The Journal of Symbolic Logic
Volume 00, Number 0, XXX 0000

SOME MODEL THEORY OF GUARDED NEGATION QUERIES

VINCE BÁRÁNY, MICHAEL BENEDIKT, AND BALDER TEN CATE

Abstract. The Guarded Negation Fragment (GNFO) is a fragment of first-order logic that contains all positive existential formulas, can express the first-order translations of basic modal logic and of many description logics, along with many sentences that arise in databases. It has been shown that the syntax of GNFO is restrictive enough so that computational problems such as validity and satisfiability are still decidable. This suggests that, in spite of its expressive power, GNFO formulas are amenable to novel optimizations. In this paper we study the model theory of GNFO formulas. Our results include effective preservation theorems for GNFO, effective Craig Interpolation and Beth Definability results, and the ability to express the certain answers of queries with respect to a large class of GNFO sentences within very restricted logics.

§1. Introduction. The guarded negation fragment (GNFO) is a syntactic fragment of first-order logic, introduced in [11] as an extension to the much-studied guarded fragment of first-order logic [2, 29]. Both fragments restrict the use of certain syntactic constructs by requiring the presence of *guards*, with the aim of taming the language from an algorithmic point of view, with an acceptable compromise on expressiveness. The guarded fragment is obtained by requiring all quantification to be guarded. This idea has its roots in modal logic and, accordingly, the model theory of the resulting fragment has a very similar flavour to that of modal logic. The guarded negation fragment is obtained instead by requiring all use of negation to be guarded. As it turns out, the latter use of guards is more general than the former. Formally, every sentence of the guarded fragment can be equivalently expressed in the guarded negation fragment [8]. GNFO also properly contains the positive existential fragment of FO.

GFO constitutes a rich formalism that captures many of the integrity constraint languages and schema-mapping languages proposed in databases [33, 23], and also many of the description logics [3] proposed in knowledge representation. But GNFO is more suitable than GFO for expressing database *queries*; that is, mappings from structures to relations. Indeed, as noted above, GNFO properly contains all positive existential formulas. These are the most common SQL queries, built up using the basic SELECT FROM WHERE construct and UNION.

Bárány's work done while affiliated with TU Darmstadt.
Benedikt was supported by EPSRC grant EP/H017690/1
ten Cate was supported by NSF Grants IIS-0905276 IIS-1217869.

0022-4812/00/0000-0000/$00.00

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 atomic formula or an equality atom, 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 also supported; this can be seen as a special case of guarded negation through the use of a vacuous equality guard $x = x$.

It was shown in [8] that GNFO possesses a number of desirable computational properties. For example, every satisfiable GNFO formula has a finite model (*finite model property*), as well as a, typically infinite, model of bounded tree-width (*tree-like model property*). It follows that satisfiability and entailment (hence, by the finite model property, satisfiability and entailment in the finite) of GNFO formulas are decidable.

In [10] the implications of GNFO for database theory are explored. For example, an SQL-based syntax for GNFO is defined, and an analogously constrained variant of stratified Datalog is also presented. Several computational problems concerning GNFO formulas (e.g. the "boundedness problem" for a fragment of the fixpoint extension of GNFO) are shown to be decidable.

In this work we investigate model-theoretic properties of GNFO. We first present results showing that GNFO formulas satisfying specific semantic properties can be rewritten into restricted syntactic forms. For example, we show that every GNFO formula that is closed under extensions can be effectively rewritten as an existential GNFO formula. We give an analogous result for queries closed under homomorphisms.

Next we consider GNFO sentences that can also be expressed as a kind of generalized Horn sentence known in the database community as a tuple-generating dependencies (TGD). We provide a syntactic characterization of the GNFO sentences that are equivalent to a finite set of TGDs and give a similar result for sentences in the guarded fragment.

We then turn to model theoretic results concerning explicit and implicit definability. The Projective Beth Definability theorem states that for any property that is implicitly defined by a first-order theory there is a first-order formula that explicitly defines the property. We show the analogous result with first-order replaced by GNFO. Following ideas of Marx [37] we establish a Craig Interpolation Theorem for GNFO and from this conclude the Projective Beth Definability theorem for GNFO. This is in contrast with the situation for the Guarded Fragment, which does enjoy the simpler Beth definability property [32]. Contradicting claims made in earlier work [37] we show that Projective Beth fails for the so-called packed fragment.

Finally, we study definability issues related to the "open world query answering" problem for GNFO. Open world query answering concerns determining which results of formulas are implied by partial information about the underlying structure, in the form of a subset of the interpretations of relations and a logical theory constraining the completion. More formally, the input to this problem is a set Σ of GNFO sentences, a finite structure F, and a positive existential formula Q. The goal is to determine the values of Q that hold in every

structure extending the interpretations of relations in F and satisfying Σ. These values are sometimes referred to as "the certain answers to Q under Σ". The complexity of open world query answering has already been identified for several GNFO-based languages in [10]. Here we show that GNFO sentences that are equivalent to a set of TGDs have additional attractive properties from the point of view of open world query answering. Specifically, we extend and correct results of Baget et. al. [6] by showing that the certain answers can always be determined by evaluating a sentence in a small fragment of (guarded negation) fixpoint logic, Guarded Negation Datalog, for which boundedness was shown decidable in [10]. From this we conclude that first-order definability of certain answers of GNFO TGDs is decidable.

An extended abstract of the present paper appeared in [7]. Related work both prior to and subsequent to [7] is discussed in Section 6.

Organization: Section 2 contains preliminaries. Section 3 looks at rewriting for restricted fragments of GNFO, while Section 4 looks at rewriting of queries with respect to views, via results on Craig interpolation and Beth definability. Section 5 presents our results on rewriting the certain answers of conjunctive queries with respect to GNFO TGDs. Section 6 covers conclusions and related work.

§2. Definitions and Preliminaries. We work with fragments of first-order logic (FO) with equality and with its usual semantics, restricting attention to finite signatures consisting of relation symbols and constant symbols and no function symbols.

We assume familiarity with basic notions from model theory, such as a *reduct* of a structure (restricting the signature), an *expansion* of a structure, and a *type* (a satisfiable set of formulas in a collection of variables, possibly with parameters from a structure); and will only rely on material that can be found in the first few chapters of a standard model theory textbook, such as Chang and Keisler [19]. For example, we will make use of the Compactness Theorem and work with saturated elementary extensions. We briefly review the notion of saturation that we need in this work. A structure $\mathfrak{B}$ is an *elementary extension* of a structure $\mathfrak{A}$, denoted $\mathfrak{A} \preceq \mathfrak{B}$, if $\mathfrak{B}$ is an extension of $\mathfrak{A}$ and every FO sentence with parameters from $\mathfrak{A}$ that is true in $\mathfrak{A}$ is also true in $\mathfrak{B}$. A structure $\mathfrak{A}$ is *ω-saturated* if for every set of formulas $\Gamma(\mathbf{x})$ (where $\mathbf{x} = x_1, \ldots, x_n$) containing finitely many parameters from $\mathfrak{A}$, if every finite subset of $\Gamma(\mathbf{x})$ is realized by some n-tuple in $\mathfrak{A}$, then the entire set $\Gamma(\mathbf{x})$ is realized by an n-tuple in $\mathfrak{A}$. The conclusion means that there is a tuple $\mathbf{c}$ of elements of the domain of $\mathfrak{A}$ such that $\mathfrak{A} \models \gamma(\mathbf{c})$ for all $\gamma(\mathbf{x}) \in \Gamma(\mathbf{x})$. A first-order structure is *recursively saturated* if the conclusion above holds when the collection Γ is further required to be recursive (or, in other words, decidable). A basic result in model theory is that every structure has an ω-saturated elementary extension, and every countable structure (in a countable signature) has a countable recursively-saturated elementary extension.

