Finite Controllability of Conjunctive Query Answering with Existential Rules: Two Steps Forward

Giovanni Amendola, Nicola Leone and Marco Manna
University of Calabria
amendola@mat.unical.it, leone@mat.unical.it, manna@mat.unical.it

Abstract
Reasoning with existential rules typically consists of checking whether a Boolean conjunctive query is satisfied by all models of a first-order sentence having the form of a conjunction of Datalog rules extended with existential quantifiers in rule-heads. To guarantee decidability, five basic decidable classes—linear, weakly-acyclic, guarded, sticky, and shy—have been singled out, together with several generalizations and combinations. For all basic classes, except shy, the important property of finite controllability has been proved, ensuring that a query is satisfied by all models of the sentence if, and only if, it is satisfied by all of its finite models. This paper takes two steps forward: (i) devise a general technique to facilitate the process of (dis)proving finite controllability of an arbitrary class of existential rules; and (ii) specialize the technique to complete the picture for the five mentioned classes, by showing that also shy is finitely controllable.

1 Introduction
The problem of answering a Boolean query \( q \) against a logical theory consisting of an extensional database \( D \) and an ontology \( \Sigma \) is attracting the attention of various fields of Computer Science, from Artificial Intelligence [Baget et al., 2011; Calvanese et al., 2013; Gottlob et al., 2014; Amendola et al., 2018] to Database Theory [Bienvenu et al., 2014; Alviano and Pieris, 2015] and Logic [Pérez-Urbina et al., 2010]. This problem, known as ontology-based query answering (OBQA), is usually stated as \( D \cup \Sigma \models_q q \), and it is equivalent to checking whether \( q \) is satisfied by all models of \( D \cup \Sigma \) according to the “open-world” semantics of first-order logic (where \( \models_q \) is the entailment under arbitrary models).

Both Description Logics [Baader et al., 2002] and Datalog☆ [Calì et al., 2009a] have been recognized as effective families of formal knowledge representation languages to specify \( \Sigma \), while union of (Boolean) conjunctive queries, \( U(B)CQs \) for short, is the most common and studied formalism to express \( q \). Unfortunately, for both these families, OBQA is generally undecidable [Johnson and Klug, 1984; Rosati, 2007; Calì et al., 2013]. Hence, a number of syntactic decidable classes of the above languages have been singled out. But decidability alone is not the only desideratum. For example, a good balance between computational complexity and expressiveness is of high importance too. But there is another property that is turning out to be as interesting as challenging to prove: finite controllability [Johnson and Klug, 1984]. A class \( C \) of ontologies is finitely controllable if, for each triple \( (D, \Sigma, q) \) with \( \Sigma \in C \), it holds that \( D \cup \Sigma \models_q q \) if, and only if, \( D \cup \Sigma \models_q q \), where \( \models_q \) is the entailment under finite models only. Note that this is equivalent to \( D \cup \Sigma \models_q q \) implies there is a finite model \( M \) of \( D \cup \Sigma \) such that \( M \models_q q \), as the “only if” direction is trivially true. And there are applications, both in Databases [Johnson and Klug, 1984; Rosati, 2006] and Knowledge Representation [Rosati, 2008; Ibáñez-García et al., 2014], where reasoning over finite models is preferred.

In this paper we focus on the Datalog☆ family, and consider for \( \Sigma \) a set of existential rules of the form \( \forall X \forall Y (\phi(X,Y) \rightarrow \exists Z p(X,Z)) \), where the body \( \phi(X,Y) \) is a conjunction of atoms, and the head \( p(X,Z) \) is a single atom. The main decidable classes rely on the following basic five syntactic properties: weak-acyclicity [Fagin et al., 2005], linearity [Calì et al., 2009b], guardedness [Calì et al., 2013], stickiness [Calì et al., 2010], and shyness [Leone et al., 2012], which underlie the basic classes called weakly-acyclic, linear, guarded, sticky, and shy, respectively. Several variants and combinations of these classes have been defined and studied too [Baget et al., 2010; Krötzsch and Rudolph, 2011; Calì et al., 2012; Gottlob et al., 2013; 2018], as well as semantic properties subsuming the syntactic ones [Baget et al., 2011; Leone et al., 2012].

