PhD course on

View-based query processing

Data integration – lecture 3

Riccardo Rosati

Dipartimento di Informatica e Sistemistica

Università di Roma "La Sapienza"

{rosati}@dis.uniroma1.it

Course overview

1. Introduction to view-based query processing [Lenzerini]
2. Conjunctive query evaluation [Gottlob]
3. Data exchange [Gottlob]
4. **Data integration** [De Giacomo, Rosati]
5. Data integration through ontologies [De Giacomo]
6. View-based query processing over semistructured data [Calvanese]
7. Reasoning about views [Lenzerini]

Lecture overview

- exclusion dependencies (EDs)
- separation properties for EDs
- query reformulation under KDs, IDs, and EDs:
  - GAV mapping
  - LAV mapping
  - complexity and expressiveness issues
- inconsistency tolerance (consistent query answering)
- the loosely-sound semantics for information integration
- query answering under loosely-sound semantics

Integrity constraints for relational schemas

Most important ICs for the relational model:

- key dependencies (KDs)
- functional dependencies (FDs)
- inclusion dependencies (IDs)
- foreign keys (FKs)
- exclusion dependencies (EDs)

Exclusion dependencies (EDs)

- an ED states that the presence of a tuple t in a relation implies the **absence** of a tuple t' in another relation such that t' contains a projection of the values contained in t
- syntax: $r[i_1, \ldots, i_k] \cap s[j_1, \ldots, j_k] = \emptyset$
- e.g., the ED $r[1] \cap s[2] = \emptyset$

  corresponds to the FOL sentence

$$\forall x, y, z, x', z'. r(x, y, z) \rightarrow \neg s(x', x, z')$$

- EDs are a special form of **denial dependencies** (a.k.a. denial constraints)

Query answering under IDs and EDs

under EDs and IDs:

- possibility of inconsistencies
- when $ret(\mathcal{I}, \mathcal{C})$ violates the EDs, no legal database exists and query answering becomes trivial!
- Is query answering decidable?
- Is query answering separable?

Example

Global schema: player$(Pname, YOB, Pteam)$

team$(Tname, Tcity, Tleader)$

coach$(Cname, Cteam)$

Constraints: team$[Tleader,\ Tname] \subseteq$ player$[Pname,\ Pteam]$

coach$[Cname] \cap$ player$[Pname] = \emptyset$

Mapping:

$$\text{player} \quad \rightsquigarrow \quad \begin{cases} \text{player}(X, Y, Z) & \leftarrow \quad \text{s}_1(X, Y, Z) \\ \text{player}(X, Y, Z) & \leftarrow \quad \text{s}_3(X, Y, Z) \end{cases}$$

$$\text{team} \quad \rightsquigarrow \quad \text{team}(X, Y, Z) \quad \leftarrow \quad \text{s}_2(X, Y, Z)$$

$$\text{coach} \quad \rightsquigarrow \quad \text{coach}(X, Y) \quad \leftarrow \quad \text{s}_4(X, Y)$$

Example (cont'd)

Source database $\mathcal{C}$

s$_1$:

Totti	1971	Roma

s$_2$:

Juve	Torino	Del Piero

s$_3$:

Vieri	1970	Inter

s$_4$:

Del Piero	Viterbese

Retrieved global database $ret(\mathcal{I}, \mathcal{C})$

player:

Totti	1971	Roma
Vieri	1970	Inter

team:

Juve	Torino	Del Piero

coach:

Del Piero	Viterbese

Example (cont'd)

player:

Totti	1971	Roma
Vieri	1970	Inter

team:

Juve	Torino	Del Piero

coach:

Del Piero	Viterbese

violation of team$[Tleader,\ Tname] \subseteq$ player$[Pname,\ Pteam]$

Example (cont'd)

player:

Totti	1971	Roma
Vieri	1970	Inter
Del Piero	α	Juve

team:

Juve	Torino	Del Piero

coach:

Del Piero	Viterbese

"repair" of team$[Tleader,\ Tname] \subseteq$ player$[Pname,\ Pteam]$

Example (cont'd)

player:

Totti	1971	Roma
Vieri	1970	Inter
Del Piero	α	Juve

team:

Juve	Torino	Del Piero

coach:

Del Piero	Viterbese

violation of coach$[Cname] \cap$ player$[Pname] = \emptyset$

Example (cont'd)

player:

Totti	1971	Roma
Vieri	1970	Inter
Del Piero	α	Juve

team:

Juve	Torino	Del Piero

coach:

Del Piero	Viterbese

violation of coach$[Cname]$ $\cap$ player$[Pname] = \emptyset$

nonetheless, a form of separability holds for IDs and EDs!