A *homomorphism* $h : \mathfrak{A} \to \mathfrak{B}$ between structures $\mathfrak{A}$ and $\mathfrak{B}$ is a map from the domain of $\mathfrak{A}$ to the domain of $\mathfrak{B}$ that preserves the relations (i.e., $(a_1, \ldots, a_n) \in R^{\mathfrak{A}}$ implies $(h(a_1), \ldots, h(a_n)) \in R^{\mathfrak{B}}$) as well as the interpretation of all constant symbols (i.e., $h(c^{\mathfrak{A}}) = c^{\mathfrak{B}}$).

The primary focus of this paper is on finite structures. Finite model theory is concerned with logical semantics restricted to finite structures. When working

with both classical and finite model semantics additional care must be taken to make it clear in each instance which semantics is meant. Crucially, both GFO and GNFO possess the finite model property (every satisfiable sentence has a finite model), which for most purposes voids the distinction between the two semantics and allows us to employ classical tools in the service of finite model theory. But at times, when working with different formalisms, we will need to be more specific as to which semantics is meant. We shall use the shorthand "(Both classically and in the finite.)" in formal assertions to signify that the statement holds equally true when semantic entailment is unrestricted and when it is restricted to finite structures.

Database query languages and constraint languages. One motivation for this work is to explore how well GNFO is suited for database applications. Accordingly, we will work with several logics and that are common in database theory, introduced below.

- *Existential FO*, comprises formulas $\exists x_1 \dots x_n\ \phi$, where ϕ is quantifier-free.
- *Conjunctive queries* (CQ), are the subset of existential FO where the quantifier-free kernel ϕ above does not contain disjunction or negation. Equivalently, these are the first-order formulas in prenex normal form built up using only $\wedge$ and $\exists$. A *boolean* conjunctive query is a CQ without free variables, that is, expressed as a FO sentence.
- *Acyclic conjunctive queries* form an algorithmically well-behaved subclass of conjunctive queries [50, 24, 28]. The standard definition of acyclic CQ involves the notions of hypergraph acyclicity and hypergraph structure of a CQ [28]. We will not need to directly use this definition, but only the following equivalent characterization, which generalizes one in [28] for boolean acyclic CQs. A formula ϕ is *answer-guarded* if it is of the form $\phi(\mathbf{x}) = R(\mathbf{x}) \wedge \phi'$ for some ϕ' and relation symbol R. Then we have the following alternative characterization of acyclic answer-guarded CQs:

  FACT 2.1. *An answer-guarded conjunctive query is acyclic iff it is equivalent to a positive existential* GFO *formula.*

- *Tuple-generating dependencies* (TGD) 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 positive relational atoms (no equalities), and every variable from $\mathbf{x}$ occurs in at least one conjunct of ϕ. ϕ is called the *body* of the TGD, while ρ is referred to as the *head*.

In addition to the above fragments of FO, some of our arguments involve *Datalog* a language that extends positive-existential FO with a fixpoint mechanism. Datalog programs use a signature that is partitioned into "intensional relations", representing the results of a fixpoint computation, and "extensional relations" that represent an input structure. In terms of second-order logic, intensional relations can be viewed as second-order variables, while extensional relations are part of the signature of the structure over which the program is being evaluated. A Datalog program Π consists of rules $R(x_1 \dots x_n) \ :=\ \phi$, where R is an intensional relation and ϕ is a CQ over intensional and extensional relations, such that each variable x_i occurs in at least one conjunct of ϕ. Associated to the program Π is an operator that takes as input a structure $\mathfrak{A}$ in the extended signature that includes both the extensional and intensional relations and returns a structure $\mathfrak{A}'$

over the same extended signature. $\mathfrak{A}'$ agrees with $\mathfrak{A}$ on all extensional relations. For each intensional relation R, $R_{\mathfrak{A}'}$ is the set of n-tuples obtained by evaluating a rule of Π of the form $R(x_1 \dots x_n) := \phi$ (that is, evaluating ϕ in $\mathfrak{A}$ and projecting on variables $x_1 \dots x_n$). This "immediate consequence" operator on structures is monotone, and thus has a unique least fixpoint. The result of evaluating a program Π on a structure $\mathfrak{A}$ is the least fixpoint (starting with all intensional relations empty). Given a distinguished intensional predicate P (the *goal* predicate), the *output* of a Datalog program is the set of tuples belonging to the goal predicate in the least fixpoint. Datalog can be viewed as the positive-existential fragment of least-fixpoint logic.

Abiteboul, Hull, and Vianu [1] is a good reference for all of these languages.

One subtle but notable difference in the treatment of query languages in the database literature and the logic literature concerns the relationship between database instances and (finite) first-order structures. A *database instance* (or simply *instance*) I for a signature τ, assigns to every relation symbol $R \in \tau$ of arity n a collection of n-tuples, and to every constant symbol c a value, called the *interpretation* of R, and respectively of c, in I. A *fact* over a signature τ is an expression $R(a_1 \dots a_n)$, where R is a relation symbol and $a_1 \dots a_n$ are values. An interpretation of a relation R can be equivalently considered as a set of facts, namely the facts of the form $R(a_1 \dots a_n)$ where $(a_1, \dots, a_n)$ belongs to the interpretation of R. The *active domain* of an instance or a structure is the set of values that participate in some fact, or, in other words, the union of the one-dimensional projections of the relations. We write $\mathrm{adom}(\mathfrak{A})$ for the active domain of $\mathfrak{A}$. Note the difference between an instance and a relational structure: a relational structure is defined over an explicitly given domain, which can contain any number of "inactive" elements. Two structures can thus correspond to the same instance while having different domains. In database theory one is typically interested in *domain-independent* formulas, that is, formulas that do not distinguish between structures corresponding to the same instance. For example the sentence $\exists x\, U(x)$ is domain-independent, while $\forall x\, U(x)$ is not. Both CQs and Datalog are domain-independent languages. In parts of this work, we will deal with logical formulas that are domain-independent. For a domain-independent sentence ϕ we can talk about ϕ "being true on instance I", and similarly give semantics to domain-independent formulas in terms of instances rather than structures. Thus if we are dealing with questions about domain-independent formulas, it will often be convenient to perform constructions that form instances from instances, rather than constructions that form structures from structures. A homomorphism $h : I \to J$ between instances I and J is defined as with structures, but h is now defined on the active domain of I, and is required to preserve the interpretation of the relations as well as any constants occurring in the active domain of I.

Given two structures $\mathfrak{A}, \mathfrak{B}$ over the same signature τ, we write $\mathfrak{A} \subseteq^w \mathfrak{B}$ if the two structures agree on the interpretation of the constant symbols, and, for every relation $R \in \tau$, $R^{\mathfrak{A}} \subseteq R^{\mathfrak{B}}$. This can be thought of as a weak version of the usual substructure relation, where we do not require the substructure to be induced by taking a subset of the domain. Since the definition does not refer to the domains of the structures $\mathfrak{A}, \mathfrak{B}$, it is clearly also applicable to instances.

To every CQ $q(\mathbf{x}) = \exists \mathbf{y} \bigwedge_i \alpha_i$ of signature τ one can associate the τ-instance $\mathrm{CanonInst}(q)$, the *canonical instance* associated to q: the active domain of $\mathrm{CanonInst}(q)$

consists of the set of variables and constants occurring in q and the facts are the literals α_i. Evaluation of a CQ can be restated in terms of homomorphisms from CanonInst(Q): for every n-ary CQ $q(x_1 \dots, x_n)$ and every n-tuple $\mathbf{a}$ of an instance I we have that $I \models q(\mathbf{a})$ iff there exists a homomorphism $h : (\text{CanonInst}(q), \mathbf{x}) \to (I, \mathbf{a})$ [18].