The above basic classes are pairwise incomparable, except for linear which is strictly contained in guarded and shy, as reported in Figure 1. Interestingly, both weakly-acyclic and shy strictly contain datalog —the well-known class of existential-free rules of the form \( \forall X \forall Y (\phi(X,Y) \rightarrow p(X)) \). Moreover, sticky strictly contains joinless —the class collecting sets of rules where each body contains no repeated variable. The latter, introduced by [Gogacz and Marcinkowski, 2017] to prove that sticky is finitely controllable, is also exploited in this paper. Finally, both linear and joinless strictly contain inclusion-dependencies —the well-known class of relational database dependencies collecting sets of rules with one single body atom and no repeated variable.
Figure 1: Taxonomy of the basic Datalog± classes.

Under arbitrary models, OBQA can be reduced to answering q over a universal model U that can be homomorphically embedded into every other arbitrary (finite or infinite) model of D ∪ Σ. Hence, D ∪ Σ |=a q if, and only if, U |= q. A way to compute such a U is to employ the chase procedure [Beeri and Vardi, 1984]. Intuitively, starting from D, the chase “re-pairs” violations of rules by repeatedly adding new atoms — introducing fresh values, called nulls, whenever required by an existential variable — until all rules are satisfied. In the classical setting, the chase is sound and complete. But when finite model reasoning is required, then the chase is generally incomplete, unless ontologies are finitely controllable.

Finite controllability of weakly-acyclic comes for free as every ontology here admits a finite universal model [Fagin et al., 2005]. Conversely, the proof of this property for the subsequent three classes has been a very different matter. Complex, yet intriguing, constructions have been devised for linear [Rosati, 2006; Bárany et al., 2014], guarded [Bárány et al., 2014], and more recently for sticky [Gogacz and Marcinkowski, 2017]. To complete the picture, we address the same problem for shy, while taking two steps forward: ¹

We devise a general technique, called canonical rewriting, to facilitate the process of (dis)proving finite controllability of any class of existential rules. Indeed, via this technique, we can immediately (re)confirm that linear is finitely controllable since inclusion-dependencies is. Additionally, we can prove that sticky-join [Calì et al., 2012] — generalizing both sticky and linear — is finitely controllable since sticky is.

We specialize the canonical rewriting to complete the picture for the aforementioned five basic classes, and show that also shy is finitely controllable since joinless is.

2 Basics: Ontology-Based Query Answering

Basics. Let C, N and V denote pairwise disjoint discrete sets of constants, nulls and variables, respectively. A term t is an element of T = C ∪ N ∪ V. An atom α is a labeled tuple p(t₁, ..., tₘ), where p is a predicate symbol, m is the arity of both p and α, and t₁, ..., tₘ are terms. An atom is simple if it contains no repeated term, and it is propositional if it has arity 0. Given two sets A and B of atoms, a homomorphism from A to B is a mapping h : T → T such that c ∈ C implies h(c) = c, and h(A) ⊆ B with h(A) obtained from A by replacing each term t by h(t). An instance I is a discrete set of atoms where each term is either a constant or a null.

Syntax. A database D is a finite null-free instance. An (existential) rule is a logical implication of the form ∀X∀Y (ϕ(X, Y) → 9Z p(X, Z)), with X ⊆ Y ∪ Z ⊆ V, whose body (resp., head) body(ρ) = ϕ(X, Y) (resp., head(ρ) = {p(X, Z)}) is a conjunction (or set) of atoms, possibly with constants. The set X is called the frontier of ρ. An ontology Σ is a set of rules. A union of Boolean conjunctive query, UBCQ for short, is a first-order expression of the form ∃Y₁ψ₁(Y₁) ∨ ... ∨ ∃Yₜψₜ(Yₜ), where each ψⱼ(Yⱼ) is a conjunction of atoms. Constants may occur also in q.