Deductive closure of EDs under IDs

From

$$\text{team}[Tleader, Tname] \subseteq \text{player}[Pname, Pteam]$$

$$\text{coach}[Cname] \cap \text{player}[Pname] = \emptyset$$

it follows that

$$\text{coach}[Cname] \cap \text{team}[Tleader] = \emptyset$$

- this constraint is violated by the retrieved global database $ret(\mathcal{I}, \mathcal{C})$!
- can we saturate (close) the EDs by adding all the EDs that are logical consequence of the EDs and IDs?

Deductive closure of EDs under IDs

- derivation rule of EDs under EDs and IDs:

  from the ED $r[i_1, \ldots, i_k] \cap s[j_1, \ldots, j_k] = \emptyset$

  and the ID $t[\ell_1, \ldots, \ell_k] \subseteq s[j_1, \ldots, j_k]$

  derive the ED $r[i_1, \ldots, i_k] \cap t[\ell_1, \ldots, \ell_k] = \emptyset$

- corresponds to a simple application of **resolution** on the FOL sentences corresponding to EDs and IDs

- if the set of EDs is closed with respect to the above rule, it contains all EDs that are logical consequences of the initial EDs and IDs

Separation for IDs and EDs

Theorem (ID-ED separation): Under IDs and EDs:

if $ret(\mathcal{I}, \mathcal{C})$ satisfies all EDs derived from the IDs and the original EDs

then the EDs can be ignored wrt certain answers of a query Q

$\Rightarrow$ query answering method for GAV systems under EDs and IDs:

1. close the set of EDs with respect to the IDs
2. verify consistency of $ret(\mathcal{I}, \mathcal{C})$ with respect to EDs
3. compute ID-rewrite of the input query
4. unfold the query computed at previous step
5. evaluate the query over the sources

the ED consistency check can be done by suitable CQs (exercise)

Separation for IDs, KDs EDs

extension of the above result to the presence of KDs:

Theorem (ID-KD-ED separation): Under KDs, NKCIDs, and EDs: if $ret(\mathcal{I}, \mathcal{C})$ satisfies all the KDs and satisfies all EDs derived from the IDs and the original EDs then the KDs and the EDs can be ignored wrt certain answers of Q

Query answering method

query answering method for GAV systems under KDs, EDs and IDs:

1. close the set of EDs with respect to the IDs
2. verify consistency of $ret(\mathcal{I}, \mathcal{C})$ with respect to KDs and EDs
3. compute ID-rewrite of the input query
4. unfold the query computed at previous step
5. evaluate the query over the sources

LAV systems and integrity constraints

can we use these techniques also in LAV systems?

- semantics for LAV systems in the presence of global integrity constraints
- comparison with GAV
- the equality problem
- decidability

Semantics for LAV systems under ICs

- we refer only to databases over a fixed infinite domain Γ
- observation: under the sound assumption for the mapping, the whole integration system corresponds to a FOL theory!
- the semantics is given by the FOL models of such a theory

Semantics for LAV systems under ICs

given a source database $\mathcal{C}$ for a LAV system $\mathcal{I}$, a global database $\mathcal{B}$ is **legal** if $\mathcal{B} \cup \mathcal{C}$ is a model of the FOL theory corresponding to $\mathcal{I} \cup \mathcal{C}$

more precisely:

- theory corresponding to $\mathcal{C}$ = set of ground atoms
- the mapping $\mathcal{M}$ corresponds to a set of FOL sentences
- each IC in $\mathcal{G}$ corresponds to a FOL sentence

(see also previous lectures)

LAV systems under IDs

if the only global ICs are IDs:

