Ontology-Based Query Answering under Guarded Rules

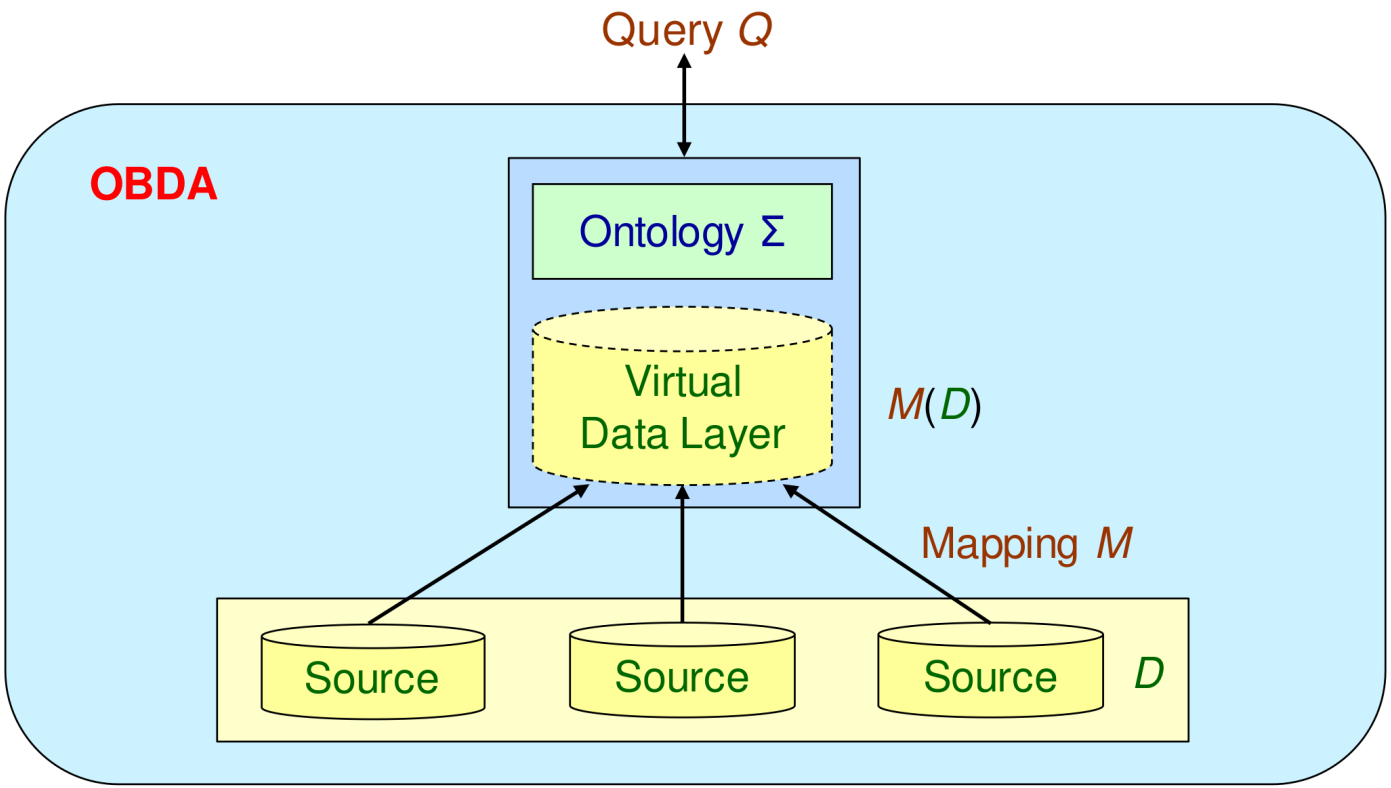
Michaël Thomazo

with thanks to Dresden, Edimbourgh and Montpellier for sharing slides

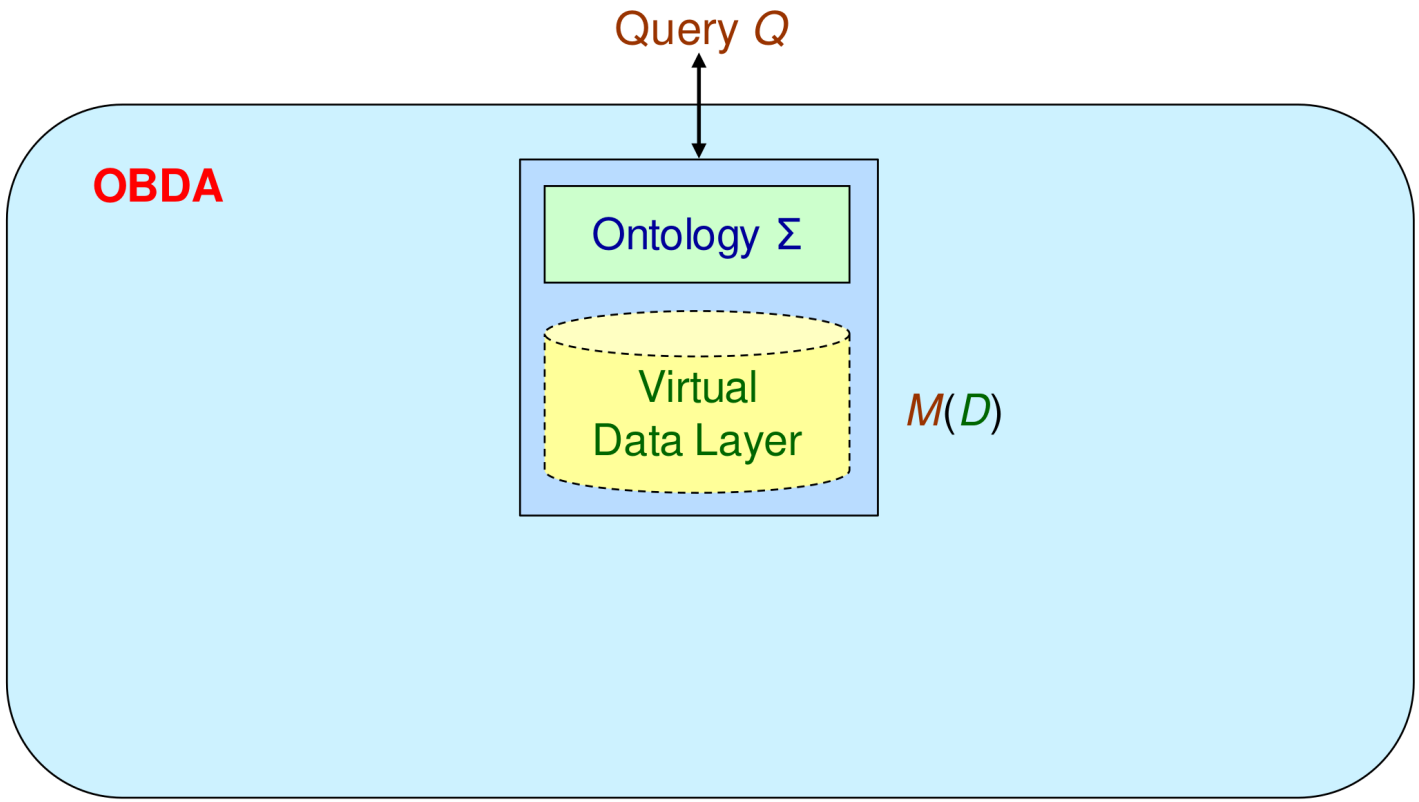
Inria, DI ENS, ENS, CNRS, University PSL, Paris, France

Journée de la logique, Montpellier

Ontology-Based Data Access



Ontology-Based Query Answering



Ontology Language

Existential rules

$$\forall \vec{x}\ \forall \vec{y}\ (\ \mathrm{Body}[\vec{x}, \vec{y}] \rightarrow \exists \vec{z}\ \mathrm{Head}[\vec{x}, \vec{z}]\ )$$

$\vec{y}$ is called the frontier of the rule.

Our Problem

Given a set of ground atoms D, a set of existential rules $\mathcal{R}$ and a Boolean conjunctive query q, does it hold that

$$D, \mathcal{R} \models q?$$

Complexity

- Data complexity: is calculated by considering only the database as part of the input, while the ontology and the query are fixed

- Combined complexity: is calculated by considering, apart from the database, also the ontology and the query as part of the input