Semantics. Consider a triple (D, Σ, q) as above. An instance I satisfies a rule ρ ∈ Σ, written I |= ρ, if whenever there is a homomorphism h from body(ρ) to I, then there is a homomorphism h′ ⊇ h|X from head(ρ) to I. Also, I satisfies Σ, written I |= Σ, if I satisfies each rule of Σ. The (arbitrary) models of D ∪ Σ, written amods(D, Σ), are the set {I : I ⊇ D and I |= Σ}. A model U is universal if, for each M ∈ amods(D, Σ), there is a homomorphism from U to M. An instance I satisfies q, written I |= q, if there is a homomorphism from some ψⱼ(Yⱼ) to I. Also, q is true over D ∪ Σ, written D ∪ Σ |=ₕ q, if each model of D ∪ Σ satisfies q.

Finite controllability. A class C of ontologies is finitely controllable if, for each triple (D, Σ, q) with Σ ∈ C, D ∪ Σ |=a q implies there is M ∈ fmods(D, Σ) s.t. M ̸|= q, where fmods(D, Σ) denotes the finite models of D ∪ Σ. This is usually stated as D ∪ Σ |=a q if, and only if, D ∪ Σ |=ₕ q.

3 Step 1: Canonical Rewriting

We design a general technique (see Figure 2) to facilitate the process of (dis)proving finite controllability of an arbitrary class of existential rules. Specifically, any triple (D, Σ, q) is rewritten in the semantically equivalent (canonical) triple (D′, Σ′, q′) that has the advantage of being structurally simpler than the original one. Indeed, D′ is a propositional database, while Σ′ and q′ are constant-free structures containing simple atoms only. Rather than giving boring formal details, we prefer to shed some light on our construction via a simple example. Let D = {p(a), p(b), r(a, b)}, and Σ consists of the following rules:

\[(p_0) \quad p(X) \rightarrow \exists Y \; r(Y, X)\]
\[(p_1) \quad r(X, Y) \rightarrow p(X)\]
\[(p_2) \quad r(X, X) \rightarrow s(X)\]

We now construct D′ = {p[a], p[b], r[a, b]} as the propositional database obtained by encoding each atom of D in a different fresh predicate. Then, from Σ we build Σ′ as follows:

\[(p_{0,0}) \quad p[a] \rightarrow \exists Y r_{[1,a]}(Y)\]
\[(p_{0,1}) \quad p[b] \rightarrow \exists Y r_{[1,b]}(Y)\]
\[(p_{0,2}) \quad p[a] \rightarrow \exists Y r_{[2,a]}(Y, X)\]
\[(p_{1,0}) \quad r_{[a,a]} \rightarrow p[a]\]
\[(p_{1,1}) \quad r_{[a,b]} \rightarrow p[a]\]
\[(p_{1,2}) \quad r_{[b,a]} \rightarrow p[b]\]
\[(p_{1,3}) \quad r_{[b,b]} \rightarrow p[b]\]
\[(p_{1,4}) \quad r_{[a,1]}(Y) \rightarrow p[a]\]
\[(p_{1,5}) \quad r_{[b,1]}(Y) \rightarrow p[b]\]
\[(p_{1,6}) \quad r_{[1,a]}(X) \rightarrow p[1](X)\]
\[(p_{1,7}) \quad r_{[1,b]}(X) \rightarrow p[1](X)\]
\[(p_{1,8}) \quad r_{[1,1]}(X) \rightarrow p[1](X)\]
\[(p_{1,9}) \quad r_{[1,2]}(X, Y) \rightarrow p[1](X)\]
\[(p_{2,0}) \quad r_{[a,a]} \rightarrow s[a]\]
\[(p_{2,1}) \quad r_{[b,b]} \rightarrow s[b]\]
\[(p_{2,2}) \quad r_{[1,1]}(X) \rightarrow s[1](X)\]

