Rewriting Guarded Negation Queries

Vince Bárány*, Michael Benedikt, and Balder ten Cate **

[1] LogicBlox Inc., Atlanta, GA
[2] Department of Computer Science, University of Oxford
[3] Department of Computer Science, UC-Santa Cruz

Abstract. The Guarded Negation Fragment (GNFO) is a fragment of first-order logic that contains all unions of conjunctive queries, a restricted form of negation that suffices for expressing some common uses of negation in SQL queries, and a large class of integrity constraints. At the same time, as was recently shown, the syntax of GNFO is restrictive enough so that static analysis problems such as query containment are still decidable. This suggests that, in spite of its expressive power, GNFO queries are amenable to novel optimizations. In this paper we provide further evidence for this, establishing that GNFO queries have distinctive features with respect to rewriting. Our results include effective preservation theorems for GNFO, Craig Interpolation and Beth Definability results, and the ability to express the certain answers of queries with respect to GNFO constraints within very restricted logics.

1 Introduction

The guarded negation fragment (GNFO) is a syntactic fragment of first-order logic, introduced in [BtCS11]. On the one hand, GNFO can be seen as a constraint language: it captures classical database referential integrity constraints (that is, inclusion dependencies), specifications of relationships between schemas given in a common schema mapping language (namely that of Local-As-View constraints [Len02, FKMP05]) and the first-order translations of ontologies specified in some of the most popular description logics [BCM+03]. It contains these prior classes by virtue of extending the Guarded Fragment of first-order logic [AvBN98]. On the other hand, GNFO is more suitable than the Guarded Fragment for defining queries: for example, it contains all positive existential queries, corresponding in expressiveness to unions of conjunctive queries. The defining characteristic of GNFO formulas is that a subformula $\psi(\mathbf{x})$ with free variables $\mathbf{x}$ can only be negated when used in conjunction with a positive literal $\alpha(\mathbf{x},\mathbf{y})$, i.e. a relational atom or an equality, containing all free variables of ψ, as in

$$\alpha(\mathbf{x},\mathbf{y}) \wedge \neg\psi(\mathbf{x}) \ ,$$

where order and repetition of variables is irrelevant. One says that the literal $\alpha(\mathbf{x},\mathbf{y})$ *guards* the negation. Unguarded negations $\neg\phi(x)$ of formulas with at most one free variable are supported through the use of an equality guard $x = x$.

It was shown in [BtCS11] that GNFO possesses a number of desirable static analysis properties. For example, every satisfiable GNFO-formula has a finite model (*finite*

[*] Work done while affiliated with TU Darmstadt.

[**] ten Cate was supported by NSF Grants IIS-0905276 and IIS-1217869. Benedikt supported by EPSRC grant EP/H017690/1

model property), as well as a, typically infinite, model of bounded tree-width (*tree-like model property*). It follows that satisfiability and implication (hence, by the finite model property, finite satisfiability and finite implication) of GNFO formulas are decidable.

In [BtCO12] the implications of GNFO for database theory are explored: for example, an SQL-based syntax for GNFO is defined, and an analog of stratified Datalog is also presented. The complexity of query evaluation and "open world query answering" (i.e. computing certain answers) is identified for several GNFO-based languages, and many important static analysis problems for queries (e.g. boundedness for the GNFO-variant of Datalog) are shown to be decidable.

In this work we investigate properties of GNFO related to *rewriting*. We first present results showing that GNFO queries or constraints satisfying additional semantic properties can be rewritten into restricted syntactic forms. For example, we show that every GNFO query that is closed under extensions can be effectively rewritten as an existential GNFO formula.We give an analogous result for queries closed under homomorphisms. We also show that the GNFO sentences that can be expressed in a common constraint language – that of tuple-generating dependencies (TGDs), are precisely those that can be rewritten into a recently-introduced class of TGDs, the frontier-guarded TGDs.

We then turn to the setting where one has views and queries both defined within GNFO, imposing an additional restriction that the free variables in the views and queries are guarded. We show that if the views and queries satisfy the semantic restriction that the views *determine* the query, then we can find a rewriting of the query in terms of the views, with the rewriting belonging again to GNFO. Following ideas of Marx [Mar07], we proceed by showing that an important model theoretic theorem for first-order logic, the Projective Beth Definability theorem, holds in GNFO. We show that, unlike in the case of the Guarded Fragment, the more general Craig Interpolation Theorem of first-order logic holds for GNFO. In contrast, we show that Craig Interpolation and Projective Beth fail for the guarded fragment, contradicting claims made in earlier work.

We also study the existence of rewritings computing the certain answer to conjunctive queries. We show that GNFO sentences that take the form of dependencies have particularly attractive properties from the point of view of open world query answering. We extend and correct results of Baget et. al. [BMRT11a] by showing that the certain answers are expressible in a small fragment of Datalog. Using this, we show that the existence of first-order rewritings can be effectively decided for dependencies in GNFO.

For space reasons, most proofs are deferred to the full version.

2 Definitions and Preliminaries

We will make use of some basic notions of database theory – in particular the notion of schema or signature, relational structures, and the following "classical" query classes: conjunctive queries (CQs), Unions of Conjunctive Queries (UCQs), first-order logic formulas (FO), existential and positive existential FO, and Datalog. Abiteboul, Hull, and Vianu [AHV95] is a good reference for all of these languages. Note that by default we allow constants in our signature (that is, in CQs, UCQs, etc.). In this work, by the *active domain* of a structure I we mean the set of values that occur in some relation of I along with the values named by constants. In our arguments we will often make use of the following basic notions from classical model theory: s a *reduct* of a structure is obtained by restricting the signature), an *expansion* of a structure (obtained by adding additional relations). By a *fact* of a structure $\mathfrak{A}$ we mean an expression $R(a_1,\dots,a_n)$ where $(a_1,\dots,a_n)$ is a tuple belonging to a relation $R^{\mathfrak{A}}$. For structures I,J, we write $I \subseteq J$ if the domain of I is contained in the domain of J, every fact of I is also a fact of J, and I and J agree on the interpretation of all constant symbols. In this case, we say that I is a *subinstance* of J and that J is a *super-instance* of I. If, furthermore, every fact of J containing only values from the domain of I belongs to I, then we say that I is an *induced substructure* of J and J is an *extension* of I.

The basics of GNFO. The Guarded Negation Fragment (GNFO) is built up inductively according to the grammar:

$$\phi ::= R(\mathbf{t}) |\ t_1 = t_2 \mid \exists x\ \phi \mid \phi \vee \phi \mid \phi \wedge \phi \mid R(\mathbf{t},\mathbf{y}) \wedge \neg\phi(\mathbf{y})$$

where R is either a relation symbol or the equality relation $x = y$, and the t_i represent either variables or constants. Notice that any use of negation must occur conjoined with an atomic relation that contains all the free variables of the negated formula – such an atomic relation is a *guard* of the formula. The purpose of allowing equalities as guards is to ensure that every formula with at most one free variable can be considered guarded, and we often write $\neg\phi$ instead of $(x = x) \wedge \neg\phi$, when ϕ has no free variables besides (possibly) x. If τ is a signature consisting of constants and predicates, GNFO$[\tau]$ denotes the GNFO formulas in signature τ.

GNFO should be compared to the *Guarded Fragment*, GFO [AvBN98], typically defined via the grammar:

$$\phi ::= R(\mathbf{x}) |\ \exists \mathbf{x}\ R(\mathbf{x},\mathbf{y}) \wedge \phi(\mathbf{x}) \mid \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi(\mathbf{y})$$

It is easy to see that every union of conjunctive queries is expressible in GNFO. It is only slightly more difficult to verify that every GFO sentence can be expressed in GNFO [BtCS11]. Turning to fragments of first-order logic that are common in database theory, consider *guarded tuple-generating dependencies*: that is, sentences of the form

$$\forall \mathbf{x}\ R(\mathbf{x}) \wedge \phi(\mathbf{x}) \rightarrow \exists \mathbf{y}\ \psi(\mathbf{x},\mathbf{y}) \ .$$

where ϕ,ψ are conjunctions of relational atomic formulas. By simply writing out such a sentence using $\exists,\neg,\wedge$, one sees that it is convertible to a GNFO sentence. In particular, every *inclusion dependency* is expressible in GNFO, and many of the common dependencies used in data integration and and in exchange (e.g. linear-guarded dependencies, also known as Local-As-View (LAV) constraints [Len02, FKMP05]) lie in GNFO.

Looking at constraints that come from Entity-Relationship and other semantic data models, we see that concept subsumption, when translated into relational database terminology, is expressible in GFO, hence in GNFO. Going further, many of the common description logic languages used in the semantic web (e.g. $\mathcal{ALC}$ and $\mathcal{ALCHIO}$ [BCM$^+$03]) are known to admit translations into GFO, and hence into GNFO.

We will frequently make use of the key result from [BtCS11]:

Theorem 1. *A* GNFO *sentence is satisfiable over all structures iff it is satisfiable over finite structures. Satisfiability and validity are decidable (and* 2ExpTime*-complete).*

We have mentioned before that GNFO can capture important integrity constraints, but in [BtCO12] it is also argued that GN-RA, and hence GNFO, captures many uses of negation in queries in practice.

3 Rewriting special GNFO queries

Preservation theorems in model theory are results that syntactically characterize the formulas within a logic that satisfy important semantic properties. Two examples from classical model theory are the Łoś-Tarski theorem, stating that the universal formulas capture all first order properties closed under taking induced substructures, and the Homomorphism Preservation theorem, stating that existential positive sentences capture all first order properties closed under homomorphism [CK90]. It is known that the Łoś-Tarski theorem fails if we consider equivalence only over finite structures [EF99], while Rossman [Ros08] has shown that the Homomorphism Preservation theorem does hold if we restrict attention to finite structures. A well-known preservation theorem from modal logic is Van Benthem's theorem, stating that basic modal logic captures precisely the fragment of first-order logic invariant under bisimulation [vB83]. The analog for finite structures was proven to hold by Rosen [Ros97], cf. also [Ott04].

Here we will investigate the analogous questions for GNFO. We will start by showing analogs of Van Benthem's theorem for GNFO. We will then identify syntactic fragments that capture the intersection of important fragments of first-order logic with GNFO – from these, new semantic characterizations will follow, including analogs of the Łoś-Tarski and Homomorphism Preservation theorems.

Characterizing GNFO within FO. We first look at the question of characterizing GNFO as the set of all first-order formulas that are invariant under certain simulation relations. In [BtCS11], *guarded-negation bisimulation* were introduced, and it was shown that GNFO captures the fragment of first-order logic that is invariant under GN-bisimulations. Here we give a characterization theorem for a simpler kind of simulation relation, which we call a *strong GN-bisimulation*. We will use this characterization as a basic tool throughout the paper – to show that a certain formula is in GNFO, to argue that two structures must agree on all GNFO formulas, and to amalgamate structures that cannot be distinguished by GN-sentences in a subsignature. The many uses of strong GN-bisimulations suggest that it is really the "right" equivalence relation for GNFO.

