Query Rewriting Under Existential Rules

Andreas Pieris

School of Informatics, University of Edinburgh, UK

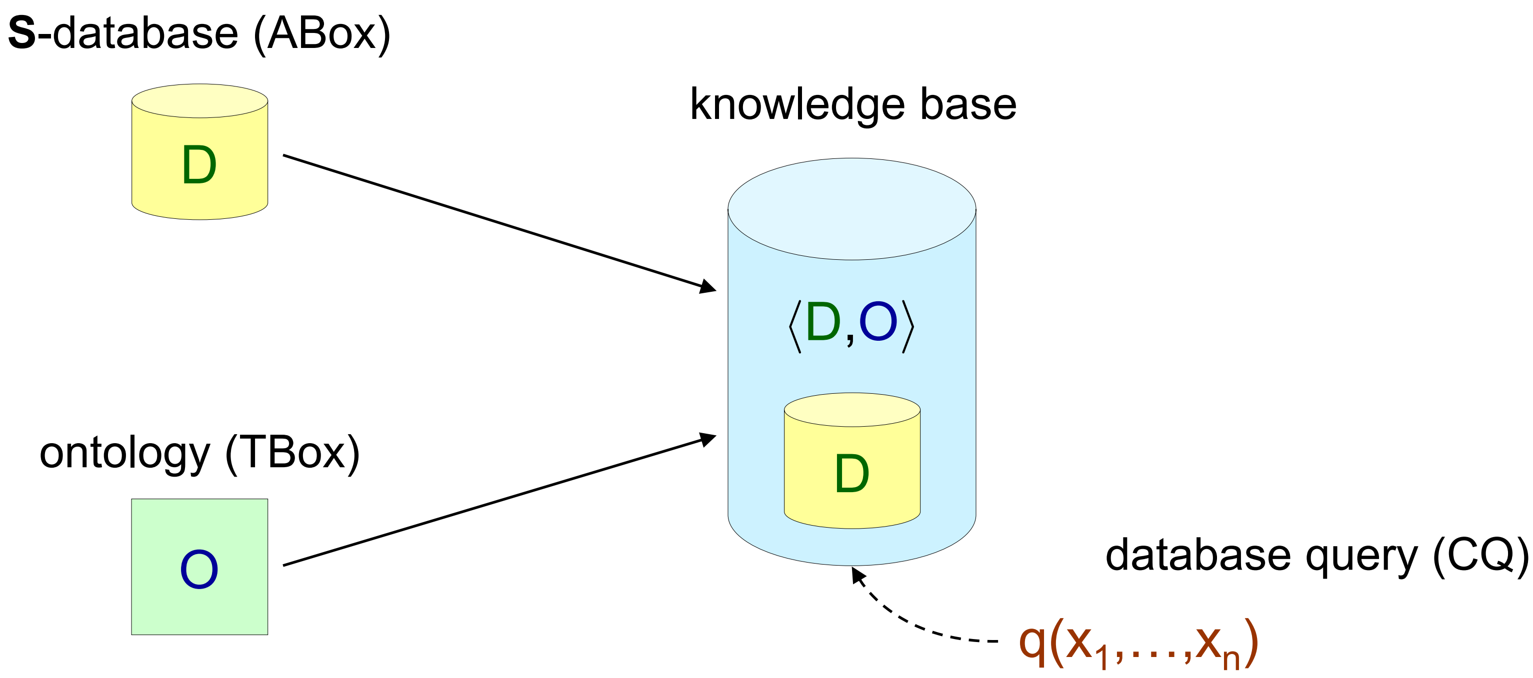
based on joint work with Pablo Barceló, Gerald Berger, Andrea Calì, Georg Gottlob, Marco Manna, Giorgio Orsi and Pierfrancesco Veltri

DL Workshop, Montpellier, France, July 18 - 21, 2017

this talk is about first-order rewritability under the

basic decidable classes of existential rules

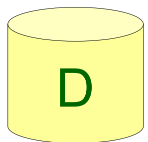
Ontology-Based Query Answering



$$\text{Certain-Answers}(q, D, O) \;=\; \{\, (c_1,\ldots,c_n) \in \text{dom}(D)^n \;\mid\; D \wedge O \models q(c_1,\ldots,c_n) \,\}$$

Ontology-Mediated Queries

S-database (ABox)



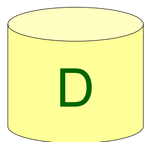
ontology-mediated query (OMQ)

$Q = (\mathbf{S}, O, q(x_1,\ldots,x_n))$

$Q(D) \;=\; \text{Certain-Answers}(q, D, O)$

Scalability in OMQ Evaluation

S-database (ABox)



ontology-mediated query (OMQ)

$Q = (\mathbf{S}, O, q(x_1,\ldots,x_n))$

?

Exploit standard RDBMSs - efficient technology for answering queries

Query Rewriting

$$Q = (\mathbf{S}, O, q(x_1,\ldots,x_n))$$

$$\Big\downarrow \text{rewrite}$$

$$Q_{rew}(x_1,\ldots,x_n)$$

a query that can be executed by a standard DBMS - **first-order query**

for every **S**-database D : $Q(D) = Q_{rew}(D)$

[Calvanese, De Giacomo, Lembo, Lenzerini & Rosati, AAAI 2005, J. Autom. Reasoning 2007]

Query Rewriting: An Example

$\{\ \forall x\ (Person(x) \rightarrow \exists y\ HasFather(x,y) \wedge Person(y)) \quad \equiv \quad Person \sqsubseteq \exists HasFather.Person\ \}$

$\{\ Person(\cdot),\ HasFather(\cdot,\cdot)\ \}$

$\exists x\ Person(x) \wedge HasFather(John,x)$

$Q = (\mathbf{S}, O, q())$

$\Big\downarrow$ rewrite

$Q_{rew} = \exists x\ Person(x) \wedge HasFather(John,x) \ \vee\ Person(John)$

First-Order Rewritability (FO-Rewritability)

(OL,QL)

an ontology language

(fragment of first-order logic)

a database query language

(sublanguage of first-order queries)

Definition: An OMQ language O is FO-Rewritable if every Q $\in$ O is FO-Rewritable

FO-Rewritability: The Main Questions

1. **Can we isolate meaningful OMQ languages that are FO-Rewritable?**

2. **For non-FO-Rewritable languages, can we decide FO-Rewritability?**

3. **What is the size of the FO rewritings? Can we do better?**

...have been extensively studied for DL- and rule-based OMQ languages

Existential Rules

(a.k.a. tuple-generating dependencies)

$$\forall \mathbf{x} \forall \mathbf{y} \, (\varphi(\mathbf{x},\mathbf{y}) \rightarrow \exists \mathbf{z} \, \psi(\mathbf{x},\mathbf{z}))$$

$\forall x \, (\text{Person}(x) \rightarrow \exists y \, \text{HasFather}(x,y) \wedge \text{Person}(y)) \quad \equiv \quad \text{Person} \sqsubseteq \exists \text{HasFather}.\text{Person}$

$\forall x \forall y \, (\text{HasChild}(x,y) \wedge \text{Human}(y) \rightarrow \text{Human}(x)) \quad \equiv \quad \exists \text{HasChild}.\text{Human} \sqsubseteq \text{Human}$

Existential Rules

(a.k.a. tuple-generating dependencies)

$$\varphi(\mathbf{x},\mathbf{y}) \rightarrow \exists \mathbf{z}\ \psi(\mathbf{x},\mathbf{z})$$

Person(x) $\rightarrow \exists$y HasFather(x,y), Person(y) ≡ Person $\sqsubseteq \exists$HasFather.Person

HasChild(x,y), Human(y) $\rightarrow$ Human(x) ≡ $\exists$HasChild.Human $\sqsubseteq$ Human

Existential Rules

(a.k.a. tuple-generating dependencies)

$$\varphi(\mathbf{x},\mathbf{y}) \rightarrow \exists \mathbf{z}\ \psi(\mathbf{x},\mathbf{z})$$

($\exists$Rules,CQ)

Guardedness

Frontier-Guarded

one body-atom contains all the $\forall$-variables in the head

$$R(\mathbf{x}), \varphi(\mathbf{x},\mathbf{y}) \rightarrow \exists \mathbf{z}\, \psi(\mathbf{x},\mathbf{z})$$

[Baget, Leclère, Mugnier & Salvat, IJCAI 2009, Artif. Intell. 2011]

↑

Guarded

one body-atom contains all the $\forall$-variables

$$R(\mathbf{x},\mathbf{y}), \varphi(\mathbf{x},\mathbf{y}) \rightarrow \exists \mathbf{z}\, \psi(\mathbf{x},\mathbf{z})$$

[Calì, Gottlob & Kifer, KR 2008, J. Artif. Intell. Res. 2013]

↑

Linear

one body-atom

$$R(\mathbf{x},\mathbf{y}) \rightarrow \exists \mathbf{z}\, \psi(\mathbf{x},\mathbf{z})$$

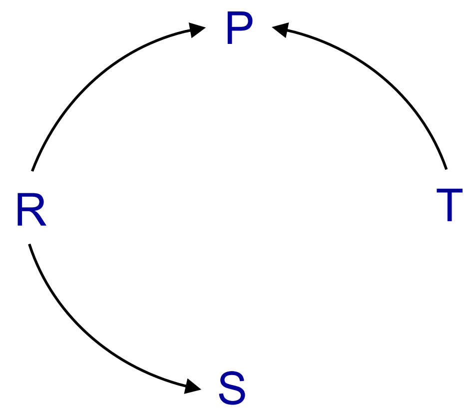
[Calì, Gottlob & Lukasiewicz, PODS 2009, J. Web Sem. 2012]