¹This paper is an abridged version of [Amendola et al., 2017].
Intuitively, new predicates encode tuples of terms consisting of constants (a and b) from $D \cup \Sigma$, and placeholders (1 and 2, where the latter is the maximum arity among predicates occurring in $\Sigma$) of arbitrary terms of $C \cup N$. Any rule $\rho_{i,j}$ of $\Sigma'$ is called child of $\rho_i$, while the latter is the parent of $\rho_{i,j}$. Finally, by following the same intuition, we apply the canonical rewriting to a set of atoms over $C \cup \Sigma \cup V$ w.r.t. the constants of $D \cup \Sigma$. E.g., given $A = \{r(a, X), p(X)\}$, we obtain the following family of sets $A^c = \{\{r[a,a], p[a]\}, \{r[a,b], p[b]\}, \{r[a,1], p[1]\}, \{r[a,1][X], p[1][X]\}\}$. Thus, given the UBCQ $q = \exists X \{r(a, X), p(X)\} \lor \exists Y \{r(Y, Y)\}$, its canonical rewriting will be $q^c = \{r[a,a], p[a]\} \lor \{r[a,b], p[b]\} \lor \exists X \{r[a,1](X), p[1](X)\} \lor \{r[a,1], p[1]\} \lor \exists Y \{r[1,1](Y)\}$.

Accordingly, we define the unpacking function $\mathfrak{R}$ from atoms over the set of canonical predicates to atoms over the set of original ones. The function generalizes in the obvious way to set of atoms, rules, and queries. For instance, if $A' = \{r[1,2](X,Y), r[1,1](\eta), r[1,1](\eta)\}$ with $\eta \in N$, then $\mathfrak{R}(A') = \{r(X,Y), r(\eta, a), r(\eta, b)\}$. Informally, the unpacking function acts like an “inverse” operator of the canonical rewriting.

Consider now the model $D = D \cup \{p(\eta), r(\eta, a), r(\eta, b), r(\eta, \eta), s(\eta)\}$ of $D \cup \Sigma$. Also, let $M' = D^c \cup \{p[1][\eta], r[1,0](\eta), r[1,0](\eta), r[1,2](\eta, \eta)\}$ be a model of $D^c \cup \Sigma'$. Note that, whenever there is a homomorphism $h$ from the body of some rule $\rho_{i,j}$ to $M'$, then $h$ also maps the body of the parent $\rho_i$ to $M$. Indeed, this is true as $\mathfrak{R}(M') \subseteq M$. However, although $M'$ is a model of $D^c \cup \Sigma'$, $\mathfrak{R}(M')$ is not a model of $D \cup \Sigma$, as $\alpha' = r[1,2](\eta, \eta)$ does not trigger any rule of $\Sigma'$, while $\mathfrak{R}(\alpha')$ triggers $\rho_2$. In fact, this happens because: there is no bijection between the placeholders $\{1, 2\}$ and $\{\eta\}$, and also $M'$ contains no atom of the form $r[1,2](\eta, \eta)$, where a bijection between $\{1\}$ and $\{\eta\}$ exists. To overcome such a mismatch, we focus on a suitable subset of $\text{amods}(D^c \cup \Sigma')$.

**Definition 1.** A model $N'$ of $D^c \cup \Sigma'$ is canonical if: (i) it is constant-free; and (ii) for each non-simple atoms $\alpha'$ of $N'$, there is a simple atom $\beta'$ of $N$ such that $\mathfrak{R}(\alpha') = \mathfrak{R}(\beta')$.