- Data complexity vs. Combined complexity
  - Data complexity tends to be a more meaningful measure - ontologies and queries tend to be small; databases tend to be large
  - Nevertheless, the combined complexity is a relevant measure - identifies the real source of complexity

Description Logics and their Translation to Existential Rules

DL-Lite: Popular family of DLs - at the basis of the OWL 2 QL profile of OWL

DL-Lite Axioms	First-order Representation
$A \sqsubseteq B$	$\forall X\ (A(X) \rightarrow B(X))$
$A \sqsubseteq \exists R$	$\forall X\ (A(X) \rightarrow \exists Y\ R(X,Y))$
$\exists R \sqsubseteq A$	$\forall X \forall Y\ (R(X,Y) \rightarrow A(X))$
$\exists R \sqsubseteq \exists P$	$\forall X \forall Y\ (R(X,Y) \rightarrow \exists Z\ P(X,Z))$
$A \sqsubseteq \exists R.B$	$\forall X\ (A(X) \rightarrow \exists Y\ (R(X,Y) \wedge B(Y)))$
$R \sqsubseteq P$	$\forall X \forall Y\ (R(X,Y) \rightarrow P(X,Y))$
$A \sqsubseteq \neg B$	$\forall X\ (A(X) \wedge B(X) \rightarrow \bot)$

DL-Lite form the core of the OWL 2 QL Semantic Web profile.