Recall that a homomorphism from a structure $\mathfrak{A}$ to a structure $\mathfrak{B}$ is a map from the domain of $\mathfrak{A}$ to the domain of $\mathfrak{B}$ that preserves the relations as well as the interpretation of the constant symbols. We say that a set, or tuple, of elements from a structure $\mathfrak{A}$ is *guarded* in $\mathfrak{A}$ if there is a fact of $\mathfrak{A}$ that contains all elements in question except possibly for those that are the interpretation of a constant symbol.

Definition 1 (Strong GN-bisimulations). *A strong GN-bisimulation between structures $\mathfrak{A}$ and $\mathfrak{B}$ is a non-empty collection Z of pairs $(\mathbf{a},\mathbf{b})$ of guarded tuples of elements of $\mathfrak{A}$ and of $\mathfrak{B}$, respectively, such that for every $(\mathbf{a},\mathbf{b}) \in Z$:*

- *there is a homomorphism $h : \mathfrak{A} \to \mathfrak{B}$ with $h(\mathbf{a}) = \mathbf{b}$ and such that "h is compatible with Z", meaning that $(\mathbf{c}, h(\mathbf{c})) \in Z$ for every guarded tuple $\mathbf{c}$ in $\mathfrak{A}$.*
- *there is a homomorphism $g : \mathfrak{B} \to \mathfrak{A}$ with $g(\mathbf{b}) = \mathbf{a}$ and such that "g is compatible with Z", meaning that $(g(\mathbf{d}), \mathbf{d}) \in Z$ for every guarded tuple $\mathbf{d}$ in $\mathfrak{B}$.*

We write $(\mathfrak{A},\boldsymbol{a}) \to^s_{GN} (\mathfrak{B},\boldsymbol{b})$ if the map $\boldsymbol{a} \mapsto \boldsymbol{b}$ extends to a homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ that is compatible with some strong GN-bisimulation between $\mathfrak{A}$ and $\mathfrak{B}$. Note that, here, $\boldsymbol{a}$ and $\boldsymbol{b}$ are not required to be guarded tuples. We write $(\mathfrak{A},\boldsymbol{a}) \sim^s_{GN} (\mathfrak{B},\boldsymbol{b})$ if, furthermore, $\boldsymbol{a}$ is a guarded tuple in $\mathfrak{A}$ (in which case we also have that $(\mathfrak{B},\boldsymbol{b}) \sim^s_{GN} (\mathfrak{A},\boldsymbol{a})$). These notations can also be indexed by a signature σ, in which case they are defined in terms of σ-reducts of the respective structures. A first-order formula $\phi(\boldsymbol{x})$ is preserved by $\sim^s_{GN}$ *if, whenever $(\mathfrak{A},\boldsymbol{a}) \to^s_{GN} (\mathfrak{B},\boldsymbol{b})$ and $\mathfrak{A} \models \phi(\boldsymbol{a})$, then $\mathfrak{A} \models \phi(\boldsymbol{b})$.*

The reader may verify as an exercise that if there exists a strong GN-bisimulation between two structures, then the respective induced substructures consisting of the elements designated by constant symbols must be isomorphic.

Our first "expressive completeness" result characterizes GNFO as the fragment of first-order logic that is invariant for strong GN-bisimulations.

Theorem 2. *A first-order formula $\phi(\boldsymbol{x})$ is preserved by $\to^s_{GN}$ (over all structures) iff it is equivalent to a* GNFO *formula.*

Strong bisimulations will play a key role in our remaining results. When we want to show that a GNFO formula ϕ can be replaced by another simpler ϕ', we will often justify this by showing that an arbitrary model of ϕ can be replaced by a strongly bisimilar structure where ϕ' holds (or vice versa). The proof of the "hard direction" of Theorem 2 uses the technique of *recursively saturated models* [CK90].

Characterizing fragments of GNFO. We now look at characterizing the intersection of GNFO with smaller fragments of first-order logic. We will start with tuple-generating dependencies (TGDs). Recall that these are sentences of the form:

$$\forall \mathbf{x}\, \phi(\mathbf{x}) \to \exists \mathbf{y}\, \rho(\mathbf{x},\mathbf{y})$$

where ϕ and ρ are conjunctions of relational atoms (not equalities). TGDs capture many classes of integrity constraints used in classical databases, in data exchange, and in ontological reasoning. Static analysis and query answering problems in the latter contexts have in recent years been driving a quest for identifying expressive yet computationally well-behaved classes of TGDs. A guarded TGD (GTGD) is one in which ϕ includes an atom containing all variables occurring in the rule. Guarded TGDs constitute an important class of TGDs at the heart of the Datalog$^{\pm}$ framework [CGL09, BGO10] for which static analysis problems are decidable. More recently, Baget, Leclère, and Mugnier [BLM10] introduced *frontier-guarded TGDs* (FGTGD), defined like guarded TGDs, but where only the variables occurring both in ϕ and in ρ (the *exported* variables) must be guarded by an atom in ϕ. All FGTGDs are equivalent to GNFO sentences, obtained just by writing them out using existential quantification, negation, and conjunction. Theorem 3 below shows that these are *exactly* the TGDs that GNFO can capture.

We need two lemmas, one about GNFO and one about TGDs. For structure I and superinstance J of I, let us denote by $J \ominus I$ the substructure of J obtained by removing all facts containing only values from the active domain of I. We say that J is a *squid-superinstance* of I if (i) every set of elements from the active domain of I that is guarded in J is already guarded in I, and (ii) $J \ominus I$ is a disjoint union of structures J' for which it holds that $(adom(J') \cap adom(I)) \setminus C$ is guarded in I, where C is the set of elements of I named by a constant symbol (intuitively, we can think of J as a squid, where each J' is one of its tentacles). The following lemma, intuitively, allows one to turn an arbitrary superinstance of a structure I into a squid-superinstance of I, modulo strong GN-bisimulation.

Lemma 1. *For every pair of structures I,J with J being a super-instance of I, there is a squid-superinstance J' of I and a homomorphism $h : J' \to J$ whose restriction to I is the identity function, such that $J' \sim^s_{GN} J$ via a strong GN-bisimulation that is compatible with h. Moreover, we can choose J' to be finite if J is.*

We will make use of Lemma 1 as a tool for bringing certain conjunctive queries into a restricted syntactic form, by exploiting the fact that, whenever a tuple from $adom(I)$

satisfies a conjunctive query in a squid-superinstance J of I, then we can partition the atoms of the query into independent subsets that are mapped into different tentacles of J.

The following lemma expresses a general property of TGDs that follows from the fact that TGDs are preserved under taking direct products of structures [Fag82].

Lemma 2. *Let Σ be any set of TGDs and suppose that $\Sigma \models \forall \mathbf{x}(\phi(\mathbf{x}) \to \bigvee_{i=1\dots n} \exists \mathbf{y}_i \psi_i(\mathbf{x},\mathbf{y}_i))$, where ϕ, ψ_i are conjunctions of atoms. Then $\Sigma \models \forall \mathbf{x}(\phi(\mathbf{x}) \to \exists \mathbf{y}_i \psi_i(\mathbf{x},\mathbf{y}_i))$ for some $i \leq n$. This holds both over finite structures and over arbitrary structures.*

We now return to describing our characterization of TGDs that lie in GNFO. Consider a TGD $\rho = \forall \mathbf{x} \beta(\mathbf{x}) \to \exists \mathbf{z} \gamma(\mathbf{xz})$. A *specialisation* of ρ is a TGD of the form $\rho^\theta = \forall \mathbf{x} \beta(\mathbf{x}) \to \exists \mathbf{z}' \gamma'(\mathbf{xz}')$ obtained from ρ by applying some substitution θ mapping the variables $\mathbf{z}$ to constant symbols or to variables among $\mathbf{x}$ and $\mathbf{z}$. The following lemma states that as far as strong GN-bisimulation invariant TGDs are concerned, we can replace any TGD by specializations of it that are equivalent to frontier-guarded TGDs. Its proof relies heavily on the two lemmas above.

Lemma 3. *[TGD specialisations] Let Σ be a set of TGDs that is strong GN-bisimulation invariant and let ρ be a TGD such that $\Sigma \models \rho$. Then there exists a specialisation ρ' of ρ such that $\Sigma \models \rho'$, and such that ρ' is logically equivalent to a conjunction of frontier-guarded TGDs. This holds both over finite structures and over arbitrary structures.*

The result above immediately implies our first main characterization:

Theorem 3. *Every* GNFO*-sentence that is equivalent to a finite set of TGDs on finite structures is equivalent to a finite set of TGDs on arbitrary structures, and such a formula is equivalent (over all structures) to a finite set of FGTGDs.*

In the light of the above result, it may seem tempting to suppose that, similarly, guarded TGDs form the intersection of TGDs and GFO. This is, however, not the case: the TGD $\forall xyz R(x,y) \wedge R(y,z) \to P(x)$ can be equivalently expressed in GFO, but not by means of a guarded TGD; and the guarded TGD $\forall x P(x) \to \exists yz\ E(x,y) \wedge E(y,z) \wedge E(z,x)$ is not expressible in GFO. Instead, we show that the intersection of GFO and TGDs is *acyclic frontier-guarded TGDs*.

Recall from [Yan81] that an *acyclic conjunctive query* is a conjunctive query whose hypergraph is acyclic. There is another equivalent characterization of acyclic conjunctive queries, which is more convenient for our present purposes: a conjunctive query is acyclic if it can be equivalently expressed by a formula of GFO built up from atomic formulas using only conjunction and guarded existential quantification [GLS03]. We say that a TGD $\rho = \forall \mathbf{xy} \beta(\mathbf{x},\mathbf{y}) \to \exists \mathbf{z} \gamma(\mathbf{x},\mathbf{z})$ is acyclic if the conjunctive queries $\exists \mathbf{y} \beta(\mathbf{x},\mathbf{y})$ and $\exists \mathbf{z} \gamma(\mathbf{x},\mathbf{z})$ are both acyclic. Using Theorem 3 above, plus the "Treeification Lemma" of [BGO10], we can characterize the GFO sentences that are equivalent to TGDs:

Theorem 4. *Every* GFO*-sentence that is equivalent to a finite set of TGDs over finite structures is equivalent to a finite set of TGDs on arbitrary structures, and such a formula is equivalent (over all structures) to a finite set of acyclic FGTGDs.*

Existential and Positive-Existential Formulas. We turn to characterizing the existential formulas that are in GNFO, establishing an analog of the Łoś-Tarski theorem. We say that a first-order formula $\phi(\mathbf{x})$ is *preserved under extensions* over a given class of structures if for all structure $\mathfrak{A}$ and $\mathfrak{B}$ from the class, such that $\mathfrak{A} \models \phi(\mathbf{a})$ and $\mathfrak{A}$ is an induced substructure of $\mathfrak{B}$, we have that $\mathfrak{B} \models \phi(\mathbf{a})$.