Acyclicity

(…or, non-recursive - the predicate graph is acyclic)

$$R(x,y), R(y,z) \rightarrow \exists w\, P(x), S(x,w)$$

$$T(x) \rightarrow P(x)$$



Stickiness

(…or, do not forget the joins)

$R(x,y), P(y,z) \rightarrow \exists w\, T(x,y,w)$

$T(x,y,z) \rightarrow \exists w\, S(y,w)$ ✓

$R(x,y), P(y,z) \rightarrow \exists w\, T(x,y,w)$

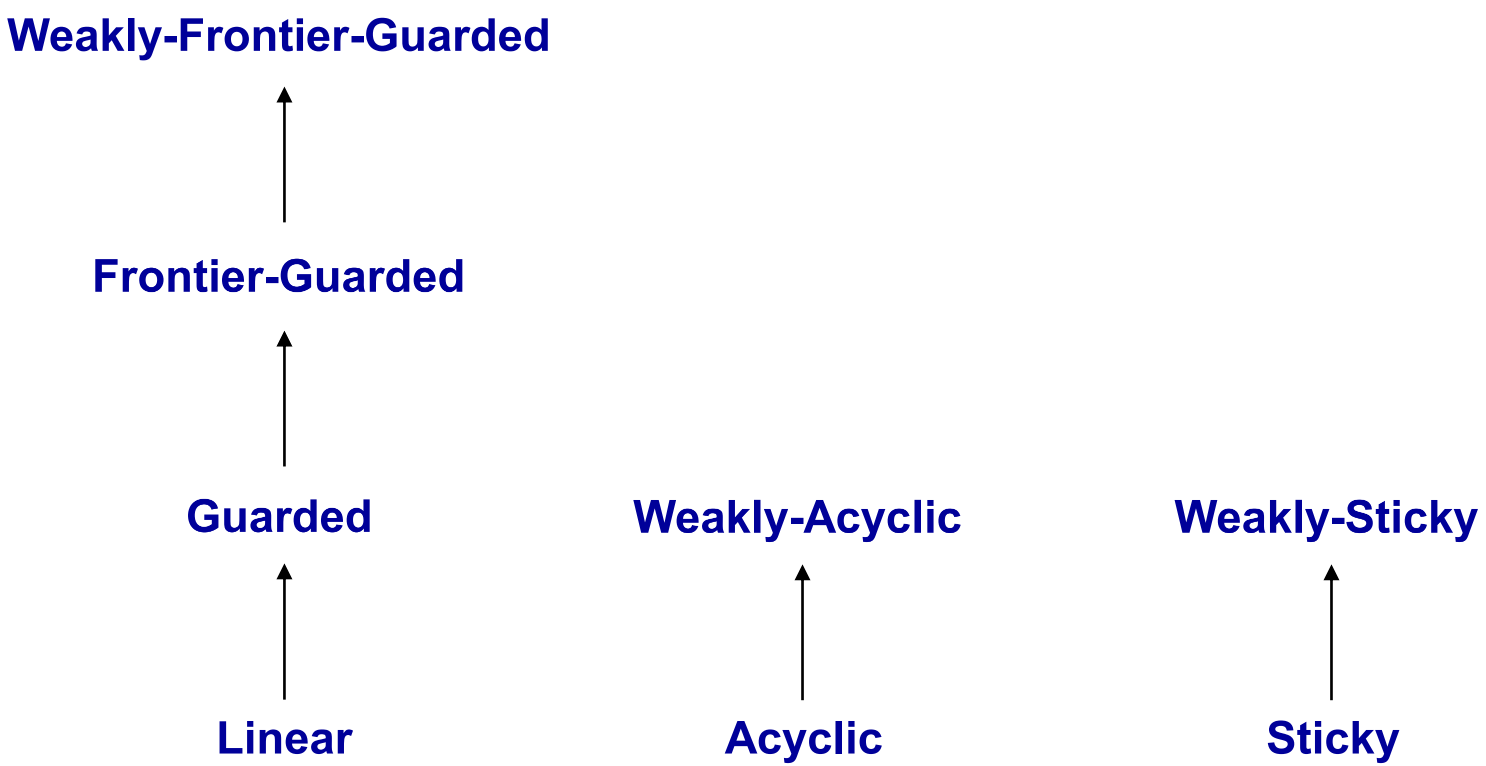
$T(x,y,z) \rightarrow \exists w\, S(x,w)$ ✗

$$R(x_1,\ldots,x_n), P(y_1,\ldots,y_m) \rightarrow T(x_1,\ldots,x_n,y_1,\ldots,y_m)$$ ✓

[Calì, Gottlob & P., PVLDB 2010, Artif. Intell. 2012]

Classes of Existential Rules

(a.k.a. Datalog$^{\pm}$ languages)



Classes of Existential Rules

(a.k.a. Datalog$^{\pm}$ languages)

What about FO-Rewritability?

Weakly-Frontier-Guarded

Frontier-Guarded

Guarded

Linear

Weakly-Acyclic

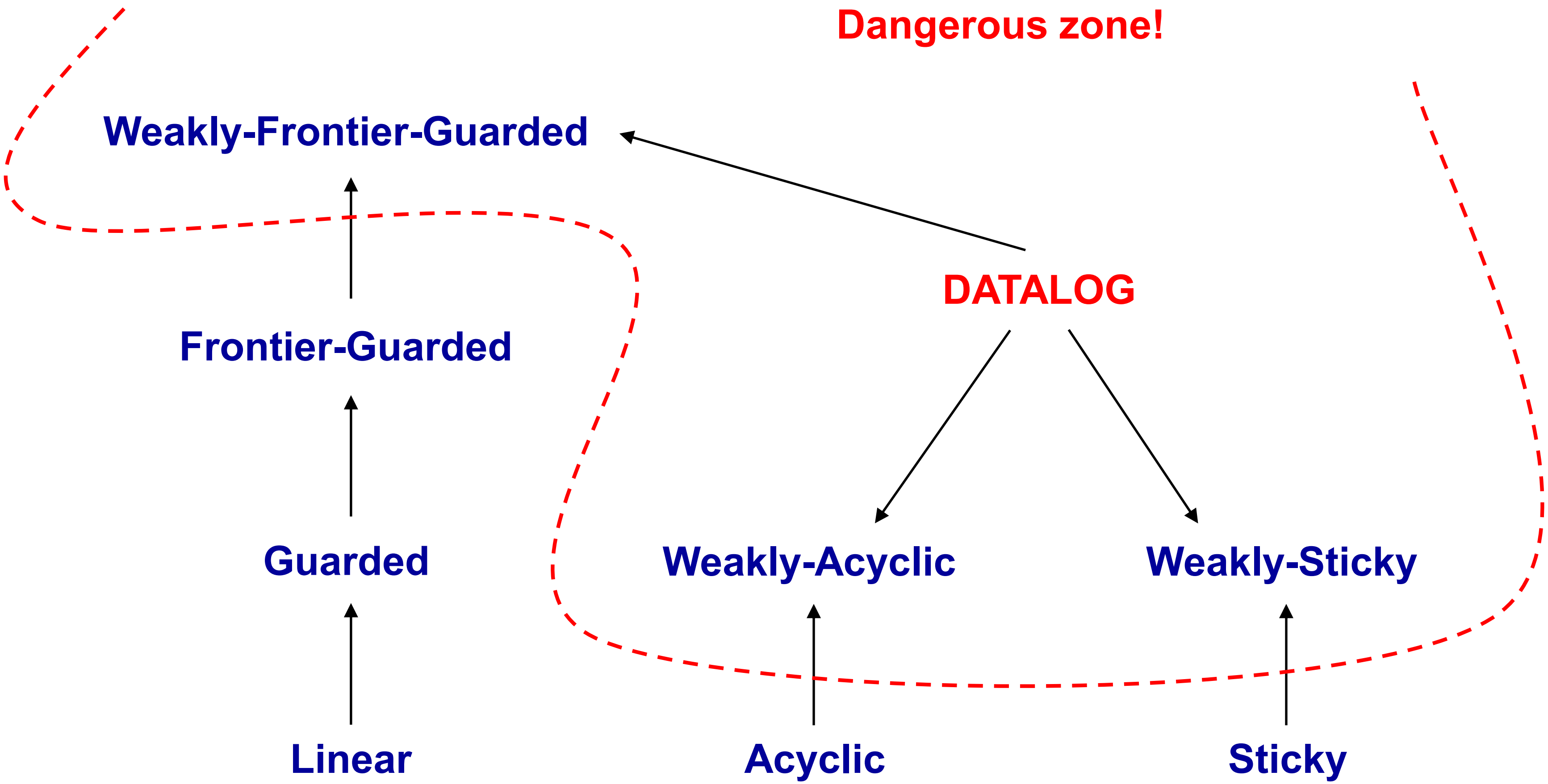
Acyclic

Weakly-Sticky

Sticky

Classes of Existential Rules

(a.k.a. Datalog$^{\pm}$ languages)



Guardedness and FO-Rewritability

Theorem: (**Guarded**,**CQ**) is not FO-Rewritable

$Q = (\{P, R\},\ \{R(x,y), P(y) \rightarrow P(x)\},\ P(c_n))$

$D \supseteq \{P(c_1)\}$, and contains no other P-atom

Q_{rew} has to check for the existence of an R-path in D of unbounded length

$$c_n \xrightarrow{R} \#_{n-1} \xrightarrow{R} \#_{n-2} \xrightarrow{R} \cdots \xrightarrow{R} \#_2 \xrightarrow{R} c_1$$

compute the transitive closure of R - not possible via a first-order query

FO-Rewritable OMQ Languages

Theorem: (**L**,**CQ**), where **L** $\in$ { **Linear**, **Acyclic**, **Sticky** }, is FO-Rewritable