The Guarded-Negation Fragment. The Guarded Negation Fragment (GNFO) is a syntactic fragment of first-order logic, from which it inherits the usual semantics. The formulas of GNFO are built up inductively according to the grammar[1]

$$\phi ::= R(t_1, \dots, t_n) \mid t_1 = t_2 \mid \exists x\,(\phi) \mid (\phi \vee \phi) \mid (\phi \wedge \phi) \mid (\alpha \wedge \neg\phi)$$

where R is a relation symbol, each t_i is a variable or a constant symbol, and, in the last clause, α is an atomic formula (possibly an equality) in which all free variables of the negated formula ϕ occur. That is, each use of negation must occur conjoined with an atomic formula that contains all the free variables of the negated formula. The atomic formula α that witnesses this is called a *guard* for $\neg\phi$. Since we allow equalities as guards, every formula with at most one free variable can be trivially guarded, and we often write $\neg\phi$ instead of $((x = x) \wedge \neg\phi)$, when ϕ has no free variables besides (possibly) x. For τ a signature consisting of constant symbols and relation symbols, GNFO[τ] denotes the GNFO formulas in signature τ.

GNFO should be compared to the *Guarded Fragment* (GFO) of first-order logic [2, 29] typically defined via the grammar

$$\phi ::= R(t_1, \dots, t_n) \mid t_1 = t_2 \mid \exists \mathbf{x}\,(\alpha \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \wedge \phi) \mid \neg\phi$$

where, in the third clause, α is again an atomic formula in which all free variables of ϕ occur (and $\mathbf{x}$ may be a sequence of variables). Note that, in GFO formulas, all quantification must occur in conjunction with a guard, while there is no restriction on the use of negation.

Since GNFO is closed under conjunction and existential quantifications, every conjunctive query is expressible in GNFO. It is not much more difficult to verify that every GFO sentence can also be equivalently expressed in GNFO [8]. Turning to fragments of first-order logic that are common in database theory, consider *guarded tuple-generating dependencies*: that is, sentences of the form

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

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* (i.e. every formula $\forall \mathbf{x}\,(R(\mathbf{x}) \to \exists \mathbf{y}\,S(\mathbf{x}, \mathbf{y}))$, where the atomic formulas $R(\mathbf{x})$ and $S(\mathbf{x}, \mathbf{y})$ have no constants and no repeated variables) is expressible in GNFO. As mentioned in the introduction, many of the common dependencies used to describe relationships between schemas (e.g. see [33, 23]) are expressible in GNFO. In addition, many of the common description logic languages used in the semantic web (e.g. $\mathcal{ALC}$ and $\mathcal{ALCHIO}$ [3]) are known to admit translations into GFO and hence into GNFO.

We will frequently make use of the key result from [8] showing that GNFO is decidable and has the finite model property:

[1] In practice, the parentheses are often omitted and parsing ambiguity is resolved with the help of the standard order of precedence of logical connectives.

THEOREM 2.2. *A* GNFO *sentence is satisfiable over all structures iff it is satisfiable over finite structures. Satisfiability and validity are decidable (and* 2ExpTime*-complete).*

It was shown in [10] that GNFO can be equivalently restated as a fragment of Codd's relational algebra, and of the standard database query language SQL. More specifically, in [10], a fragment of relational algebra, called Guarded-Negation Relation Algebra (GN-RA) is introduced, and is shown to capture domain-independent GNFO. It is worth noting also that we can actually decide whether a given GNFO formula is domain-independent (and hence whether it can be converted to GN-RA). This is in contrast to the well-known fact that domain-independence is undecidable for first-order logic [1]. To see the decidability, we simply note that the statement expressing that a GNFO formula is domain-independent can be expressed as the validity of a GNFO sentence, and then apply Theorem 2.2.

Note that if we have two GNFO open formulas $\phi_1(\mathbf{x})$ and $\phi_2(\mathbf{x})$, the sentence stating that they are equivalent, or that one implies the other, is not necessarily a GNFO sentence. This does hold, however, if ϕ_1 and ϕ_2 are answer-guarded. We will need to require answer-guardedness in some of our results involving open formulas.[2] Most results about GNFO sentences trivially generalize to answer-guarded GNFO formulas. For instance, the observation from [8] that every GFO sentence can be equivalently transcribed into GNFO extends to answer-guarded GFO formulas.

Guarded sets and tuples. Let $\mathfrak{A}$ be a structure and $e_1, \ldots, e_k$ be the interpretation of all constants in the signature of $\mathfrak{A}$. A subset X of the domain of $\mathfrak{A}$ is *guarded* if there is a fact (in some relation) in which all members of $X \setminus \{e_1, \ldots, e_k\}$ occur together. We will sometimes apply the same notion to tuples: a tuple of values from the domain of a structure is guarded (in the structure), if the set of all elements of the tuple is guarded. Note that an answer-guarded query can only be satisfied by guarded tuples.

Tree-like model property. Satisfiable GFO formulas always have models that are "tree-like": this is the *tree-like model property* of GFO [2, 29]. For any relational structure $\mathfrak{A}$ with constants, and any guarded tuple $\mathbf{a}$ there is a *guarded unravelling* [2] $(\mathfrak{A}^*_{\mathbf{a}}, \langle \mathbf{a} \rangle)$ of $\mathfrak{A}$ at $\mathbf{a}$, a structure and tuple such that:

(i) $\mathfrak{A}^*_{\mathbf{a}}$ is *tree like* in the sense that it has a tree decomposition with guarded bags [30];
(ii) $\mathfrak{A}^*_{\mathbf{a}} \models \varphi(\langle \mathbf{a} \rangle)$ if and only if $\mathfrak{A} \models \varphi(\mathbf{a})$ for all $\varphi(\mathbf{x}) \in$ GFO.

We conclude this section by recalling an important result about approximating arbitrary answer-guarded conjunctive queries by conjunctive queries that are in GFO, which is proven using the unravellings above.

Paraphrasing [9] we define the *treeification* $T(q)$ of an answer-guarded CQ q as the collection of minimal acyclic CQ that imply q. From [9] we know that $T(q)$ is finite if the signature is finite. We will thus sometimes identify the treeification with the (answer-guarded) UCQ $\bigvee T(q)$.

[2] Note, however, that the equivalence problem and the entailment problem are decidable in 2ExpTime even for *non*-answer-guarded GNFO formulas (as follows from a easy reduction in which free variables are replaced by constant symbols). See, for example, Corollary 5.10.

The next fact is a simple consequence of the definition of treeification and of the properties of guarded unravellings. It was first observed in [9] in the case of boolean CQs, but the same reasoning applies to answer-guarded CQs.

FACT 2.3 (Treeification). *For every answer-guarded CQ* $q(\mathbf{x})$*, every structure* $\mathfrak{A}$ *and guarded tuple* $\mathbf{a}$ *of* M *it holds that* $\mathfrak{A}^*_{\mathbf{a}} \models q(\langle\mathbf{a}\rangle)$ *iff* $\mathfrak{A}^*_{\mathbf{a}} \models \bigvee T(q)(\langle\mathbf{a}\rangle)$*. Consequently, for every answer-guarded* GFO *formula* $\phi(\mathbf{x})$ *and answer-guarded conjunctive query* $q(\mathbf{x})$ *it holds that* $\phi(\mathbf{x}) \models q(\mathbf{x})$ *iff* $\phi(\mathbf{x}) \models \bigvee T(q)(\mathbf{x})$*.*

We note that guarded unravellings are typically infinite and that it takes considerably more work to show that the last claim remains valid when restricting attention to finite structures [9] and add that this very claim is what underpins the argument in [8] establishing the finite model property of GNFO.

§3. Characterization and Preservation theorems. Preservation theorems are results showing that every property definable within a certain logic and which additionally satisfies some important semantic invariance can be expressed by a formula in the logic whose syntactic form guarantees that invariance. One example from classical model theory is the Łoś-Tarski theorem, stating that a property of structures definable in first-order logic is definable by a universal formula if and only if it is closed under taking substructures. A second example is the Homomorphism Preservation theorem, stating that a property of structures definable in first-order logic is expressible by an existential positive sentences if and only if it is closed under homomorphism [19]. One can consider the "finite model theory analogs" of each of these statements: for example, the finite model theory analog of Łoś-Tarski would be that a property of finite structures definable in first-order logic that is closed under taking substructures must be definable by a universal formula of first-order logic. This analog is known to fail [21]. Rossman [45] has shown that the finite analog of the Homomorphism Preservation theorem does hold.

