Efficient Query Containment Checking Using Logical Reasoning Engines

Sergey Paramonov

Technical University of Vienna

EMCL Workshop 2012 Vienna

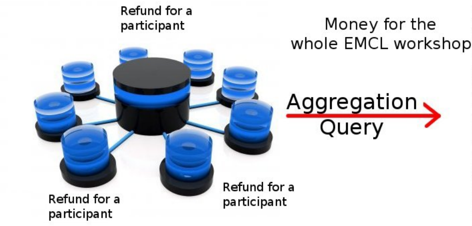
Historical perspective

- Query completeness problem has roots in the development of school system in Bolzano.
- Central school database is needed for administration, final grades, statistical reports etc.
- Teachers and admnistraters have only local records.

Settings

- People involved:
    - the KRDB group in Bolzano
    - the KBS group in Vienna
- Bolzano: developed theory of query completeness
- Vienna: developed a powerful disjunctive datalog engine (DLV)
- shortcoming of current theory lack of implementations
- Our goal: put theory into practise.

Motivation for Query Completeness



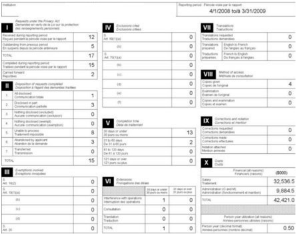
Refund for a participant

Refund for a participant

Refund for a participant

Money for the whole EMCL workshop

Aggregation Query



Institution

Reporting period Période visée par le rapport
4/1/2008 to/à 3/31/2009

I Requests under the Privacy Act
Demandes en vertu de la Loi sur la protection des renseignements personnels

Received during reporting period Reçues pendant la période visée par le rapport	12
Outstanding from previous period En suspens depuis la période antérieure	5
TOTAL	17
Completed during reporting period Traitées pendant la période visée par le rapport	15
Carried forward Reportées	2

II Disposition of requests completed
Disposition à l'égard des demandes traitées

1.	All disclosed Communication totale	1
2.	Disclosed in part Communication partielle	3
3.	Nothing disclosed (excluded) Aucune communication (exclusion)	0
4.	Nothing disclosed (exempt) Aucune communication (exemption)	0
5.	Unable to process Traitement impossible	8
6.	Abandoned by applicant Abandon de la demande	3
7.	Transferred Transmission	0
TOTAL		15

III Exemptions invoked
Exceptions invoquées

S. Art. 18(2)	0
S. Art. 19(1)(a)	0
(b)	0
(c)	0
(d)	0
S. Art. 20	0

IV Exclusions cited
Exclusions citées

S. Art. 69(1)(a)	0
(b)	0
S. Art. 70(1)(a)	0
(b)	0
(c)	0
(d)	0
(e)	0
(f)	0

V Completion time
Délai de traitement

30 days or under 30 jours ou moins	13
31 to 60 days De 31 à 60 jours	2
61 to 120 days De 61 à 120 jours	0
121 days or over 121 jours ou plus	0

VI Extensions
Prorogations des délais

	30 days or under 30 jours ou moins	31 days or over 31 jours ou plus
Interference with operations Interruption des opérations	1	0
Consultation	0	0
Translation Traduction	0	0
TOTAL	1	0

VII Translations
Traductions

Translations requested Traductions demandées		0
Translations prepared Traductions préparées	English to French De l'anglais au français	0
	French to English Du français à l'anglais	0

VIII Method of access
Méthode de consultation

Copies given Copies de l'original	4
Examination Examen de l'original	0
Copies and examination Copies et examen	0

IX Corrections and notation
Corrections et mention

Corrections requested Corrections demandées	0
Corrections made Corrections effectuées	0
Notation attached Mention annexée	0

X Costs
Coûts

Financial (all reasons) Financiers (raisons)	($000)
Salary Traitement	32,536.5
Administration (O and M) Administration (fonctionnement et maintien)	9,884.5
TOTAL	42,421.0
Person year utilization (all reasons) Années-personnes utilisées (raisons)	
Person year (decimal format) Années-personnes (nombre décimal)	0.50

- When does query completeness matter?
- in data integration
- if several people, institutions independently contribute data
- some data are final and others provisional

Query Completeness

- What does it mean for a query to be complete?
- Intuitevely it captures in the answer all tuples.
- Could you imagine that EMCL administration is missing you personal record?
- Now we can verify that everything is in the right place![1]

[1] "Beware! I have only proved it correct, not tried it." Donald Knuth

Formalization [Motro 89]

Definition (Partial Database)

A partial database is a pair $D = (D^i, D^a)$ of two instances,

- the ideal database D^i
- the available database D^a

such that $D^a \subseteq D^i$

Intuition:

- D^i reflects real world, what is really true
- D^a reflects data we physically store

Note (We make validity assumption)

there is no "wrong" data in the available database.