Via the Bounded Derivation Depth Property (BDDP)

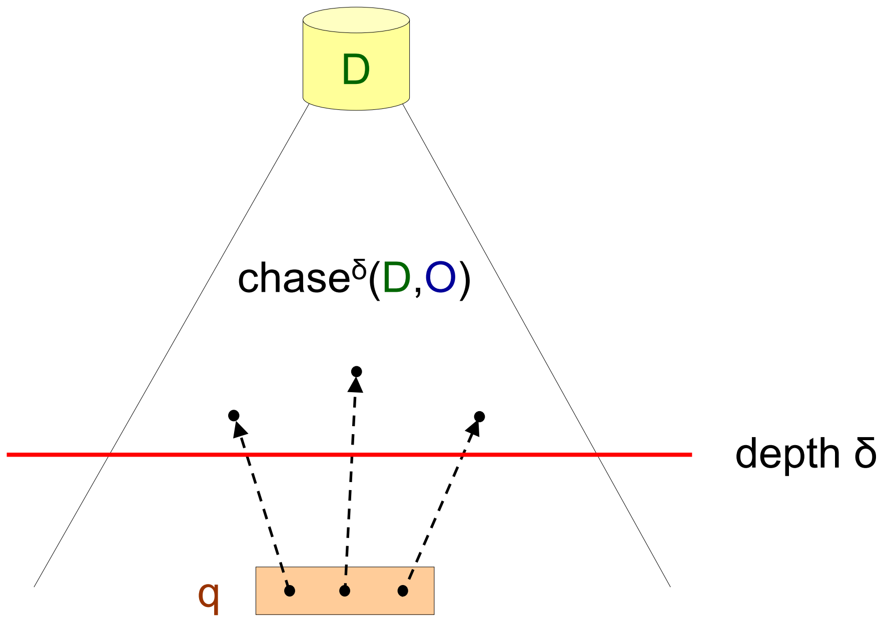
[Calì, Gottlob & Lukasiewicz, PODS 2009, J. Web Sem. 2012] + [Calì, Gottlob & P., PVLDB 2010, Artif. Intell. 2012]

Bounded Derivation Depth Property (BDDP)

Definition: (**L**,**CQ**) enjoys the BDDP if:

for every Q = (**S**, O, q) ∈ (**L**,**CQ**), there exists $\delta \geq 0$ such that,

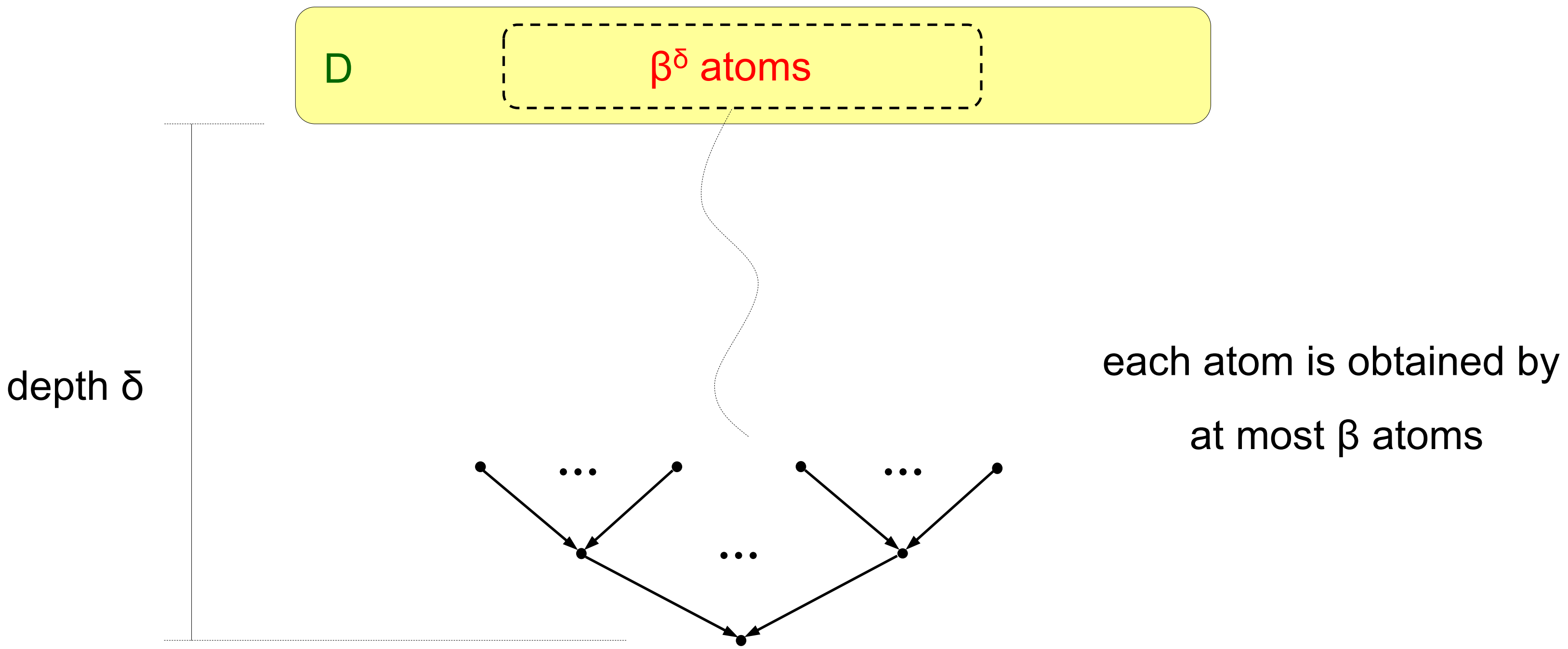
for every **S**-database D, Q(D) = q(chase$^{\delta}$(D,O))



[Calì, Gottlob & Lukasiewicz, PODS 2009, J. Web Sem. 2012]

Bounded Derivation Depth Property (BDDP)

Proposition: BDDP $\Rightarrow$ FO-Rewritability



$\Rightarrow$ to entail a CQ q we need at most $|q| \cdot \beta^{\delta}$ database atoms

Bounded Derivation Depth Property (BDDP)

Proposition: BDDP $\Rightarrow$ FO-Rewritability

Given an OMQ (**S**, O, q):

- $\mathbf{D}_{\beta,\delta,q}$ be the set of all possible **S**-databases of size at most $|q| \cdot \beta^{\delta}$
- $\mathbf{C} = \{ D \in \mathbf{D}_{\beta,\delta,q} \mid q(\mathrm{chase}(D,O)) \text{ is non-empty} \}$
- Convert **C** into a UCQ

…in fact, the other direction also holds - FO-Rewritability $\Leftrightarrow$ BDDP

FO-Rewritable OMQ Languages

Theorem: (**L**,**CQ**), where **L** $\in$ { **Linear**, **Acyclic**, **Sticky** }, is FO-Rewritable

Via the Bounded Derivation Depth Property (BDDP)

but, the BDDP-based algorithm is very expensive

can we do better?

[Calì, Gottlob & Lukasiewicz, PODS 2009, J. Web Sem. 2012] + [Calì, Gottlob & P., PVLDB 2010, Artif. Intell. 2012]

Perfect Reformulation

Fig. 2 The algorithm PerfectRef

Algorithm PerfectRef $(q, \mathcal{T})$
Input: conjunctive query q, TBox $\mathcal{T}$
Output: union of conjunctive queries PR
$PR := \{q\}$;
repeat
  $PR' := PR$;
  for each $q \in PR'$ **do**
  (a) **for each** g in q **do**
    for each PI I in $\mathcal{T}$ **do**
      if I is applicable to g
      then $PR := PR \cup \{q[g/gr(g, I)]\}$
  (b) **for each** g_1, g_2 in q **do**
      if g_1 and g_2 unify
      then $PR := PR \cup \{\tau(reduce(q, g_1, g_2))\}$;
until $PR' = PR$;
return PR

rewriting step

reduction step

Applicability → Soundness

Reduction → Completeness

[Calvanese, De Giacomo, Lembo, Lenzerini & Rosati, AAAI 2005, J. Autom. Reasoning 2007]

Perfect Reformulation for Existential Rules

$$\begin{tikzcd}
R(y,x), P(y) \rightarrow \exists z\, T(z,x,x) \arrow[dr] & & \exists u \exists v \exists w\, T(u,v,w), P(w) \arrow[dl, "{g = \{u \rightarrow z,\ v \rightarrow x,\ w \rightarrow x\}}"] \\
& T(z,x,x) &
\end{tikzcd}$$

thus, we can simulate a chase step by applying a backward resolution step

$$\exists u \exists v \exists w\, T(u,v,w), P(w) \ \vee\ \exists x \exists y\, R(y,x), P(y), P(x)$$

Perfect Reformulation for Existential Rules

$R(y,x), P(y) \rightarrow \exists z\, T(z,x,x)$ $\qquad\qquad$ $\exists u \exists v \exists w\, T(u,v,w), \mathbf{P(u)}$

