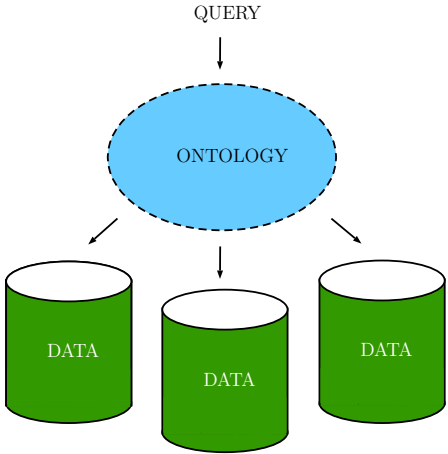
Ontology-Mediated Query Answering and Heterogeneous Data

Federico Ulliana

Université de Montpellier, LIRMM, INRIA GraphIK

Joint work with
Marie-Laure Mugnier, Marie-Christine Rousset, Michel Leclère,
Meghyn Bienvenu, Pierre Bourhis, Sophie Tison

Ontology-Mediated Query Answering



Ontological knowledge :

1. Enriched query vocabulary
2. Incomplete information
3. Unified view of data

Applications : Data-Integration, Semantic Web (see next talks)

Ontology-Mediated Query Answering

Fact : "*Alice* is a **Professor**"

Query : "does *Alice* have a **Contact** ?"

Ontology-Mediated Query Answering

Fact : "*Alice* is a **Professor**"

Query : "does *Alice* have a **Contact** ?" ✗

Ontology-Mediated Query Answering

Know. : "Every **Professor** has a **Contact**" Fact : "*Alice* is a **Professor**"

Query : "does *Alice* have a **Contact** ?"

Ontology-Mediated Query Answering

Know. : "Every **Professor** has a **Contact**" Fact : "*Alice* is a **Professor**"

Query : "does *Alice* have a **Contact** ?" ✓

Ontology-Mediated Query Answering

Know. : "Every **Professor** has a **Contact**" Fact : "*Alice* is a **Professor**"

Query : "does *Alice* have a **Contact** ?" ✓

This approach applies to scientific data (e.g., medicine, biology, chemistry), bibliographical data, governmental (open) data

Knowledge Base := (Database,Ontology)

Ontologies model (some aspects of) the world

- Introduce vocabulary relevant to a domain
  - (typically) Classes : **Professor**, **Contact** Relations : is a, has a
    Hierarchies : **Professor** $\leq$ **Faculty** $\leq$ **Person**

- Formally specifies (rich) semantics and meaning of the vocabulary
  - (typically) in terms of Rules
    "Every **Professor** has a **Contact**"
    "**Professors** are disjoint from **Students**"
    "Every **Class** it taught by a single **Professor**"

Ontologies vs. Database Schemas

Ontology $\sim$ DB schema : describe structure and constraints on data

Schema define legal DB states

Ontologies entail implicit facts

DB schemas do not influence query answering (beside optim.), ontologies do.

Closed world assumption (CWA)

- Missing information $\sim$ false

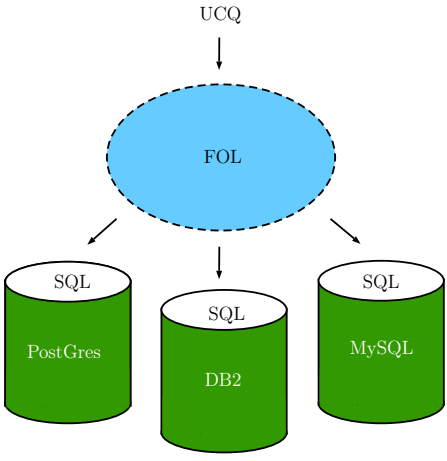
Open world assumption (OWA)

- Missing inform. $\sim$ unknown

OMQA and Heterogeneous Data

- Relational Data
- RDF Data (very partially covered today)
- NOSQL Data

OMQA for Relational Databases



- Relational technology ubiquitous

  Boost for OMQA investigations

- Ontologies : Description Logics, Existential Rules

Logical View of Databases

teach	
alice	Logics
alice	DB
bob	z_0

prof
alice
charles

$$\exists z_0 \Big(\; teach(alice, Logics) \wedge prof(alice) \wedge teach(alice, DB) \wedge prof(charles) \wedge teach(bob, z_0) \; \Big)$$