From any $M'$ it is possible to build a canonical model $N'$ that can be homomorphically mapped to $M'$: (1) set $N'$ to $M'$; (2) replace each constant $c$ of $N'$ by a fresh null $\eta_c$; (3) let $E$ be the set of nulls occurring in atoms of $N'$ that violate condition (ii); (4) for each null $\eta$ of $E$, let $\text{pad}(\eta) = \{\eta_0, \ldots, \eta_k\}$, where $k$ is the maximum number of times that $\eta$ occurs in an atom of $N'$; (5) for each $\eta$ of $E$, replace each atom $\alpha'$ of $N'$ containing $\eta$ with new atoms obtained from $\alpha'$ by arranging the elements of $\text{pad}(\eta)$ in the positions where $\eta$ occurs, while avoiding repetitions; and (6) remove all atoms of $N'$ violating condition (ii).

Hence, in our running example, $N' = D^c \cup \{p[1][\eta_1], p[1][\eta_2], r[1,0](\eta_1), r[1,0](\eta_1), r[1,0](\eta_2), r[1,0](\eta_1), r[1,0](\eta_2), r[1,2](\eta_1), r[1,2](\eta_2), r[1,2](\eta_1)\}$. Now, from the above intuitions, we collect some relevant properties.

**Proposition 1.** Let $A$ be a set of atoms over $C \cup V$ w.r.t. the constants of $D \cup \Sigma$, $M'$ be a model of $D^c \cup \Sigma'$, and $N'$ be a canonical model of $D^c \cup \Sigma'$. It holds that:

a. If, for some $A' \in A^c$, $M' \models A'$, then $\text{th}(M') \models A$;

b. If $\text{th}(N') \models A$, then $N' \models A'$, for some $A' \in A^c$;

c. There is $N'' \in \text{ams}(D^c, \Sigma')$ that can be homomorphically mapped to $M'$. Also, if $M'$ is finite, then $N''$ is.

d. $\text{th}(N')$ is a model of $D \cup \Sigma$;

e. $\text{amods}(D, \Sigma) \subseteq \text{amods}(D^c, \Sigma')$.

With these properties in hand we are able to provide a formal argument for the implications depicted in Figure 2.

**Theorem 1.** It holds that:

a. $D \cup \Sigma \models_a q$ if, and only if, $D^c \cup \Sigma' \models_a q^c$; and

b. $D \cup \Sigma \models_i q$ if, and only if, $D^c \cup \Sigma' \models_i q^c$.

**Proof sketch.** We now describe the proof sketch for finite model reasoning (b). We start with the “only-if” direction.

$$D^c \cup \Sigma' \models_i q^c \quad \text{by contradiction}$$

$$\exists N' \in \text{cfmods}(D^c, \Sigma') \text{ s.t. } N' \neq q^c \quad \text{(by Prop. 1.c)}$$

$$\forall q^c \in q^c \text{ it holds that } N' \neq q^c \quad \text{(by definition)}$$

$$\mathfrak{R}(N') \neq q \quad \text{(by Prop. 1.b)}$$

$$D \cup \Sigma \models_i q \quad \text{(by Prop. 1.d)}$$

Now, we sketch the proof for the “if” direction.

$$D \cup \Sigma \models_i q \quad \text{(by contradiction)}$$

$$\exists M' \in \text{fmods}(D, \Sigma) \text{ s.t. } M \neq q \quad \text{(by definition)}$$

$$\exists M' \in \text{fmods}(D^c, \Sigma') \text{ s.t. } M' = M \quad \text{(by Prop. 1.e)}$$

$$\forall q^c \in q^c \text{ it holds that } M' \neq q^c \quad \text{(by Prop. 1.a)}$$

$$D^c \cup \Sigma' \models_i q^c \quad \text{(by definition)}$$

The implication chains hold also for arbitrary model reasoning (a), by replacing $\models_i$ by $\models_a$, and $\text{cfmods}$ by $\text{amods}$. □

By exploiting our tool, we can reprove that linear is finitely controllable. In fact, the canonical rewriting of a linear ontology belongs to inclusion-dependencies. Moreover, since the canonical rewriting of a sticky-join [Calli et al., 2012] ontology belongs to sticky, we prove (for the first time) that:

**Theorem 2.** sticky-join is finitely controllable.

Note that the definitions of linear and sticky-join in [Gogacz and Marcinkowski, 2017] are not standard (actually stricter): there, in linear repeated variables are admitted only in rule heads, and the difference between sticky and sticky-join “can only be seen if repeated variables in the heads of the rules are allowed”. Hence, the finite controllability of sticky-join was unknown before our work. One can verify that the proof of their Lemma 4 breaks when moving to the linear (hence sticky-join) ontology $\Sigma = \{p(X, X) \rightarrow r(X); r(X) \rightarrow Yr(Y)\}$ paired with the database $D = \{p(c, c)\}$. 