$g = \{u \rightarrow z,\ v \rightarrow x,\ w \rightarrow x\}$

$T(z,x,x)$

thus, we can simulate a chase step ~~by~~ applying a backward resolution step

$\exists u \exists v \exists w\, T(u,v,w), P(u) \;\vee\; \exists x \exists y \exists u\, R(x,y), P(x), \mathbf{P(u)}$

unsound rewriting

Perfect Reformulation for Existential Rules

$$\begin{tikzcd}
R(y,x), P(y) \rightarrow \exists z\, T(z,x,x) \arrow[dr] & & \exists u \exists v \exists w\, T(u,v,w), \mathbf{P(u)} \arrow[dl, "g = \{u \rightarrow z,\ v \rightarrow x,\ w \rightarrow x\}"] \\
& T(z,x,x) &
\end{tikzcd}$$

Applicability condition: constants, join variables and free variables in the query do NOT unify with ∃-variables

…but, it may destroy completeness

Perfect Reformulation for Existential Rules

$R(y,x), P(y) \rightarrow \exists z\ T(z,x,x)$ $\exists u \exists v \exists w\ T(u,v,w), P(u)$

$T(x,y,z) \rightarrow P(x)$

$\exists u \exists v \exists w\ T(u,v,w), P(u)\ \vee$

$\exists u \exists v \exists w \exists y \exists z\ T(u,v,w), T(u,y,z)\ \vee$

(by the reduction step) $\exists u \exists v \exists w\ T(u,v,w)\ \vee$

(by the rewriting step) $\exists x \exists y\ R(x,y), P(x)$

XRewrite

ALGORITHM 1: The algorithm XRewrite

Input: a CQ q over a schema $\mathcal{R}$ and a set Σ of TGDs over $\mathcal{R}$
Output: the perfect rewriting of q w.r.t. Σ

$i := 0;$
$Q_{\text{REW}} := \{\langle q, \mathsf{r}, \mathsf{u}\rangle\};$
repeat
  $Q_{\text{TEMP}} := Q_{\text{REW}};$
  foreach $\langle q, x, \mathsf{u}\rangle \in Q_{\text{TEMP}}$, where $x \in \{\mathsf{r}, \mathsf{f}\}$ **do**
    foreach $\sigma \in \Sigma$ **do**
      /* rewriting step
      foreach $S \subseteq body(q)$ such that σ is $\boxed{\text{applicable}}$ to S **do**
        $i := i + 1;$
        $q' := \gamma_{S,\sigma^i}(q[S/body(\sigma^i)]);$
        if there is no $\langle q'', \mathsf{r}, \star\rangle \in Q_{\text{REW}}$ such that $q' \simeq q''$ **then**
          $Q_{\text{REW}} := Q_{\text{REW}} \cup \{\langle q', \mathsf{r}, \mathsf{u}\rangle\};$
        end
      end
      /* factorization step
      foreach $S \subseteq body(q)$ which is $\boxed{\text{factorizable}}$ w.r.t. σ **do**
        $q' := \gamma_S(q);$
        if there is no $\langle q'', \star, \star\rangle \in Q_{\text{REW}}$ such that $q' \simeq q''$ **then**
          $Q_{\text{REW}} := Q_{\text{REW}} \cup \{\langle q', \mathsf{f}, \mathsf{u}\rangle\};$
        end
      end
    end
    /* query q is now explored
    $Q_{\text{REW}} := (Q_{\text{REW}} \setminus \{\langle q, x, \mathsf{u}\rangle\}) \cup \{\langle q, x, \mathsf{e}\rangle\};$
  end
until $Q_{\text{TEMP}} = Q_{\text{REW}};$
$Q_{\text{FIN}} := \{q \mid \langle q, \mathsf{r}, \mathsf{e}\rangle \in Q_{\text{REW}}\};$
return Q_{FIN}

applicability condition for existential rules

apply only useful reduction steps

[Gottlob, Orsi & P., ICDE 2011, ACM Trans. Database Syst. 2014]

FO-Rewritable OMQ Languages

Theorem: (**L**,**CQ**), where **L** $\in$ { **Linear**, **Acyclic**, **Sticky** }, is FO-Rewritable

Via the Bounded Derivation Depth Property (BDDP)

but, the BDDP-based algorithm is very expensive

can we do better?

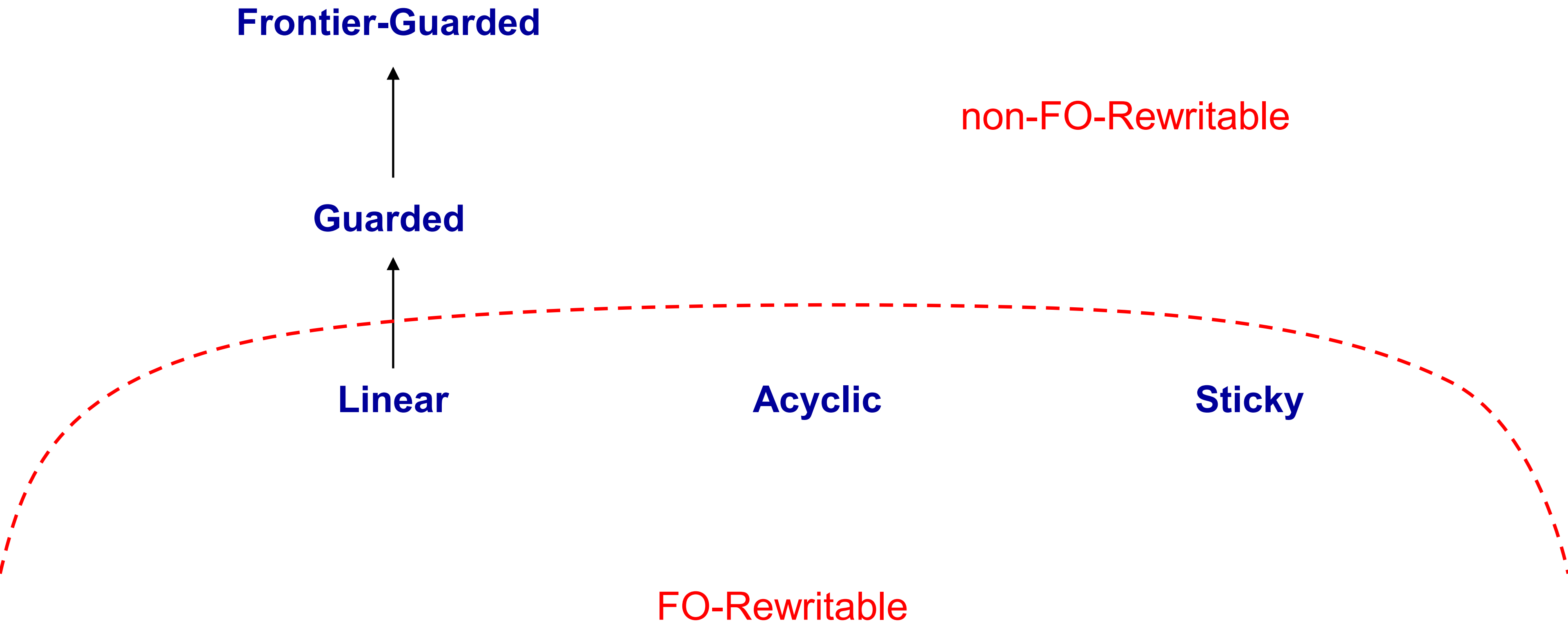
use the XRewrite algorithm

Piece-based rewriting - based on a refined notion of unification

[König, Leclère, Mugnier & Thomazo, RR 2012, Semantic Web 2015]

Recap

What about deciding FO-Rewritability?



Deciding FO-Rewritability

$\{\, R(x,y),\ S(y) \rightarrow S(x),\quad R(x,y),\ P(x) \rightarrow S(y) \,\}$

$\{\, P(\cdot),\ R(\cdot,\cdot),\ S(\cdot) \,\}$

$P(x) \wedge R(x,y) \wedge S(y)$

$Q = (\mathbf{S},\ O,\ q(x,y))$

rewrite

$P(x) \wedge R(x,y)$

Deciding FO-Rewritability

$\{ R(x,y), S(y) \rightarrow S(x), \quad R(x,y), P(x) \rightarrow S(y) \}$

$\{ P(\cdot), R(\cdot,\cdot), S(\cdot) \}$

$S(y)$

$Q = (\mathbf{S}, O, q(y))$

rewrite

✘

Deciding FO-Rewritability

FORew(L,QL)

Input: an OMQ Q $\in$ (**L**,**QL**)

Question: is Q FO-Rewritable?

What is the complexity of **FORew(Guarded,CQ)** and **FORew(Frontier-Guarded,CQ)**?

☹

Deciding FO-Rewritability

FORew(L,QL)

Input: an OMQ Q $\in$ (**L**,**QL**)

Question: is Q FO-Rewritable?