Partial Database Example

- $D = (D^a, D^i)$
  is partial database with two students (Oliver & Wu) in two different classes (2b & 2a).

- **Ideal** Database $D^i = \{$

$$Student(Oliver, ''EMCL''), Class(Oliver, 2, b),$$
$$Student(Wu, ''ICCL''), Class(Wu, 2, a)\}$$

- **Available** Database $D^a = D^i \setminus Class(Oliver, 2, a)$

Note

Available database is missing the fact that Oliver is a second year student.

Formalism. Completeness

What does it mean for a query Q to be complete?

Definition

Q is said to be complete written as $Compl(Q)$:

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

Intuition: a query Q is complete if query evaluation over available database is the same as over ideal one.

Completeness Statements [Levy 96]

Peter confirmed:

> *"Workshop database contains all 2 year students "*[2]

We formalize this as a **table completeness statement**:

$$Student^i(N,M), Class^i(N,2,C) \rightarrow Student^a(N,M)$$

or shortly **Compl(student(N,M) ; class(N,2,C))**
General notation:

$$Compl(R(\bar{s}); G)$$

where query $Q(\bar{s}) = R(\bar{s}), G$ is safe

[2] It is actually not true, right Martin?

TC-QC

Main question in the project how to implement the problem:

When completeness of small parts of the database entail completeness of the query?

Formally:
TC-QC: table completeness entails query completeness

$$Compl(R_1, G_1), \ldots, Compl(R_n, G_n) \models Compl(Q)$$

Example

All students in Dresden, Vienna, Bolzano and Lisbon are good, does it mean that all ECML students are good?

Query Containment

- Definition (Query Containment: Q_1 is contained in Q_2 written as: $Q_1 \subseteq Q_2$)

$$Q_1(D) \subseteq Q_2(D) \quad \forall D \text{ - db instances}$$

  - Studied for conjunctive queries (**CQ**).
    - Correspond to single-block select-from-where SQL query
    - Query that ask for good EMCL students:

$$Q(Name) \leftarrow Student(Name, "EMCL"), Good(Name).$$

  - Extensions: CQs with comparisons($\geq, >$), finite domains, unions of CQs.
  - Complexity: from NP to Π_2^P.[3]

[3] Free Complexity Class tonight in the pub

Containment example

Given two queries Q_1 and Q_2

$$\begin{array}{rcl} Q_1(Name) & \leftarrow & Student(Name,''EMCL''), Good(Name). \\ Q_2(Name) & \leftarrow & Student(Name,''EMCL''). \\ Q_1 & \subseteq & Q_2 \quad ? \end{array}$$

The question whether all good EMCL student are among EMCL student?

And the answer is, of course, yes.

Opposite does not hold:

It is hard to beleive but there might exist not good EMCL students.

Algorithm for the TC-QC

- TC-QC problem can be reduced to the variants of query containment.

  Intuition:

  - Query needs parts $\{P_i\}$ of the relation R_i to be complete
  - Is P_i contained in the parts $S_1, \ldots, S_n$ stated to be complete?

  so containment:

$$P_i \subseteq S_1 \cup S_2 \cup \cdots \cup S_n$$

- Query containment can be in reduced to evalution task of different reasoning engines.

Implementation

Query containment can be in principle reduced to the

- ASP: done in DLV for Relational Case
- SMT: partially studied for comparisons in Z3.
- QBF: alternative approach in the future.

Future Work

- Investigate different faces of the problem e.g. finite domain contraint (now in progress)
- Develop different implementations: SMT, DLV, ASP+Difference logic, QBF.
- Create a uniform benchmark for different classes of languages(RQ,LQ,CQ,UCQ)

Evaluation of the project

A detailed report with complete results is going to be submitted to ESSLLI 2012 as an article and a poster.

Questions time

`<joke>`

- **Sir Humphrey**: If local authorities don't send us statistics, Government figures will be a nonsense.
- **Hacker**: Why?
- **Sir Humphrey**: They'll be incomplete.
- **Hacker**: Government figures are a nonsense, anyway.
- **Bernard**: I think Sir Humphrey wants to ensure they're a complete nonsense.

`</joke>`

Thank you for your attention.