$\pi_{[1]}(\texttt{teach}) \bowtie \texttt{prof}$

$$Q(x) :- \; \exists x, y \left(teaches(x, y) \wedge \mathsf{Professor}(x) \right)$$

$\pi_{\emptyset}(\sigma_{[1=\texttt{bob}]}\texttt{teach})$

$$Q_{bool}() :- \; \exists x \left(teaches(bob, x) \right)$$

Theorem (Codd'72)

Relational Algebra $\approx$ *FO without function symbols*

Homomorphism and Query Answering

Variable substitution $h : Q \longrightarrow F$

$Q(x) :- \ \exists x, y \left(teaches(x, y) \wedge \mathsf{Professor}(x) \right)$ $\quad Q_{bool} :- \ \exists x \left(teaches(bob, x) \right)$

$h_1 = \{x \mapsto alice, y \mapsto Logics\}$ $\quad h_3 = \{x \mapsto z_0\}$

$h_2 = \{x \mapsto alice, y \mapsto DB\}$

$Answ(Q, F) = \{(Alice)\}$ $\quad Answ(Q_{bool}, F) =$ "Yes"

$$F \models Q$$

Ontology Languages

Description Logics

- DL-Lite [Calvanese et al., 2007]
- EL [Baader et al., 2005]
- Horn DLs [Grosof et al., 2003]
- Foundations of OWL 2 Profiles [Hitzler et al., 2009]

Positive Existential Rules [Baget et al., 2009]

- Same as Tuple Generating Dependencies (TGDs) [Beeri and Vardi, 1984]
- See also Datalog+/- [Cali et al., 2010]
- See Conceptual Graphs [Sowa, 1984, Chein and Mugnier, 1992]
- Generalize Horn DLs

Positive Existential Rules

$$\forall \vec{x}, \vec{y} \left(\mathsf{Body}(\vec{x}, \vec{y}) \longrightarrow \exists \vec{z}\ \mathsf{Head}(\vec{y}, \vec{z}) \right)$$

Body, Head positive conjunctions of atoms, without function symbols.

$$\forall x \Big(\mathsf{Professor}(x) \longrightarrow \exists Z\ \mathsf{hasContact}(x, Z) \wedge \mathsf{Contact}(Z) \Big)$$

Existentially quantified variables enable "value invention".

Generalize DL-Lite / OWL 2 QL and RDF(S)

DL-Lite Axioms	Existential Rules
$A \sqsubseteq B$	$\forall x\big(\mathsf{A}(x) \longrightarrow \mathsf{B}(x)\big)$
$A \sqsubseteq \exists R$	$\forall x\big(\mathsf{A}(x) \longrightarrow \exists z.\mathsf{R}(x,z)\big)$
$\exists R \sqsubseteq A$	$\forall xy\big(\mathsf{R}(x,y) \longrightarrow \mathsf{A}(x)\big)$
$\exists R \sqsubseteq \exists P$	$\forall xy\big(\mathsf{R}(x,y) \longrightarrow \exists z.\mathsf{P}(x,z)\big)$
$A \sqsubseteq \exists R.B$	$\forall x\big(\mathsf{A}(x) \longrightarrow \exists z.\mathsf{R}(x,z) \wedge \mathsf{B}(z)\big)$
$P \sqsubseteq Q$	$\forall xy\big(\mathsf{P}(x,y) \longrightarrow \mathsf{Q}(x,y)\big)$
$A \sqsubseteq \neg B$	$\forall x\big(\mathsf{A}(x) \wedge \mathsf{B}(x) \longrightarrow \bot\big)$

Generalize EL Description Logics / OWL 2 EL

EL-Lite Axioms	Existential Rules
$A \sqsubseteq B$	$\forall x\big(\mathsf{A}(x) \longrightarrow \mathsf{B}(x)\big)$
$A \sqcap B \sqsubseteq C$	$\forall x\big(\mathsf{A}(x) \wedge B(x) \longrightarrow \mathsf{C}(x)\big)$
$A \sqsubseteq \exists R.B$	$\forall x\big(\mathsf{A}(x) \longrightarrow \exists z.\mathsf{R}(x,z) \wedge B(z)\big)$
$\exists R.B \sqsubseteq A$	$\forall xy\big(\mathsf{R}(x,y) \wedge \mathsf{B}(y) \longrightarrow \mathsf{A}(x)\big)$