- it is possible to turn the LAV mapping into a GAV mapping
- more precisely: transformation of a LAV integration system with IDs $\mathcal{I} = (\mathcal{G}, \mathcal{S}, \mathcal{M})$ into a GAV system $\mathcal{I}' = (\mathcal{G}', \mathcal{S}, \mathcal{M}')$
- with respect to $\mathcal{I}$, the transformed system $\mathcal{I}'$ contains auxiliary IDs and auxiliary global relation symbols
- the transformation is query-preserving:

  for every CQ q and for every source database $\mathcal{C}$, the certain answers to q in $(\mathcal{I}, \mathcal{C})$ are equal to the certain answers to q in $(\mathcal{I}', \mathcal{C})$

Transforming LAV into GAV: example

initial LAV mapping:

$$s(X, Y) \ \text{:-}\ r_1(X, Z), r_2(Y, W)$$

$$t(X, Y) \ \text{:-}\ r_1(X, Z), r_3(Y, X)$$

transformed GAV mapping:

$$s_i(X, Y) \ \text{:-}\ s(X, Y)$$

$$t_i(X, Y) \ \text{:-}\ t(X, Y)$$

additional IDs generated by the transformation: ($s_e/4$, $t_e/3$)

$$s_i[1, 2] \subseteq s_e[1, 2] \qquad s_e[1, 3] \subseteq r_1[1, 2]$$

$$s_e[2, 4] \subseteq r_2[1, 2] \qquad t_i[1, 2] \subseteq t_e[1, 2]$$

$$t_e[1, 3] \subseteq r_1[1, 2] \qquad t_e[2, 1] \subseteq r_3[1, 2]$$

Query answering in LAV systems under IDs

method for query answering in LAV system $\mathcal{I}$ with IDs:

1. transform $\mathcal{I}$ into a GAV system $\mathcal{I}'$
2. apply the query answering method for GAV systems under IDs

   (the unfolding step must be slightly changed due to the presence of auxiliary global symbols)

LAV systems under IDs and EDs

what happens if we have also EDs in the global schema?

- the above transformation of LAV into GAV is still correct in the presence of EDs
- it is thus possible to first turn the LAV system into a GAV one and then compute query answering in the transformed system
- the addition of EDs is completely modular (we just need to add auxiliary steps in the query answering technique)

Query answering in LAV systems under IDs and EDs

method for query answering in LAV system $\mathcal{I}$ with IDs and EDs:

1. transform $\mathcal{I}$ into a GAV system $\mathcal{I}'$
2. apply the query answering method for GAV systems under IDs and EDs

   (the unfolding step must be slightly changed due to the presence of auxiliary global symbols)

LAV systems and KDs

what happens in LAV systems with KDs in the global schema?

we consider a LAV system with only KDs:

- the transformation of LAV into GAV is still correct in the presence of KDs
- more precisely, starting from a LAV system $\mathcal{I}$ with KDs we obtain a GAV system $\mathcal{I}'$ with KDs and IDs
- but in general $\mathcal{I}'$ is such that the IDs added by the transformation are key-conflicting IDs
- i.e., these IDs are not NKCIDs

  $\Rightarrow$ KDs and IDs in $\mathcal{I}'$ are not separable

LAV systems and KDs

- therefore, it is not possible to apply the query answering method for LAV systems under separable KDs and IDs
- can we find some analogous query answering method based on query rewriting?

A negative result

- problem: KDs and LAV mappings derive new equality-generating dependencies (not simple KDs)
- (Duschka et al., 1998): we cannot do query answering by FOL query reformulation in LAV systems under KDs
- i.e., we cannot find a first-order rewriting of a CQ in LAV systems under KDs, because it does not exist!
- we have to resort to more powerful relational query languages (e.g., Datalog)

Summary

query answering in integration systems by first-order (UCQ) rewriting?

- GAV, IDs + EDs: yes
- GAV, IDs + KDs + EDs: only if KDs and IDs are separable
- LAV, IDs + EDs: yes
- LAV, KDs: no

Inconsistency tolerance

under the “classical” (i.e., first-order) semantics considered so far:

- if data at the sources violate (through the mapping) a single KD or ED, the integration system has no legal databases (i.e., no models)
- consequently, the certain answers to any query of arity n are all the n-tuples of constants of Γ (ex falso quodlibet)
- non-interesting case for query answering

