time"**

- **Data Stability** is concerned whether **data** is **stable** in **some period of time**
  - "Is the number of students a stable figure during 2nd semester?"
  - "What is the longest periods for which the student numbers are stable?"

Business processes that **manipulate data** can tell us when the data (or query answer) is **stable**

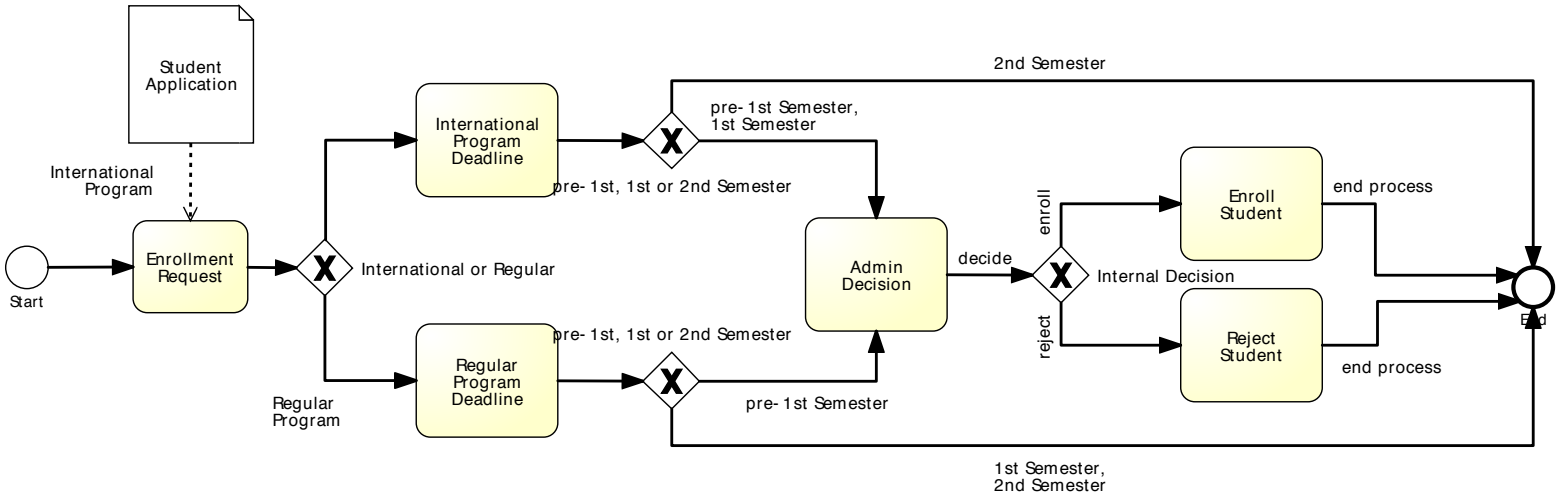
What are Business Processes?

- **Business Processes (BPs)** are set of activities organized to accomplish a **specific goal**
  - E.g., Student registration, Car production, etc.

- **Business Processes**
  - Used for **documenting** how the companies and organizations operate their business using languages such as **BPMN**
  - **Can be executed** using **Business Process Engines**
  - Allow different types of **analysis** (simulation, verification, process mining, etc.)