Theorem 5. *Every* GNFO *formula that is preserved under extensions over finite structures has the same property over all structures, and such a formula is equivalent (over all structures) to an existential formula in* GNFO. *Furthermore, we can decide whether a formula has this property, and also find the existential* GNFO *formula effectively.*

The first part of the first statement follows from the fact that the property of preservation of a GNFO formula can be expressed as a GNFO sentence, along with the finite model property for GNFO. The second part uses the classical Łoś-Tarski theorem to show that a sentence is rewritable as an existential, and then uses our previous infrastructure (e.g. strong bisimulations) to show that any unguarded negations in the existential formula can be removed.

Finally, we consider the situation for GNFO formulas that are positive-existential (for short, $\exists^+$), i.e., that do not contain any negation (and hence, also, only existential quantification) Since GNFO contains all $\exists^+$ formulas, Rossman's theorem [Ros08] implies that the $\exists^+$ formulas are exactly the formulas in GNFO preserved by homomorphism, over all structures or (equivalently, by the finite model property for GNFO) over finite structures. In addition, using the proof of Rossman's theorem plus the decidability of GNFO we can decide whether a GNFO formula can be written in $\exists^+$.

Theorem 6. *There is an effective algorithm for testing whether a given* GNFO *formula is equivalent to a UCQ and, if so, computing such a UCQ.*

4 Determinacy and Rewriting for Queries With Respect To Views

We now investigate properties pertaining to view-based query rewriting for GNFO.

Suppose V is a finite set of relation names, and we have FO formulas $\{\phi_v : v \in V\}$ over a signature S that is disjoint from V. we can consider each ϕ_v as defining a view v that is to be made accessible to a user, where given a finite structure I, this view is the set $\phi_v(I)$ of all tuples of elements satisfying ϕ_v in I. Suppose ϕ_Q is another first-order formula over the signature S. We say that the views ϕ_v's *determine* ϕ_Q if: for all finite structures I and I' with $\phi_v(I) = \phi_v(I')$ for all $v \in V$, we have $\phi_Q(I) = \phi_Q(I')$. Determinacy states that the query result can be recovered from the results of the views, via some function. Note that in this paper, when we talk about a set of views determining a query, we will *always* be working only over finite structures. Segoufin and Vianu initiated a study of determinacy for queries, including the question of when the assumption of determinacy implies that the recovery function is realized by a query. A *rewriting of* ϕ_Q *over* $\{\phi_v : v \in V\}$ is a formula ρ over the signature V (where the arity of a relation $v \in V$ is the number of arguments of ϕ_v), such that for every structure I for signature S, ρ applied to the view structure is the same as $\phi_Q(I)$. The view structure is the structure whose domain is the set of all elements occurring in $\phi_v(I)$ for some $v \in V$, and that interprets each $v \in V$ by $\phi_v(I)$. It is known that determinacy for unions of conjunctive queries is undecidable [NSV10], and that for UCQs determinacy does not imply rewritability even in first-order logic.

In contrast, we will show that whenever GNFO $\{\phi_v : v \in V\}$ determines GNFO ϕ_Q, then there is a rewriting, with the additional assumption that both $\{\phi_v : v \in V\}$ and ϕ_Q are *answer-guarded* – for FO formulas, we mean by this that they are of the form $\phi(\mathbf{x}) = R(\mathbf{x}) \wedge \phi'$ for some ϕ' and relation symbol R. That is, we show that determinacy implies rewritability for GNFO queries and views whose free variables are guarded. Note that rewritings, when they exist, can always be taken to be domain-independent queries, since the recovery function is (by definition) dependent only on the view extent.

Nash, Segoufin, and Vianu [NSV10] showed that these notions of determinacy and rewritability are closely related to interpolation and definability theorems in classical model theory. The Craig Interpolation theorem for first-order logic can be stated as follows: given formulas ϕ, ψ such that $\phi \models \psi$, there is a formula χ such that (i) $\phi \models \chi$, and $\chi \models \psi$ (ii) all relations occurring in χ occur in both ϕ and ψ (iii) all constants occurring in χ occur in both ϕ and ψ (iv) all free variables of χ are free variables of both ϕ and ψ.

The Craig Interpolation theorem has a number of important consequences, including the *Projective Beth definability theorem*. Suppose that we have a sentence ϕ over a first-order signature of the form $S \cup \{G\}$, where G is an n-ary predicate, and suppose S' is a subset of S. We say that ϕ *implicitly defines predicate G over S'* if: for every S'-structure I, every expansion to an $S \cup \{G\}$-structure I' satisfying ϕ has the same restriction to G up to isomorphism. Informally, the S' structure and the sentence ϕ determine a unique value for G. We say that an n-ary predicate G is *explicitly definable over S' for models of* ϕ if there is another formula $\rho(x_1 \dots x_n)$ using only predicates from S' such that $\phi \models \forall \mathbf{x}\, \rho(\mathbf{x}) \leftrightarrow G(\mathbf{x})$. It is easy to see that whenever G is explicitly definable over S' for models of ϕ, then ϕ implicitly defines G over S'. The Projective Beth Definability theorem states the converse: if ϕ implicitly defines G over S', then G is explicitly definable over S' for models of ϕ. In the special case where $S' = S$, this is called simply the Beth Definability theorem.

A proof of the Craig Interpolation theorem can be found in any model theory textbook (e.g. [CK90]). The proof is not effective, and it has been shown that it cannot be made effective [Fri76]. The Projective Beth Definability theorem follows from the Craig Interpolation theorem. Both theorems fail when restricted to finite structures.

We say that a fragment of first-order logic has the Craig Interpolation Property (CIP) if for all $\phi \models \psi$ in the fragment, the result above holds relative to the fragment. We similarly say that a fragment satisfies the Projective Beth Definability Property (PBDP) if the Projective Beth Definability theorem holds relativized to the fragment – that is, if ϕ in the hypothesis of the theorem lies in the fragment then there is a corresponding formula ρ lying in the fragment as well. We talk about the Beth Definability Property (BDP) for a fragment in the same way. The argument for first-order logic applies to any fragment with reasonable closure properties [Hoo00] to show that CIP implies PBDP.

As shown by Nash, Segoufin, and Vianu, the PBDP easily implies that whenever an FO query is determined by a set of FO views over all models, it is rewritable in FO. The fact that determinacy of FO queries does not imply FO rewritability over finite structures is related to the fact that CIP, PBDP, and BDP all fail for FO when implication is considered over finite structures [EF99]. Hence it is of particular interest to look at fragments of FO that have the finite model property, since there equivalence over finite structures can be replaced by equivalence over all structures. Hoogland, Marx, and Otto [HMO99] showed that the Guarded Fragment satisfies BDP but lacks CIP. Marx [Mar07] went on to explore determinacy and rewriting for the Guarded Fragment and its extensions. He argues that the PBDP holds for an extension of GFO called the Packed Fragment; using this, he concludes that determinacy implies rewritability for queries and views in the Packed Fragment. The definition of the Packed Fragment is not important for this work, but at the end of this section we show that PBDP fails for GFO, and also (contrary to [Mar07]) for the Packed Fragment. But we will adapt ideas of Marx to show that CIP and PBDP do hold for GNFO. Using this we will conclude that determinacy implies rewritability for answer-guarded GNFO views and queries.

Craig Interpolation and Beth Definability for GNFO. We now present the main technical result of this section. It is proven following a common approach in modal logic (see, in particular, Hoogland, Marx, and Otto [HMO99]), via a result saying that we can take two structures over different signatures, behaving similarly in the common signature, and *amalgamate* them to get a structure that is simultaneously similar to both of them. The amalgamation results in turn rely on the notion of strong GN-bisimulation, and use the proof of Theorem 2 to construct equivalent structures.

Theorem 7 (GNFO has Craig interpolation). *For each pair of* GNFO*-formulas* ϕ, ψ *such that* $\phi \models \psi$*, there is a* GNFO*-formula* χ *such that (i)* $\phi \models \chi$*, and* $\chi \models \psi$*, (ii) all relations occurring in* χ *occur in both* ϕ *and* ψ*, (iii) all constants occurring in* χ *occur in* ϕ *or* ψ *(or both), (iv) all free variables of* χ *are free variables of both* ϕ *and* ψ*.*

Projective Beth Definability for GNFO follows by standard arguments [Hoo00]:

Theorem 8. *If a* GNFO*-sentence* ϕ *in signature* σ *implicitly defines a relation symbol* G *in terms of a signature* $\tau \subset \sigma$*, and* τ *includes all constants from* σ*, then there is an explicit definition of* G *in terms of* τ *relative to* ϕ*.*

Observe that in Theorem 7, the interpolant is allowed to contain constant symbols outside of the common language. Indeed, this must be so, for GNFO lacks the stronger version of interpolation where the interpolant can only contain constant symbols occurring both in the antecedent and in the consequent. Recall that, in GNFO, as well as GFO, constant symbols are allowed to occur freely in formulas, and that their occurrence is not governed by guardedness conditions. In particular, for example, the formula $\forall y R(c,y)$ belongs to GFO (and to GNFO), while the formula $\forall y R(x,y)$ does not. Now, consider the valid entailment $(x = c) \wedge \forall y R(c,y) \models (x = d) \rightarrow \forall y R(d,y)$. It is not hard to show that any interpolant $\phi(x)$ not containing the constants c and d must be equivalent to $\forall y R(x,y)$. This shows that there are valid GFO-implications for which interpolants cannot be found in GNFO, if the interpolants are required to contain only constant symbols occurring both in the antecedent and the consequent. In fact, in [tC05] it was shown that, in a precise sense, every extension of GFO with this strong form of interpolation has full first-order expressive power and is undecidable for satisfiability.

Applications to rewriting. We can now state the consequence of the PBDP for determinacy-and-rewriting (relying again on the finite model property of GNFO). Note also that GNFO views V can check integrity constraints (e.g. inclusion dependencies) as well as return results. Using the above, we can get:

Theorem 9. *Suppose a set of answer-guarded* GNFO *views* $\{\phi_v : v \in V\}$ *determine an answer-guarded* GNFO ϕ_Q *on finite structures satisfying a set of* GNFO *sentences* Σ*. Then there is a* GNFO *rewriting of* ϕ_Q *using* $\{\phi_v : v \in V\}$ *that is valid over structures satisfying* Σ*. Furthermore, there is an algorithm that, given* ϕ_i*'s and* ϕ_Q *and* Σ *satisfying the hypothesis, effectively finds such a formula* ρ*.*

In particular, this holds when the view definitions ϕ_v *are answer-guarded UCQs,* ϕ_Q *is an answer-guarded* UCQ*, and* Σ *consists of inclusion dependencies and LAV constraints.*

Note also that "$\{\phi_v : v \in V\}$ determine ϕ_Q" (when the ϕ_v and ϕ_Q are answer-guarded and in GNFO) can be checked in 2ExpTime, since the property can again be expressed as a GNFO sentence, after which Theorem 1 can be applied.

Negative results for the guarded and packed fragments. We now prove that PBDP fails for the guarded fragment. This shows, intuitively, that if we want to express explicit definitions even for GFO implicitly-definable relations, we will need to use all of GNFO.

Theorem 10. *The PBDP fails for* GFO.