A well-known preservation theorem from modal logic is Van Benthem's theorem, stating that basic modal logic can express precisely the properties expressible in first-order logic invariant under bisimulation [49]. Rosen [44] has shown that Van Benthem's theorem also remains valid if one restricts attention to finite structures, cf. also [42]. Analogous results on arbitrary structures have been established for both GFO [2] and GNFO [8]. In the context of finite model theory, Otto [43, 41] provided Van Benthem-style characterizations of GFO and of the "k-bounded fragment of GNFO" indexed by a number k. Central to these results are the notions of *guarded bisimulation* and *guarded negation bisimulation* that play similar roles in the model theory of GFO, respectively, GNFO as does bisimulation in the model theory of modal logic. For a comprehensive survey the interested reader should turn to [30].

3.1. Characterizing GNFO within FO. We first look at the question of characterizing GNFO as a fragment of first-order logic invariant under certain simulation relations. In [8] *guarded-negation bisimulations* (*GN-bisimulations*) were introduced, and it was shown that GNFO expresses the first-order logic properties that are invariant under GN-bisimulations. A related characterization over finite structures for the k-variable fragment of GNFO is given in [41]. Here we will work over all structures, giving 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 equivalent to one in GNFO, to argue that two structures must agree on all GNFO formulas and to amalgamate structures that cannot be distinguished by GNFO sentences in a sub-signature. 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. Recall also 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 within the fact except possibly those that are the interpretation of some constant symbol.

DEFINITION 3.1 (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 $h\colon \mathfrak{A} \to \mathfrak{B}$ such that $h(\mathbf{a}) = \mathbf{b}$ and "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 $g\colon \mathfrak{B} \to \mathfrak{A}$ such that $g(\mathbf{b}) = \mathbf{a}$ and "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.

It is easy to see that if there exists a strong GN-bisimulation between two structures, then the respective substructures consisting of the elements designated by constant symbols must be isomorphic.

The key distinction between strong GN-bisimulation and the GN-bisimulation of [8] is that the homomorphisms whose existence is postulated in the back-and-forth properties of GN-bisimulation are only required to be "local", that is, defined on arbitrary finite neighbourhoods of the guarded tuple in question, while our definition above asks for a single "global" homomorphism that is defined on the entire domain of the respective structure, i.e. one that is uniformly appropriate for all neighbourhoods according to the requirements of GN-bisimulations of [11]. This is a very significant strengthening of requirements, which makes strong GN-bisimulation more powerful as a tool in our proofs.

Another distinction between the notions is that while GN-bisimulations are only defined on guarded tuples, our notion of strong GN-bisimulation is meaningful on arbitrary tuples. It is an equivalence relation on guarded tuples, but is asymmetric on general tuples.

In [8] it was shown that GNFO corresponds to the GN-bisimulation-invariant fragment of first-order logic. In light of our previous remark, it follows that GNFO formulas are also invariant under strong GN-bisimulations as far as guarded tuples are concerned. In fact, for arbitrary tuples one can verify via structural induction on the construction of formulas that all GNFO formulas are preserved by strong GN-bisimulations. That is, one can show that $\to^{s}_{GN}$ implies $\Rrightarrow_{GN}$, where the notation

$$(\mathfrak{A}, \mathbf{a}) \Rrightarrow_{GN} (\mathfrak{B}, \mathbf{b})$$

expresses that, for every GNFO formula $\phi(\mathbf{x})$, $\mathfrak{A} \models \phi(\mathbf{a})$ implies $\mathfrak{B} \models \phi(\mathbf{b})$.

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

THEOREM 3.2. *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 GN-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 3.2 relies on the following lemma asserting, in essence, that $\Rightarrow_{GN}$ can always be lifted to $\to^s_{GN}$ by passing from a pair of structures to suitable elementary extensions. This step is established using the technique of *recursively saturated models* [19] and will be equally instrumental in our proof of Craig Interpolation for GNFO presented in Section 4.

LEMMA 3.3.

1. *If $(\mathfrak{A}, \boldsymbol{a}) \to^s_{GN[\sigma]} (\mathfrak{B}, \boldsymbol{b})$ then $(\mathfrak{A}, \boldsymbol{a}) \Rightarrow_{GN[\sigma]} (\mathfrak{B}, \boldsymbol{b})$.*
2. *If $(\mathfrak{A}, \boldsymbol{a}) \Rightarrow_{GN[\sigma]} (\mathfrak{B}, \boldsymbol{b})$ and both structures are countable, then there are countable elementary extensions $(\widehat{\mathfrak{A}}, \boldsymbol{a})$ and $(\widehat{\mathfrak{B}}, \boldsymbol{b})$, respectively, such that $(\widehat{\mathfrak{A}}, \boldsymbol{a}) \to^s_{GN[\sigma]} (\widehat{\mathfrak{B}}, \boldsymbol{b})$.*

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

Consider the pair of countable 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 finite 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 finite 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 and 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_i, and $\mathbf{d}$ an enumeration of the range of f_i, corresponding to the enumeration of $\mathbf{c}$, and let $\Sigma(x)$ be the set of all first-order formulas of the form

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

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_i-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 $(\widehat{\mathfrak{A}}, \widehat{\mathfrak{B}})$. Note that in the argument above we are only relying on the closure of GNFO under conjunction and existential quantification.

By compactness, therefore, $\Sigma(x) \cup \{Q(x)\}$ is consistent and, by virtue of recursive saturation, it is realized by some element d_{i+1}. It follows from the construction that the partial map $f_{i+1} = f_i \cup \{(c_{i+1}, d_{i+1})\}$ preserves truth of all GNFO formulas. ⊣

This concludes the proof of the lemma. ⊣

Proof of Theorem 3.2.. We prove only the harder direction, following the template often used in preservation theorems in classical model theory. Let $\phi(\mathbf{x})$ be preserved by $\to^s_{GN}$, and let $\Psi(\mathbf{x})$ be the set of all GNFO formulas it entails. Thanks to compactness, it is enough to show that $\Psi(\mathbf{x}) \models \phi(\mathbf{x})$.

Let $\mathfrak{B} \models \Psi(\mathbf{b})$, and let $\Gamma_{\mathfrak{B},\mathbf{b}}(\mathbf{x})$ 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 $\mathfrak{B}$. Note that we may assume that both $\mathfrak{A}$ and $\mathfrak{B}$ are countable. Using Lemma 3.3, we can find elementary equivalent extensions completing the following diagram.

$$\begin{array}{ccc} (\widehat{\mathfrak{A}}, \mathbf{a}) & \to^s_{GN} & (\widehat{\mathfrak{B}}, \mathbf{b}) \\ \overset{\preceq}{} & & \overset{\preceq}{} \\ (\mathfrak{A}, \mathbf{a}) & \Rrightarrow_{GN} & (\mathfrak{B}, \mathbf{b}) \end{array}$$

By virtue of ϕ being invariant under elementary equivalence and being preserved by strong GN-bisimulations, we can chase it around the diagram starting from $\mathfrak{A} \models \phi(\mathbf{a})$ and concluding $\mathfrak{B} \models \phi(\mathbf{b})$. Given that $\mathfrak{B} \models \Psi(\mathbf{b})$ was arbitrary, this shows that $\Psi(\mathbf{x}) \models \phi(\mathbf{x})$ and so the theorem follows. ⊣

We now look at characterizing the intersection of GNFO with smaller fragments of first-order logic. We will start with tuple-generating dependencies.

3.2. Tuple-generating dependencies within GNFO. Recall that a *tuple-generating dependency (TGD)* is a sentence 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 atomic formulas (not equalities). TGDs arise in databases, as a way of specifying natural restrictions on data and as a way of capturing relationships between different datasources. They also arise in ontological reasoning. Static analysis and query answering problems have motivated research to identify expressive yet computationally well-behaved classes of TGDs. A guarded TGD (GTGD) is one in which ϕ includes an atomic formula containing all variables occurring in the TGD. Guarded TGDs constitute an important class of TGDs at the heart of the Datalog$^{\pm}$ framework [17, 9]

for which many computational problems are decidable. More recently, Baget, Leclère, and Mugnier [5] introduced *frontier-guarded TGDs* (FGTGDs), defined like guarded TGDs, but where only the variables occurring both in ϕ and in ρ (the *exported* variables) must be guarded by an atomic formula in ϕ. Every FGTGD is equivalent to a GNFO sentence, obtained just by writing it out using existential quantification, negation, and conjunction. Theorem 3.7 below shows that these are *exactly* the TGDs that GNFO can express.

