Ontologies and Large Databases
Querying Algorithms for the Web of Data

Jean-François Baget (baget@lirmm.fr)
Artificial Intelligence meets the Web of Data, Montpellier, 2013
ONTOMETRY-BASED DATA ACCESS FOR THE WEB OF DATA
Ontology-based Query Answering

Knowledge Base

Data/Facts

Ontology

Query

Answers?
Data / Facts

Relational Database

<table>
<thead>
<tr>
<th>parentOf</th>
<th>Male</th>
<th>Fem.</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>b</td>
</tr>
<tr>
<td>a</td>
<td>c</td>
<td>a</td>
</tr>
<tr>
<td>c</td>
<td>?</td>
<td></td>
</tr>
</tbody>
</table>

RDF (Semantic Web)

<table>
<thead>
<tr>
<th>F</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdf:type</td>
<td>rdf:type</td>
</tr>
<tr>
<td>ex:a</td>
<td>ex:b</td>
</tr>
<tr>
<td>ex:a</td>
<td>ex:c</td>
</tr>
<tr>
<td>ex:b</td>
<td></td>
</tr>
</tbody>
</table>

Abstraction in first-order logic

\[ \exists x ( \text{parentOf}(a,b) \land \text{parentOf}(a,c) \land \text{parentOf}(c,x) \land F(a) \land M(b) ) \]

Or in graphs / hypergraphs

A conjunction of facts ≈ a fact
A (boolean) CQ and a fact have the same form
Ontology: Existential Rules

\[ \forall X \forall Y ( B[X, Y] \rightarrow \exists Z H[X, Z] ) \]

- **Body:** \[ B[X, Y] \]
- **Head:** \[ \exists Z H[X, Z] \]

**X, Y, Z:** tuples of variables

Any conjunction of atoms (on variables and constants)

\[ \forall x \forall y \left( \text{ siblingOf}(x,y) \rightarrow \exists z \left( \text{ parentOf}(z,x) \land \text{ parentOf}(z,y) \right) \right) \]

*Simplified form:* siblingOf(x,y) \rightarrow parentOf(z,x) \land parentOf(z,y)

- Same as Tuple Generating Dependencies
  - [+ Equality Generating Dependencies, Negative Constraints]
- See also Datalog+/-
- Same as the logical translation of Conceptual Graph rules
- Generalize light Description Logics used for OBDA (DL-Lite, $\mathcal{EL}$, …)
Generating Fresh Variables

\[ R = \forall x \forall y \ (\text{siblingOf}(x,y) \rightarrow \exists z \ (\text{parentOf}(z,x) \land \text{parentOf}(z,y))) \]

\[ F = \text{siblingOf}(a,b) \]

A rule \( \text{body} \rightarrow \text{head} \) is applicable to a fact \( F \) if there is a homomorphism \( h: \text{body} \rightarrow F \)

Then \( h(\text{head}) \) can be « added » to \( F \) with renaming existential variables of head

\[ F' = \exists z_0 \ (\text{siblingOf}(a,b) \land \text{parentOf}(z_0,a) \land \text{parentOf}(z_0,a)) \]
Basic Decision Problem

Given a KB $\mathcal{K} = (F, R)$ and a (Boolean) conjunctive query $Q$, is $Q$ entailed by $\mathcal{K}$?
**Forward vs Backward Chaining**

**FC**  
Fact saturation (« bottom-up »)

\[ F, R \models Q \iff \text{there is a homomorphism from } Q \text{ to } F', \text{ where } F' \text{ is obtained by a derivation sequence from } F \text{ with } R \]

**BC**  
Query rewriting (« top-down »)

[Decomposition into 2 steps: DL-Lite]

\[ F, R \models Q \iff \text{there is a homomorphism from } Q' \text{ to } F, \text{ where } Q' \text{ is obtained by a rewriting sequence from } Q \text{ with } R \]
Decidability Issues