$\mathcal{EL}$ as Existential Rules

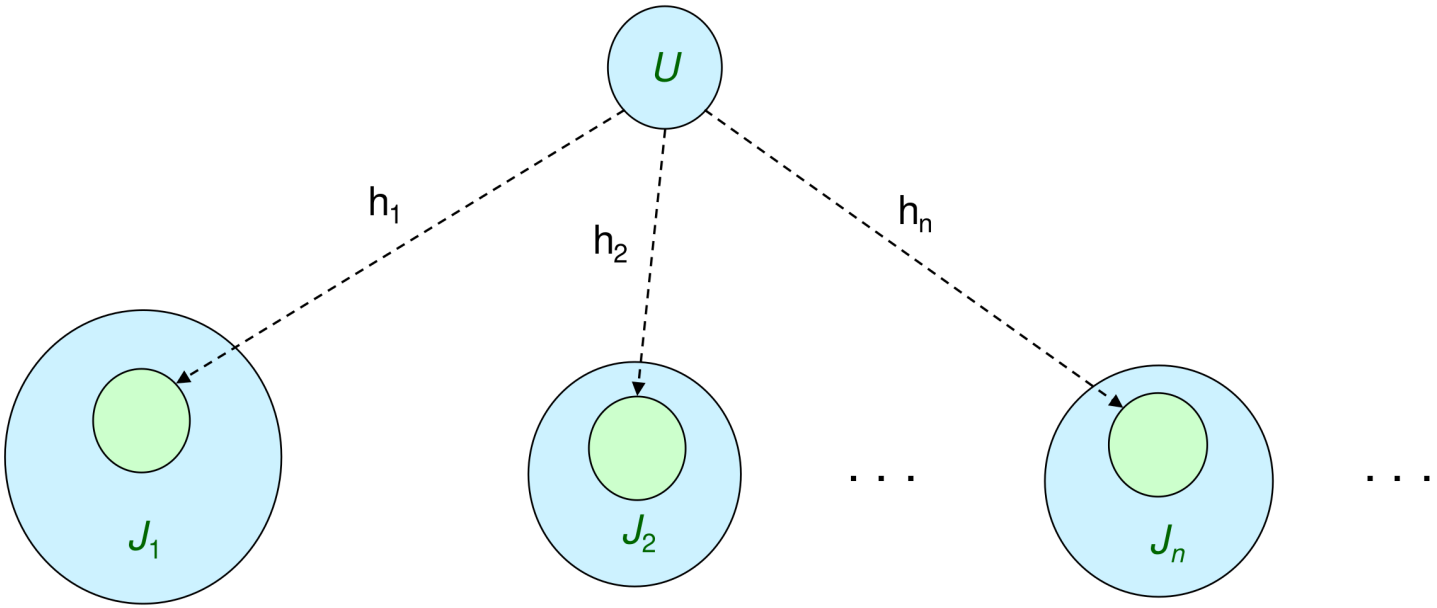
EL: Popular DL for biological applications - at the basis of OWL 2 EL profile

EL Axioms	First-order Representation
$A \sqsubseteq B$	$\forall X\ (A(X) \rightarrow B(X))$
$A \sqcap B \sqsubseteq C$	$\forall X\ (A(X) \wedge B(X) \rightarrow C(X))$
$A \sqsubseteq \exists R.B$	$\forall X\ (A(X) \rightarrow \exists Y\ (R(X,Y) \wedge B(Y)))$
$\exists R.B \sqsubseteq A$	$\forall X \forall Y\ (R(X,Y) \wedge B(Y) \rightarrow A(X))$

Outline of the Talk

1. Limitations of terminating chase and first-order rewritability
2. Guarded Rules: Definition and Properties of the Chase
3. An Algorithm for Guarded Rules
4. Other Approaches

Universal Models



An instance U is a universal model of $D \wedge \Sigma$ if the following holds:

1. U is a model of $D \wedge \Sigma$
2. $\forall J \in$ models$(D \wedge \Sigma)$, there exists a homomorphism h_J such that $h_J(U) \subseteq J$

(Non)-Finiteness of the Chase

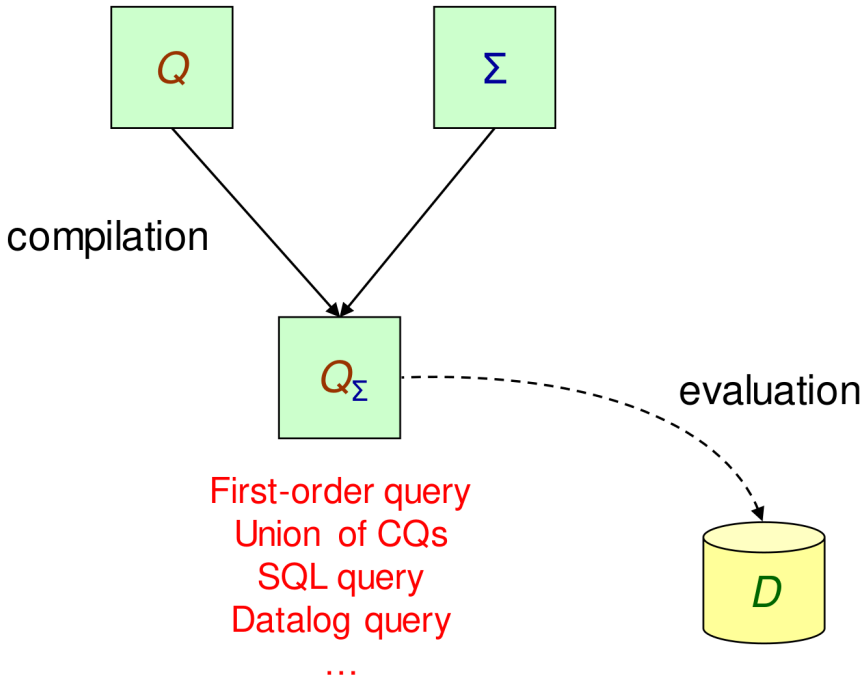
Σ

$$\forall X\,(Person(X) \rightarrow \exists Y\,(hasParent(X,Y) \wedge Person(Y)))$$

$$chase(D,\Sigma) = D \cup \{hasParent(Alice, z_1), Person(z_1),$$
$$hasParent(z_1, z_2), Person(z_2),$$
$$hasParent(z_2, z_3), Person(z_3), \ldots$$

1. **Existential quantification**
2. **Recursive definitions**

Existence of a First-Order Rewriting



$\forall D : D \wedge \Sigma \models Q \quad \Leftrightarrow \quad D \models Q_\Sigma$

evaluated and optimized by

exploiting existing technology

Ensures small data complexity: in $AC_0 \subseteq$ LogSpace

An "Easy" Example

Consider the following ruleset:

$$p(x) \to \exists y\ r(x, y) \wedge p(y)$$
$$p(x) \wedge r(x, y) \to p(y)$$

None of the mentionned approaches are applicable:

- chase does not terminate because of the first rule;
- first-order rewriting does not exist because of the second rule.