We need two lemmas: one about GNFO and one about TGDs. For two structures $\mathfrak{A} \subseteq^w \mathfrak{B}$, let us denote by $\mathfrak{B} \ominus \mathfrak{A}$ the structure obtained from $\mathfrak{B}$ by removing all facts containing only values from the active domain of $\mathfrak{A}$. We say that $\mathfrak{B}$ is a *squid-extension* of $\mathfrak{A}$ if

(i) every set of elements from the active domain of $\mathfrak{A}$ that is guarded in $\mathfrak{B}$ is already guarded in $\mathfrak{A}$; and
(ii) $\mathfrak{B} \ominus \mathfrak{A}$ is a union of structures $\mathfrak{B}'_i$ such that: for two distinct $\mathfrak{B}'_i$ and $\mathfrak{B}'_j$ their active domains overlap only in $\mathrm{adom}(\mathfrak{A}) \cup C$, and each $(\mathrm{adom}(\mathfrak{B}'_i) \cap \mathrm{adom}(\mathfrak{A})) \setminus C$ is guarded in $\mathfrak{A}$, where C is the set of elements of $\mathfrak{A}$ named by a constant symbol.

Intuitively, we can think of $\mathfrak{B}$ as a squid, where each $\mathfrak{B}'_i$ is one of its tentacles. We refer to the $\mathfrak{B}_i$ as the tentacles, and the partition into $\mathfrak{B}_i$ as a *squid decomposition* of $\mathfrak{B}$.

We extend the notation to instances in the obvious way (since it does not depend on the domain of $\mathfrak{A}$ or $\mathfrak{B}$). The following lemma allows one to turn an arbitrary extension of a structure $\mathfrak{A}$ into a squid-extension of $\mathfrak{A}$, modulo strong GN-bisimulation.

LEMMA 3.4. *For every pair of structures $\mathfrak{A}, \mathfrak{B}$ with $\mathfrak{A} \subseteq^w \mathfrak{B}$, there is a squid-extension $\mathfrak{B}'$ of $\mathfrak{A}$ and a homomorphism $h : \mathfrak{B}' \to \mathfrak{B}$ whose restriction to $\mathfrak{A}$ is the identity function, such that $\mathfrak{B}' \sim^s_{GN} \mathfrak{B}$ via a strong GN-bisimulation that is compatible with h. Moreover, we can choose $\mathfrak{B}'$ to be finite if $\mathfrak{B}$ is.*

We will make use of Lemma 3.4 as a tool for bringing certain conjunctive queries into a restricted syntactic form, by exploiting the fact that, whenever a tuple from $\mathrm{adom}(\mathfrak{A})$ satisfies a conjunctive query in a squid-extension $\mathfrak{B}$ of $\mathfrak{A}$, then we can partition the atomic formulas of the query into independent subsets that are mapped into different tentacles of $\mathfrak{B}$.

PROOF. For every set X of elements that is guarded in $\mathfrak{A}$, we create a structure $\mathfrak{B}_X$ that is a fresh isomorphic copy of $\mathfrak{B}$ 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 elements named by a constant symbol. We define $\mathfrak{B}'$ to be the union of all such $\mathfrak{B}_X$. Clearly, $\mathfrak{B}'$ is a squid-extension of $\mathfrak{A}$, and the natural projection $h : \mathfrak{B}' \to \mathfrak{B}$ is a homomorphism. Furthermore, we claim that $\mathfrak{B}' \sim^s_{GN} \mathfrak{B}$ via a strong GN-bisimulation that is compatible with h. The claimed strong GN-bisimulation consists of all pairs $(\mathbf{a}, h(\mathbf{a}))$ where $\mathbf{a}$ is a guarded tuple of $\mathfrak{B}'$. ⊣

The following lemma expresses a general property of TGDs that follows from the fact that TGDs are preserved under taking direct products of structures [22].

LEMMA 3.5. *(Both classically and in the finite.) 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 and the ψ_i are conjunctions of atomic formulas. Then $\Sigma \models \forall \mathbf{x}(\phi(\mathbf{x}) \to \exists \mathbf{y}_i \psi_i(\mathbf{x}, \mathbf{y}_i))$ for some $i \leq n$.*

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

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

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 $h_1 : J \rightarrow I_1$ and $h_2 : J \rightarrow I_2$ are homomorphisms, 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 $J \models \Sigma$.

Because J is finite if both I_1 and I_2 are, the above argument is equally valid over finite structures as over arbitrary structures. ⊣

We now return to describing our characterization of TGDs that are equivalent to some GNFO sentence. Consider a TGD $\rho = \forall \mathbf{x}\,(\beta(\mathbf{x}) \rightarrow \exists \mathbf{z} \gamma(\mathbf{x}\mathbf{z}))$. A *specialization* of ρ is a TGD of the form $\rho^\theta = \forall \mathbf{x}\,(\beta(\mathbf{x}) \rightarrow \exists \mathbf{z}' \gamma'(\mathbf{x}\mathbf{z}'))$ 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 3.6. *[TGD specializations] (Both classically and in the finite.) 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 specialization ρ' of ρ such that $\Sigma \models \rho'$, and such that ρ' is logically equivalent to a conjunction of frontier-guarded TGDs.*

PROOF. First we introduce the notion of a *quasi-frontier guarded TGD*. By the *graph of a TGD* $\rho = \forall \mathbf{x}\,(\beta(\mathbf{x}) \rightarrow \exists \mathbf{z} \gamma(\mathbf{x}, \mathbf{z}))$ 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 if the graph of ρ is not connected, then ρ can be decomposed into several TGDs, 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 atomic formulas belonging to that component is guarded by some atomic formula in the TGD body β. This is equivalent to saying that the decomposition into TGDs just mentioned yields a set of frontier-guarded TGDs.

We will show that, if Σ is a set of TGDs that is strong GN-bisimulation invariant and ρ is 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.

Thus fix $\rho = \forall \mathbf{x}\,(\beta(\mathbf{x}) \rightarrow \exists \mathbf{z} \gamma(\mathbf{x}, \mathbf{z}))$ such that $\Sigma \models \rho$.

Consider any structure $J \models \Sigma$ and homomorphism $h : \text{CanonInst}(\beta(\mathbf{x})) \to J$. Let B be the image of h. By Lemma 3.4, 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, $J' \models \Sigma$. Therefore since $\Sigma \models \rho$, $J' \models \rho$. In particular, h can be extended to a homomorphism h' from $\text{CanonInst}(\exists \mathbf{z}\gamma(\mathbf{x}, \mathbf{z}))$ 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, 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

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

is a specialization of ρ such that the CQ $\exists \mathbf{z}'\gamma'(\mathbf{x}, \mathbf{z}')$ is satisfied in J', hence also in J, under the assignment h for the universally quantified variables $\mathbf{x}$. We first show that each ρ' is quasi-frontier-guarded. Consider the decomposition of ρ'

$$\rho' \equiv \bigwedge_j \forall \mathbf{x} \left(\beta(\mathbf{x}) \to \exists \mathbf{z}'\gamma'_j(\mathbf{x}, \mathbf{z}')\right)$$

such that the graphs of $\rho'_j = \forall \mathbf{x} \left(\beta(\mathbf{x}) \to \exists \mathbf{z}'\gamma'_j(\mathbf{x}, \mathbf{z}')\right)$ enumerate the connected components of the graph of ρ' and let j be arbitrary.