- Entailment is not decidable
- Many decidable classes exhibited in databases and KR
- Three generic kinds of properties ensuring decidability:
  - Saturation by Forward Chaining halts (« finite expansion set », \( fes \))
  - Query rewriting by Backward Chaining halts (« finite unification set », \( fus \))
  - Saturation by Forward Chaining may not halt but the generated facts have a tree-like structure (« bounded treewidth set », \( bts \))

These properties are not recognizable [Baget+ KR 10] but they provide generic algorithms
(Partial) Inclusion Map of Decidable Classes

- Datalog
- weakly-acyclic
- wa-GRD
- jointly acyclic
- weakly-acyclic
- Inclusion dependency
- atomic body
- domain-r

Key:
- BTS
- FUS
- glut fg
- jointly fg
- weakly fg
- weakly-guarded
- frontier-guarded
- frontier-1
- guarded
- w-sticky-join
- w-sticky
- sticky join
- sticky
- (Partial) Inclusion Map of Decidable Classes
FINITE EXPANSION SETS
Breadth-First FC / Chase

\[ F^0 = F \]
\[ F^i \ (i > 1) \] obtained from \( F^{i-1} \) by performing all possible rule applications on \( F^{i-1} \)

Any atom with rank \( i \) can be obtained by a derivation sequence of length \( \leq \max (|\text{rule body}|)^i \)

\( F, R \models Q \) iff there is \( k \) s.t. \( Q \) maps to \( F^k \)
Finite Saturation (*fes*)

**Def:** $\mathcal{R}$ is a finite expansion set (*fes*) if for all fact $F$, there is $F'$ (finitely) derived from $F$ s.t. any rule application on $F'$ leads to a fact equivalent to $F'$

$\mathcal{R}$ is *fes* iff for all fact $F$, there is $k$ s.t. $F^{k+1} \equiv F^k$ = *core chase*

Note: $k$ is not independent from $F$

$fes \neq \text{« bounded » set of rules (cf. this notion in Datalog)}$

$\mathcal{R}$ is bounded if there is $k$ s.t. for all $F$, $F^{k+1} \equiv F^k$

*(ex: acyclic GRD)*
Main Recognizable Classes with Finite Saturation (fes)

Semantic condition
[Cuenca Grau+ KR 12]

Acyclic existential dependency graph
[Krötzsch+ IJCAI 11]

Acyclic position dependency graph
[Deutsch+ ICDT 03]
[Fagin+ ICDT 03]

No existential variables

Super-weak acyclicity

Joint-acyclicity

Weak-acyclicity

Datalog

MFA

fes-GRD

GRD with any kind of fes s.c.c.
[Baget KR 04]

GRD with wa strongly connected components
[Deutsch+ PODS 08]

aGRD
[Baget KR 04]

Acyclic chase graph
[Deutsch+ PODS 08]

Position dependency graph:
nodes are positions in predicates
edges show how existential variables are propagated

Graph of Rule Dependencies:
( GRD – also chase graph)
nodes are rules
edges express that a rule may lead to trigger a rule
FINITE UNIFICATION SETS
Remark: If $Q_1$ is more general than $Q_2$ (i.e. $Q_1$ maps to $Q_2$) then $Q_2$ is useless.

**Def:** $R$ is a **finite unification set (fus)** if, for any query $Q$, the set of *most general* rewritings of $Q$ with $R$ is finite.

[dropping « most general » would weaken the notion]

**Prop:** [König+ RR 2011] When $R$ is fus, there is a *unique* sound and complete set of *most general* rewritings of $Q$.  