Figure 2: Application of the canonical rewriting.
4 Step 2: Finite Controllability of Shy

Differently from linear and sticky-join, the canonical rewriting of a shy ontology—although it is still a shy ontology—falls in none of the known finitely controllable classes. Hence, to prove the finite controllability of shy, we are going to specialize the canonical rewriting. But before, let us recall the key properties of this class. Fix a triple \((D, \Sigma, q)\):

**Proposition 2** ([Leone et al., 2012]). Assume \(\Sigma \in \text{shy} \). Given a universal model \(U \supseteq D \cup \Sigma\), a rule \(\rho \in \Sigma\), and a homomorphism \(h\) from the atoms of \(\rho\) to \(U\), it holds that:

a. if a variable \(X\) of \(\rho\) occurs in two different atoms of \(\text{body}(\rho)\), then \(h(X) \in C\); and

b. if two frontier variables \(X, Y\) of \(\rho\) occur in two different atoms of \(\text{body}(\rho)\) and \(h(X) = h(Y)\), then \(h(X) \in C\).

Consider now the canonical rewriting \(\langle D^c, \Sigma^c, \subseteq^c \rangle\) of \(\langle D, \Sigma, q^c \rangle\). Inspired by well-supported interpretations [Fages, 1991], we define a suitable subset of all finite models:

**Definition 2.** A finite model \(M' \in D^c \cup \Sigma^c\) is well-supported if there is an ordering \(\alpha_1, \ldots, \alpha_n\) of its atoms s.t., for each \(j \in \{1, \ldots, n\}\), it holds that: (i) \(\alpha_j \in D^c\), or (ii) there is \(p \in \Sigma^c\) and a homomorphism from the atoms of \(\rho\) to \(\alpha_1, \ldots, \alpha_j\) mapping \(\text{body}(\rho)\) to \(\alpha_1, \ldots, \alpha_{j-1}\) and \(\text{head}(\rho)\) to \(\alpha_j\).

Subsequently, we define \(\Sigma^c_{\text{j}}\) as the set of all joinless rules of \(\Sigma^c\). For instance, consider the ontology \(\Sigma = \{p_0 = p(X) = r(X, Y), p_1 = p(X) = s(X, Y), p_2 = r(X, Y), s(X, Y) \rightarrow t(X)\}\) and the database \(D = \{p(a)\}\). According to the canonical rewriting, \(\Sigma^c = \{p_0 = p[a] \rightarrow r_{[1]}(Y); p_{0,1} = p_1(X) \rightarrow r_{[1,2]}(X, Y); p_{1,0} = p[a] \rightarrow s_{[1]}(Y); p_{1,1} = p_2(Y) = s_{[1]}(X); p_{2,0} = r_{[1]}(a), s_{[1]}(a) \rightarrow t_{[1]}(a); p_{2,1} = r_{[1]}(a), s_{[1]}(a) \rightarrow t_{[1]}(X); p_{2,3} = r_{[1,2]}(X, Y), s_{[1,2]}(X, Y) \rightarrow t_{[1]}(X)\}\) and \(D^c = \{p[a]\}\). Hence, \(\Sigma^c_{\text{j}} = \{p_{0,0}, p_{0,1}, p_{1,0}, p_{1,1}, p_{2,0}\}\) and \(M' = \{p[a], r_{[1]}(a), s_{[1]}(a), t_{[1]}(a)\}\) is a well-supported model of \(D^c \cup \Sigma^c\). Note that here is where the shyness of \(\Sigma^c\) is crucial. Also, \(N_0'\) can be homomorphically embedded into \(N'\) by mapping \(\langle \eta, 2, 1 \rangle\) to \(\eta\), and \(\langle \eta, 3, 1 \rangle\) to \(\eta\). These insights are the basis of the following properties.