Proof. Consider the GF sentence ϕ that is the conjunction of the following:

$$\begin{aligned} \forall x\, C(x) &\to \exists yzu(G(x,y,z,u) \wedge E(x,y) \wedge E(y,z) \wedge E(z,u) \wedge E(u,x)) \\ \forall xy\, E(x,y) \wedge \neg C(x) &\to P_0(x) \wedge \neg P_1(x) \wedge \neg P_2(x) \\ \forall xy\, P_i(x) \wedge E(x,y) &\to P_{(i+1 \bmod 3)}(y) \quad \text{for all } 0 \leq i < 3 \end{aligned}$$

The first sentence forces that if $C(x)$ holds, then x lies on a directed E-cycle of length 4. The remaining two sentences force that if $\neg C(x)$ holds, then x only lies on directed E-cycles whose length is a multiple of 3. Clearly, the relation C is implicitly defined in terms of E. However, there is no explicit definition in GFO in terms of E, because no formula of GFO can distinguish the directed E-cycle of length k from the directed E-cycle of length ℓ for $3 \leq k < \ell$ [AvBN98]. □

It follows from Theorem 10 that GFO lacks CIP as well, which was already known [HMO99]. Furthermore, the above argument can be adapted to show that determinacy does not imply rewritability for views and queries defined in GFO: consider the set of views $\{\phi_{v_1}, \phi_{v_2}\}$, where $\phi_{v_1} = \phi$ and $\phi_{v_2}(x,y) = E(x,y)$. Clearly, $\{\phi_{v_1}, \phi_{v_2}\}$ determine the query $Q(x) = \phi \wedge C(x)$. On the other hand, any rewriting would constitute an explicit definition in GFO of C in terms of E, relative to ϕ, which we know does not exist.

In [Mar07, Lemma 4.4] it was asserted that PBDP holds for an extension of the Guarded Fragment, called the *Packed Fragment*, in which a guard $R(\mathbf{x})$ may be a conjunction of atomic formulas, as long as every pair of variables from $\mathbf{x}$ co-occurs in one of these conjuncts. The proof of Theorem 10, however, shows that PBDP fails for the Packed Fragment, because known results (cf. [Mar07]) imply that no formula of the Packed Fragment can distinguish the cycle of length k from the cycle of length ℓ for $4 \leq k < \ell$. Indeed, it turns out that there is a flaw in the proof of Lemma 4.4 in [Mar07].

5 Rewriting GNFO dependencies

Given a finite structure I, a set of integrity constraints Σ, and a query $Q(x_1 \dots x_k)$, the *certain answers to Q on I (under Σ)* are the set of tuples $c_1 \dots c_k \in I$ such that $\mathbf{c} \in Q(M)$ for every M containing I and satisfying Σ. Calculating the certain answers is a central problem in information integration and ontologies (in the former case one restricts to M finite, but for our constraints there will be no distinction). One of the key advantages of GNFO is that one can compute the certain answers for every Q and Σ in GNFO, and thus in particular for every Σ in GNFO and conjunctive query Q [BtCO12]. Baget et al. [BLM10] proved that for every set of frontier-guarded dependencies Σ and conjunctive query Q, the certain answers can be computed in polynomial time in I. However, it is known that there are guarded TGDs and conjunctive queries such that the certain answers can not be computed by a first-order query. We say that conjunctive query Q is *first-order rewritable* under constraints Σ if there is a first-order formula ϕ such that on any finite structure I $\phi(I)$ is exactly the certain answer to Q on I under Σ. Our next goal will be to show that we can decide, given a set Σ of frontier-guarded TGDs and a conjunctive query Q, whether or not Q is first-order rewritable. We will proceed by first capturing the certain answers in a fragment of Datalog. In proving this, we will follow (and correct) the approach of Baget et al. [BMRT11a], who argued that the certain answers of conjunctive queries under frontier-guarded TGDs are rewritable in Datalog. For guarded TGDs, this result had been announced by Marnette [Mar11].

The proof of Baget et al. [BMRT11b] revolves around a "bounded base lemma" showing that whenever a set of facts is not closed under "chasing" with FGTGDs, there is a small subset that is not closed (Lemma 4 of [BMRT11b]). However both the exact statement of that lemma and its proof are flawed. Our proof corrects the argument, making use of model-theoretic techniques (including Lemma 1) to prove the bounded base lemma. It then follows the rest of the argument in [BMRT11b] to show not only Datalog-rewritability, but rewritability into a Datalog program comprised of frontier-guarded rules. A conjunctive query is *answer-guarded* if it includes an atom that guards all free variables. In particular all Boolean conjunctive queries are answer-guarded.

Theorem 11. *For every set Σ of frontier-guarded TGDs, and for every answer-guarded conjunctive query $Q(\boldsymbol{x})$, one can effectively find a frontier-guarded Datalog program that computes the certain answers to Q.*

Note that entailment can be interpreted either in the classical sense or in the finite sense, since we have the finite model property. Indeed, in our proofs, we use constructions that make use of infinite structures, but the conclusion hold in the finite.

We state a special case of the Datalog-rewritability result for guarded TGDs – that is, TGDs in which the bodies are guarded. By a "guarded rule" in a Datalog program we mean a rule whose body contains an atom that guards *all* variables (not just the exported ones). A Guarded Datalog program is a Datalog program in which each rule is guarded. A simpler argument shows the following:

Theorem 12. *For every set Σ of guarded TGDs, and for every answer-guarded conjunctive query Q, one can effectively find a guarded Datalog program that computes on any structure I the certain answer to Q under Σ.*

In [BtCO12], a fragment of Datalog, denoted *GN-Datalog* was defined, and it was shown that for this fragment one can decide whether a query is equivalent to a first-order query (equivalently, as shown in [BtCO12], to some query obtained by unfolding the Datalog rules a finite number of times). Since GN-Datalog contains frontier-guarded Datalog, we can couple the decision procedure from [BtCO12] with the algorithm in Theorem 11 to obtain:

Corollary 1. *FO-rewritability of conjunctive queries under sets of frontier-guarded TGDs is decidable.*

6 Related Work and Conclusions

We have investigated rewriting of GNFO queries in several contexts, building on the decidability results for GNFO established in [BtCS11], and the complexity results for open- and closed-world querying established in [BtCO12].

Quite a number of issues related to decidability and complexity of rewritings are left open in this work – for example, the complexity of finding rewritings over views, and the effectiveness of finding rewritings within TGDs. Although we do not discuss the exact complexity of the decision problem for FO-rewritability under frontier-guarded TGDs in this paper, we believe that an elementary bound can be extracted from analysis of [BtCO12]. An open question concerning semantic characterizations is whether strong GN-bisimulations capture GNFO over finite structures.

Acknowledgements. The authors gratefully acknowledge their debt to Martin Otto for enlightening discussions on the subject. We have also benefited from helpful discussions with Maarten Marx.

Bibliography

[AHV95] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Add.-Wesley, 1995.

[AvBN98] H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *J. Phil. Logic*, 27:217–274, 1998.

[BCM$^+$03] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The description logic handbook*. Cambridge University Press, 2003.

[BGO10] V. Bárány, G. Gottlob, and M. Otto. Querying the guarded fragment. In *LICS*, 2010.

[BLM10] J.-F. Baget, M. Leclère, and M.-L. Mugnier. Walking the decidability line for rules with existential variables. In *KR*, 2010.

[BMRT11a] J.-F. Baget, M.-L. Mugnier, S. Rudolph, and M. Thomazo. Walking the complexity lines for generalized guarded existential rules. In *IJCAI*, 2011.

[BMRT11b] Jean-François Baget, Marie-Laure Mugnier, Sebastian Rudolph, and Michaël Thomazo. Complexity boundaries for generalized guarded existential rules, 2011. Research Report LIRMM 11006.

[BtCO12] V. Bárány, B. ten Cate, and M. Otto. Queries with guarded negation. In *VLDB*, 2012.

[BtCS11] V. Bárány, B. ten Cate, and L. Segoufin. Guarded negation. In *ICALP*, 2011.

[CGL09] A. Calì, G. Gottlob, and T. Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. In *PODS*, 2009.

[CK90] C. C. Chang and J. Keisler. *Model Theory*. North-Holland, 1990.

[EF99] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, 1999.

[Fag82] Ronald Fagin. Horn clauses and database dependencies. *J. ACM*, 29(4):952–985, 1982.

[FKMP05] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: Semantics and query answering. *TCS*, 336(1):89–124, 2005.

[Fri76] H. Friedman. The complexity of explicit definitions. *AIM*, 20(1):18 – 29, 1976.

[GLS03] G. Gottlob, N. Leone, and F. Scarcello. Robbers, marshals, and guards: game theoretic and logical characterizations of hypertree width. *J. Comput. Syst. Sci.*, 66(4):775–808, 2003.

[HMO99] E. Hoogland, M. Marx, and M. Otto. Beth definability for the guarded fragment. In *LPAR*, 1999.

[Hoo00] E. Hoogland. *Definability and interpolation: model-theoretic investigations*. PhD thesis, University of Amsterdam, 2000.

[Len02] M. Lenzerini. Data integration: A theoretical perspective. In *PODS*, 2002.

[Mar07] Maarten Marx. Queries determined by views: pack your views. In *PODS*, 2007.

[Mar11] B. Marnette. Resolution and datalog rewriting under value invention and equality constraints. Technical report, 2011. `http://arxiv.org/abs/1212.0254`.

[MV97] M. Marx and Y. Venema. *Multidimensional Modal Logic*. Kluwer, 1997.

[NSV10] A. Nash, L. Segoufin, and V. Vianu. Views and queries: Determinacy and rewriting. *ACM Trans. Database Syst.*, 35(3), 2010.

[Ott04] M. Otto. Modal and guarded characterisation theorems over finite transition systems. *APAL*, 130:173–205, 2004.

[Ros97] E. Rosen. Modal logic over finite structures. *JLLI*, 6(4):427–439, 1997.

[Ros08] B. Rossman. Homomorphism preservation theorems. *J. ACM*, 55(3), 2008.

[tC05] B. ten Cate. Interpolation for extended modal languages. *JSL*, 70(1):223–234, 2005.

[vB83] J. F. A. K. van Benthem. *Modal Logic and Classical Logic*. Humanities Pr, 1983.

[Yan81] M. Yannakakis. Algorithms for acyclic database schemes. In *VLDB*, 1981.

Proof of Theorem 2

Recall the statement:

A first-order formula $\phi(\mathbf{x})$ is preserved by $\rightarrow^{s}_{GN}$ (over all structures) iff it is equivalent to a GNFO formula.

Recall that the notation $(\mathfrak{A},\mathbf{a}) \Rrightarrow_{GN} (\mathfrak{B},\mathbf{b})$ is used to express that, for every GNFO formula $\phi(\mathbf{x})$, $\mathfrak{A} \models \phi(\mathbf{a})$ implies $\mathfrak{B} \models \phi(\mathbf{b})$. This notation may be indexed by a signature σ, in which case it is defined in terms of σ-reducts of the respective structures.

We first prove the following lemma (which will be used later on as well):

Lemma 4.