[unique if each query in the set is made non redundant]
Main Recognizable Classes with Finite Query Rewriting (fus)

- **Sticky-join**
  - Body restricted to a single atom
  - [Baget+ IJCAI 09]
  - Restricts multiple occurrences of body variables that do not occur in all head atoms
  - [Cali+ RR 10]

- **Sticky**
  - Each head atom contains all or none of the body variables
  - [Baget+ IJCAI 09]
  - [Cali+ PVLDB 2010]

- **Domain-restricted**
  - Each head atom contains all the body variables

- **Atomic-body**
  - = linear Datalog+/
  - [Cali+ PODS 09]

E.g. necessary properties of concepts / relations

E.g. concept product
- elephant(x) \( \land \) mouse(y) \( \rightarrow \) bigger-than(x,y)
(GREEDY) BOUNDED TREEWIDTH SETS
Decomposition Tree / Treewidth

\[ p(a,b) \ q(b,z0) \ r(a,b,t0) \ p(b,t0) \ q(t0,z1) \ r(b,t0,t1) \ p(t0,t1) \]

Decomposition tree:
1) each node (term) appears in a bag
2) each hyperedge (atom) has all its nodes in a bag
3) for each node \( x \), the subgraph induced by the bags containing \( x \) is connected

Width of a tree decomposition = \( \max \) number of nodes in a bag (minus 1)
Treewidth of a graph = \( \min \) width over all decomposition trees of this graph
Bounded treewidth and decidability

Definition: A set $R$ of existential rules is a bounded treewidth set ($bts$) if, for any fact $F$, there exists a bound $b$ such that, for any $F'$ derived from $F$, $\text{treewidth}(F') < b$.

Theorem (basically [Cali+, KR 2008]): If $R$ is $bts$, then entailment is decidable.

Proof: Direct consequence of [Courcelles90] + [Thomas88] for compactness.
(Partial) Inclusion Map of Decidable Classes

- Datalog
- acyclic GRD
- wa-GRD
- jointly acyclic
- weakly-acyclic
- Inclusion dependency
- atomic body
- domain-r
- Sticky
- w-sticky
- w-sticky-join
- Sticky join
- Frontier-1
- Guarded
- Frontier-guarded
- Weakly-guarded
- Jointly-guarded
- Glut fg
Some Recognizable bts (and not fes) Classes of Rules

**Frontier:** variables shared by the body and the head

Guard only the *frontier*

- **[Baget+ KR’ 10]**
  - \( r(x,y) \land r(y,z) \rightarrow r(y,u) \)
  - \( r(y,u) \land r(z,u) \)

The *frontier* has size 1

- **[Baget+ IJCAI’ 09]**

Guard only affected variables from the *frontier*

- **[Baget+ KR’ 10]**
  - \( r(x,y) \land r(y,z) \rightarrow r(y,u) \land r(z,u) \)

Guard only affected variables (i.e. possibly mapped to new existentials)

- **[Cali+ KR’ 08]**

An atom in the body *guards* all the body variables

- **[Cali+ KR’ 08]**

**datalog**
From BTS to GBTS

Recognizability: As fes and fus, bts is an unrecognizable class.

Algorithms: Can we, as done for fes and fus, present a generic algorithm for all bts subclasses?

- **Problem 1:** we have to be able to compute the bound $b = f_c(F, \mathcal{R})$ for each of the bts subclasses.
- **Problem 2:** even in that case, the enumeration of every interpretation of treewidth lesser than $b$ (and of sufficient depth) is not reasonable.
The GBTS class: main ideas

Goal: Ensure that we can greedily build a decomposition tree of bounded width along the chase.

\[ F = \mathbf{B}_0 \]

\[ T_0 = \text{Terms}(F) \cup \{\text{possible constants}\} \]

\[ B_1 = \pi_j(B_i) \]

\[ T_1 = \text{Terms}(B_1) \cup T_0 \]

\[ B_2 = \pi_{j'}(B_{i'}) \]

\[ T_2 = \text{Terms}(B_2) \cup T_0 \]

\[ B_3 = \pi_k(B_q) \]

\[ T_3 = \text{Terms}(B_3) \cup T_0 \]

Is there a bag \( B_p \) such that \( \pi_k \) maps terms of frontier \( (R_q) \) to \( T_p \)?

YES

NO

The procedure halts and \text{FAILS}.
The GBTS class: main ideas

Definition: A set of existential rules $\mathcal{R}$ is gbts if, for any fact $F$, for any $\mathcal{R}$-derivation sequence from $F$, this greedy algorithm does not return FAILS (in that case, it effectively builds a decomposition tree of width $\leq |F| + \max(|R_i|)$).