Note that, by construction, all existential variables $\mathbf{z}'$ are mapped by h' to elements that neither belong to $\text{adom}(B)$ nor interpret any constant symbol: if h' had mapped an existential variable to $\text{adom}(B)$, then this variable would have been removed and replaced by a universal variable. Next note that the active domains of the tentacles of J' overlap only on elements of $\text{adom}(B)$. Using connectivity of γ'_j, we see that the existential variables must map to the active domain of a single tentacle. From connectedness of the graph of γ'_j, we know there are two possibilities: if there are no existential variables in γ'_j, then γ'_j consists of a single atom. In this case the universal variables map into a guarded set of B. If there is any existential variable in γ'_j, then every universal variable lies in some atom with an existential variable. Since the existential variables do not map into $\text{adom}(B)$, it follows that the image of $\text{CanonInst}(\gamma'_j)$ under h' must be entirely contained in a single tentacle of J'. Now the subset of the universally-quantified variables $\mathbf{x}$ occurring in γ'_j is mapped into B, since h mapped into B and h' extended h. Thus the variables $\mathbf{x}$ must be mapped by h' to the intersection of a tentacle and the active domain of B, hence (by the properties of a squid decomposition) again we can conclude that $\mathbf{x}$ maps to a guarded set of elements of B. And since h' agrees with h on these variables, the same statement holds with h substituting for h'. Since B was defined as the h-image of $\text{CanonInst}(\beta)$, we can conclude that the universally-quantified variables occurring in γ'_j are guarded in β, viz. ρ'_j is frontier-guarded. Since j was arbitrary, this shows that ρ' is indeed quasi-frontier-guarded.

Now we need to show that one such ρ' is entailed by Σ. What we have shown thus far is that any J that is satisfied by Σ satisfies one such ρ'. But there are only finitely many such ρ', and thus by Lemma 3.5 we can conclude that Σ entails one such ρ'. ⊣

The result above immediately implies our first main characterization:

THEOREM 3.7. *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 light of the above result, it may seem tempting to suppose that, similarly, guarded TGDs can express all that can be expressed both by TGDs and in GFO. This is, however, not the case: the TGD $\forall xyz\ \ (R(x,y) \wedge R(y,z) \rightarrow P(x))$ can be equivalently expressed in GFO, but not by means of a guarded TGD; and the guarded TGD $\forall x\ \ (P(x) \rightarrow \exists yz\ E(x,y) \wedge E(y,z) \wedge E(z,x))$ is not expressible in GFO. Instead, we show that every property expressible both in GFO and by a finite set of TGDs is in fact expressible by a finite set of *acyclic frontier-guarded TGDs*.

Recall from Section 2 that a CQ is answer-guarded if its free variables co-occur in one of its atomic sub-formulas and that such a CQ is acyclic if it is equivalent to a positive-existential GFO formula. We say that a frontier-guarded TGD $\rho = \forall \mathbf{xy}\,(\beta(\mathbf{x},\mathbf{y}) \rightarrow \exists \mathbf{z}\,\gamma(\mathbf{x},\mathbf{z}))$ is acyclic if the answer-guarded CQ $\exists \mathbf{y}\beta(\mathbf{x},\mathbf{y})$ and the answer-guarded CQ $\exists \mathbf{yz}\,\beta(\mathbf{x},\mathbf{y}) \wedge \gamma(\mathbf{x},\mathbf{z})$ are both acyclic. Note that both CQs are indeed answer-guarded, by virtue of ρ being frontier-guarded.

THEOREM 3.8. *Every* GFO *sentence that is equivalent to a finite set of TGDs over finite structures is equivalent (over all structures) to a finite set of acyclic FGTGDs.*

PROOF. Let ϕ be any GFO sentence that is equivalent to a finite set of TGDs over finite structures. Then, by Theorem 3.7, ϕ is equivalent to a finite set Σ of FGTGDs over arbitrary structures.

Recall the notion of guarded unravelling $\mathfrak{A}^*$ of a structure $\mathfrak{A}$ and the notion of treeification of an answer-guarded CQ from Section 2. Note that for each TGD in Σ, its left-hand side is answer-guarded by definition, and its right-hand side can be assumed answer-guarded as well. Consider the set Σ' of disjunctive GTGDs obtained by replacing the head and body of each TGD by its treeification, and expanding out the disjunction in the left-hand side.

We claim that Σ is equivalent to Σ'. Note that since ϕ is in GFO, for any structure $\mathfrak{A}$, $\mathfrak{A} \models \phi \leftrightarrow \mathfrak{A}^* \models \phi$. Similarly, since Σ' is in GFO, $\mathfrak{A} \models \phi \leftrightarrow \mathfrak{A}^* \models \phi$. Thus it is enough to show equivalence of ϕ and Σ' on guarded unravellings. But from Fact 2.3 we see that each formula is equivalent to its treeification on guarded unravellings, and so our claim is proven.

Now by Lemma 3.5, we obtain that each disjunctive TGD in Σ' is equivalent to one of the GTGDs obtained by replacing the disjunction in its head by one of the disjuncts. Since the head and body of each such TGD are acyclic, each such TGD is acyclic. ⊣

3.3. Existential and Positive-Existential Formulas. We turn to characterizing the existential formulas within GNFO, establishing an analog of the Łoś-Tarski theorem.

THEOREM 3.9. *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 an equivalent 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 $\Phi = \bigwedge_{c\in\mathbf{c}\cup\mathbf{d}} P(c) \wedge \phi^P(\mathbf{d}) \to \phi(\mathbf{d})$ 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.

As to the effective content of the claim, note that once an equivalent existential formula is known to exist in GNFO, we can find it by exhaustive search relying on the decidability of equivalence of GNFO formulas.

By the classical Łoś-Tarski theorem, if a first-order formula is preserved under extensions over all structures, it is equivalent to an existential formula ϕ'. Thus, to complete the proof, it suffices to show that every GNFO formula ϕ that is equivalent to an existential formula ϕ' is also equivalent to an existential GNFO formula ϕ''. We can assume that ϕ is satisfiable (since otherwise it is clearly equivalent to a GNFO formula). We can convert ϕ' into the form $\bigvee_i \phi'_i$, where $\phi'_i(\mathbf{x}) = \exists\mathbf{y}\left(\varepsilon'_i \wedge \bigwedge_j \psi'_{ij}\right)$ with each ψ'_{ij} a possibly negated relational atom and where ε_i is the conjunction of (in)equalities of a complete equality type on $\mathbf{cxy}$, viz. a maximal satisfiable set of (in)equalities involving the constants $\mathbf{c}$ and variables $\mathbf{xy}$.

In general, some of the negated atomic formulas and inequalities in ϕ' may not be guarded. Let ϕ'' be obtained from ϕ' by removing all conjuncts that are unguarded negative atomic formulas or unguarded inequalities.

We claim that ϕ' and ϕ'' are equivalent. One direction is obvious, since ϕ' clearly implies ϕ''. In the remainder of the proof, we show that ϕ'' implies ϕ'.

Consider an arbitrary structure $\mathfrak{A}$ and tuple $\mathbf{a}$ such that $\mathfrak{A} \models \phi''(\mathbf{a})$. It is our task to show that $\mathfrak{A} \models \phi'(\mathbf{a})$. Our general approach will be to construct another structure $\mathfrak{A}'$ and tuple $\mathbf{b}$ such that $\mathfrak{A}' \models \phi'(\mathbf{b})$. In addition, we will show that $(\mathfrak{A}', \mathbf{b}) \to^s_{GN} (\mathfrak{A}, \mathbf{a})$. By Theorem 3.2, this will allow us to conclude $\mathfrak{A} \models \phi'(\mathbf{a})$ as needed, since ϕ' is logically equivalent to $\phi \in$ GNFO.