Theorem: FORew(L,CQ), where **L** $\in$ { **Guarded**, **Frontier-Guarded** } is in 3EXPTIME, and 2EXPTIME-hard even for bounded arity

[Barceló, Berger & P., 2017]

Deciding FO-Rewritability

Theorem: FORew(Guarded,BCQ) is in 3EXPTIME and 2EXPTIME-hard even for bounded arity

Upper Bound:

- Characterize FO-Rewritability via the finiteness of a set of certain "tree-like" databases

- Construct an alternating tree automaton **A**, with double-exponentially many states, such that the OMQ is FO-Rewritable iff the language of **A** is finite

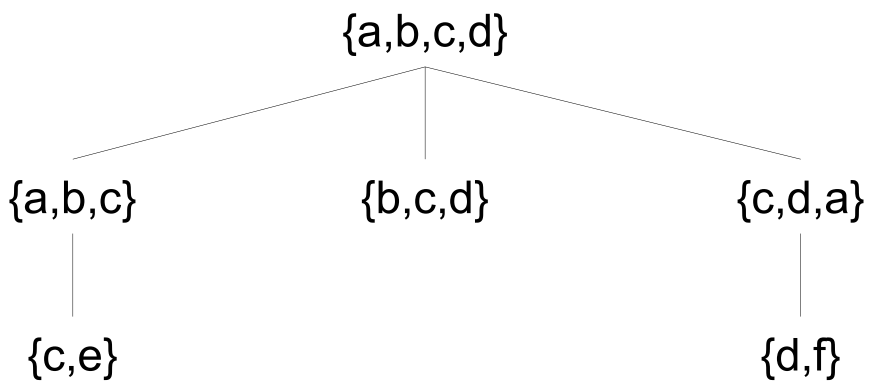
Lower Bound:

- Inherited from **FORew(ELI,CQ)**
  [Bienvenu, Hansen, Lutz & Wolter, IJCAI 2016]

[Barceló, Berger & P., 2017]

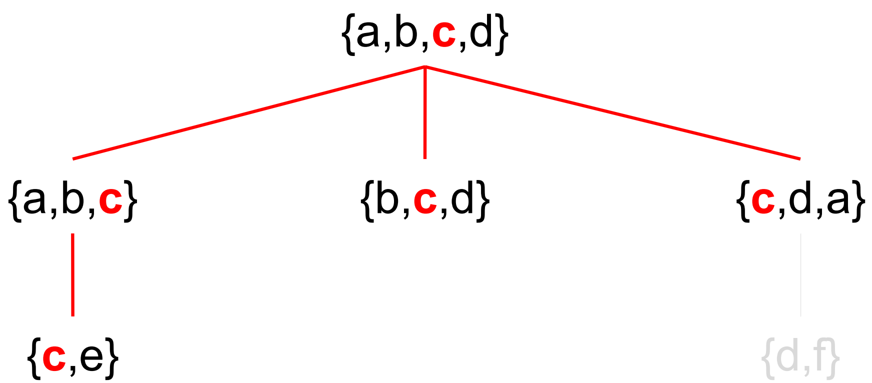
Tree Decomposition

D = { R(a,b,c), T(c,e), R(b,c,d), S(c,d,a), P(d,f), T(f,f) }



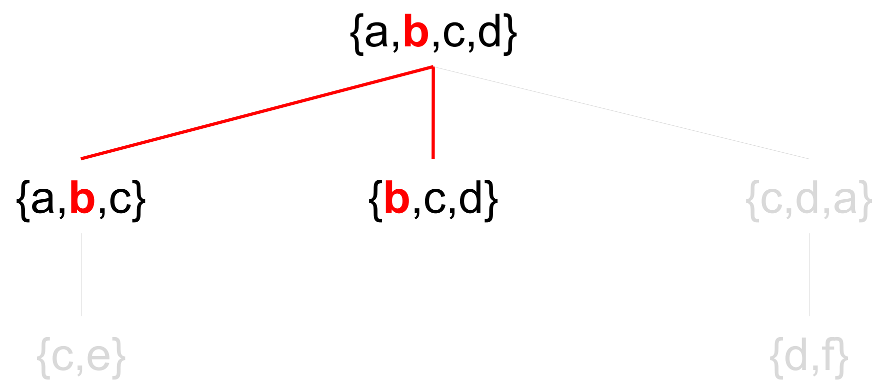
Tree Decomposition

D = { R(a,b,c), T(c,e), R(b,c,d), S(c,d,a), P(d,f), T(f,f) }



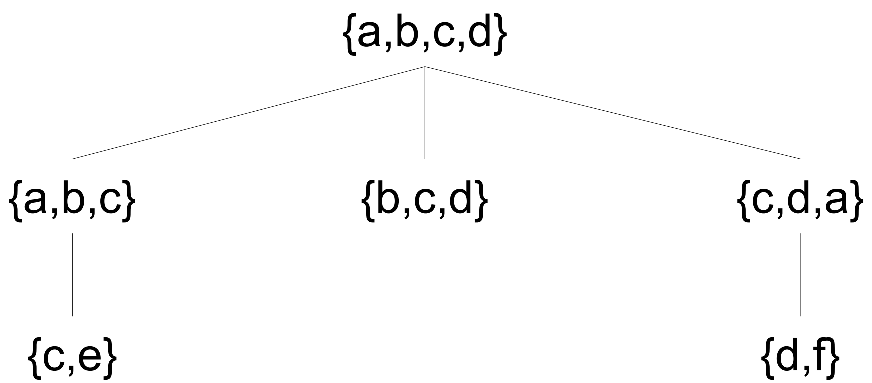
Tree Decomposition

D = { R(a,b,c), T(c,e), R(b,c,d), S(c,d,a), P(d,f), T(f,f) }



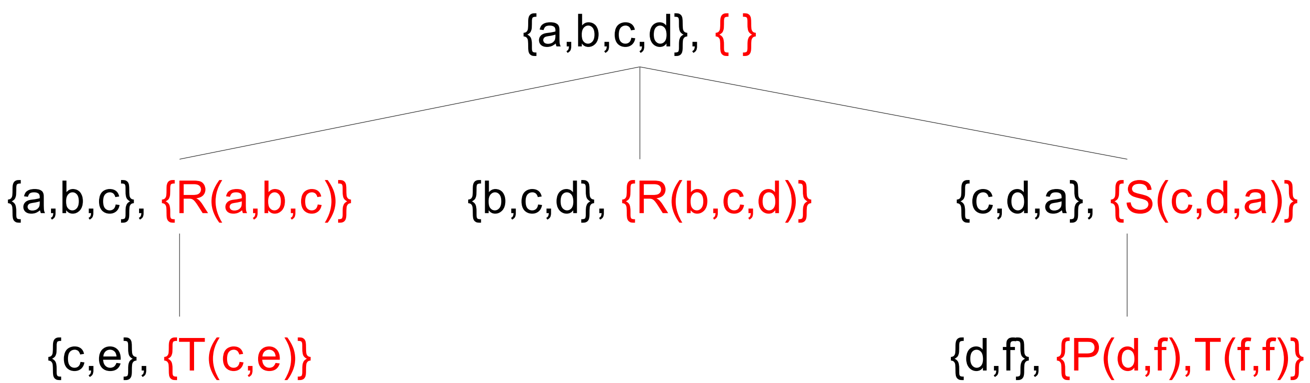
Tree Decomposition

D = { R(a,b,c), T(c,e), R(b,c,d), S(c,d,a), P(d,f), T(f,f) }



Tree Decomposition

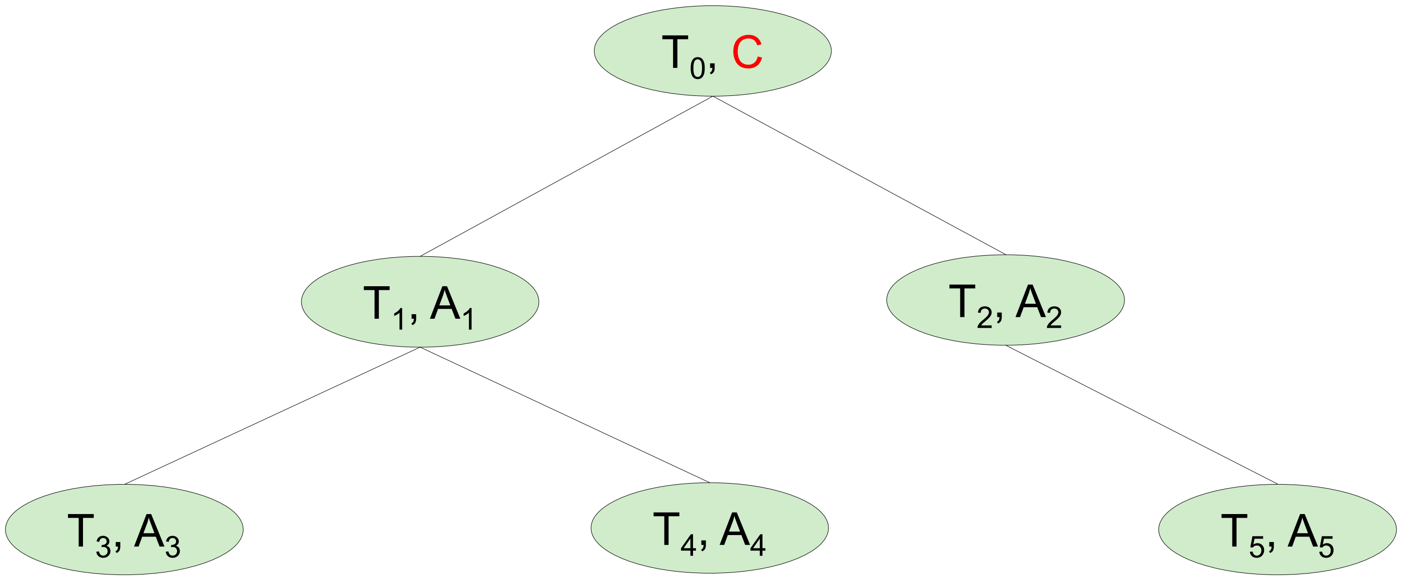
D = { R(a,b,c), T(c,e), R(b,c,d), S(c,d,a), P(d,f), T(f,f) }



C-Tree Databases

(…or, almost "tree-like" databases)

Definition: An **S**-database D is a C-tree, where C $\subseteq$ D, if it has the form:



for each i > 0, $|T_i| \leq$ arity(**S**)