Question 1: Is gbts an « interesting subset » of bts?

Question 2: Is there a generic algorithm for gbts?
Which known BTS classes are GBTS?

- glut fg
- jointly fg
- weakly fg
- weakly-guarded
- frontier-guarded
- frontier-1
- guarded
- jointly acyclic
- weakly-acyclic
- Datalog
- wa-GRD
- acyclic GRD
- atomic body
- inclusion dependency
- domain-r
- sticky
- w-sticky
- w-sticky-join
- sticky join
Overview of the gbts algorithm
Step 1: a finite encoding of the chase

Let us suppose that \( R \) is gbts

\[
F = B_0
\]

\[
T_0
\]

Take \( B_i \) and \( B_j \) created by a same rule \( R \) (resp. by \( \pi_i \) and \( \pi_j \)), s.t. there is a bijection \( \psi_{ij} \) between \( T_i \) and \( T_j \) with \( \psi_{ij}(t) = t' \) iff \( t \) and \( t' \) have been generated from the same term of \( R \).

If, moreover, \( \psi_{ij} \) determines an isomorphism between the atoms of the subtrees respectively rooted in \( B_i \) and \( B_j \)

Then \( B_i \) and \( B_j \) are equivalent.
(and so are their children)

Consider an equivalence class of bags.

Choose a (high) representant

Delete subtrees of all others!
Computing the blocked tree

To each bag we associate a pattern

A pattern of a bag (at step p) encodes all ways of mapping a subset of any rule body to the fact obtained at step p, while using some terms from this bag.

Equivalence of patterns at some step p implies that copies (as defined before) under equivalent bags belong to the chase.
Overview of the gbts algorithm

Step 2: querying a blocked tree

Let us suppose that we have built a finite blocked tree

This blocked tree encodes the chase.

Suppose Q maps to the chase.

Choose a set of bags (in the chase) that supports all atoms of Q.

There exists an Atom-Tree decomposition $Q_t$ of $Q$.

There exists an ATD mapping $\Gamma$ of $Q_t$.

$\Gamma$ is valid.
Overview of the gbts algorithm
Step 2: querying a blocked tree

A complicated algorithm to check if $Q$ maps to $R$-chase($F$)

For every ATD $Q_t$ of $Q$
  • For every ATD-mapping $\Gamma$ of $Q_t$ in chase($F$)
    If $\Gamma$ is Valid in chase($F$)
      Return YES

Return NO

Problem: we don’t have the chase, only the blocked tree…

For every ATD $Q_t$ of $Q$
  • For every ATD-mapping $\Gamma$ of $Q_t$ in the blocked tree
    If $\Gamma$ is Valid in the blocked tree
      Return YES

Return NO
Overview of the gbts algorithm

Step 2: querying a blocked tree

We have to find an ATD mapping in the blocked tree…
Overview of the gbts algorithm

Step 2: querying a blocked tree

... and prove it corresponds to a Valid one in one completion.

\[ F = B_0 \]

\[ T_0 \]

\[ B_1 \quad T_1 \]

\[ B_2 \quad T_2 \]

\[ B_3 \quad T_3 \]

\[ B_4 \quad T_4 \]

\[ B_5 \quad T_5 \]

\[ B_6 \quad T_6 \]

\[ B_8 \quad T_8 \]

\[ B_{10} \quad T_{10} \]

\[ B_{12} \quad T_{12} \]

\[ B_{14} \quad T_{14} \]

\[ Q_0 \]

\[ Q_3 \]

\[ Q_{11} \]

\[ Q_{13} \]

\[ Q_{15} \]

\[ \varphi_{3,4} \]

\[ \varphi_{3,4} \quad \varphi_{4,10} \]

\[ \varphi_{14,8} \circ \varphi_{14} \]

\[ \pi_0 \]

\[ \pi_3 \]

\[ \pi_{10} \]

\[ \pi_{12} \]

\[ \pi_{14} \]

\[ \psi_{12,6} \circ \varphi_{12} \]

\[ \psi_{14,8} \circ \varphi_{14} \]
Overview of the gbts algorithm

Step 2: querying a blocked tree

Validation is backtrack-free

Generating the required bag is done at most in $b^f$ steps
### Complexity results