Let h be a variable assignment from an appropriate $\phi''_i(\mathbf{x}) = \exists\mathbf{y}\left(\varepsilon''_i \wedge \bigwedge_j \psi''_{ij}\right)$ to elements of $\mathfrak{A}$, witnessing $\mathfrak{A} \models \phi''(\mathbf{a})$. In particular, ε''_i is in general an incomplete equality type on $\mathbf{cxy}$ that only includes an equality or inequality of every pair of variables that co-occur in a positive relational atom in some ψ''_{ij}. We need to show that $\mathfrak{A} \models \phi'_i(h(\mathbf{x}))$. The main obstacles to overcome are:

(i) the possibility that h maps two variables u, v to the same element of $\mathfrak{A}$ while ε'_i includes the (unguarded) inequality $u \neq v$.
(ii) the possibility that $\mathfrak{A}$ 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 $\mathfrak{A}'$ 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 $\mathfrak{A}$ at $\mathbf{a}$ and (ii) making sure that every fact satisfied in $\mathfrak{A}'$ whose values are in the range of h is guarded by a fact that is an h-image of a positive atomic formula of ϕ'_i.

The precise construction is as follows. Let X be the set of constants and all variables occurring, free or bound, in ϕ'_i. Further let $\equiv$ be the equivalence relation on X generated by all pairs of constants or variables (u, v) such that ε''_i

contains the equality $u = v$. Let $f : X \to X/_{\equiv}$ be the natural map that sends each variable to its equivalence class. We define the structure $\mathfrak{A}^*$ with domain $X/_{\equiv}$ and, for each relation symbol R, the relation $R^{\mathfrak{A}^*}$ consisting of tuples $f(\mathbf{u})$ such that $R(\mathbf{u})$ occurs as a positive atomic sub-formula in ϕ_i'' or, what is the same, in ϕ_i'. Further let the $\equiv$-class of each constant interpret in $\mathfrak{A}^*$ the corresponding constant symbol and let $\mathbf{b} = f(\mathbf{x})$. Note that $\mathfrak{A}^*$ depends on $\mathfrak{A}$ solely through the choice of the disjunct ϕ_i' that is assumed to be satisfied at $\mathbf{a}$ in $\mathfrak{A}$ via the variable assignment h.

- Observation 1: there is a homomorphism $g : \mathfrak{A}^* \to dom(\mathfrak{A})$ such that $h = g \circ f$ and such that g is injective on guarded subsets of $\mathfrak{A}^*$, viz. it maps distinct elements co-occurring in a fact of $\mathfrak{A}^*$ to distinct elements of $\mathfrak{A}$.
- Observation 2: f assigns elements of $\mathfrak{A}^*$ to variables of ϕ' in a manner witnessing $\mathfrak{A}^* \models \phi'(\mathbf{b})$.

Observation 1 follows from the definition $\equiv$ and of $\mathfrak{A}^*$. Observation 2 follows from the construction of $\mathfrak{A}^*$ (for the equalities, inequalities, and positive atomic formulas) and from the previous observation (for the negative atomic formulas).

As a next step, we transform $\mathfrak{A}^*$ into $\mathfrak{A}'$ as follows. For each fact F of $\mathfrak{A}^*$ we make an isomorphic copy of $\mathfrak{A}$ denoted $\mathfrak{A}'_F$, where the isomorphism maps the elements belonging to the g-image of F to their, by Observation 1, unique g-preimage and maps all other elements to distinct fresh elements. We define $\mathfrak{A}'$ as the union $\mathfrak{A}^* \cup \bigcup\{\mathfrak{A}'_F \mid F \text{ a fact of } \mathfrak{A}^*\}$, and let $\widehat{g} : \mathfrak{A}' \to \mathfrak{A}$ be the map that extends g by mapping every newly-created element in some $\mathfrak{A}'_F$ to the corresponding element of $\mathfrak{A}$. Note that, by construction, $\widehat{g} : \mathfrak{A}^* \to \mathfrak{A}$ is a homomorphism.

- Observation 3: $\mathfrak{A}' \models \phi_i'(\mathbf{b})$ via the variable assignment f.
- Observation 4: $(\mathfrak{A}', \mathbf{b}) \to^{s}_{GN} (\mathfrak{A}, \mathbf{a})$.

Observation 3 follows from Observation 2, $\mathfrak{A}^* \subseteq^{w} \mathfrak{A}'$, and the observation that $\mathfrak{A}'$ does not add any new facts on elements of $\mathfrak{A}^*$. For Observation 4, it can be easily verified that the graph of $\widehat{g}$ is in fact a strong GN-bisimulation, which is compatible with the homomorphism g and $g(\mathbf{b}) = \mathbf{a}$. From Observation 4 and Theorem 3.2 we get that $\mathfrak{A} \models \phi'(\mathbf{a})$ as needed. ⊣

Note. This theorem can also be proven by refining the GNFO interpolation theorem of Section 4 to get a Lyndon-style interpolation theorem. The approach via interpolation is spelled out in the paper [12].

Finally, we consider the situation for GNFO formulas that are positive existential (for short, $\exists^+$). Since GNFO contains all $\exists^+$ formulas, Rossman's homomorphism preservation theorem [45] implies that the $\exists^+$ formulas are exactly the formulas in GNFO closed under 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 effectively decide whether a GNFO formula can be rewritten in $\exists^+$.

THEOREM 3.10. *There is an effective algorithm for testing whether a given* GNFO *formula is equivalent to a positive existential formula, and, if so, computing such a formula.*

PROOF. Rossman's proof [45] 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. ⊣

§4. Interpolation and Beth definability for GNFO. The Craig Interpolation theorem for first-order logic [20] 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* [13]. Suppose that we have a sentence ϕ over a first-order signature of the form $\sigma \cup \{G\}$, where G is an n-ary predicate, and suppose σ' is a subset of σ. A sentence ϕ *implicitly defines predicate G over σ'* if: for every σ'-structure I, every expansion to a $\sigma \cup \{G\}$-structure I' satisfying ϕ has the same restriction to G up to isomorphism. Informally, the σ' structure and the sentence ϕ determine a unique value for G. An n-ary predicate G is *explicitly definable over σ' 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 σ' for models of ϕ, then ϕ implicitly defines G over σ'. The Projective Beth Definability theorem states the converse: if ϕ implicitly defines G over σ', then G is explicitly definable over σ' for models of ϕ. In the special case where $\sigma' = \sigma$, 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. [19]). The known proofs are not effective, and it has been shown that the proof cannot be made effective [25]. 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 [31] to show that CIP implies PBDP.

CIP and PBDP do not hold when implication is restricted to finite models [21]. However, the finite and unrestricted versions of these properties are equivalent when considering fragments of FO that have the finite model property, since there equivalence (resp. consequence) over finite structures can be replaced by equivalence (resp. consequence) over all structures. Thus it is particularly natural to look at CIP and PBDP for such fragments, such as GFO and GNFO. Hoogland, Marx, and Otto [32] showed that the Guarded Fragment satisfies BDP but lacks CIP. Marx [37] went on to explore PBDP for the Guarded Fragment and its extensions. He argues that the PBDP holds for an extension of GFO called 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 [37]) for the Packed Fragment. But we will adapt ideas of Marx to show that CIP and PBDP do hold for GNFO.

The main technical result of this section is then:

THEOREM 4.1 (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*.*

Section 4.1 is dedicated to the proof of Theorem 4.1. In Section 4.2 we present further applications of the result, and in Section 4.3 we discuss failure of interpolation for the guarded fragment.

Observe that in Theorem 4.1, 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 [46] 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.

4.1. Proof of Craig interpolation for GNFO. To establish Theorem 4.1 we follow a common approach in modal logic (see, in particular, Hoogland, Marx, and Otto [32]), 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 (in the respective signatures). 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 [38]. In the lemma and claims below, $\mathbf{a}$ will range over tuples, not necessarily guarded.

LEMMA 4.2 (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

$\mathbf{c} = c_1 \dots c_k$ of elements of $\mathfrak{A}$ and for any k-tuple $\mathbf{d} = d_1 \dots d_k$ of elements of $\mathfrak{B}$, we will denote by $\langle \mathbf{c}, \mathbf{d} \rangle$ the k-tuple $((c_1, d_1), \dots, (c_k, d_k))$.