1. *If* $(\mathfrak{A},\boldsymbol{a}) \rightarrow^{s}_{GN[\sigma]} (\mathfrak{B},\boldsymbol{b})$ *then* $(\mathfrak{A},\boldsymbol{a}) \Rrightarrow_{GN[\sigma]} (\mathfrak{B},\boldsymbol{b})$.
2. *Conversely, if* $(\mathfrak{A},\boldsymbol{a}) \Rrightarrow_{GN[\sigma]} (\mathfrak{B},\boldsymbol{b})$, *then* $(\mathfrak{A},\boldsymbol{a})$ *and* $(\mathfrak{B},\boldsymbol{b})$ *have elementary extensions* $(\widehat{\mathfrak{A}},\boldsymbol{a})$ *and* $(\widehat{\mathfrak{B}},\boldsymbol{b})$, *respectively, such that* $(\widehat{\mathfrak{A}},\boldsymbol{a}) \rightarrow^{s}_{GN[\sigma]} (\widehat{\mathfrak{B}},\boldsymbol{b})$.

Proof. The first part can be proved by a straightforward formula induction. For the second part, we will use countable recursively saturated structures.

We may assume that $\mathfrak{A}$ and $\mathfrak{B}$ are countable. Consider the pair of structures $(\mathfrak{A},\mathfrak{B})$ viewed as a single structure over an extended signature with additional unary predicates P and Q to denote the domain of $\mathfrak{A}$ and of $\mathfrak{B}$, respectively. Let $(\widehat{\mathfrak{A}},\widehat{\mathfrak{B}})$ be any countable recursively saturated elementary extension of $(\mathfrak{A},\mathfrak{B})$. Let Z be the collection of all pairs of guarded tuples of $\widehat{\mathfrak{A}}$ and $\widehat{\mathfrak{B}}$ that are GNFO-indistinguishable. To establish the lemma, we need to show that Z is a strong GN-bisimulation, and that the partial map $\mathbf{a} \mapsto \mathbf{b}$ extends to a homomorphism that is compatible with Z. Both follow directly from the following claim.

Claim: every finite partial map f from $\widehat{\mathfrak{A}}$ to $\widehat{\mathfrak{B}}$ or vice versa that preserves truth of all GNFO-formulas, can be extended to a homomorphism f' compatible with Z.

Proof of claim: We assume that f is a partial map from $\widehat{\mathfrak{A}}$ to $\widehat{\mathfrak{B}}$ (the other direction is symmetric). Fix an enumeration $c_1, c_2, \ldots$ of the (countably many) elements of the domain of $\widehat{\mathfrak{A}}$ that are not in the domain of f. We will define a sequence of partial maps $f = f_0 \subseteq f_1 \subseteq f_2 \subseteq \cdots$ such that $dom(f_{i+1}) = dom(f_i) \cup \{c_{i+1}\}$, and such that each f_i preserves truth of all GNFO formulas. It then follows that $\bigcup_i f_i$ is a homomorphism extending f compatible with Z.

It remains only to show how to construct f_{i+1} from f_i. Here, we use the fact that $(\widehat{\mathfrak{A}},\widehat{\mathfrak{B}})$ is recursively saturated. Let $\mathbf{c}$ be an enumeration of the domain of f, and $\mathbf{d}$ an enumeration of the range of f, corresponding to the enumeration of $\mathbf{c}$, and let $\Sigma(x)$ be the set of all first-order formulas of the form

$$\phi(\mathbf{c}, c_{i+1}) \rightarrow \phi(\mathbf{d}, x)$$

where $\phi(\mathbf{c}, c_{i+1})$ is a GNFO formula with parameters $\mathbf{c}$ and c_{i+1}, and $\phi(\mathbf{d}, x)$ is obtained by replacing each parameter in $\mathbf{c}$ by its f-image, and replacing c_{i+1} by x. In the above definition of $\Sigma(x)$ we only consider formulas $\phi(\mathbf{c}, c_{i+1})$ that belong to GNFO even when the parameters $\mathbf{c}, c_{i+1}$ are treated as free variables (thereby excluding formulas such as $c_1 \neq c_2$).

The set $\Sigma(x) \cup \{Q(x)\}$ is clearly a recursive set. From the fact that f_i preserves truth of GNFO-formulas, it follows that every finite subset of $\Sigma(x) \cup \{P(x)\}$ is realized in $(\widehat{\mathfrak{A}},\widehat{\mathfrak{B}})$, and therefore the entire type is realized by some element d_{i+1}. It follows from the construction that the partial map $f_{i+1} = f \cup \{(c_{i+1}, d_{i+1})\}$ preserves truth of all GNFO formulas. $\dashv$

This concludes the proof of the lemma.

Proof (of Theorem 2). We prove the hard direction, following the template often used in preservation theorems in classical model theory. Let $\phi(\mathbf{x})$ be preserved by $\rightarrow^{s}_{GN}$, and let $\Psi(\mathbf{x})$ be the set of all GNFO formulas it entails. It is enough to show that $\Psi(\mathbf{x}) \models \phi(\mathbf{x})$.

Let $\mathfrak{B} \models \Psi(\mathbf{b})$, and let $\Gamma_{\mathfrak{B},\mathfrak{b}}(\mathbf{x})$ be the set of all negations of GNFO formulas false of $\mathbf{b}$ in $\mathfrak{B}$. We claim that $\Gamma_{\mathfrak{B},\mathbf{b}}(\mathbf{x}) \cup \{\phi(\mathbf{x})\}$ is consistent. Suppose it were not consistent. then, by the Compactness Theorem, we would have that $\phi(\mathbf{x})$ implies $\gamma(\mathbf{x})$, where $\gamma(\mathbf{x})$ is the negation of some finite conjunction of formulas from $\Gamma_{\mathfrak{B},\mathbf{b}}(\mathbf{x})$. It follows from the construction of $\Gamma_{\mathfrak{B},\mathbf{b}}(\mathbf{x})$ that $\gamma(\mathbf{x})$ is (up to logical equivalence) a GNFO formula, which therefore must belong to $\Psi(\mathbf{x})$. This yields a contradiction because we have that $\mathfrak{B} \models \Psi(\mathbf{b})$ and $\mathfrak{B} \not\models \gamma(\mathbf{b})$.

Thus there is $\mathfrak{A}$ and $\mathbf{a}$ such that $\mathfrak{A} \models \Gamma_{\mathfrak{B},\mathbf{b}}(\mathbf{a}) \wedge \phi(\mathbf{a})$. By construction, every GNFO formula true of $\mathbf{a}$ in $\mathfrak{A}$ is also true of $\mathbf{b}$ in $\mathfrak{B}$. Therefore, by Lemma 4, we can lift $\mathfrak{A},\mathbf{a}$ and $\mathfrak{B},\mathbf{b}$ to elementary extensions that are strongly GN-bisimilar. By the fact that ϕ is invariant for strong GN-bisimulations, we get that the extension of $\mathfrak{B},\mathbf{b}$ satisfies ϕ. Now by the fact that ϕ is invariant for elementary extensions, we conclude $\mathfrak{B} \models \phi(\mathbf{b})$.

Proof of Lemma 1

Recall the statement:

Let Σ be a set of TGDs that is strong GN-bisimulation invariant and let ρ be a TGD such that $\Sigma \models \rho$. Then there exists a specialisation ρ' of ρ such that $\Sigma \models \rho'$, and such that ρ' is logically equivalent to a conjunction of frontier-guarded TGDs. This holds both over finite structures and over arbitrary structures.

For every pair of structures I,J with $I \subseteq J$, there is a squid-extension J' of I and a homomorphism $h : J' \to J$ whose restriction to I is the identity function, such that $J' \sim^{s}_{GN} J$ via a strong GN-bisimulation that is compatible with h. Moreover, we can choose J' to be finite if J is.

Proof. For every set X of elements that is guarded in I, we create a structure J_X that is a fresh isomorphic copy of J in which only the elements of $X \cup C$ are kept constant (i.e., mapped to themselves by the isomorphism), where C is the set of all element named by a constant symbol. We define J' to be the union of all such J_X. Clearly, J' is a squid-extension of I, and the natural projection $h : J' \to J$ is a homomorphism. Furthermore, we claim that $J' \sim^{s}_{GN} J$ via a strong GN-bisimulation that is compatible with h. The strong GN-bisimulation in question consists of all pairs $(\mathbf{a}, h(\mathbf{a}))$ where $\mathbf{a}$ is a guarded tuple of J'.

Proof of Lemma 2

Recall the statement:

Let Σ be any set of TGDs and suppose that

$$\Sigma \models \forall \mathbf{x}(\phi(\mathbf{x}) \to \bigvee_{i=1\ldots n} \exists \mathbf{y}_i \psi_i(\mathbf{x}, \mathbf{y}_i))$$

where ϕ, ψ_i are conjunctions of atoms. Then $\Sigma \models \forall \mathbf{x}(\phi(\mathbf{x}) \to \exists \mathbf{y}_i \psi_i(\mathbf{x}, \mathbf{y}_i))$ for some $i \leq n$. This holds both over finite structures and over arbitrary structures.

To simplify the presentation, we consider the case where $n = 2$. Let

$$\Sigma \models \forall \mathbf{x}(\phi(\mathbf{x}) \to \exists \mathbf{y}_1 \psi_1(\mathbf{x}, \mathbf{y}_1) \vee \exists \mathbf{y}_2 \psi_2(\mathbf{x}, \mathbf{y}_2))$$

and suppose for the sake of a contradiction that there are structures $I_1 \models \Sigma$ and $I_2 \models \Sigma$ such that $I_i \models \phi(\mathbf{a}_i) \wedge \neg \exists \mathbf{y}_i \psi_i(\mathbf{a}_i, \mathbf{y}_i)$. Let J be the direct product $I_1 \times I_2$, that is, the structure whose domain is the cartesian product of the domains of I_1 and I_2 and such that a tuple of pairs belong to a relation in J if and only if the tuple of first-projections belongs to the corresponding relation in I_1 and the tuple of second-projections belongs to the corresponding relation in I_2. If a constant symbol denotes a in I_1 and b in I_2, it denotes the pair (a, b) in J. Since TGDs are closed under taking direct products, we have that $J \models \Sigma$. It also follows from the construction that (i) the natural projections $h_1 : J \to I_1$ and $h_2 : J \to I_2$ are homomorphism, and (ii) whenever $\phi(\mathbf{x})$ is satisfied by tuples $\mathbf{a}_1$ in I_1 and $\mathbf{a}_2$ in I_2, then the tuple of pairs $\mathbf{a}$ whose first-projections are $\mathbf{a}_1$ and whose second projections are $\mathbf{a}_2$ also satisfies $\phi(\mathbf{x})$ in J. Putting this together, we obtain that $J \models \phi(\mathbf{a}) \wedge \bigwedge_i \neg \exists \mathbf{y}_i \psi_i(\mathbf{a}, \mathbf{y}_i)$, which contradicts the fact that $J \models \Sigma$.

The last sentence holds because J is finite if I_1 and I_2 are.

Proof of Lemma 3

Recall the statement:

Let Σ be a set of TGDs that is strong GN-bisimulation invariant and let ρ be a TGD such that $\Sigma \models \rho$. Then there exists a specialisation ρ' of ρ such that $\Sigma \models \rho'$, and such that ρ' is logically equivalent to a conjunction of frontier-guarded TGDs. This holds both over finite structures and over arbitrary structures.