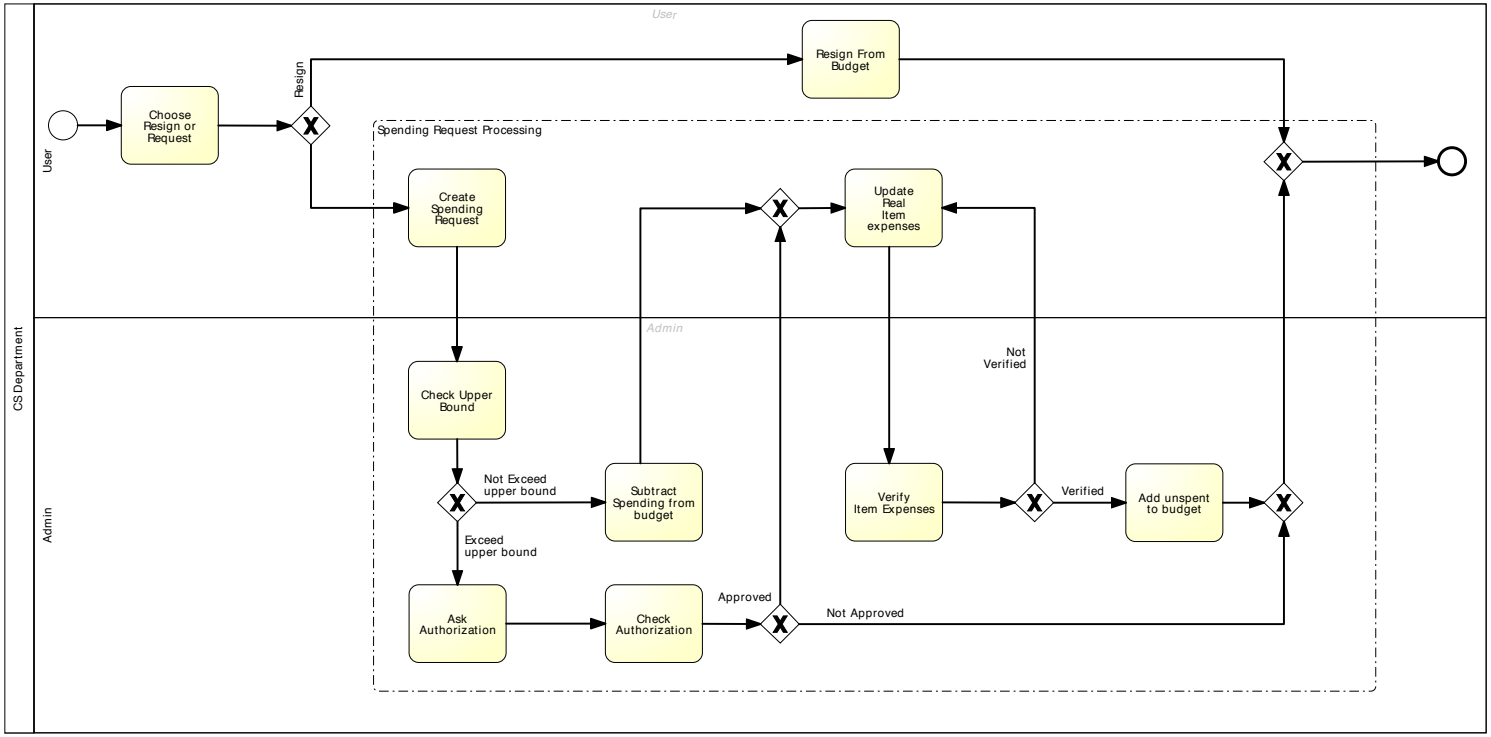
Ex: Student Registration (cont'd)

- Student Registration managed by a Business Process (in BPMN)



- Some questions:
  - How final or reliable are the figures that we see?
  - For which programs the figures become final eventually?
  - Which are the time periods in which the figures are stable?

Ex: Purchasing Information System @ Univ. Bolzano



- Is the available amount of the budget a stable figure?
- What is the maximal and what minimal amount in the budget that is not going to be spent? Etc.

Process and Data: Two sides of the same coin

- In current BP languages,
  the **interaction** **between BP**s and **Data** is very **limited**

- Recently, several models have been proposed
  - Relational transducers [Abiteboul et al., 1998]
  - Data-driven Web Systems [Deutsch et al., 2004; 2007]
  - Data-Centric Dynamic Systems [Calvanese et. al., 2012; 2013]

Data-aware Business Process

- **Data-aware Business Process (DABP)** is defined as a tuple $\mathcal{D}=(N, L, D, O)$
  - $N = (P,T)$ – **process net** represented as a finite **graph**
  - $L = (L_e,L_w)$ – **labeling functions** that label edges t from T
    - $L_e(t)$ – **execution condition,** Boolean Conjunctive Query over D, O
    - $L_w(t)$ – **writing action,** a horn rule $Head_t$:- $Body_t$
  - D – **relational database**
  - O – set of (**business**) **objects,** where each o has associated
    - $M(o) \in P$ – **place** of the process net
    - $Val(o) = R(\underline{s})$ – **value** as a record of fixed size

- $\mathcal{P} = (N, L)$ – **process part** (static part)
- $C = (D, O)$ – **configuration part** (dynamic part)