Rule (Model) Semantics

$(F \models \sigma)$ F is a model of σ when any homomorphism $h(\mathsf{Body}(\sigma)) \subseteq F$ can be extended to some $h' \supseteq h$ such that $h'(\mathsf{Head}(\sigma)) \subseteq F$

$$\sigma : \forall x \big(\mathsf{Professor}(x) \longrightarrow \exists Z.\mathsf{hasContact}(x, Z)\big)$$

$$\mathsf{Professor}(\mathit{alice}) \not\models \sigma$$

$$\mathsf{Professor}(\mathit{alice}) \wedge \mathsf{hasContact}(\mathit{alice}, \mathit{5\text{-}252}) \models \sigma$$

Rule Application

A rule σ apply on F if there is an homomorphism $\quad h(\mathsf{Body}(\sigma)) \subseteq F$

Effective application adds $\quad h^{\mathbf{safe}}(\mathsf{Head}(\sigma))$

$$\sigma : \forall x \big(\mathsf{Professor}(x) \longrightarrow \exists Z.\mathsf{hasContact}(x, Z)\big) \qquad F : \mathsf{Professor}(alice)$$

$$h^{\mathbf{safe}} = \{\ x \mapsto alice,\ Z \mapsto z_0\ \}$$

$$\exists z_0 \big(\ \mathsf{Professor}(alice) \wedge \mathsf{hasContact}(alice, z_0)\ \big)$$

The Chase Procedure

Algorithmic tool for constructing universal models (can answer any query).

Until possible$^{(*)}$ apply the rules of Σ on the factbase F.

$$F, \Sigma \models Q \quad \Longleftrightarrow \quad \mathit{Chase}(F, \Sigma) \models Q$$

Saturation (also called materialization) approach.

Why Materializing Data ?

Pro :

- Materialization is independent from queries.

Cons :

- Possible data blowup
- Needs maintenance under updates
- Requires writing access rights to sources

Query Rewriting

Algorithmic tool for computing an equivalent of the input query Q (under Σ).

Until possible$^{(*)}$, propagate the rules of Σ into the query Q.

$$\forall x(\mathsf{Professor}(x) \longrightarrow \exists Z\ \mathsf{hasContact}(x, Z))$$

piece based unification $\updownarrow$ $\updownarrow$

$$Q' :- \mathsf{Professor}(alice) \qquad Q :- \exists w.\mathsf{hasContact}(alice, w)$$

$$Rew(Q, \Sigma) = Q' \vee Q$$

Query Rewriting

$$F, \Sigma \models Q \quad \iff \quad F \models Rew(Q, \Sigma)$$

Duality

$$Chase(F, \Sigma) \models Q \quad \iff \quad F \models Rew(Q, \Sigma)$$

Why Rewriting Queries ?

Pro :

- Rewriting is independent from data
- Resilient to updates
- Deals with read-only sources

Cons :

- Query blowup (exponential in many practical cases)
- Calls for compact rewritings (many investigations)
- Use mixed approaches

Looking for Terminating Conditions

Chase can be infinite

$$\forall x, y\big(\ \mathsf{hasParent}(x, y) \rightarrow \exists z.\mathsf{hasParent}(y, z)\ \big)$$

$$F : \mathsf{hasParent}(alice, bob) \qquad \exists z_0, z_1 \ldots \big(\mathsf{hasParent}(alice, bob) \wedge \mathsf{hasParent}(bob, z_0) \wedge \mathsf{hasParent}(z_0, z_1) \ldots\big)$$

Rewriting can be infinite

$$\forall x, y, z(\mathsf{partOf}(x, y) \wedge \mathsf{partOf}(y, z) \rightarrow \mathsf{partOf}(x, z))$$

$$Q : \mathsf{partOf}(foot, body) \qquad \bigvee_{i=0}^{n} \exists x_0 \ldots x_n\big(\mathsf{partOf}(foot, x_0) \wedge \cdots \wedge \mathsf{partOf}(x_n, body)\big)$$

Finite Expansion and Unification

Σ finite expansion $\quad \forall F.\ \exists k.\ \mathit{Chase}^{(k)}(F, \Sigma) \equiv \mathit{Chase}^{(k+1)}(F, \Sigma)$

$$\forall x, y, z(\mathsf{partOf}(x, y) \wedge \mathsf{partOf}(y, z) \rightarrow \mathsf{partOf}(x, z))$$

Σ finite unification $\quad \forall Q.\ \exists k.\ \mathit{Rew}^{(k)}(Q, \Sigma) \equiv \mathit{Rew}^{(k+1)}(Q, \Sigma)$

$$\forall x, y\big(\mathsf{hasParent}(x, y) \rightarrow \exists z.\mathsf{hasParent}(y, z)\big)$$

Abstract classes for which query answering is decidable.