Guarded Rules

Guard

A *guard* in a rule is an atom of its body that contains all the body variables.

Guarded Rule

A rule is guarded is it admits a guard. A ruleset is guarded if all its rules are guarded.

Guarded Rules

Guard

A *guard* in a rule is an atom of its body that contains all the body variables.

Guarded Rule

A rule is guarded is it admits a guard. A ruleset is guarded if all its rules are guarded.

The following rule is guarded:

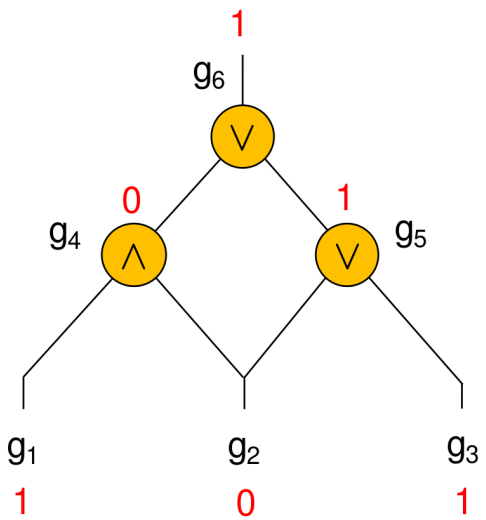
$$p(x) \rightarrow \exists y\ r(x, y) \wedge p(y)$$

The following rule is not guarded:

$$r(\mathbf{x}, y) \wedge r(y, \mathbf{z}) \rightarrow r(x, z)$$

Data Complexity of Guarded Rules

We can show PTime hardness in data complexity of the reasoning with guarded rules.



Does the circuit evaluate to *true*?

encoding of the circuit as a database D

$T(g_1)$ $T(g_3)$

$AND(g_4,g_1,g_2)$ $OR(g_5,g_2,g_3)$ $OR(g_6,g_4,g_5)$

evaluation of the circuit via a *fixed* set Σ

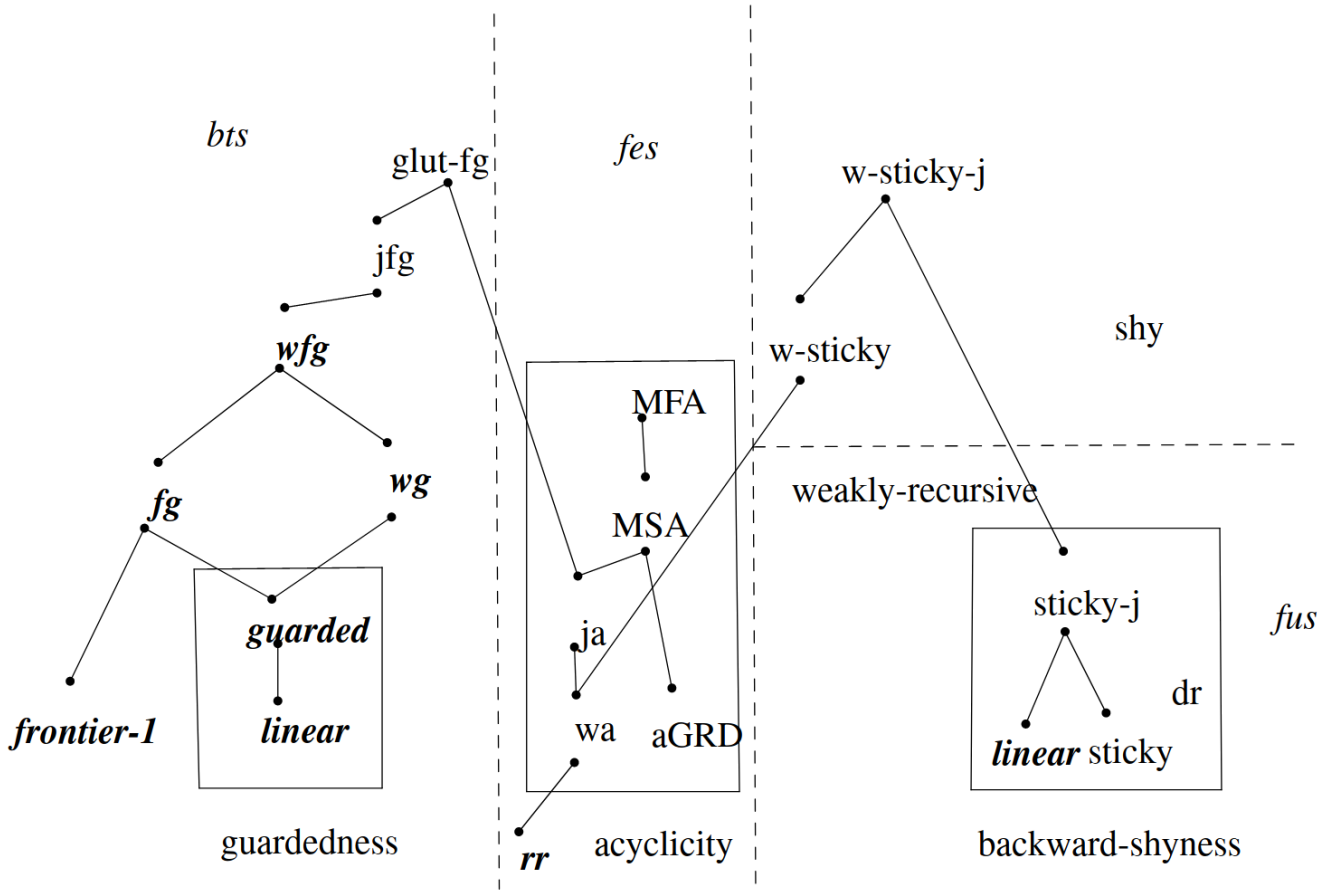
$\forall X \forall Y \forall Z\ (T(X) \wedge OR(Z,X,Y) \rightarrow T(Z))$