**Proposition 3.** It holds that:

a. The ontology \(\Sigma_{\text{glut}}\) is finitely controllable [Gogacz and Marcinkowski, 2017];

b. For each \(M' \in \text{fmods}(D^c \cup \Sigma^c)\) there exists a well-supported finite model \(N' \in D^c \cup \Sigma^c\) s.t. \(N' \subseteq M'\);

c. Whenever \(\Sigma\) is shy, for each \(M' \in \text{wsfmods}(D^c, \Sigma^c)\), there is a model \(N' \in \text{wsfmods}(D^c, \Sigma^c)\) that can be homomorphically embedded into \(M'\).

With these properties in hand, we can now provide a formal argument to show the main result of the section.

**Theorem 3.** Under shy ontologies, it holds that:

\[ D^c \cup \Sigma^c \models q^c,\text{ if, and only if, } D^c \cup \Sigma^c \models q^c. \]

**Proof sketch.** We prove the “if” direction.

\[ D^c \cup \Sigma^c \models q^c \]

(by contradiction)

\[ \Rightarrow \exists M'_1 \in \text{amods}(D^c, \Sigma^c) \text{ s.t. } M'_1 \models q^c \]

(by definition)

\[ \Rightarrow \exists M'_1 \in \text{amods}(D^c, \Sigma^c) \text{ s.t. } M'_1 \subseteq M' \]

(by Prop. 3a)

\[ \Rightarrow \exists M'_2 \in \text{wsfmods}(D^c, \Sigma^c) \text{ s.t. } M'_2 \models q^c \]

(by Prop. 3b)

\[ \Rightarrow \exists M'_3 \in \text{wsfmods}(D^c, \Sigma^c) \text{ s.t. } M'_3 \models q^c \]

(by Prop. 3c)

\[ \Rightarrow M'_1 \in \text{fmods}(D^c, \Sigma^c) \text{ s.t. } M'_1 \models q^c \]

(by definition)

\[ D^c \cup \Sigma^c \models q^c \]

(by definition)

Therefore, we can conclude that the statement is true since the “only if” direction trivially holds.

Finally, the finite controllability of shy follows by combining Theorem 3 with Theorem 1.

**Theorem 4.** shy is finitely controllable.

5 Conclusion

To complete the related works started with the Introduction, we recall that finite controllability in OBQA was formalized for the first time by [Rosati, 2006].

While he was working on a question left open two decades before by [Johnson and Klug, 1984] about containment of conjunctive queries in case of both arbitrary and finite databases.

Basically, using our terminology, they proved that ontologies mixing both inclusion-dependencies and functional-dependencies are not finitely controllable, by leaving open the case in which ontologies contain inclusion-dependencies only.

Rosati then answered positively this question. And a decade later, by demonstrating that shy is finitely controllable (Step 2), we complete an important picture around the basic decidable Datalog\(^+\) classes.

But we believe that the techniques developed here could have future further applications for. For example, it would be interesting to extend the canonical rewriting (Step 1) by encoding in the predicates also a bounded number of nulls (apart from constants and placeholders of nulls).

This requires more complex techniques, which however would apply to both guarded [Körtgesch and Rudolph, 2011] (extending guarded and weakly-acyclic), and weakly-sticky-join [Cali et al., 2012] (extending sticky-join and weakly-acyclic).

5192
Acknowledgments
The paper was partially supported by MISE under project “S2BDW” (F050389/01-03/X32)-“Horizon2020” PON I&C 2014-20, and by Regione Calabria under project “DLV Large Scale” (CUP J28C17000220006)-POR Calabria 2014-20.

References