Boundedness and Optimization [DL'16]

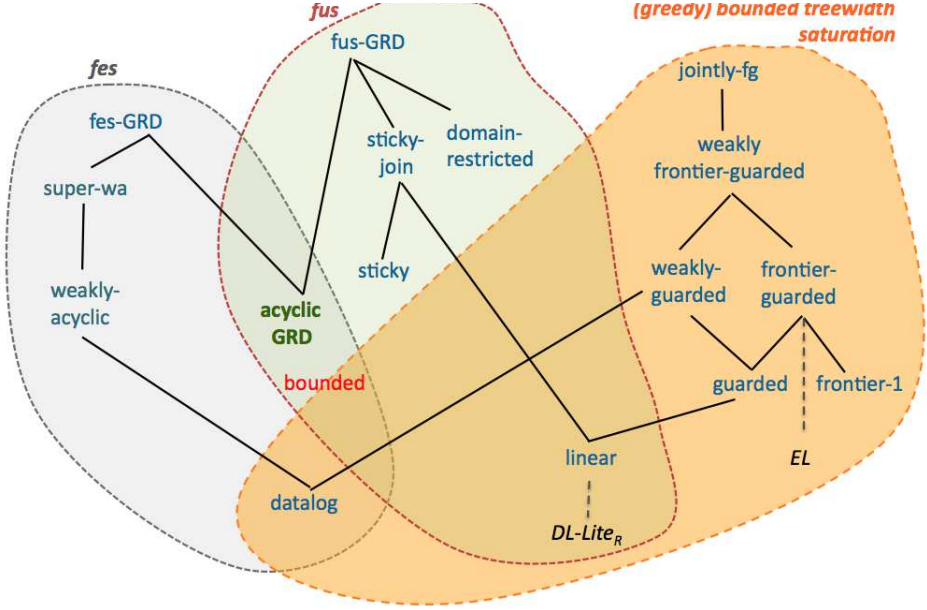
Σ bounded $\qquad \exists \mathbf{k}.\ \forall F.\quad Chase^{(\mathbf{k})}(F, \Sigma) \equiv Chase^{(\mathbf{k}+1)}(F, \Sigma)$

$$\forall x, y \big(\mathsf{knows}(x, y) \longrightarrow \exists z. \mathsf{knows}(y, z), \mathsf{knows}(z, y) \big)$$

$$\mathsf{knows}(alice, bob)$$
$$\downarrow$$
$$\mathsf{knows}(bob, z_0) \wedge \mathsf{knows}(z_0, bob)$$
$$\swarrow \qquad \searrow$$
$$\mathsf{knows}(z_1, z_0) \wedge \mathsf{knows}(z_0, z_1) \quad \wedge \quad \mathsf{knows}(bob, z_2) \wedge \mathsf{knows}(z_2, bob)$$
$$\ldots$$

Theorem

Σ *bounded* $\iff$ Σ *both finite expansion and unification*


fus
(greedy) bounded treewidth saturation
fes
fus-GRD
jointly-fg
fes-GRD
sticky-join
domain-restricted
weakly frontier-guarded
super-wa
sticky
weakly-guarded
frontier-guarded
weakly-acyclic
acyclic GRD
bounded
guarded
frontier-1
linear
EL
datalog
DL-Lite$_R$