$\forall X \forall Y \forall Z\ (T(Y) \wedge OR(Z,X,Y) \rightarrow T(Z))$

$\forall X \forall Y \forall Z\ (T(X) \wedge T(Y) \wedge AND(Z,X,Y) \rightarrow T(Z))$

Circuit evaluates to *true* iff $D \wedge \Sigma \models T(g_6)$

Relationships between Guarded Rules and Other Classes



Guarded Rules and $\mathcal{EL}$

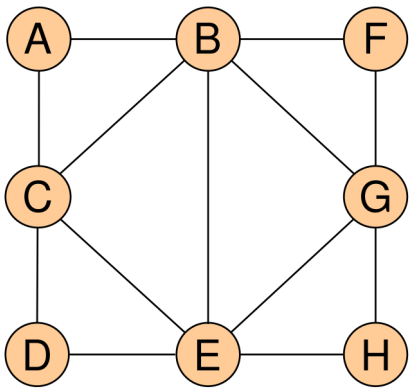
EL: Popular DL for biological applications - at the basis of OWL 2 EL profile

EL Axioms	First-order Representation
$A \sqsubseteq B$	$\forall X\ (A(X) \rightarrow B(X))$
$A \sqcap B \sqsubseteq C$	$\forall X\ (A(X) \wedge B(X) \rightarrow C(X))$
$A \sqsubseteq \exists R.B$	$\forall X\ (A(X) \rightarrow \exists Y\ (R(X,Y) \wedge B(Y)))$
$\exists R.B \sqsubseteq A$	$\forall X \forall Y\ (R(X,Y) \wedge B(Y) \rightarrow A(X))$

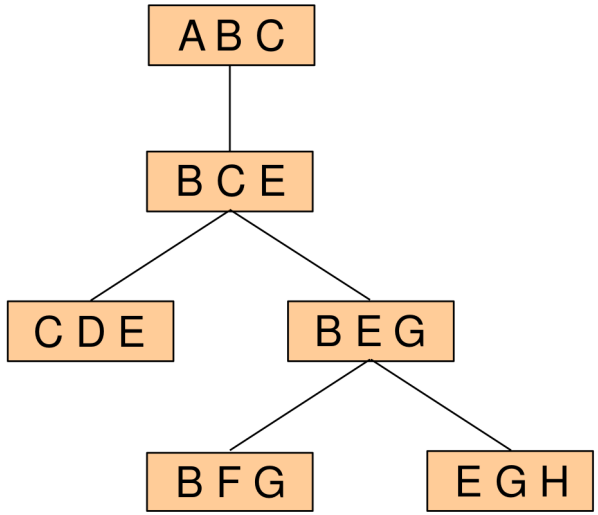
Tree Decomposition

Tree decomposition - a mapping of a graph into a tree

Graph $G = (\mathbf{V},\mathbf{E})$



Tree decomposition $T = (\mathbf{V'},\mathbf{E'})$ of G



1. For each v $\in$ **V**, there exists u $\in$ **V'** such that v $\in$ u
2. For each (v,w) $\in$ **E,** there exists u $\in$ **V'** such that {v,w} $\subseteq$ u
3. For each v $\in$ **V**, {u | v $\in$ u} induces a connected subtree

Treewidth and Consequences

Example

- $w(T) = \mathsf{max}_{B \in T}(\mathsf{size}(B)) - 1$
- $tw(G) = \mathsf{min}_{T \text{ for } G}(w(T))$

Small Treewidth Makes (Some) Problems Easy

Many problems that are hard on general graphs become easy on graphs of bounded treewidth.

Gaifman Graph

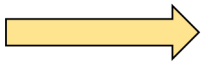
A.k.a Primal Graph

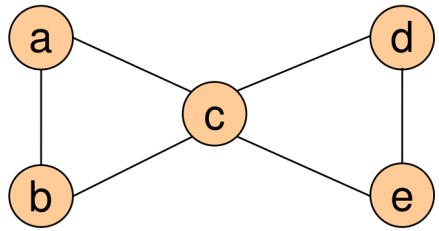
- An instance J can be represented as a graph $\mathcal{G}_J$ - Gaifman graph

R(a,b,c)

S(c,d)

T(c,d,e)





- The treewidth of J, denoted tw(J), is defined as tw($\mathcal{G}_J$)
- Thus, we can talk about the treewidth of the chase