Proof. Before we start, we introduce the notion of a *quasi-frontier guarded TGD*. By the *graph of a TGD* ρ we mean the undirected graph whose nodes are the conjuncts of γ and where two conjuncts are connected by an edge if they share an existentially quantified variable. Observe that when the graph of ρ is not connected, then ρ can be decomposed into several rules, one for each connected component. We say that ρ is *quasi-frontier guarded* if, for each connected component of its graph, the set of universally quantified variables occurring in atoms belonging to that component is guarded by some atom in the rule body β. This is equivalent to saying that the aforementioned decomposition yields a set of frontier-guarded TGDs. We will show that, if Σ be a set of TGDs that is strong GN-bisimulation invariant and let ρ be a TGD such that $\Sigma \models \rho$, then there exists a specialisation ρ' of ρ such that $\Sigma \models \rho'$, and such that ρ' is quasi-frontier guarded.

Let $\rho = \forall \mathbf{x} \beta(\mathbf{x}) \to \exists \mathbf{z} \gamma(\mathbf{x}, \mathbf{z})$, and let $\rho_1, \ldots, \rho_n$ be all quasi-frontier guarded specialisations of ρ. Each ρ_i is of the form $\forall \mathbf{x} \beta(\mathbf{x}) \to \exists \mathbf{z}_i \gamma_i(\mathbf{x}, \mathbf{z}_i)$ Let ρ' be the "disjunctive TGD"

$$\rho' \ = \ \forall \mathbf{x} \beta(\mathbf{x}) \to \bigvee_i \exists \mathbf{z}_i \gamma_i(\mathbf{x}, \mathbf{z}_i) \ .$$

By Lemma 2 it is enough to show that Σ entails ρ'.

Consider any structure $J \models \Sigma$ and homomorphism h from $\beta(\mathbf{x})$ to J. Let B be the image of $\beta(\mathbf{x})$ under the homomorphism h. By Lemma 1, B has a squid extension J' such that $J' \sim^s_{GN} J$ via some strong GN-bisimulation that is compatible with a homomorphism

$g : J' \to J$ whose restriction to B is the identity function. Since Σ is invariant for strong GN-bisimulations, $J' \models \Sigma$, and therefore also $J' \models \rho$. In particular, h can be extended to a homomorphism h' from $\exists \mathbf{z} \gamma(\mathbf{x}, \mathbf{z})$ to J'. We can extract from h' a substitution θ, namely the one that sends a variable z_i to a constant symbol c if $h'(z_i)$ is the interpretation of c (if $h'(z_i)$ is the interpretation of several constant symbols we choose one arbitrarily), or else θ sends z_i to an arbitrary x_j for which $h'(z_i) = h(x_j)$ if there is such x_j, or otherwise θ sends z_i to z_i. Applying θ to the conjunctive query $\exists \mathbf{z} \gamma(\mathbf{x}, \mathbf{z})$ yields another conjunctive query $\exists \mathbf{z}' \gamma'(\mathbf{x}, \mathbf{z}')$ (where $\mathbf{z}'$ is a subset of $\mathbf{z}$). By construction we have that

$$\rho' = \forall \mathbf{x} \beta(\mathbf{x}) \to \exists \mathbf{z}' \gamma'(\mathbf{x}, \mathbf{z}')$$

is a specialization of ρ that is satisfied in J', and hence also in J, under the assignment h for the universally quantified variables. It only remains to show that ρ' is quasi-frontier-guarded. First note that, by construction, all existential variables in $\mathbf{x}'$ are mapped by h' to elements that do not belong to B. Now, recall the definition of quasi-frontier-guardedness and consider any connected component in the graph of ρ'. By connectedness and by the above observation about the existential variables in $\mathbf{x}$, we have that the image of ρ' under h' must be entirely contained in a single "tentacle" of J'. In particular, this means that the universally quantified variables occurring in the conjunct belonging to this component must be mapped by h to a guarded set of elements of B. Since B was defined as the h-image of β, this means that the universally quantified variables in question are guarded in β.

Proof of Theorem 4

Recall the statement:

Every GFO-sentence that is equivalent to a finite set of TGDs over finite structures is equivalent to a finite set of TGDs on arbitrary structures, and such a formula is equivalent (over all structures) to a finite set of acyclic FGTGDs.

We will first need to recall an important prior result. A CQ is *answer-guarded* if the free variables co-occur in some atom. The following key lemma from [BGO10] shows that, in the context of the Guarded Fragment, answer-guarded conjunctive queries can be reduced to acyclic answer-guarded conjunctive queries.

Lemma 5 (Treeification Lemma [BGO10]). *Fix a finite schema. For every answer-guarded conjunctive query q, there is a finite set of acyclic answer-guarded conjunctive queries $T(q)$, called the* treeification of q, *such that the following holds:*

> *for every structure M, there is a structure M^* that satisfies the same* GFO*-sentences as M and such that, for all answer-guarded conjunctive queries q, q and $\bigvee T(q)$ yield the same answers on M^*.*

In particular, for all GFO*-sentences ϕ and Boolean conjunctive queries q, $\phi \models q$ if and only if $\phi \models \bigcup T(q)$.*

We now begin the proof of Theorem 4:

Proof. Let ϕ be any GFO-sentence that is equivalent to a finite set of TGDs over finite structures. Then, by Theorem 3, ϕ is equivalent to a finite set Σ of FGTGDs over arbitrary structures. By Lemma 2, it suffices to show that Σ is equivalent to be the set Σ' of disjunctive GTGDs obtained by replacing the left-hand side and the right-hand side of each rule by its treeification. This follows from Lemma 5: for any structure M, we have that $M \models \Sigma$ if and only if $M^* \models \Sigma$ if and only if $M^* \models \Sigma'$ if and only $M \models \Sigma'$.

Proof of Theorem 5

Recall the statement:

Every GNFO formula which is preserved under extensions over finite structures has the same property over all structures, and such a formula is equivalent (over all structures) to an existential formula in GNFO. Furthermore, we can decide whether a formula has this property, and also find the existential GNFO formula effectively.

Proof. Let ϕ be a GNFO formula containing constants $\mathbf{c}$ and with free variables $\mathbf{x}$. Let $\mathbf{d}$ be fresh constants, one for each variable in $\mathbf{x}$. Then ϕ is preserved under extensions over finite structures iff the GNFO sentence $\bigwedge_{c\in\mathbf{c}\cup\mathbf{d}} P(c) \wedge \phi^P(\mathbf{d}) \rightarrow \phi(\mathbf{d})$ is a validity over finite structures, where ϕ^P is the relativization of ϕ to a new unary predicate P. Since ϕ is a GNFO formula, it is a validity over finite structures iff it is a validity over all structures. Also, the decidability of GNFO allows us to decide this validity. By the classical Łoś-Tarski theorem, if a formula is preserved over all structures, it is equivalent to an existential formula ϕ'. We can convert ϕ' into the form $\bigvee_i \phi'_i$, where $\phi'_i = \exists\mathbf{z} \bigwedge_j \psi'_{ij}$ and each ψ'_{ij} is a possibly negated atomic formula or equality. In general, some of the negated atomic formulas and equalities may not be guarded. Let ϕ'' be obtained from ϕ' by removing all conjuncts that are unguarded negative atomic formulas or equalities. We claim that ϕ' and ϕ'' are equivalent. In one direction, ϕ' clearly implies ϕ''.

For the converse, consider an arbitrary structure M and tuple $\mathbf{a}$ such that $M \models \phi''(\mathbf{a})$. It is our task to show that $M \models \phi(\mathbf{a})$. Our general approach will be to construct another structure M' and tuple $\mathbf{b}$ such that $M' \models \phi'(\mathbf{b})$. In addition, we will show that $(M', \mathbf{b}) \rightarrow^s_{GN} (M, \mathbf{a})$). This, then, allows us to conclude that $M \models \phi(\mathbf{a})$, since ϕ and ϕ' are logically equivalent and ϕ belongs to GNFO.

Let h be a witnessing satisfying variable assignment from some $\phi''_i = \exists\mathbf{y} \bigwedge_j \psi''_{ij}(\mathbf{x}, \mathbf{y})$ to M, where ψ''_{ij} is the modification of ψ'_{ij} above. We want to show that the tuple $h(\mathbf{x})$ satisfies ϕ'_i as well. The main obstacles that we have to overcome are

(i) the possibility that h maps two variables u, v to the same element of M whereas ϕ'_i includes the (unguarded) inequality $u \neq v$.
(ii) the possibility that M contains a fact that is the h-image of an atomic formula occurring under an (unguarded) negation in ϕ'_i.

Based on these considerations, our construction of M' and $\mathbf{b}$ will, intuitively, involve (i) making sure that only those equalities are satisfied that are either explicitly contained in ϕ'_i or that follow (by transitivity) from guarded equalities true in M at $\mathbf{a}$ and (ii) making sure that every fact of M' whose values are in the range of h is guarded by a fact that is an h-image of a positive atom of ϕ'_i.

The precise construction is as follows. First, we may assume without loss of generality that ϕ' is satisfiable, and that ϕ' includes an equality or inequality between each pair of variables u, v for which it is the case that u and v co-occur in a positive relational atom in ϕ'. Let X be the set of all variables occurring, free or bound, in ϕ'_i. Furthermore let $\equiv$ be the equivalence relation on X generated by all pairs of variables (u, v) such that ϕ'' contains the equality $u = v$. Let $f : X \rightarrow X/_{\equiv}$ be the natural map that sends each variable to its equivalence class. As a first step, we construct a structure M^* whose domain is $X/_{\equiv}$ and whose facts are the f-images of the positive atoms of ϕ'_i (or, equivalently, of ϕ''). Let $\mathbf{b} = f(\mathbf{x})$.

- Observation 1: the function h factors though f, i.e., $h = f \cdot g$ for some function $g : X/_\equiv \to dom(M)$. Moreover, $g : M^* \to M$ is a homomorphism and g is injective on guarded subsets (i.e., it maps distinct elements co-occurring in a fact of M^* to distinct elements of M).
- Observation 2: f is a satisfying variable assignment for ϕ' in M^*, showing that $M^* \models \phi''(\mathbf{b})$.

Observation 1 follows from the fact whenever $u \equiv v$, then $h(u) = h(v)$ (which is immediate from the definition of the equivalence relation $\equiv$ and the fact that h is a satisfying assignment). Observation 2 follows from the construction of M^* (for the equalities, inequalities, and positive atoms) and from the previous observation (for the negative atoms).

As a next step, we transform M^* into M' as follows: for each fact F of M^* we make an isomorphic copy of M denoted M'_F, where the isomorphism in question maps the elements belonging to the g-image of F to their g-preimage and maps all other elements to distinct fresh elements. Note that we are using here the fact that g is injective on guarded subsets, making the above a well-defined construction. We define M' as the union $M^* \cup \bigcup\{M'_F \mid F \text{ a fact of } M^*\}$, and we let $\widehat{g} : M' \to M$ be the map that extends g by mapping every newly-created element in some M'_F to the corresponding element of M. Note that, by construction, $\widehat{g} : M^* \to M$ is a homomorphism.