Implementations

DL-Lite

- OnTop, Rapid, Iqaros, Presto/Mastro, Requiem, HermiT

EL

- Rapid, ELK

Existential Rules / RDF Rules

- RDFOX **Graal** graphik-team.github.io/graal/

OMQA and Heterogeneous Data

- Relational Data
- RDF Data (very partially covered today)
- NOSQL Data

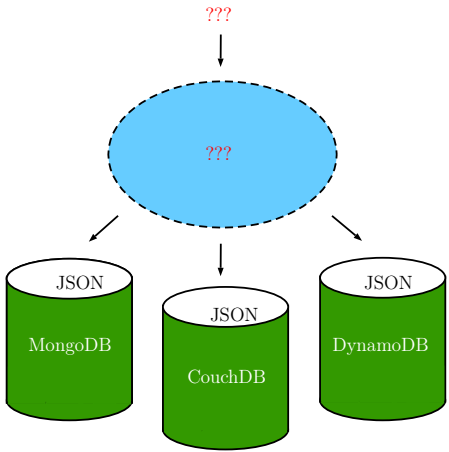
What if data is NOT in relational form ?

Web Data $\sim$ Trees (e.g., JSON key-value records,XML)

Manageable by *efficient* Key-Value stores (e.g., MongoDB, CouchDB)

- Naive solution : relational translation (store optimizations are lost)
- Define (virtual) mappings [Botoeva et al., , Michel et al., 2016]
- Our approach : add ontology on top of data ; do query reformulation

OMQA for Key-Value Records [AAAI-16]



- Reformulate within store native query language

  (saturation not practical)

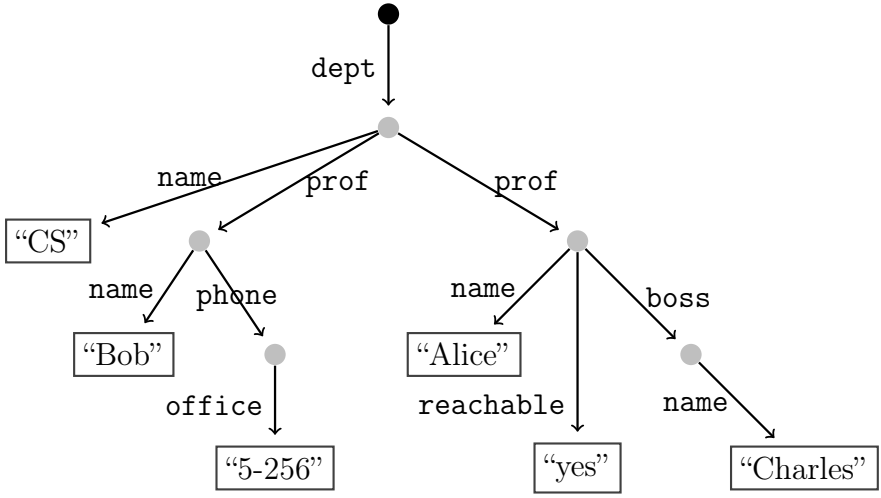
- Which kind of ontological rules can we imagine ?

Key-value Records (JSON Records)

```
{
dept : {
 name : "Computer Science",
 professor : [
                { name : "Alice",   reachable : "yes",   boss : "Charles" }
                { name : "Bob",        phone : {office  : "5-256"}       }  ]
 course : [  ["C123", "Java"]  ,  [ "C310" , "C++" ]  ]  }
}
```