Characterizing FO-Rewritability

Proposition: Let Q = (**S**, O, q) $\in$ (**Guarded**,**BCQ**):

Q is FO-Rewritable

unravelling and compactness $\uparrow$ $\Updownarrow$ $\downarrow$ Q is UCQ-Rewritable

there exists k ≥ 0 such that, for every C-tree D over **S**,

with |dom(C)| ≤ (arity(**S**,O) $\cdot$ |q|), it holds that:

D $\models$ Q $\Rightarrow$ there exists D' $\subseteq$ D with |D'| ≤ k such that D' $\models$ Q

Characterizing FO-Rewritability

Proposition: Let Q = (**S**, O, q) $\in$ (**Guarded**,**BCQ**):

Q is FO-Rewritable

$\Updownarrow$

there exist finitely many (non-isomorphic)

C-trees D over **S**, with |dom(C)| ≤ (arity(**S**,O) $\cdot$ |q|), such that:

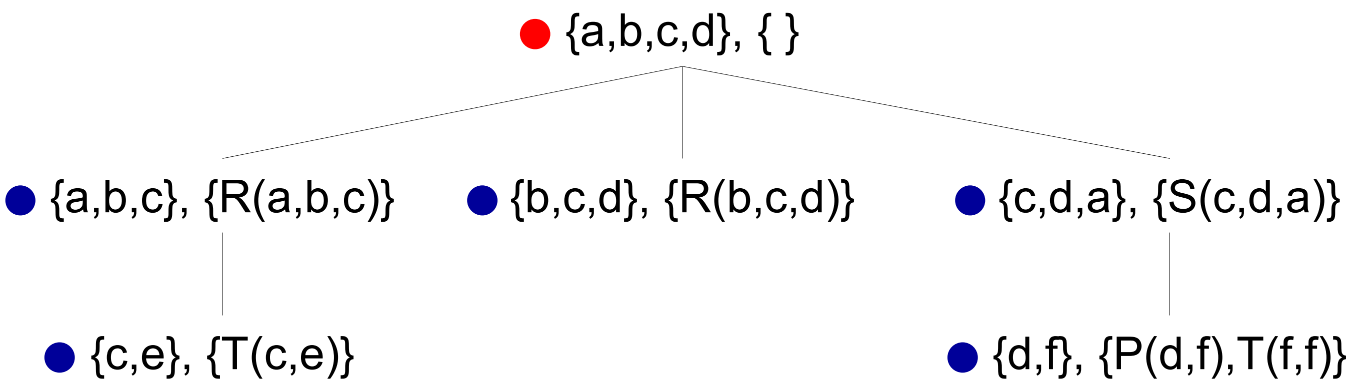
(i) D $\models$ Q

(ii) remove an atom from D $\Rightarrow$ Q is violated

(iii) D is non-redundant

Well-Colored Tree Decomposition

D = { R(a,b,c), T(c,e), R(b,c,d), S(c,d,a), P(d,f), T(f,f) }



node v is red $\Rightarrow$ v is the least common ancestor of a non-empty set of blue nodes

Characterizing FO-Rewritability

Proposition: Let Q = (**S**, O, q) $\in$ (**Guarded**,**BCQ**):

Q is FO-Rewritable

$\Updownarrow$

there exist finitely many (non-isomorphic)

C-trees D over **S**, with |dom(C)| $\leq$ (arity(**S**,O) $\cdot$ |q|), such that:

(i) D $\models$ Q

(ii) remove an atom from D $\Rightarrow$ Q is violated

(iii) D is well-colored

the language of an alternating tree automaton **A**

with double-exponentially many states

Characterizing FO-Rewritability

Proposition: Let Q = (**S**, O, q) $\in$ (**Guarded**,**BCQ**):

Q is FO-Rewritable

$\Updownarrow$

the language of **A** is finite

(which is feasible in exponential time in the number of states)

Deciding FO-Rewritability

Theorem: FORew(Frontier-Guarded,BCQ) is in 3EXPTIME

Q $\in$ (**Frontier-Guarded**,**BCQ**)

a BCQ is a frontier-guarded rule

Q' $\in$ (**Frontier-Guarded**,**BAQ**)

by treeifying the rule-bodies

[Bárány, ten Cate & Segoufin, ICALP 2011, J. ACM 2015]

Q'' $\in$ (**Guarded**,**BAQ**)

Q is FO-Rewritable $\Leftrightarrow$ Q'' is FO-Rewritable

[Barceló, Berger & P., 2017]

Deciding FO-Rewritability: Next Steps

- Practical rewriting algorithms for (**Frontier-Guarded**,**CQ**)

- Such a practical algorithm exists for (**EL**,**AQ**)

  [Hansen, Lutz, Seylan & Wolter, IJCAI 2015]

- …and it has been recently extended to (**EL**,**CQ**)

  [Hansen & Lutz, DL 2017]

Recap

What about the size of the FO rewritings?

Frontier-Guarded

can be checked in 3EXPTIME

Guarded

Linear **Acyclic** **Sticky**

FO-Rewritable

Height/Size of XRewrite(Q)

Given an OMQ $Q = (\mathbf{S}, O, q) \in (\mathbf{L}, \mathbf{CQ})$

worst-case optimal

L	Height	Size
Linear	$\lvert q \rvert$	$\lvert \mathbf{S} \rvert^{\lvert q \rvert} \cdot (\mathrm{arity}(\mathbf{S}) \cdot \lvert q \rvert)^{\mathrm{arity}(\mathbf{S}) \cdot \lvert q \rvert}$
Acyclic	$\lvert q \rvert \cdot \mathrm{body}(O)^{\#\mathrm{pred}(O)}$	$2\hat{}(\lvert \mathbf{S} \rvert \cdot (\lvert q \rvert \cdot \mathrm{body}(O)^{\#\mathrm{pred}(O)} \cdot \mathrm{arity}(\mathbf{S}))^{\mathrm{arity}(\mathbf{S})})$
Sticky	$\lvert \mathbf{S} \rvert \cdot (\#\mathrm{terms}(q) + 1)^{\mathrm{arity}(\mathbf{S})}$	$2\hat{}(\lvert \mathbf{S} \rvert \cdot (\#\mathrm{terms}(q) + 1)^{\mathrm{arity}(\mathbf{S})})$

- **Linear**: the rewriting step replaces an atom with one atom
- **Acyclic**: the rewriting can be seen as a tree of depth at most #pred(O)
- **Sticky**: only variables of q occur more than once in a disjunct

Upper/Lower Bound for **Frontier-Guarded**

- The automata-based approach provides a UCQ-rewriting - disjunction of the trees accepted by the automaton (very large - 5EXP)

- Triple-exponential lower bound for the size of UCQ-rewritings for (**EL**,**CQ**)

  [Bienvenu, Lutz & Wolter, IJCAI 2013]

Target More Succinct Query Languages

In particular, what about

- Positive existential queries (PE)
- Non-recursive Datalog queries (NDL)
- First-order queries (FO)

Even for (**DL-Lite$_R$**,**CQ**)

- No PE/NDL-rewriting of polynomial size
- No FO-rewriting of polynomial size (unless the PH collapses)

…it holds even for (**Acyclic**,**CQ**)

[Gottlob, Kikot, Kontchakov, Podolskii, Schwentick & Zakharyaschev, Artif. Intell. 2014]

FO-Rewritability: Pure Approach

Two crucial limitations:

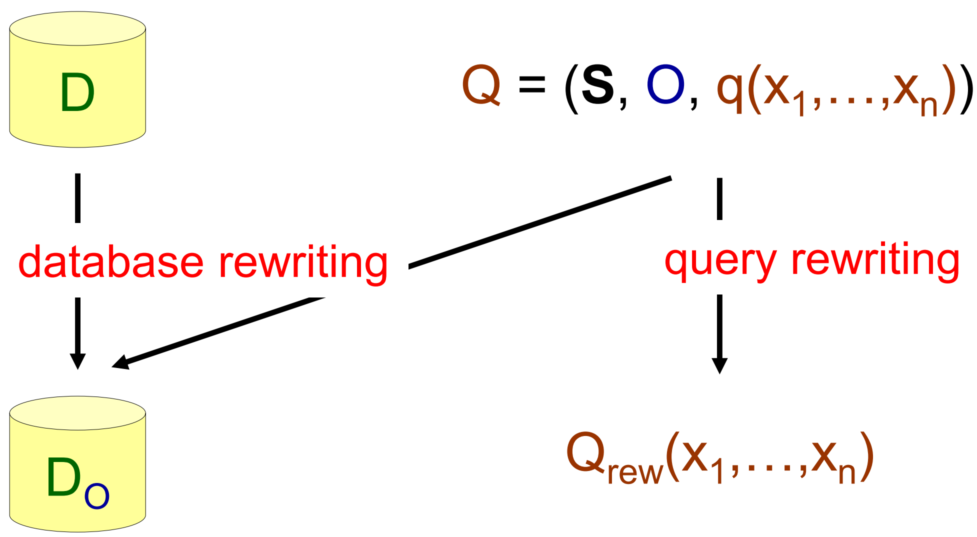
- No small rewritings - even for lightweight languages like **Linear** or **DL-Lite$_R$**

- Simple OMQs are immediately excluded, e.g.,

({HasChild, Human}, {HasChild(x,y), Human(y) $\rightarrow$ Human(x)}, Human(x))

a more refined approach is needed

FO-Rewritability: Combined Approach



both steps in polynomial time!!!

for every **S**-database D : Q(D) = $Q_{rew}(D_O)$

[Lutz, Toman & Wolter, IJCAI 2009]

FO-Rewritability: Combined Approach

schema assumptions

Size	Arity	**Linear**	**Acyclic**	**Sticky**	**Guarded**	**Fr-Guarded**
∞	∞	✓	[×]	[[×]]	×	×
∞	$\leq k$	✓	[×]	✓	[[×]]	×
$\leq k$	∞	✓	✓	[[×]]	×	×
$\leq k$	$\leq k$	✓	✓	✓	✓	?