Guardedness and Tree Decomposition

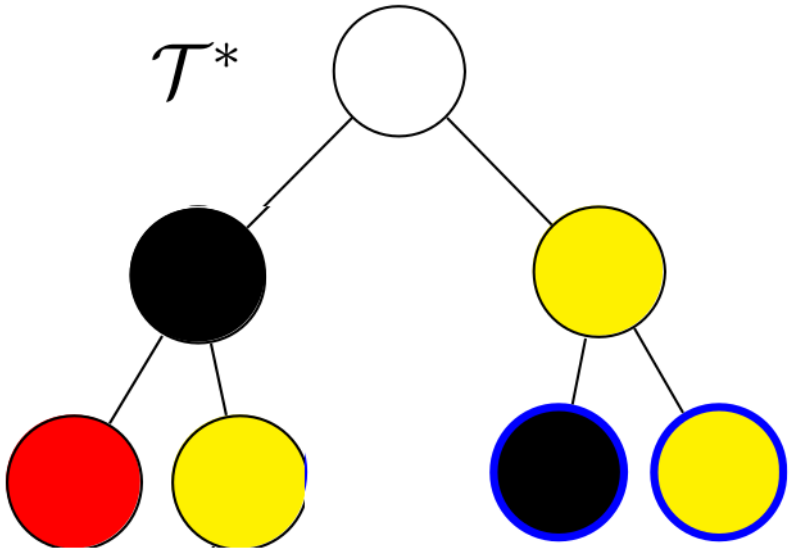
Property

For any guarded rule set, for any database, there exists b such that, for any k, $\mathsf{chase}_k(D, \mathcal{R})$ has treewidth at most b.

Key idea: maintain a tree decomposition at each step of the chase.

Blocking Technique: Finite Representation of the Chase

Finite representation of the chase:



Creating this Finite Representation is not Trivial

- defining what it means that two bags are equivalent;
- actually computing this equivalence relation.

On an easy example:

- $R_1 : p(x) \rightarrow \exists y\ r(x, y)$
- $R_2 : r(x, y) \rightarrow \exists z\ s(y, z)$
- $R_3 : q(x) \wedge r(x, y) \rightarrow q(y)$
- $R_4 : q(x) \wedge s(x, y) \wedge r(z, x) \wedge h(x)$

and consider $D = \{q(a_1), r(a_1, b_1), r(b_1, c_1), p(c_1), r(a_2, b_2), r(b_2, c_2), p(c_2)\}$

Ultimate Similarity

$$\frac{}{\mathbb{P} \rightsquigarrow \pi_{|\sigma(fr(R))}.\mathbb{P}_{\text{init}}(\mathbb{B}(R,\sigma))}\ \text{Prop. 5}$$

$$\frac{\mathbb{P}_1 \rightsquigarrow \lambda.\mathbb{P}_2}{\mathbb{P}_1 \rightsquigarrow \lambda.\mathit{Join}_{\text{l}}(\mathbb{P}_1,\lambda,\mathbb{P}_2)}\ \text{Prop. 6} \qquad \frac{\mathbb{P}_1 \rightsquigarrow \lambda.\mathbb{P}_2}{\mathbb{P}_1 \rightsquigarrow \mathit{Join}_{\text{u}}(\mathbb{P}_1,\lambda,\mathbb{P}_2)}\ \text{Prop. 7}$$

$$\frac{\mathbb{P}_1 \rightsquigarrow \mathbb{P}_2 \quad \mathbb{P}_2 \rightsquigarrow \mathbb{P}_3}{\mathbb{P}_1 \rightsquigarrow \mathbb{P}_3}\ \text{Prop. 8}$$

$$\frac{\mathbb{P}_1 \rightsquigarrow \mathbb{P}_2 \quad \mathbb{P}_1 \rightsquigarrow \lambda.\mathbb{P}_3}{\mathbb{P}_2 \rightsquigarrow \lambda.\mathbb{P}_3}\ \text{Prop. 9} \qquad \frac{\mathbb{P}_1 \rightsquigarrow \lambda.\mathbb{P}_2 \quad \mathbb{P}_2 \rightsquigarrow \mathbb{P}_3}{\mathbb{P}_1 \rightsquigarrow \lambda.\mathbb{P}_3}\ \text{Prop. 10}$$

Figure 5: Deduction calculus for the pattern saturation rules. For the first inference rule, $\mathbb{P}$ is any pattern, $R \in \mathcal{R} \cup \{\rightarrow I\}$, π is a homomorphism from $\mathit{fr}(R)$ to $\mathit{terms}(\mathit{support}(\mathbb{P}))$ such that $(\mathit{body}(R),\pi) \in \mathbb{P}$, and σ is the fusion of $\mathit{fr}(R)$ induced by π.

Other Approaches for Guarded Rules

- chase does not terminate, but one can decide whether a guarded ruleset ensures universal chase termination;
- query rewriting may not always exist, but one can decide whether a query is first-order rewritable;
- another interesting approach: the combined approach.

The Combined Approach

- An approach proposed in the context of description logics
- Several promising results - applied on (extensions of) EL, and members of the DL-Lite family

$D = \{P(a), S_1(a), P(b), S_2(b)\}$

$\Sigma = \{\forall X\,(P(X) \rightarrow \exists Y\,(R(X,Y) \wedge P(Y)))\}$

$Q = \exists X \exists Y \exists Z\,(R(X,Y) \wedge R(Z,Y) \wedge S_1(X) \wedge S_2(Z))$