Inconsistency tolerance: example

example:

$$\mathcal{G} = \{r/2,\ key(r) = \{1\}\}, \quad \mathcal{S} = \{s/2, t/2\}$$

$$\mathcal{M} = \{r(X,Y)\ \text{:-}\ s(X,Y),\ \ r(X,Y)\ \text{:-}\ t(X,Y)\}$$

$$\mathcal{C} = \{s(a,b), t(a,c)\}$$

$$ret(\mathcal{I},\mathcal{C}) = \{r(a,b), r(a,c)\}$$

$$q(X)\ \text{:-}\ r(X,Y)$$

there is no legal databases for $(\mathcal{I},\mathcal{C})$ ($ret(\mathcal{I},\mathcal{C})$ violates the KD on r)

$\Rightarrow cert(q,\mathcal{I},\mathcal{C}) = \{c \mid c \in \Gamma\}$

however: we would like the only certain answer to q to be a

Consistent query answering (CQA)

- study of methods and techniques for “repairing” a database instance that is inconsistent with the integrity constraints declared on its schema [Arenas et al., 2000]
- peculiarity of CQA: repair is virtual and based on a logical/declarative semantics
- $\Rightarrow$ data are not changed, not a material repair (as in data cleaning)
- the CQA principles and methods can be extended to data integration scenarios
- in the following, we only consider GAV mapping

The loosely-sound semantics

here we introduce one particular semantics for inconsistency tolerance in GAV integration systems, the **loosely-sound semantics** [Calì et al., 2005]

- the loosely-sound semantics principle: add as much as you like (as with sound semantics), and throw away only a minimal set of tuples

Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be two global databases that satisfy constraints on the global schema. Then, $\mathcal{B}_1$ is **better** than $\mathcal{B}_2$, denoted $\mathcal{B}_1 \gg_{(\mathcal{I},\mathcal{C})} \mathcal{B}_2$, iff

$$\mathcal{B}_1 \cap ret(\mathcal{I},\mathcal{C}) \supset \mathcal{B}_2 \cap ret(\mathcal{I},\mathcal{C})$$

The answers $cert_\ell(Q,\mathcal{I},\mathcal{C})$ to a query are those that are true on **all** "best" legal global databases w.r.t. $\gg_{(\mathcal{I},\mathcal{C})}$

Example

Global schema: player$(Pname, YOB, Pteam)$

team$(Tname, Tcity, Tleader)$

Constraints: team$[Tleader, Tname] \subseteq$ player$[Pname, Pteam]$

$key(\textsf{player}) = \{Pname\}$

Mapping:

$$\textsf{player} \rightsquigarrow \begin{cases} \textsf{player}(X, Y, Z) \leftarrow \textsf{s}_1(X, Y, Z) \\ \textsf{player}(X, Y, Z) \leftarrow \textsf{s}_3(X, Y, Z) \end{cases}$$

$$\textsf{team} \rightsquigarrow \textsf{team}(X, Y, Z) \leftarrow \textsf{s}_2(X, Y, Z)$$

Example (cont'd)

Source database $\mathcal{C}$

s_1:

Totti	1971	Roma
Vieri	1950	Inter

s_2:

Juve	Torino	Del Piero

s_3:

Vieri	1970	Inter

Example (IDs – KDs)

Retrieved global database $ret(\mathcal{I}, \mathcal{C})$

player:

Totti	1971	Roma
Vieri	1970	Inter
Vieri	1950	Inter

team:

Juve	Torino	Del Piero

in $ret(\mathcal{I}, \mathcal{C})$ there is a violation of the KD and a violation of the ID

there are two possible ways of repairing the violation of the KD with a minimum deletion of tuples:

Example (cont'd)

First form

player:

Totti	1971	Roma
Vieri	1970	Inter
Del Piero	α	Juve

team:

Juve	Torino	Del Piero

Second form

player:

Totti	1971	Roma
Vieri	1950	Inter
Del Piero	α	Juve

team:

Juve	Torino	Del Piero

Consider again the query $\mathsf{q}(X, Z) \leftarrow \mathsf{player}(X, Y, Z)$: we obtain

$$cert_{\ell}(\mathsf{q}, \mathcal{I}, \mathcal{C}) = \{\langle \mathsf{Totti}, \mathsf{Roma} \rangle, \langle \mathsf{Vieri}, \mathsf{Inter} \rangle, \langle \mathsf{Del\ Piero}, \mathsf{Juve} \rangle\}$$