[×] - assuming PSPACE $\neq$ NEXPTIME

[[×]] - assuming PSPACE $\neq$ EXPTIME

FO-Rewritability: Combined Approach

schema assumptions

Size	Arity	Linear	Acyclic	Sticky	Guarded	Fr-Guarded
∞	∞	✓	[×]	[[×]]	×	×
∞	≤ k	✓	[×]	✓	[[×]]	×
≤ k	∞	✓	✓	[[×]]	×	×
≤ k	≤ k	✓	✓	✓	✓	?

via the Polynomial Witness Property

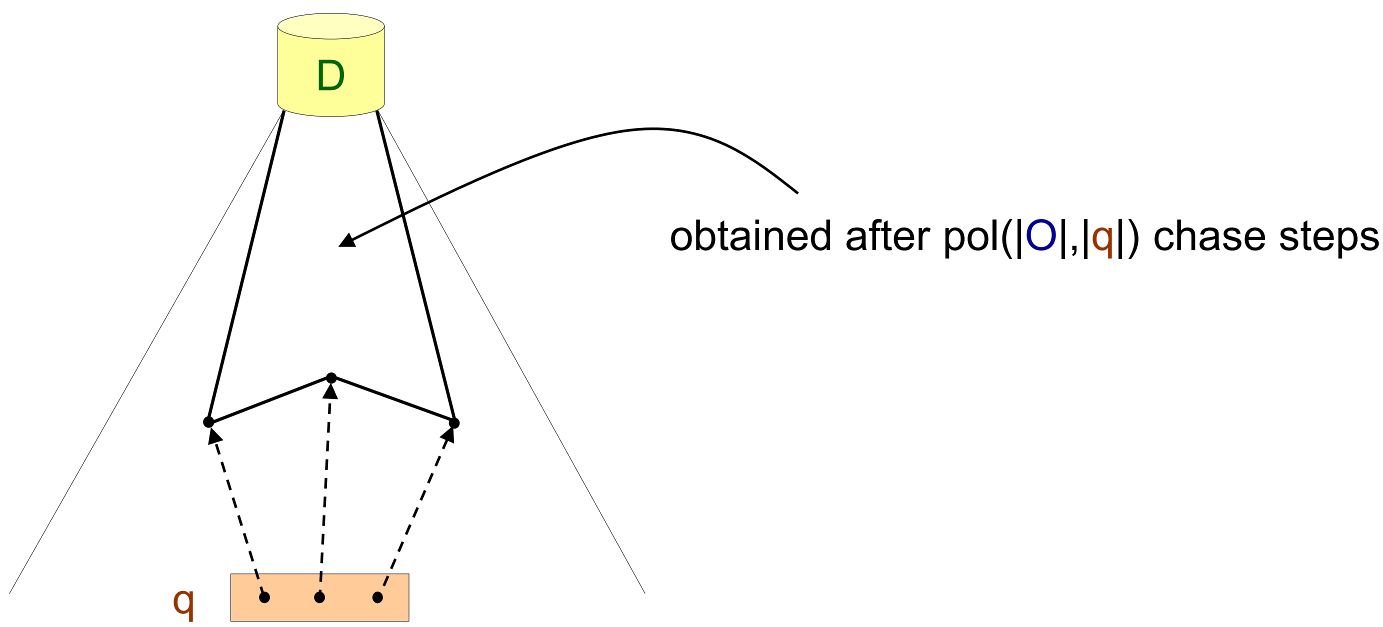
[Gottlob, Kikot, Kontchakov, Podolskii, Schwentick & Zakharyaschev, Artif. Intell. 2014] + [Gottlob, Manna & P., KR 2014]

Polynomial Witness Property (PWP)

Definition: (**L**,**CQ**) enjoys the PWP if there exists a polynomial pol(·) such that

for every Q = (**S**, O, q(**x**)) ∈ (**L**,**CQ**), **S**-database D, and **t** ∈ dom(D)$^{|\mathbf{x}|}$

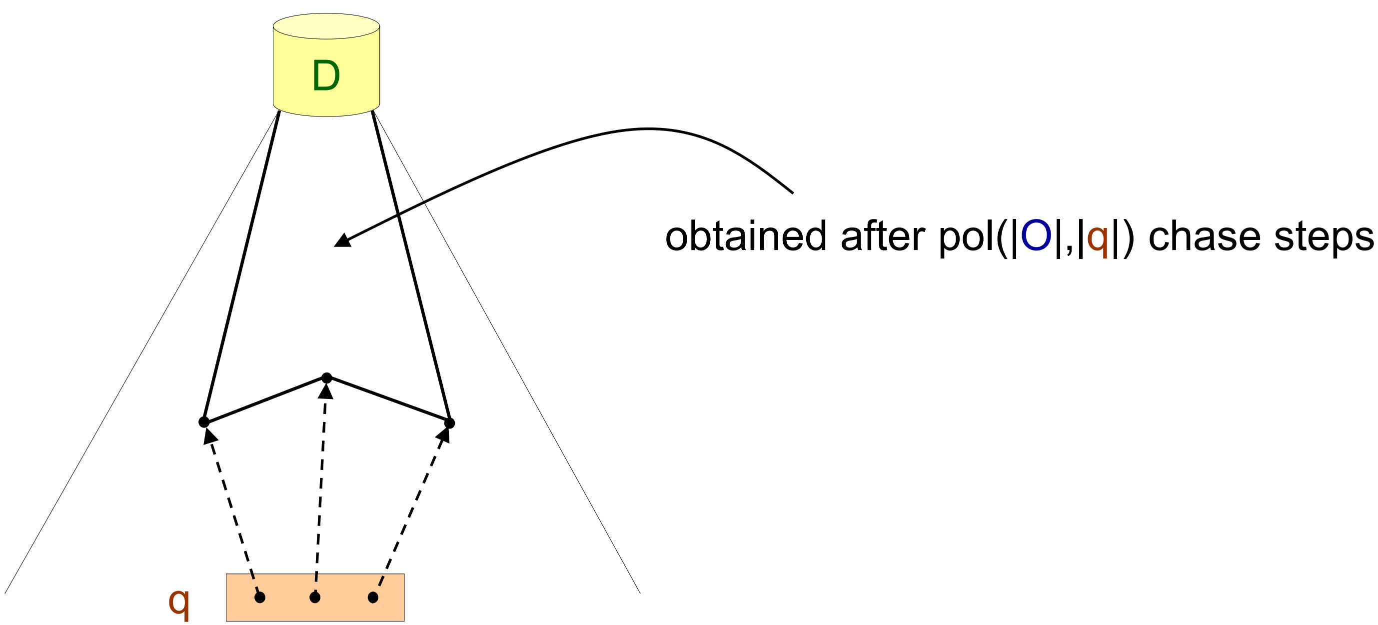
t ∈ Q(D) ⇒ q(**t**) can be entailed after pol(|O|,|q|) chase steps



[Gottlob, Kikot, Kontchakov, Podolskii, Schwentick & Zakharyaschev, Artif. Intell. 2014]

Polynomial Witness Property (PWP)

Proposition: PWP $\Rightarrow$ PE/NDL-rewritings constructible in polynomial time, assuming databases with at least two constants



[Gottlob, Kikot, Kontchakov, Podolskii, Schwentick & Zakharyaschev, Artif. Intell. 2014]

FO-Rewritability: Combined Approach

schema assumptions

Size	Arity	**Linear**	**Acyclic**	**Sticky**	**Guarded**	**Fr-Guarded**
∞	∞	✓	[×]	[[×]]	×	×
∞	≤ k	✓	[×]	✓	[[×]]	×
≤ k	∞	✓	✓	[[×]]	×	×
≤ k	≤ k	✓	✓	✓	✓	?

via the Polynomial Witness Property

[Gottlob, Kikot, Kontchakov, Podolskii, Schwentick & Zakharyaschev, Artif. Intell. 2014] + [Gottlob, Manna & P., KR 2014]

FO-Rewritability: Combined Approach

schema assumptions

Size	Arity	Linear	Acyclic	Sticky	Guarded	Fr-Guarded
∞	∞	✓	[×]	[[×]]	×	×
∞	≤ k	✓	[×]	✓	[[×]]	×
≤ k	∞	✓	✓	[[×]]	×	×
≤ k	≤ k	✓	✓	✓	✓	?

via the Polynomial Witness Property?