- Observation 3: $M^* \models \phi'_i(\mathbf{b})$ via the variable assignment f.
- Observation 4: $(M', \mathbf{b}) \to^s_{GN} (M, \mathbf{a})$. In particular, we get that $M \models \phi(\mathbf{a})$.

Observation 3 follows from the fact that $M' \subseteq M^*$, together with the fact (used for negative atoms in ϕ') that $\widehat{f}$ is a homomorphism and that $h = \widehat{f} \cdot g$ is a satisfying variable assignment for ϕ'' in M.

For Observation 4, the graph of $\widehat{g}$ is in fact a strong GN-bisimulation which is compatible with the homomorphism g. Note that that $g(\mathbf{a}) = \mathbf{b}$.

Finally, we observe that once we know that an equivalent existential formula in GNFO exists, we can find it by exhaustive search, using the fact that equivalence of GNFO formulas is decidable.

Proof of Theorem 6

Recall the statement:

There is an effective algorithm for testing whether a given GNFO formula is equivalent to a UCQ and, if so, computing such a UCQ.

Proof. Rossman's proof [Ros08] shows that if an arbitrary FO formula ϕ is equivalent to an $\exists^+$ formula, it is equivalent to one of the same quantifier rank as ϕ. If ϕ is in GNFO, we can test equivalence of a given $\exists^+$ formula ϕ' with ϕ, using the decidability of GNFO. We can thus test all $\exists^+$ formulas with quantifier rank bounded by the quantifier rank of ϕ, giving an effective procedure.

Proof of Theorem 7

Recall the statement:

For each pair of GNFO-formulas ϕ, ψ such that $\phi \models \psi$, there is a GNFO-formula χ such that (i) $\phi \models \chi$, and $\chi \models \psi$, (ii) all relations occurring in χ occur in both ϕ and ψ, (iii) all constants occurring in χ occur in ϕ or ψ (or both), (iv) all free variables of χ are free variables of both ϕ and ψ.

We follow a common approach in modal logic (see, in particular, Hoogland, Marx, and Otto [HMO99]), making use of a result saying that we can take two structures over different signatures, behaving similarly in the common signature, and *amalgamate* them to get a structure that is simultaneously similar to both of them. The precise statement of the theorem will be in terms of the notion of strong GN-bisimulation introduced in Section 3, and the proof will make use of the results there. Our specific amalgamation construction is inspired by the *zig-zag products* introduced by Marx and Venema in [MV97].

Lemma 6 (Amalgamation).
Let σ and τ be signatures containing the same constant symbols but possibly different relation symbols. If $(\mathfrak{A}, \boldsymbol{a}) \rightarrow^{s}_{GN[\sigma\cap\tau]} (\mathfrak{B}, \boldsymbol{b})$, then there is a structure $(\mathfrak{U}, \boldsymbol{u})$ such that $(\mathfrak{A}, \boldsymbol{a}) \rightarrow^{s}_{GN[\sigma]} (\mathfrak{U}, \boldsymbol{u}) \rightarrow^{s}_{GN[\tau]} (\mathfrak{B}, \boldsymbol{b})$

Proof. Let Z be the strong GN-bisimulation between $\mathfrak{A}$ and $\mathfrak{B}$ witnessing the fact that $(\mathfrak{A}, \mathbf{a}) \rightarrow^{s}_{GN[\sigma\cap\tau]} (\mathfrak{B}, \mathbf{b})$. Below, for any partial map f from $\mathfrak{A}$ to $\mathfrak{B}$ or vice versa, with a slight abuse of notation, we will write $f \in Z$ if f can be extended to a homomorphism that is compatible with Z. In particular, we have $(\mathbf{a} \mapsto \mathbf{b}) \in Z$. Note that, for individual elements c and d, $(c \mapsto d) \in Z$ if and only if $(d \mapsto c) \in Z$. In addition, with some further abuse of notation, for any k-tuple $\mathbf{c} = c_1 \ldots c_k$ of elements of $\mathfrak{A}$ and for any k-tuple $\mathbf{d} = d_1 \ldots d_k$ of elements of $\mathfrak{B}$, we will denote by $\langle \mathbf{c}, \mathbf{d} \rangle$ the k-tuple $((c_1, d_1), \ldots, (c_k, d_k))$.

We define the amalgam $(\mathfrak{U}, \mathbf{u})$ as follows.

$$\begin{aligned} &\text{The domain of } \mathfrak{U} \text{ is } \{(c,d) \in \mathfrak{A} \times \mathfrak{B} \mid (c \mapsto d) \in Z\}\ , \\ &R^{\mathfrak{U}} = \{\langle \mathbf{c}, \mathbf{d} \rangle \mid \mathbf{c} \in R^{\mathfrak{A}} \text{ and } (\mathbf{c} \mapsto \mathbf{d}) \in Z\} \text{ for all } R \in \sigma \\ &S^{\mathfrak{U}} = \{\langle \mathbf{c}, \mathbf{d} \rangle \mid \mathbf{d} \in S^{\mathfrak{B}} \text{ and } (\mathbf{d} \mapsto \mathbf{c}) \in Z\} \text{ for all } S \in \tau \\ &c^{\mathfrak{U}} = (c^{\mathfrak{A}}, c^{\mathfrak{B}}) \text{ for all constant symbols } c \\ &\mathbf{u} = \langle \mathbf{a}, \mathbf{b} \rangle \end{aligned}$$

To see that this is a proper definition, note that for $R \in \sigma \cap \tau$, if $\mathbf{c} \in R^{\mathfrak{A}}$ and $(\mathbf{c} \mapsto \mathbf{d}) \in Z$ then also $\mathbf{d} \in R^{\mathfrak{B}}$ and $(\mathbf{d} \mapsto \mathbf{c}) \in Z$, and vice versa.

Claim 1: $(\mathfrak{A}, \mathbf{a}) \rightarrow^{s}_{GN[\sigma]} (\mathfrak{U}, \mathbf{u})$
Proof of claim: Let Z' be the collection of all pairs $(\mathbf{v}, \langle \mathbf{v}, \mathbf{w} \rangle)$ for $(\mathbf{v} \mapsto \mathbf{w}) \in Z$ and $\mathbf{v}$ guarded (by a σ-atom) in M. We will show that Z' is a strong GN-bisimulation between $\mathfrak{A}$ and $\mathfrak{U}$, and that $(\mathbf{a} \mapsto \mathbf{u}) \in Z'$.

Consider any pair $(\mathbf{v}, \langle \mathbf{v}, \mathbf{w} \rangle) \in Z'$. By construction, we have that $(\mathbf{v}, \mathbf{w}) \in Z$ and hence, there is a homomorphism $h : \mathfrak{A} \to \mathfrak{B}$ that is compatible with Z, and such that $h(\mathbf{v}) = \mathbf{w}$. Let $\widehat{h}(a) = (a, h(a))$ for all $a \in \mathfrak{A}$. It can easily be verified that $\widehat{h}$ is a homomorphism from $\mathfrak{A}$ to $\mathfrak{U}$ that is compatible with Z', and that $\widehat{h}(\mathbf{v}) = \langle \mathbf{v}, \mathbf{w} \rangle$. Conversely, we also need to show that there is a homomorphism from $\mathfrak{U}$ to $\mathfrak{A}$ that is compatible with Z' and that maps $\langle \mathbf{v}, \mathbf{w} \rangle$ to $\mathbf{v}$. Here, we can simply choose the natural projection as our homomorphism. It is easy to verify that this satisfies the required conditions.

Finally, we need to show that $(\mathbf{a} \mapsto \mathbf{u}) \in Z'$, i.e., that there is a homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ that is compatible with Z' and that sends $\mathbf{a}$ to $\mathbf{u}$. Recall that $\mathbf{u} = \langle \mathbf{a}, \mathbf{b} \rangle$. Let h be a homomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ that is compatible with Z and that sends $\mathbf{a}$ to $\mathbf{b}$, and let $\widehat{h}$ be defined by $\widehat{h}(a) = (a, h(a))$ for all $a \in \mathfrak{A}$. It is easy to verify that $\widehat{h}$ satisfies the required conditions.

Claim 2: $(\mathfrak{U}, \mathbf{u}) \rightarrow^{s}_{GN[\tau]} (\mathfrak{B}, \mathbf{b})$

Proof of claim: the relevant strong GN-bisimulation Z'' is constructed analogously. Note that, in this case, we do not get that $(\mathbf{b} \mapsto \mathbf{u}) \in Z''$ but we get that $(\mathbf{u} \mapsto \mathbf{b}) \in Z''$ because this partial map is included in the natural projection from $\mathfrak{U}$ to $\mathfrak{B}$, which is compatible with Z''. □

We are now ready to prove Theorem 7.

Proof. Let $\phi(\mathbf{x})$ and $\psi(\mathbf{x})$ be GNFO-formulas over signatures σ and τ, such that $\models \forall \mathbf{x}(\phi(\mathbf{x}) \rightarrow \psi(\mathbf{x}))$. We also assume that all ϕ and ψ have the same free variables (because, we can existentially quantify out the free variables of ϕ that are not free variables of ψ, and we can universally quantify out the free variables of ψ that are not free variables of ϕ, introducing a fresh guard relation if necessary). We may further assume that the signatures contain the same constant symbols.

Suppose for the sake of contradiction that there is no $\sigma \cap \tau$-interpolant.

As a first step, using a standard Compactness argument, we establish the existence of two structures $(\mathfrak{A}, \mathbf{a})$ and $(\mathfrak{B}, \mathbf{b})$ such that $\mathfrak{A} \models \phi(\mathbf{a})$, $\mathfrak{B} \models \neg\psi(\mathbf{b})$, and $(\mathfrak{A}, \mathbf{a}) \Rrightarrow_{GN[\sigma\cap\tau]} (\mathfrak{B}, \mathbf{b})$. The precise reasoning is as follows: let $\Phi(\mathbf{x})$ be the set of all valid GNFO$[\sigma \cap \tau]$ consequences of $\phi(\mathbf{x})$. Due to the Compactness theorem, we know that $\Phi(\mathbf{x})$ cannot imply $\psi(\mathbf{x})$. Therefore, there is a structure $\mathfrak{B} \models \Phi(\mathbf{b}) \wedge \psi(\mathbf{b})$. Next, let $\Psi(\mathbf{x})$ be the set of all negations of GNFO$[\sigma \cap \tau]$-formulas that are falsified by $\mathbf{b}$ in $\mathfrak{B}$. Then again, due to Compactness, $\Psi(\mathbf{x})$ cannot imply $\neg\phi(\mathbf{x})$ (for, otherwise, a negation of a finite conjunction of formulas from $\Psi(\mathbf{x})$ would have belonged to $\Phi(\mathbf{x})$, contradicting the fact that $\mathfrak{B} \models \Phi(\mathbf{b})$). Therefore, there is a structure $\mathfrak{A} \models \Psi(\mathbf{a}) \wedge \phi(\mathbf{a})$. By construction, we have that $(\mathfrak{A}, \mathbf{a}) \Rrightarrow_{GN[\sigma\cap\tau]} (\mathfrak{B}, \mathbf{b})$.