<table>
<thead>
<tr>
<th>Class</th>
<th>arity unbounded</th>
<th>arity bounded</th>
<th>Data Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>gbts</td>
<td>$2\text{ExpTime-c}$</td>
<td>$2\text{ExpTime-c}^*$</td>
<td>$\text{ExpTime-c}^*$</td>
</tr>
<tr>
<td>wfg</td>
<td>$2\text{ExpTime-c}^*$</td>
<td>$2\text{ExpTime-c}^*$</td>
<td>$\text{ExpTime-c}^*$</td>
</tr>
<tr>
<td>fg</td>
<td>$2\text{ExpTime-c}^*$</td>
<td>$2\text{ExpTime-c}^*$</td>
<td>PTIME-c ^*</td>
</tr>
<tr>
<td>fr1</td>
<td>$2\text{ExpTime-c}^*$</td>
<td>$2\text{ExpTime-c}^*$</td>
<td>PTIME-c ^*</td>
</tr>
<tr>
<td>wg</td>
<td>$2\text{ExpTime-c}$</td>
<td>$\text{ExpTime-c}$</td>
<td>$\text{ExpTime-c}$</td>
</tr>
<tr>
<td>guarded</td>
<td>$2\text{ExpTime-c}$</td>
<td>$\text{ExpTime-c}$</td>
<td>PTIME-c</td>
</tr>
<tr>
<td>ba-fg</td>
<td>$2\text{ExpTime-c}^*$</td>
<td>$\text{ExpTime-c}^*$</td>
<td>PTIME-c ^*</td>
</tr>
<tr>
<td>ba-fr1</td>
<td>$\text{ExpTime-hard}^{(1)}$</td>
<td>$\text{ExpTime-c}^*$</td>
<td>PTIME-c ^*</td>
</tr>
<tr>
<td>Datalog</td>
<td>$\text{ExpTime-c}$</td>
<td>NP-c</td>
<td>PTIME-c</td>
</tr>
</tbody>
</table>

^{(1)} Indicates a different complexity class.
On the recognizability of gbts

\[ R_{\text{new}}: \text{in}(x, B_0), \text{in}(y, z) \rightarrow \text{in}(x, z) \]

Suppose that there is a derivation from \((F, \mathcal{R})\) that fails.

- There is a derivation \(\mathcal{R}\)-derivation from \(F\) to \(G\) that does not fail, and a failing mapping from some body \(B\) to \(G\).
- There is a \(\mathcal{R}'\)-derivation from \(F'\) to \(G'\), and a failing mapping from some body \(B\) to \(G'\).
- There is a failing validation of some body \(B\) to the blocked tree of \((F', \mathcal{R}')\).

If a derivation from \((F, \mathcal{R})\) fails, then a derivation from \((U, \mathcal{R})\) fails.
ON FORWARD AND BACKWARD CHAININGS
Theorem [Salvat 96]: the following assertions are equivalent.

- Q is semantic consequence of F and \( R \)
- Q maps to some F’ finitely \( R \)-derived from F
- Some finite \( R \)-rewriting Q’ of Q maps to F
FINITE EXPANSION & UNIFICATION SETS

Saturation by Forward Chaining halts (« finite expansion set », $fes$)

Query rewriting by Backward Chaining halts (« finite unification set », $fus$)
INITIAL MISUNDERSTANDING

Definition [Cali & al. 2010]: A set of existential rules $\mathcal{R}$ has the bounded derivation-depth property (BDDP) iff, for every facts $F$ and $Q$, whenever $Q$ can be deduced from $F$ and $\mathcal{R}$, $Q$ is entailed by the $\mathcal{R}$-saturation of $F$ at rank $\gamma$, where $\gamma = f(\mathcal{R}, Q)$.