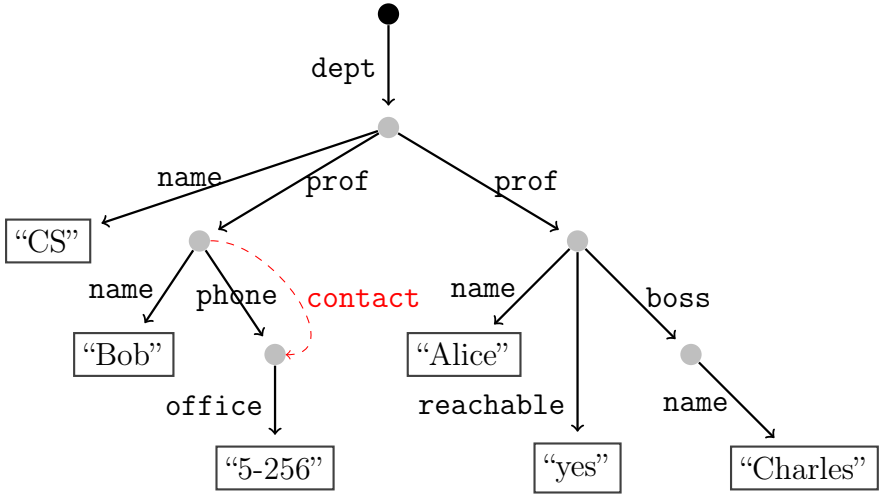
Q_1 : get(dept.prof.contact.office) ✗

Q_2 : check(dept.director) ✗

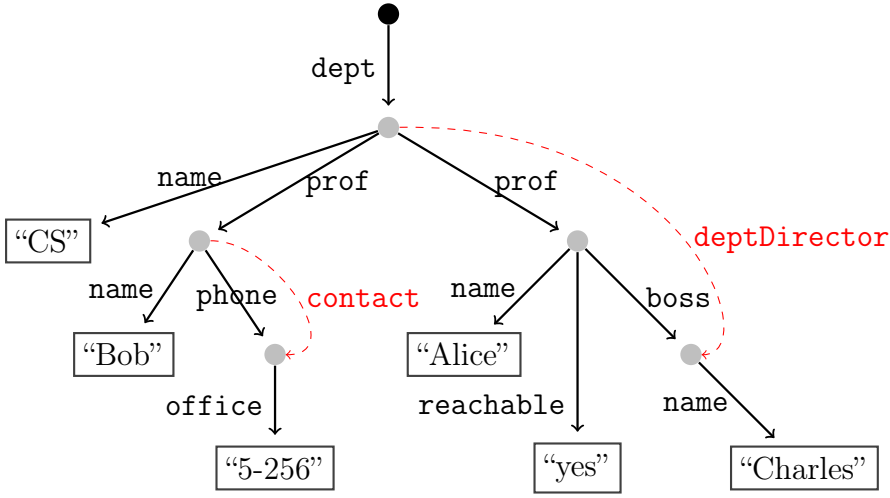


`phone` $\longrightarrow$ `contact`

`professor.boss` $\longrightarrow$ `director`



`phone` $\longrightarrow$ `contact` `professor.boss` $\longrightarrow$ `director`



`phone` $\longrightarrow$ `contact` `professor.boss` $\longrightarrow$ `director`

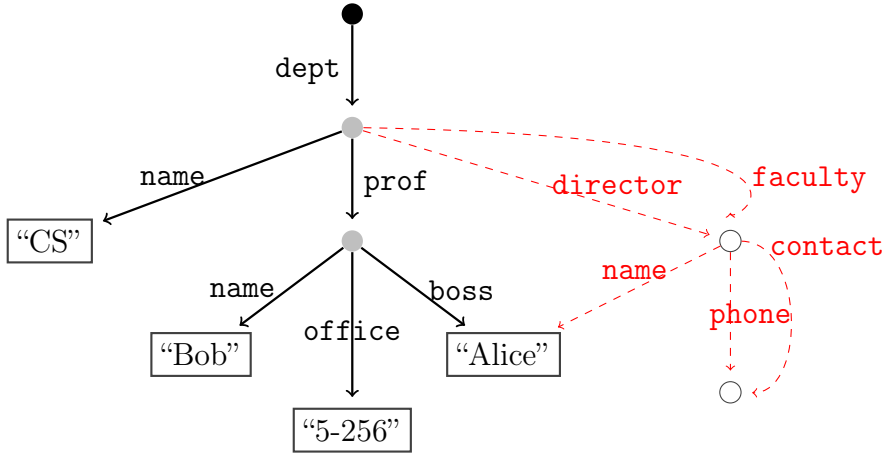
Contextual Pointed Path-Rules [IJCAI-17]

QA soon undecidable for Path-Rules. Need novel useful decidable fragments.

	$k_1 \longrightarrow k_2$	(inclusion between keys)
(*)	$K_1.value \longrightarrow K_2$	(inclusion between valued paths)
	$K_1 \longrightarrow \exists K_2$	(mandatory path)

Moreover, rules are equipped with a *context* for selective application.

$$\mathbf{C} \ : \ K_1[.value] \longrightarrow [\exists]K_2$$



$\texttt{dept} : \texttt{prof.boss}.\mathit{val} \rightarrow \texttt{director.name}$

$\texttt{dept} : \texttt{director} \rightarrow \texttt{faculty}$

$\texttt{dept.faculty} : \epsilon \rightarrow \exists\,\texttt{phone}$

$\texttt{phone} \rightarrow \texttt{contact}$

Query Reformulation

Q : check(`dept.faculty.contact`)

Q' : check(`dept.faculty.phone`)

Q'' : check(`dept.faculty`)

Q''' : check(`dept.director`)

Q''' : check(`dept.prof.boss`)

`phone` $\longrightarrow$ `contact`

`dept.faculity` : $\epsilon \longrightarrow$ `phone`

`dept` : `director` $\longrightarrow$ `faculty`

`dept` : `prof.boss.`$val \rightarrow$ `director.name`

Query answering reduces to a (suffix) word rewriting problem.