auxiliary constant for satisfying $\exists$-variables

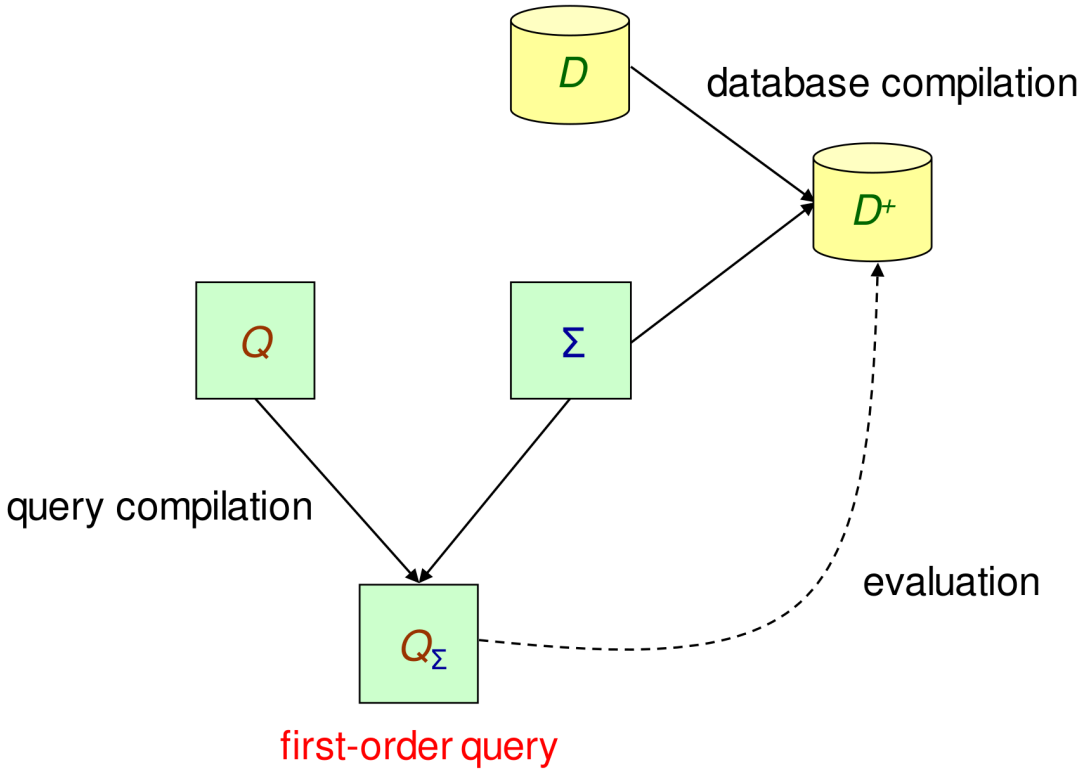
Step 1: Saturate the database, without inventing new nulls

$$D^+ = \{P(a), S_1(a), P(b), S_2(b)\} \cup \{Ex(c)\} \cup \{R(a,c), R(b,c), P(c), R(c,c)\}$$

Step 2: Eliminate unsound answers by rewrting the query into a FO-query

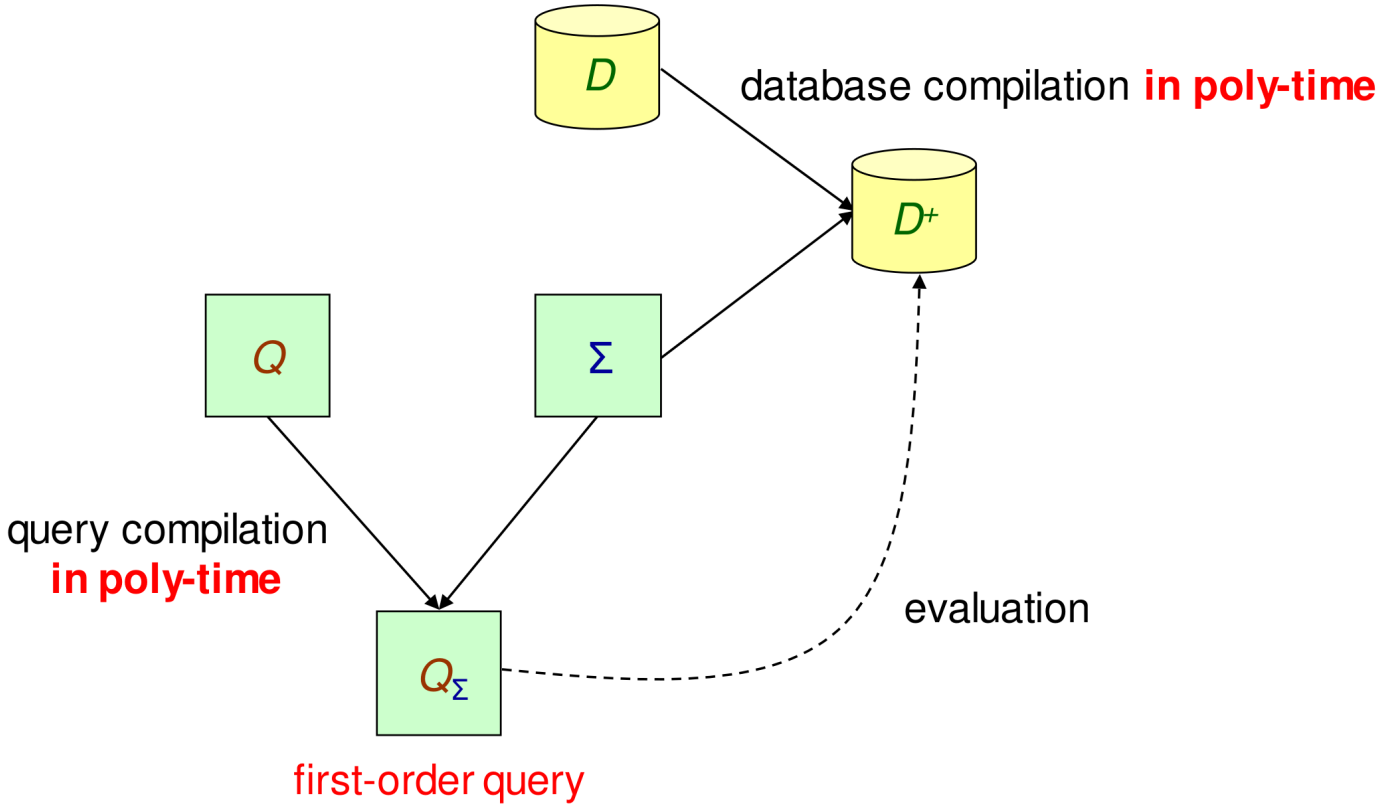
$$Q_{FO} = \exists X \exists Y \exists Z\,((R(X,Y) \wedge R(Z,Y) \wedge S_1(X) \wedge S_2(Z)) \wedge (Ex(Y) \rightarrow X = Z))$$