Next, using Lemma 4, we lift the $\Rrightarrow_{GN[\sigma\cap\tau]}$ relationship between $(\mathfrak{A}, \mathbf{a})$ and $(\mathfrak{B}, \mathbf{b})$ to a $\rightarrow^{s}_{GN[\sigma\cap\tau]}$ relationship between elementary extensions $(\widehat{\mathfrak{A}}, \mathbf{a})$ and $(\widehat{\mathfrak{B}}, \mathbf{b})$. Finally, using Lemma 6, we obtain a structure $\mathfrak{U} \models \phi(\mathbf{u}) \wedge \neg\psi(\mathbf{u})$, contradicting our assumption that $\phi(\mathbf{x})$ implies $\psi(\mathbf{x})$.

Proof of Corollary ??

Recall the statement:

Suppose a set of answer-guarded GNFO queries $\{\phi_v : v \in V\}$ determines an answer-guarded GNFO query ϕ_Q. Then there is a GNFO query ρ that is a rewriting. Furthermore, there is an algorithm that, given ϕ_i's and ϕ_Q satisfying the hypothesis, effectively finds such a formula ρ.

Proof. Extend the vocabulary with predicates v for each ϕ_v and a predicate Q for ϕ_Q. Now consider a sentence stating that each v contains exactly the tuples satisfying ϕ_v and that Q contains exactly the tuples satisfying ϕ_Q. The hypotheses imply that this sentence is in GNFO, and that it implicitly defines Q with respect to the signature containing only the symbols in V Applying the PBDP for GNFO, we get an explicit definition of Q in GNFO. By unwinding the definitions we see that this is a rewriting.

The rewriting can be found effectively by simply enumerating every possible ρ and checking whether ϕ_Q is logically equivalent to $\rho(V_1/\phi_1 \dots V_n/\phi_n))$; the check is effective using the decidability of equivalence for GNFO [BtCS11].

More details on Theorem 10

Theorem 10 follows from the argument given in the body, once one understands that these logics cannot distinguish between cycles of certain lengths. We will now explain this, making use of the notions of guarded bisimulation [BGO10], which characterizes equivalence in the guarded fragment, and packed bisimulation [Mar07], which does the same for the packed fragment.

Fix a binary relation symbol E, let C_k be the directed E-cycle of length k. Let $3 \leq k, \ell$, and let Z be the binary relation containing all pairs $((a,b),(c,d))$ such that $(a,b) \in E^{C_k}$ and $(c,d) \in E^{C_\ell}$. It can be shown that Z is a guarded-bisimulation between C_k and C_ℓ.

Moreover, if $4 \leq k, \ell$, then in fact Z is a packed-bisimulation between C_k and C_ℓ. This shows that no sentence of the packed fragment can distinguish directed E-cycles of different length. Incidentally, the packed fragment sentence $\exists xyz(Rxy \wedge Ryz \wedge Rzx \wedge \top)$ distinguishes C_3 from C_4.

Proof of Theorem 11

Recall the statement:

For every set Σ of frontier-guarded TGDs, and for every answer-guarded conjunctive query $Q(\mathbf{x})$, one can effectively find a frontier-guarded Datalog program that computes the certain answers to Q.

Proof. We can assume without loss of generality that Q is an atomic query (by extending Σ with an extra "answer rule" containing the query. This rule is frontier-guarded because Q is answer-guarded). We may also assume that for the body of each rule the graph connecting variables whenever they appear together in an atom is connected. This can be ensured by introducing new zero-ary predicates when needed.

Let k be the maximal number of symbols in a TGD. For each answer-guarded conjunctive query $q(x_1 \dots x_j)$ with at most k variables, let $R_q(x_1 \dots x_j)$ be a new relation symbol. Let FGTGD$_k$ be all the frontier-guarded TGDs in this extended signature such that both the bodies and the heads have size at most k. Let Σ' be all TGDs in FGTGD$_k$ that are consequences of

$$\Sigma \cup \{\forall \mathbf{x} R_q \leftrightarrow q \mid q \text{ answer-guarded CQ with} \leq k \text{ variables}\}.$$

For a structure I, let C_I be the set of elements of I named by constant symbols. We say that a structure I is *saturated* (with respect to Σ') if every possible fact over $adom(I) \cup C_I$ entailed by the facts of I together with Σ' belongs to I. We say that I is *guardedly-saturated* (with respect to Σ') if every possible fact over $adom(I) \cup C_I$ entailed by the facts of I together with Σ', *such that the values occurring in the fact form a guarded set in I*, belongs to I. In the absence of constants, essentially, saturation means that every entailed fact over $adom(I)$ belongs to I, while guardedly-saturated means that no fact over $adom(I)$ guarded by an existing fact of I is entailed.Note that entailment can be interpreted either in the classical sense or in the finite sense, since we have the finite model property. Indeed, in what follows, we will use constructions that make use of infinite structures, but the conclusion will hold in the finite.

Given a structure I and a subset X of the domain of I, we will denote by I_X the substructure of I induced by X, that is, the substructure of I whose domain consists of X plus the elements of I named by constant symbols, and containing all facts over this domain that belong to I.

We now claim the following "bounded base lemma" for FGTGDs:

Claim 1: whenever a structure I is not saturated, then there is a subset X of the domain of I, with $|X| \leq k$, and such that the induced substructure I_X is not saturated.
Claim 2: a structure is saturated if and only if it is guardedly saturated.

The result easily follows from these two claims: simply define the Datalog program to be the program consisting of all frontier-guarded rules (in the above expanded signature) that are implied by Σ' and that have at most k variables. A lemma similar to Claim 1 occurs in Marnette's unpublished work [Mar11] (Marnette's "bounded depth property").

The second claim is easy to prove. One direction is trivial and the other direction makes use of the standard "Chase construction" for TGDs (see, e.g. [FKMP05]): given a structure I for schema S and a finite set of TGDs Σ, the chase construction produces a structure J satisfying Σ, containing all tuples in I, and such that a conjunctive query with constants from I is satisfied in J exactly when it is implied by Σ and the facts in I. The chase is formed just by repeatedly throwing in fresh witnesses for the heads of unsatisfied TGDs. One can check that for frontier-guarded TGDs, this procedure never introduces a new fact over the active domain of I plus the elements named by constants, if that fact was not already guarded in I.

The first claim is proved by contraposition. Suppose that every substructure I_X of I with $|X| \leq k$ is saturated. Then by applying Lemma 1 to the result of the chase on I_X, we obtain that each such I_X has a (possibly infinite) squid-extension $\widehat{I_X}$ satisfying Σ. By the last property of the chase listed above, together with the fact that I_X is saturated, we have that $\widehat{I_X}$ does not contain any additional facts over the set X plus the set of elements named by constant symbols. We now define J to be the union of all these $\widehat{I_X}$ (overlapping only on $adom(I)$ and the elements named by constant symbols). By construction, J extends I and contains no new guarded facts over $adom(I)$ and the elements named by constant symbols. Moreover, J is a squid-extension of I.

It remains to show that $J \models \Sigma'$. Suppose, for the sake of contradiction, that there is a frontier-guarded TGD σ in Σ' of the form $\forall \mathbf{x}(\phi(\mathbf{x}) \rightarrow \exists \mathbf{y}\psi(\mathbf{x},\mathbf{y}))$ that is not satisfied. We may assume that σ is among the smallest ones in Σ' that are not satisfied. Now, consider any map $h : \{\mathbf{x}\} \rightarrow adom(J)$ witnessing the fact that the TGD is false in J.

We claim that the h-image of ϕ must be entirely contained in either I or in one of the tentacles J' of J, if we disregard facts that consist entirely of constants. Suppose this were not the case. The free variables of ϕ all map into some tentacle J', since they are guarded. Consider the subquery ϕ_1 of ϕ formed by removing all atoms that map into J'. ϕ_1 contains at least two atoms, one mapping to the frontier of J' and I and one atom mapping outside J'. We modify ϕ_1 so that its free variables are those variables mapping onto the frontier between I and J', resulting in another answer-guarded query. Since ϕ_1

is smaller than ϕ, by minimality of σ the FGTGD $\forall \mathbf{x}\phi \to R_\phi$, which is in Σ', must hold in J. Now let ϕ_2 be formed from ϕ by replacing the subquery ϕ_1 by R_ϕ, and consider the FGTGD $\sigma' = \forall \mathbf{x}(\phi_2(\mathbf{x}) \to \exists \mathbf{y}\psi(\mathbf{x},\mathbf{y}))$. It is easy to see that $\sigma' \in \Sigma'$, since the use of definitions is conservative. Since σ' is also smaller than σ, by minimality of σ it holds in J. But then σ must hold in J as well, a contradiction. This completes the proof of the claim.

If the h-image of ϕ is entirely contained in some tentacle, then this tentacle belongs to some $\widehat{I_X}$ which we know by construction satisfies Σ and hence the right-hand side of the TGD is satisfied, which we have assumed is not the case. If on the other hand, the h-image of ϕ is entirely contained in I, then we take the subset I' of I of size at most k that contains all elements in the h-image of ϕ, and use the fact that $I' \subseteq \widehat{I'} \models \Sigma$. Either way, we reach a contradiction.

Proof of Theorem 12

Recall the statement:

For every set Σ of guarded TGDs, and for every answer-guarded conjunctive query Q, one can effectively find a guarded Datalog program that computes on any structure I the certain answer to Q under Σ.

Proof. First consider the special case where Q is an atomic predicate. As in the proof of Theorem 11, we say that a structure I is *saturated* (with respect to Σ) if no new fact over the active domain of I plus the elements named by constant symbols is entailed by the facts of I together with Σ. The following is a version of the "bounded base lemma" referred to in the proof of Theorem 11, specialized to guarded TGDs:

> whenever a structure I is not saturated, then there is a guarded subset X of the domain of I such that the induced substructure I_X is not saturated.

Theorem 12 in this special case easily follows from this claim: just take the Datalog program containing all guarded rules that are implied by Σ.

The above claim is easily proven as follows: assume that every induced substructure I_X, for X a guarded subset, is saturated, and let J be constructed from I by chasing each I_X independently, and taking the union of the results. By construction, J does not contain any new facts over the domain of I. Moreover, it is easy to see that every guarded TGD in Σ is satisfied in J.

Next, we consider proving Theorem 12 in the more general case where Q is an answer-guarded conjunctive query. By the Treeification Lemma of Bárány, Gottlob and Otto (Lemma 5), there is a union of acyclic (answer-guarded) conjunctive queries χ_Q such that a GNFO sentence Σ implies Q iff it implies χ_Q. Introducing a fresh atomic predicate for each answer-guarded subquery of a conjunctive query in χ_Q, we can eliminate these queries, turning them into rules that are added to Σ, firing an atomic goal predicate on a tuple exactly when the original query Q returned that tuple in its certain answers. We thus have reduced to reasoning with a set of guarded TGDs Σ' and an atomic conjunctive query, where the result follows from the argument above.