[Gottlob, Kikot, Kontchakov, Podolskii, Schwentick & Zakharyaschev, Artif. Intell. 2014] + [Gottlob, Manna & P., KR 2014]

FO-Rewritability: Combined Approach

schema assumptions

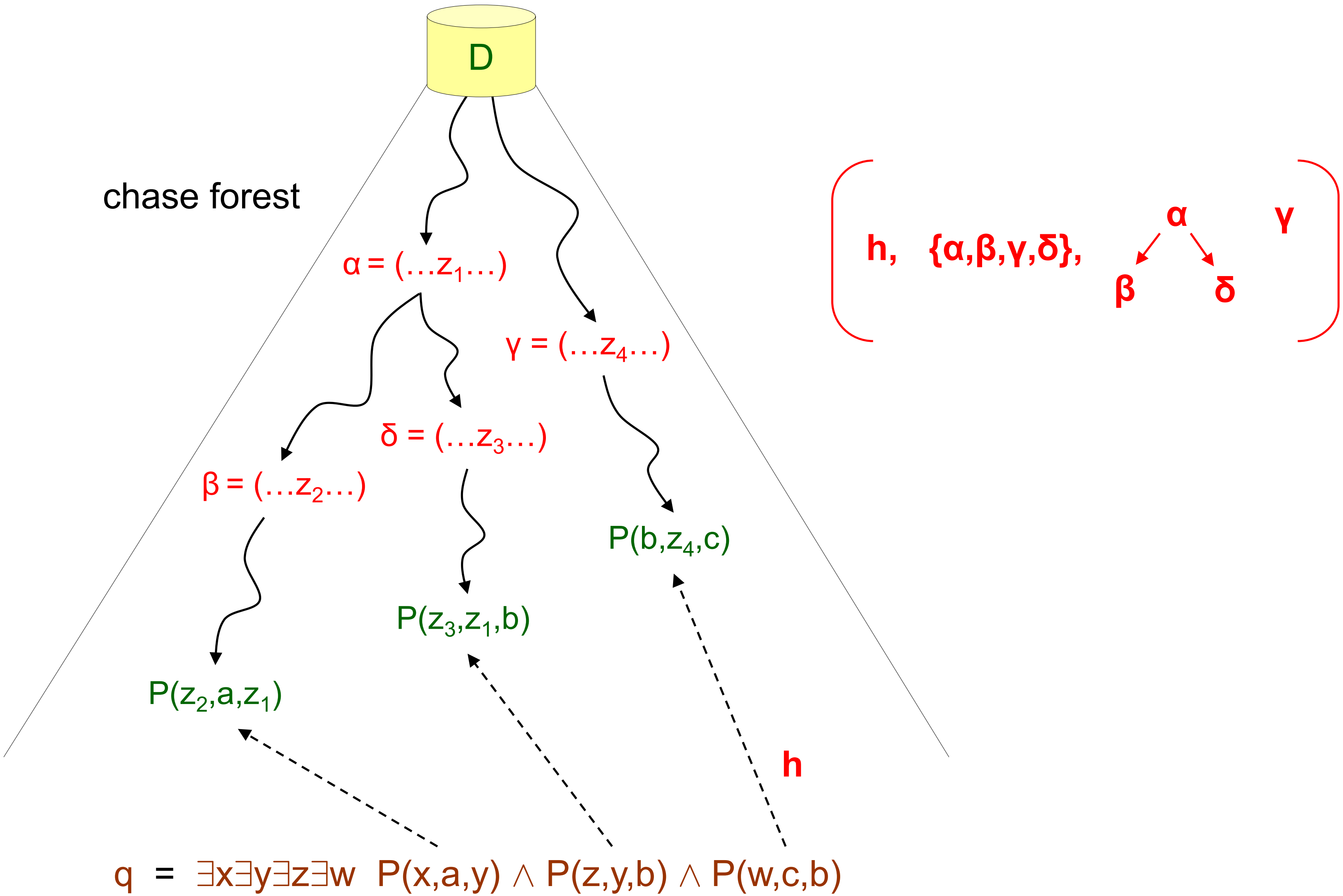
Size	Arity	Linear	Acyclic	Sticky	Guarded	Fr-Guarded
∞	∞	✓	[×]	[[×]]	×	×
∞	≤ k	✓	[×]	✓	[[×]]	×
≤ k	∞	✓	✓	[[×]]	×	×
≤ k	≤ k	✓	✓	✓	✓	?

via proof generators

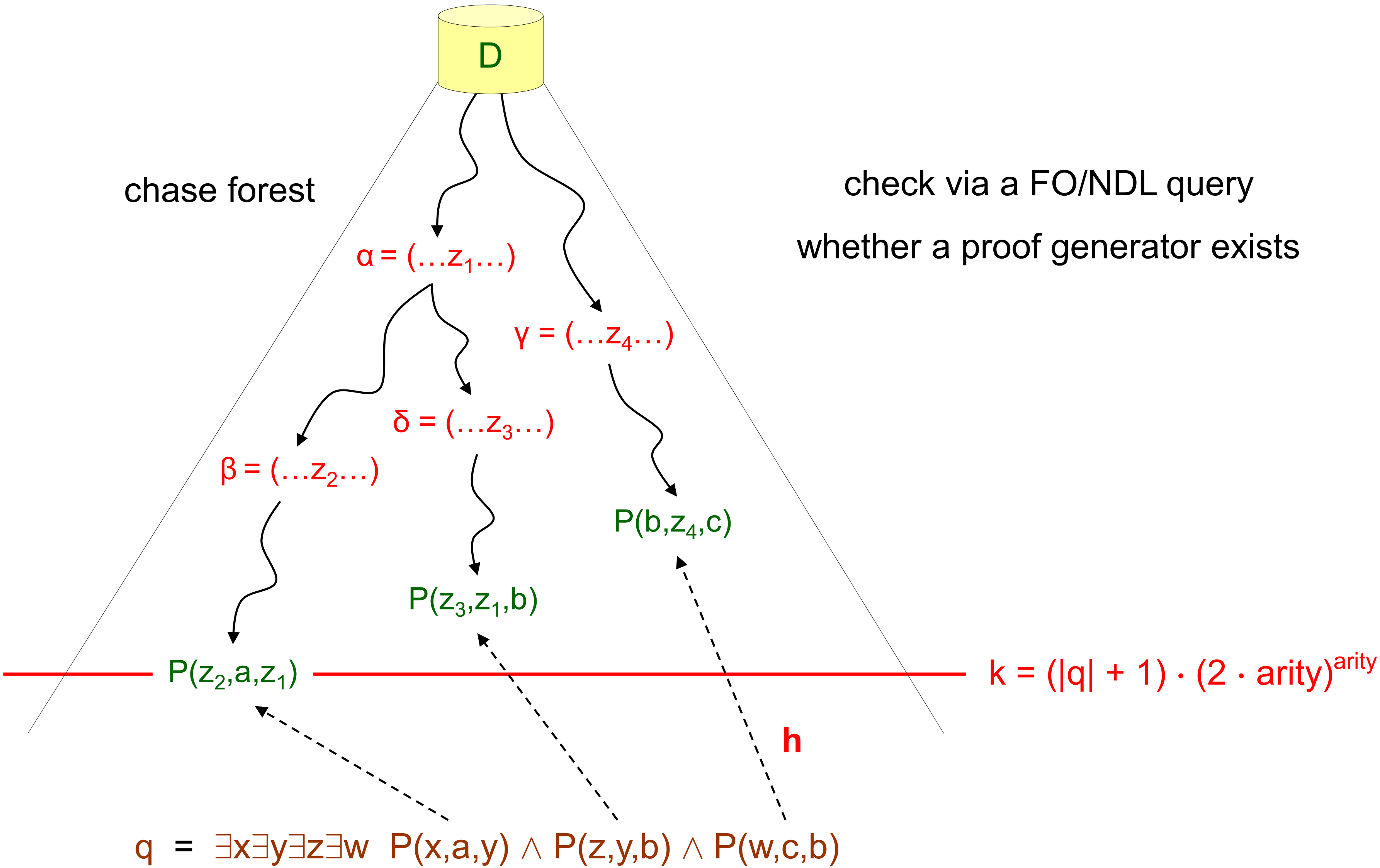
a compact representation of an exponentially-sized witness

[Gottlob, Manna & P., IJCAI 2015]

Proof Generator



Proof Generator



FO-Rewritability: Combined Approach

schema assumptions

Size	Arity	Linear	Acyclic	Sticky	Guarded	Fr-Guarded
∞	∞	✓	[×]	[[×]]	×	×
∞	$\leq k$	✓	[×]	✓	[[×]]	×
$\leq k$	∞	✓	✓	[[×]]	×	×
$\leq k$	$\leq k$	✓	✓	✓	✓	?

a unique positive case without polynomially-sized witnesses

[Gottlob, Manna & P., IJCAI 2015]

FO-Rewritability: Combined Approach

schema assumptions

Size	Arity	Linear	Acyclic	Sticky	Guarded	Fr-Guarded
∞	∞	✓	[×]	[[×]]	×	×
∞	≤ k	✓	[×]	✓	[[×]]	×
≤ k	∞	✓	✓	[[×]]	×	×
≤ k	≤ k	✓	✓	✓	✓	?

via linearization

encode the type of the guard-atom in a single predicate

[Gottlob, Manna & P., KR 2014]

FO-Rewritability: Combined Approach

schema assumptions

Size	Arity	**Linear**	**Acyclic**	**Sticky**	**Guarded**	**Fr-Guarded**
∞	∞	✓	[×]	[[×]]	×	×
∞	$\leq k$	✓	[×]	✓	[[×]]	×
$\leq k$	∞	✓	✓	[[×]]	×	×
$\leq k$	$\leq k$	✓	✓	✓	✓	?

fixing the schema is not enough

we should fix the ontology, and then adapt the linearization technique

[Thomazo, Personal Communication 2017]

Some Final Remarks

- **FO-Rewritable languages**
  - Practical resolution-based algorithms exist (XRewrite, Piece-based rewriting)
  - Prototype systems exist (Nyaya, Graal)
- **Far from practical algorithms for checking FO rewritability**
  - Notable exception the algorithm for (**EL**,**CQ**)
  - Prototype system Grind
- **Polynomial combined FO rewriting algorithms are of theoretical nature**
  - Can we construct compact UCQs?

Thank you!