Query rewriting under the loosely-sound semantics

query language: Datalog$^\neg$ under stable model semantics

Rewriting under KDs: set of rules Π_{KD} that take KDs into account

for each KD $key(r) = \{X_1, \ldots X_n\}$ in $\mathcal{G}$:

$$
\begin{array}{l}
r(\mathbf{x}, \mathbf{y}) \leftarrow r_{\mathcal{C}}(\mathbf{x}, \mathbf{y}) \ , \ not \ \overline{r}(\mathbf{x}, \mathbf{y}) \\
\overline{r}(\mathbf{x}, \mathbf{y}) \leftarrow r_{\mathcal{C}}(\mathbf{x}, \mathbf{y}) \ , \ r(\mathbf{x}, \mathbf{z}) \ , \ Y_1 \neq Z_1 \\
\cdots \\
\overline{r}(\mathbf{x}, \mathbf{y}) \leftarrow r_{\mathcal{C}}(\mathbf{x}, \mathbf{y}) \ , \ r(\mathbf{x}, \mathbf{z}) \ , \ Y_m \neq Z_m
\end{array}
$$

where $\mathbf{x} = X_1, \ldots, X_n, \mathbf{y} = Y_1, \ldots, Y_m$ and $\mathbf{z} = Z_1, \ldots, Z_m$

Query rewriting under the loosely-sound semantics

Theorem: $\Pi_{ID} \cup \Pi_{KD} \cup \Pi_{\mathcal{MC}}$ is a perfect rewriting of q

where:

- $\Pi_{\mathcal{MC}}$ = rules obtained from the mapping rules $\Pi_{\mathcal{M}}$ by replacing each r with $r_{\mathcal{C}}$
- Π_{ID} = rewriting of the query q obtained by the algorithm ID-rewrite

Remark: $\Pi_{ID} \cup \Pi_{KD} \cup \Pi_{\mathcal{MC}}$ is a Datalog$^{\neg}$ program (and is interpreted under stable model semantics)

What about EDs?

We extend the previous example with an ED:

Global schema: $\mathsf{player}(Pname, YOB, Pteam), \mathsf{team}(Tname, Tcity, Tleader)$

$\mathsf{coach}(Cname, Cteam)$

Constraints: $\mathsf{team}[Tleader,\ Tname] \subseteq \mathsf{player}[Pname,\ Pteam]$

$\mathsf{coach}[Cname] \cap \mathsf{player}[Pname] = \emptyset$

$key(player) = \{Pname, Pteam\}$

$key(team) = \{Tname\} \quad key(coach) = \{Cname\}$

Mapping: $\mathsf{player}(X, Y, Z) \leftarrow \mathsf{s}_1(X, Y, Z)$

$\mathsf{player}(X, Y, Z) \leftarrow \mathsf{s}_3(X, Y, Z)$

$\mathsf{team}(X, Y, Z) \leftarrow \mathsf{s}_2(X, Y, Z)$

$\mathsf{coach}(X, Y) \leftarrow \mathsf{s}_4(X, Y)$

Example (cont'd)

Source database $\mathcal{C}$

s_1:

Totti	1971	Roma

s_2:

Juve	Torino	Del Piero

s_3:

Vieri	1970	Inter

s_4:

Del Piero	Viterbese

Example IDs – EDs

Retrieved global database $ret(\mathcal{I}, \mathcal{C})$

player:

Totti	1971	Roma
Vieri	1970	Inter

team:

Juve	Torino	Del Piero

coach:

Del Piero	Viterbese

There are two possible ways of repairing the violation with a minimum deletion of tuples: $\Rightarrow$

Example (cont'd)

First form

player:

Totti	1971	Roma
Vieri	1950	Inter
Del Piero	α	Juve

team:

Juve	Torino	Del Piero

Second form

player:

Totti	1971	Roma
Vieri	1950	Inter

coach:

Del Piero	Viterbese

for the query $\mathsf{q}(X, Z) \leftarrow \mathsf{player}(X, Y, Z)$

$$cert_\ell(\mathsf{q}, \mathcal{I}, \mathcal{C}) = \{\langle \mathsf{Totti}, \mathsf{Roma} \rangle, \langle \mathsf{Vieri}, \mathsf{Inter} \rangle\}$$