The Combined Approach



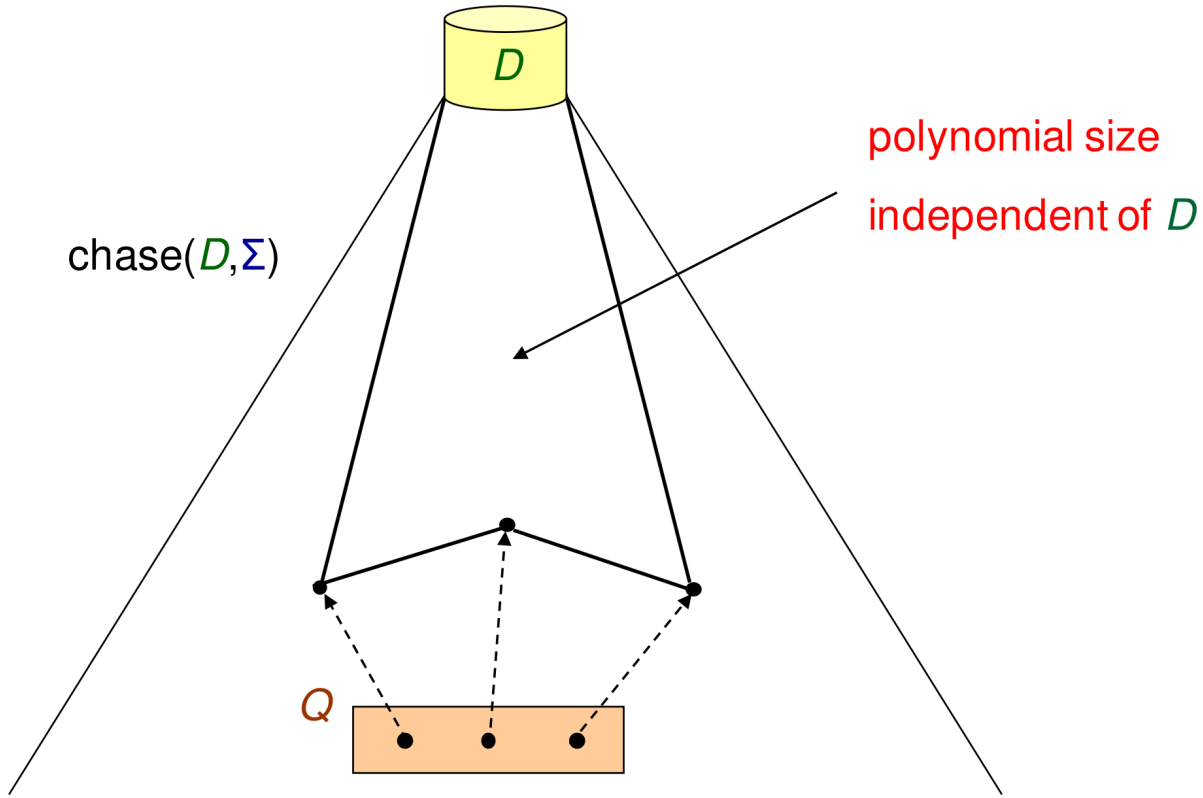
$$\forall D : D \wedge \Sigma \models Q \quad \Leftrightarrow \quad D^{+} \models Q_{\Sigma}$$

Polynomially Combined First-Order Rewritability



$$\forall D : D \wedge \Sigma \models Q \quad \Leftrightarrow \quad D^{+} \models Q_{\Sigma}$$

The Polynomial Witness Property



chase(D,Σ) $\models Q$ $\Rightarrow$ the query admits a small witness

The Polynomial Witness Property

Theorem: The PWP implies that BCQ-Answering is polynomially combined FO-rewritable

Proof (hint):

- We simulate the polynomially sized witness via a polynomially sized first-order query (query compilation)
- Notice that the number of nulls that appear in the witness depends on the query, and thus can not be explicitly added in the database
- We simulate these nulls via tuples of 0s and 1s - the constants 0 and 1 are explicitly added in the database (database compilation)

Applicability

- applicable for a subset of guarded rules, namely *linear* rules
- further conditions required for guarded rules.
- of practical interest for $\mathcal{EL}$, less so for linear rules.

Conclusion

- Guarded rules ensure a well-behaved chase: bounded treewidth
- Working with tree decompositions allow to finitely represent the chase
- Not seen today, but guardedness ensure good properties beyond existential rules

Conclusion

- Guarded rules ensure a well-behaved chase: bounded treewidth
- Working with tree decompositions allow to finitely represent the chase
- Not seen today, but guardedness ensure good properties beyond existential rules