Query Answering

Theorem

- *$Rew(Q, \Sigma)$ is a regular language*
- *Query Answering is NLSpace (data-compl.) for* Contextual Path-Rules

Related Works - Reasoning on Nested Structures

- Path Constraints [Abiteboul and Vianu, 1999, Buneman et al., 2000, Calvanese et al., 2016]
- Frame Logic [Kifer et al., 1995]
- Elog rule language [Baumgartner et al., 2001]
- Active XML [Abiteboul et al., 2004]

We propose an OMQ approach highly biased towards query reformulation.

Perspectives

OMQA well studied for Relational and RDF databases, but still many open questions.

Next big challenges :

- **Heterogeneity** : OMQA beyond relational data and query languages (e.g., NOSQL, Graph Databases, Streaming Data)

- **Federation** : OMQA on collections of possibly heterogeneous datasources (polystores, single-ontology/multi-ontology)

References I


Abiteboul, S., Benjelloun, O., and Milo, T. (2004).
Positive active xml.
In *Proceedings of the Twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*.


Abiteboul, S. and Vianu, V. (1999).
Regular path queries with constraints.
Journal of Computer and System Sciences, 58(3):428–452.


Baader, F., Brand, S., and Lutz, C. (2005).
Pushing the el envelope.
In *In Proc. of IJCAI 2005*.


Baget, J.-F., Leclère, M., Mugnier, M.-L., and Salvat, E. (2009).
Extending decidable cases for rules with existential variables.
In *IJCAI'09: 21st International Joint Conference on Artificial Intelligence*, pages 677–682. AAAI.


Baumgartner, R., Flesca, S., Gottlob, G., and Koch, C. (2001).
Visual web information extraction with lixto.
In *VLDB*.


Beeri, C. and Vardi, M. Y. (1984).
A proof procedure for data dependencies.
J. ACM, 31(4):718–741.

References II


Botoeva, E., Calvanese, D., Cogrel, B., Rezk, M., and Xiao, G.
Obda beyond relational dbs: A study for mongodb.
birth, 1926:08–27.


Buneman, P., Fan, W., and Weinstein, S. (2000).
Path constraints in semistructured databases.
J. Comput. Syst. Sci., 61:146–193.


Cali, A., Gottlob, G., Lukasiewicz, T., Marnette, B., and Pieris, A. (2010).
Datalog+/-: A family of logical knowledge representation and query languages for new applications.
In *2010 25th Annual IEEE Symposium on Logic in Computer Science*, pages 228–242.


Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., and Rosati, R. (2007).
Tractable reasoning and efficient query answering in description logics: The dl-lite family.
Journal of Automated reasoning, 39(3):385–429.


Calvanese, D., Ortiz, M., and vSimkus, M. (2016).
Verification of evolving graph-structured data under expressive path constraints.
In *19th International Conference on Database Theory*.


Chein, M. and Mugnier, M.-l. (1992).
Conceptual graphs: fundamental notions.

References III


Grosof, B. N., Horrocks, I., Volz, R., and Decker, S. (2003).
Description logic programs: combining logic programs with description logic.
In *Proceedings of the 12th international conference on World Wide Web*, pages 48–57. ACM.


Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P. F., and Rudolph, S., editors (27 October 2009).
OWL 2 Web Ontology Language: Primer.
W3C Recommendation.
Available at http://www.w3.org/TR/owl2-primer/.


Kifer, M., Lausen, G., and Wu, J. (1995).
Logical foundations of object-oriented and frame-based languages.
Journal of the ACM.


Michel, F., Faron-Zucker, C., and Montagnat, J. (2016).
A mapping-based method to query mongodb documents with sparql.
In *International Conference on Database and Expert Systems Applications*, pages 52–67. Springer.


Sowa, J. (1984).
Conceptual structures: information processing in mind and machine.