Query rewriting for EDs

set of rules Π_{ED} that take EDs into account

for each exclusion dependency $r[\mathbf{A}] \cap s[\mathbf{B}] = \emptyset$ in the closure of EDs wrt logical implication by IDs and EDs:

$$
\begin{array}{rcl}
r(\mathbf{x}, \mathbf{y}) & \leftarrow & r_{\mathcal{C}}(\mathbf{x}, \mathbf{y}) \ , \ not \ \overline{r}(\mathbf{x}, \mathbf{y}) \\
s(\mathbf{x}, \mathbf{y}) & \leftarrow & s_{\mathcal{C}}(\mathbf{x}, \mathbf{y}) \ , \ not \ \overline{s}(\mathbf{x}, \mathbf{y}) \\
\overline{r}(\mathbf{x}, \mathbf{y}) & \leftarrow & r_{\mathcal{C}}(\mathbf{x}, \mathbf{y}) \ , \ s(\mathbf{x}, \mathbf{z}) \\
\overline{s}(\mathbf{x}, \mathbf{y}) & \leftarrow & s_{\mathcal{C}}(\mathbf{x}, \mathbf{y}) \ , \ r(\mathbf{x}, \mathbf{z})
\end{array}
$$

where in $r(\mathbf{x}, \mathbf{z})$ the variables in $\mathbf{x}$ correspond to the sequence of attributes $\mathbf{A}$ of r, and in $s(\mathbf{x}, \mathbf{z})$ the variables in $\mathbf{x}$ correspond to the sequence of attributes $\mathbf{B}$ of s.

Query rewriting under the loosely-sound semantics:

IDs, KDs and EDs

Theorem: $\Pi_{ID} \cup \Pi_{KD} \cup \Pi_{ED} \cup \Pi_{\mathcal{MC}}$ is a perfect rewriting of Q.

Example

Global schema: player$(Pname, YOB, Pteam)$, team$(Tname, Tcity, Tleader)$

coach$(Cname, Cteam)$

Constraints: team$[Tleader,\ Tname] \subseteq$ player$[Pname,\ Pteam]$

coach$[Cname] \cap$ player$[Pname] = \emptyset$

$key(player) = \{Pname, Pteam\}$

$key(team) = \{Tname\} \quad key(coach) = \{Cname\}$

Mapping: player$(X, Y, Z) \leftarrow$ s$_1(X, Y, Z)$

player$(X, Y, Z) \leftarrow$ s$_3(X, Y, Z)$

team$(X, Y, Z) \leftarrow$ s$_2(X, Y, Z)$

coach$(X, Y) \leftarrow$ s$_4(X, Y)$

Query: q$(X, Z) \leftarrow$ player(X, Y, Z)

Example (cont'd)

rewriting of the query q:

$$
\begin{array}{rcl}
\mathsf{q}(X,Z) & \leftarrow & \mathsf{player}(X,Y,Z) \\
\mathsf{q}(X,Z) & \leftarrow & \mathsf{team}(Z,Y,X) \\
\mathsf{player}(X,Y,Z) & \leftarrow & \mathsf{player}_{\mathcal{C}}(X,Y,Z)\ ,\ not\ \overline{\mathsf{player}}(X,Y,Z) \\
\overline{\mathsf{player}}(X,Y,Z) & \leftarrow & \mathsf{player}_{D}(X,Y,Z)\ ,\ \mathsf{player}(X,W,Z)\ ,\ Y \neq W \\
\mathsf{team}(X,Y,Z) & \leftarrow & \mathsf{team}_{\mathcal{C}}(X,Y,Z)\ ,\ not\ \overline{\mathsf{team}}(X,Y,Z) \\
\overline{\mathsf{team}}(X,Y,Z) & \leftarrow & \mathsf{team}_{\mathcal{C}}(X,Y,Z)\ ,\ \mathsf{team}(X,V,W)\ ,\ Y \neq V \\
\overline{\mathsf{team}}(X,Y,Z) & \leftarrow & \mathsf{team}_{\mathcal{C}}(X,Y,Z)\ ,\ \mathsf{team}(X,V,W)\ ,\ Z \neq W \\
\mathsf{coach}(X,Y) & \leftarrow & \mathsf{coach}_{\mathcal{C}}(X,Y)\ ,\ not\ \overline{\mathsf{coach}}(X,Y) \\
\overline{\mathsf{coach}}(X,Y) & \leftarrow & \mathsf{coach}_{D}(X,Y)\ ,\ \mathsf{coach}(X,Z)\ ,\ Y \neq Z
\end{array}
$$