Execution in DABP

- **An execution step** in DABP is either
  a) **Traversal** of an **edge** by an **object**,
  Object o can traverse t from T iff $D,Val(o) \models L_e(t)$
  b) **Creation** of a **new object** in the start place of a process
  which is "always possible"

- Result of an execution step is a new configuration $C' = (D', O')$, written $C \rightarrow C'$
  a) If o traverses t then $D' = D \cup \{\alpha\ Head_t \mid \alpha\ Body_t \subseteq D,Val(o)\}$
  $M(o)$ points to the new place, and the rest remains the same
  b) If o is a newly introduced object
  then $O' = O \cup \{o\}$ and $M(o) = start$, and the rest remains the same

Query Stability in DABP

- Given DABP $\mathcal{D}=(\mathbf{N}, \mathbf{L}, \mathbf{D}, \mathbf{O})$, configuration $C' = (\mathbf{D'}, \mathbf{O'})$ is **reachable,** if there exist execution steps

$$C = (\mathbf{D}, \mathbf{O}) \rightarrow C_1 \rightarrow \dots \rightarrow C_n \rightarrow C'$$

- **Query Stability**
  Conjunctive query $\mathrm{Q}(\underline{\mathbf{x}})$:- $\mathrm{A1},\dots,\mathrm{An}$ is stable in DABP $\mathcal{D}=(\mathbf{N}, \mathbf{L}, \mathbf{D}, \mathbf{O})$, if for every configuration $C' = (\mathbf{D'}, \mathbf{O'})$ reachable from $C = (\mathbf{D}, \mathbf{O})$

$$\mathrm{Q}(\mathbf{D}) = \mathrm{Q}(\mathbf{D'})$$

Ex: Student Registration (cont'd)

- "Do we have stable figures for the students of informatics?"

  Q(Name) ← *student(Name,'informatics')*

  No! Because we can still enroll new students

- "Do we have stable figures for the students of economics?"

  Q(Name) ← *student(Name,'economics')*

  Yes! Because we cannot enroll any new student

Ex: Student Registration (cont'd)

Faculty of Economics	Student places 2012 / 2013	Enrollments 2011 / 2012	Enrollments 2012 / 2013	%	Stable?
Bachelor in Economics and Management	[illegible]	[illegible]	[illegible]	20.78%	YES!
Bachelor in Tourism, Sport and Event Management	[illegible]	[illegible]	[illegible]	74.51%	YES!
Bachelor in Economics and Social Sciences	[illegible]	[illegible]	[illegible]	17.24%	YES!
Master in Entrepreneurship and Innovation	[illegible]	[illegible]	[illegible]	114.29%	YES!
Master in Economics and Management of the public sector	[illegible]	[illegible]	[illegible]	22.73%	YES!
Sum	[illegible]	[illegible]	[illegible]	41.45%	YES!

Faculty of Computer Science	Student Places 2012 / 2013	Enrollments 2011 / 2012	Enrollments 2012 / 2013	%	Stable?
Bachelor in Computer Science and Engineering	[illegible]	[illegible]	[illegible]	30.43%	YES!
Master of Science in Computer Science	[illegible]	[illegible]	[illegible]	-44.44%	NO!
PhD in Computer Science	[illegible]	[illegible]	[illegible]	#VALUE!	NO!
Sum	[illegible]	[illegible]	[illegible]	-25.37%	NO!

Checking Stability in DABP

- 3 DABP dialects:
  - **DABP^{core}** forbids recursion in the horn writing rules
  - **DABP^{chron}** assumes no objects initially
  - **DABP^{recur}** no restrictions

- Computational Complexity of checking stability for CQ

complexity	DABP^{core}	DABP^{chron}	DABP^{recur}
data	AC^0	PTIME	CO-NP
combined	CO-NP ?	EXPTIME	CO-NEXPTIME
char. query lang.	CQ	DATALOG	DATALOG^{neg}

- data is measured in the size of the configuration
- combined is measured in the sizes of DABP and query

Conclusion

- 2 "**subtle**" aspects of data quality
  - **query completeness**
  - **stability of queries answers**

- Asses quality of query answers using **metadata**
  - **completeness statements, constraints**
  - **business processes**

- **Characterization** of query completeness under **constraints**
  - Basis for implementation in **ASP**

- **Modeling** query stability problem
  - Reasoning techniques for conjunctive queries

Thank you!