A formulation, expressed in Forward Chaining, that is very similar to FES / BTS.

BUT

Known BDDP classes are FUS, not FES nor BTS!
OVERLOOKED LEMMAS [Salvat, 96]
Definition [Cali & al. 2010]: A set of existential rules $R$ has the (Q) bounded derivation-depth property (QBDDP) iff, for every facts $F$ and $Q$, whenever $Q$ can be deduced from $F$ and $R$, $Q$ is entailed by the $R$-saturation of $F$ at rank $\gamma$, where $\gamma = f(R, Q)$

**Property:** If $R$ has the QBDDP, then $R$ is f.u.s. Moreover, for any $Q$, the depth of the $R$-rewriting tree of $Q$ is $N^\gamma$ (where $N$ is the max size of $Q$ and rule bodies).

For any $Q'$ rewritable from $Q$, there is a $Q''$ in the rewriting tree of depth $N^\gamma$ that is more general than $Q'$. 
BOUNDED DERIVATION DEPTH & FUS (2)

Definition [Cali & al. 2010]: A set of existential rules $\mathcal{R}$ has the (Q) bounded derivation-depth property (QBDDP) iff, for every facts $F$ and $Q$, whenever $Q$ can be deduced from $F$ and $\mathcal{R}$, $Q$ is entailed by the $\mathcal{R}$-saturation of $F$ at rank $\gamma$, where $\gamma = f(\mathcal{R}, Q)$

Property: If $\mathcal{R}$ is f.u.s., then $\mathcal{R}$ has the QBDDP.

Suppose $Q$ can be deduced from $F$ and $\mathcal{R}$

$$\gamma = f(\mathcal{R}, Q)$$
**Definition [Oxford 2012]**: A set of existential rules \( R \) has the (F) bounded derivation-depth property (FBDDP) iff, for every facts \( F \) and \( Q \), whenever \( Q \) can be deduced from \( F \) and \( R \), \( Q \) is entailed by the \( R \)-saturation of \( F \) at rank \( \gamma \), where \( \gamma = f(R, F) \)

**Property**: \( R \) is f.e.s. iff \( R \) has the FBDDP.

\[ (\Rightarrow) \]

\[ \gamma = f(R, F) \]

\[ \Rightarrow \]

\[ F_{\gamma+1} \]

\[ (\Leftarrow) \]

\[ \gamma = f(R, F) \]
Alternate Definition: A set of existential rules $\mathcal{R}$ has the bounded deduction length property (BDLP) iff, for every facts $F$ and $Q$, whenever $Q$ can be deduced from $F$ and $\mathcal{R}$, there exists $\gamma$ s.t. one of the following equivalent assertions holds:

- $Q$ is entailed by an $\mathcal{R}$-derivation of $F$ of length $\gamma$.
- $F$ entails an $\mathcal{R}$-rewriting of $Q$ of length $\gamma$.

Moreover:

- If $\gamma = f(F, \mathcal{R})$, we say that $\mathcal{R}$ has the Fact BDLP (FBDLP).
- If $\gamma = f(Q, \mathcal{R})$, we say that $\mathcal{R}$ has the Query BDLP (QBDLP).
CONCLUSION (2)

Let \( P \) be the following property on \((R,F,Q,k)\):
\[
F, R \models Q \iff Q \text{ maps to } F^k
\]

Any \( R \) satisfies:
\[
\forall Q \forall F \exists k \mid P
\]

\( R \) is \texts{fes} iff \( R \) satisfies F-BDDP
\[
\forall F \exists k \mid \forall Q: P
\]

\( R \) is \texts{fus} iff \( R \) satisfies Q-BDDP
\[
\forall Q \exists k \mid \forall F: P
\]

\( R \) is bounded iff \( R \) satisfies « strong BDDP »:
\[
\exists k \mid \forall F \forall Q: P
\]

Moreover: \( R \) is \texts{fus} iff \( R \) is FO-rewritable
[Rudolph Thomazo]
Thank you

baget@lirmm.fr