Example (continued)

rewriting of the query q (continued):

$$\begin{array}{rcl}
\overline{\mathsf{player}}(X,Y,Z) & \leftarrow & \mathsf{player}_{\mathcal{C}}(X,Y,Z)\ ,\ \mathsf{coach}(X,V) \\
\overline{\mathsf{coach}}(X,Y) & \leftarrow & \mathsf{coach}_{\mathcal{C}}(X,Y)\ ,\ \mathsf{player}(X,Z,V) \\
\overline{\mathsf{coach}}(X,Y) & \leftarrow & \mathsf{coach}_{\mathcal{C}}(X,Y)\ ,\ \mathsf{team}(Z,V,X) \\
\overline{\mathsf{team}}(X,Y,Z) & \leftarrow & \mathsf{team}_{\mathcal{C}}(X,Y,Z)\ ,\ \mathsf{coach}(Z,V) \\
\mathsf{player}_{\mathcal{C}}(X,Y,Z) & \leftarrow & \mathsf{s}_1(X,Y,Z) \\
\mathsf{player}_{\mathcal{C}}(X,Y,Z) & \leftarrow & \mathsf{s}_3(X,Y,Z) \\
\mathsf{team}_{\mathcal{C}}(X,Y,Z) & \leftarrow & \mathsf{s}_2(X,Y,Z) \\
\mathsf{coach}_{\mathcal{C}}(X,Y) & \leftarrow & \mathsf{s}_4(X,Y)
\end{array}$$

Complexity of consistent query answering

- with respect to standard strictly-sound semantics, the loosely-sound semantics adds complexity to query answering
- intuitive explanation: the number of repairs of a system with even a single KD may be exponential in the size of the source database
- consequence: query answering is not tractable in data complexity (while it is tractable under strictly-sound semantics)
- such increase of complexity is general (shared by all approaches to consistent query answering)

Summary of complexity results

EDs	KDs	IDs	strictly-sound	loosely-sound
no	no	GEN	PTIME/PSPACE	PTIME/PSPACE
yes-no	yes	no	PTIME/NP	coNP/Π_2^p
yes	yes-no	no	PTIME/NP	coNP/Π_2^p
yes-no	yes	NKC	PTIME/PSPACE	coNP/PSPACE
yes	no	GEN	PTIME/PSPACE	coNP/PSPACE
yes-no	yes	1KC	undecidable	undecidable
yes-no	yes	GEN	undecidable	undecidable

Data/Combined complexity

Some references

[Arenas et al., 2000] M. Arenas, L. E. Bertossi, J. Chomicki. *Consistent query answers in inconsistent databases*. PODS 1999.

[Bravo & Bertossi, 2003] L. Bravo and L. Bertossi. *Logic programming for consistently querying data integration systems*. IJCAI 2003.

[Calì et al., 2002] Andrea Calì, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini. *On the Expressive Power of Data Integration Systems*. ER 2002

[Calì et al., 2003] Andrea Calì, Domenico Lembo, Riccardo Rosati. *On the decidability and complexity of query answering over inconsistent and incomplete databases*. PODS 2003.

[Calì et al., 2003b] Andrea Calì, Domenico Lembo, Riccardo Rosati. *Query rewriting and answering under constraints in data integration systems*. IJCAI 2003.

Some references

[Calvanese & Rosati, 2003] Diego Calvanese, Riccardo Rosati. *Answering Recursive Queries under Keys and Foreign Keys is Undecidable*. KRDB 2003.

[Faber et al., 2005] W. Faber, G. Greco, and N. Leone. *Magic sets and their application to data integration*. ICDT 2005.

[Fuxman et al., 2005] A. Fuxman, R. J. Miller. *First-order query rewriting for inconsistent databases*. ICDT 2005.

[Leone et al., 2005] Nicola Leone et al., *The INFOMIX system for advanced integration of incomplete and inconsistent data*. SIGMOD 2005 (demo), SEBD 2005.

INFOMIX web site: `http://sv.mat.unical.it/infomix`