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

- the domain of $\mathfrak{U}$ is $\{(c, d) \in \mathfrak{A} \times \mathfrak{B} \mid (c \mapsto d) \in Z\}$;
- $R^{\mathfrak{U}} = \{\langle \mathbf{c}, \mathbf{d} \rangle \mid \mathbf{c} \in R^{\mathfrak{A}} \text{ and } (\mathbf{c} \mapsto \mathbf{d}) \in Z\}$ for every $R \in \sigma$;
- $S^{\mathfrak{U}} = \{\langle \mathbf{c}, \mathbf{d} \rangle \mid \mathbf{d} \in S^{\mathfrak{B}} \text{ and } (\mathbf{d} \mapsto \mathbf{c}) \in Z\}$ for every $S \in \tau$;
- $c^{\mathfrak{U}} = (c^{\mathfrak{A}}, c^{\mathfrak{B}})$ for every constant symbol c;
- $\mathbf{u} = \langle \mathbf{a}, \mathbf{b} \rangle$.

To see that $\mathfrak{U}$ is thus well defined, 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: $(\mathfrak{A}, \mathbf{a}) \to^s_{GN[\sigma]} (\mathfrak{U}, \mathbf{u})$

Proof of claim 1. 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-atomic formula) 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 requirements.

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 requirements. ⊣

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

Proof of claim 2. the relevant strong GN-bisimulation Z'' is constructed analogously to Z' above. 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''. ⊣

⊣

Proof of Theorem 4.1. Without loss of generality we can assume that ϕ and ψ have the same free variables and reference the same set of constant symbols (eg. by appending vacuous identities $x_i = x_i$ or $c_j = c_j$ as conjuncts to either formula as needed). With this proviso let $\phi(\mathbf{x})$ and $\psi(\mathbf{x})$ be GNFO-formulas with free variables $\mathbf{x}$ such that $\models \forall \mathbf{x}(\phi(\mathbf{x}) \to \psi(\mathbf{x}))$; let σ and τ denote their respective signatures and suppose, for the sake of contradiction, that there is no GNFO$[\sigma \cap \tau]$-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 GNFO$[\sigma \cap \tau]$ consequences of $\phi(\mathbf{x})$. Due to compactness, we know that $\Phi(\mathbf{x})$ cannot imply $\psi(\mathbf{x})$. Therefore, there is a structure $\mathfrak{B} \models \Phi(\mathbf{b}) \wedge \neg\psi(\mathbf{b})$. Next, consider

$$\Psi(\mathbf{x}) = \{\neg\eta(\mathbf{x}) \mid \eta(\mathbf{x}) \in \text{GNFO}[\sigma \cap \tau],\ \mathfrak{B} \models \neg\eta(\mathbf{b})\}$$

and notice that $\Psi(\mathbf{x})$ does not imply $\neg\phi(\mathbf{x})$. For otherwise there would be, due to compactness, some natural k and $\neg\eta_0(\mathbf{x}), \ldots, \neg\eta_{k-1}(\mathbf{x}) \in \Psi(\mathbf{x})$ such that $\bigwedge_{j<k} \neg\eta_j(\mathbf{x}) \models \neg\phi(\mathbf{x})$ ie. $\phi(\mathbf{x}) \models \bigvee_{j<k} \eta_j(\mathbf{x})$ and thus $\bigvee_{j<k} \eta_j(\mathbf{x}) \in \Phi(\mathbf{x})$, because $\bigvee_{j<k} \eta_j(\mathbf{x}) \in \mathrm{GNFO}[\sigma \cap \tau]$, implying $\mathfrak{B} \models \bigvee_{j<k} \eta_j(\mathbf{b})$ in contradiction to the fact that $\eta_j(\mathbf{x}) \in \Psi(\mathbf{x})$ and hence $\mathfrak{B} \models \neg\eta_j(\mathbf{b})$ for each $j < k$. Therefore, there is a structure $\mathfrak{A} \models \Psi(\mathbf{a}) \wedge \phi(\mathbf{a})$. By construction, we have that $(\mathfrak{A}, \mathbf{a}) \Rrightarrow_{GN[\sigma\cap\tau]} (\mathfrak{B}, \mathbf{b})$.

Note that in the above step we can ensure that both $\mathfrak{A}$ and $\mathfrak{B}$ are countable. Thus, using Lemma 3.3, we can 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 respective elementary extensions $(\widehat{\mathfrak{A}}, \mathbf{a})$ and $(\widehat{\mathfrak{B}}, \mathbf{b})$. Applying the Amalgamation Lemma 4.2 to these extensions we obtain $(\mathfrak{U}, \mathbf{u})$ such that $(\widehat{\mathfrak{A}}, \mathbf{a}) \rightarrow^s_{GN[\sigma]} (\mathfrak{U}, \mathbf{u}) \rightarrow^s_{GN[\tau]} (\widehat{\mathfrak{B}}, \mathbf{b})$. Observe that $\mathfrak{U} \models \phi(\mathbf{u})$ follows from $\widehat{\mathfrak{A}} \models \phi(\mathbf{a})$ and $(\widehat{\mathfrak{A}}, \mathbf{a}) \rightarrow^s_{GN[\sigma]}$. Similarly, we can infer $\mathfrak{U} \models \neg\psi(\mathbf{u})$ for otherwise $(\mathfrak{U}, \mathbf{u}) \rightarrow^s_{GN[\tau]} (\widehat{\mathfrak{B}}, \mathbf{b})$ would allow us to conclude $\widehat{\mathfrak{B}} \models \psi(\mathbf{b})$ contradicting our choice of $(\widehat{\mathfrak{B}}, \mathbf{b})$. Thus we have found $\mathfrak{U} \models \phi(\mathbf{u}) \wedge \neg\psi(\mathbf{u})$ contradicting the assumption that $\phi(\mathbf{x})$ implies $\psi(\mathbf{x})$. ⊣

4.2. Applications of Interpolation. An analogue of the Projective Beth Definability theorem [13] for GNFO follows from Craig interpolation by standard arguments [31].

COROLLARY 4.3. *If a* GNFO *sentence ϕ in signature σ implicitly defines a relation symbol R in terms of a signature $\tau \subset \sigma$, and τ includes all constants from σ, then there is an explicit definition of R in terms of τ relative to ϕ.*

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 σ that is disjoint from V. Suppose ϕ_Q is another first-order formula over the signature σ. The formulas $\phi_v : v \in V$ *determine* ϕ_Q *over finite structures* if for all finite σ-structures I and I' with $\phi_v(I) = \phi_v(I')$ for all $v \in V$, we have $\phi_Q(I) = \phi_Q(I')$. Similarly, we say that the set $\{\phi_v : v \in V\}$ determine ϕ_Q over all structures if the above holds for all I and I'. Unwinding the definitions, the reader can see that the latter assertion is the same as stating that the sentences asserting

$$\forall \mathbf{x}\ \phi(\mathbf{x}) \leftrightarrow V(\mathbf{x})$$

as well as

$$\forall \mathbf{x}\ \phi_Q(\mathbf{x}) \leftrightarrow Q(\mathbf{x})$$

implicitly define the relation Q over the signature V. In the database literature, the symbols $v \in V$ are often referred to as "view relations" and the corresponding formula ϕ_v is the "view definition for v".

From the PBDP we know that when $\{\phi_v : v \in V\}$ determine ϕ_Q over all structures, there is a first-order formula ρ over V that explicitly defines Q. Such a ρ is called a *rewriting of* ϕ_Q *over* $\{\phi_v : v \in V\}$. Segoufin and Vianu initiated a study of determinacy for special classes of formulas ϕ_v and ϕ_Q, including the question of deciding when determinacy and determinacy-over-finite-structures holds, and examining when the assumption of determinacy implies that the rewriting is realized by a formula in a restricted logic. Nash, Segoufin, and Vianu showed that determinacy over finite structures for unions of conjunctive queries is undecidable [39], and that for UCQs determinacy over finite structures does not imply

rewritability even in first-order logic. More recently determinacy for conjunctive queries has been shown undecidable both over finite structures and over all structures [26, 27]. 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.

We will use the PBDP above to show that whenever GNFO $\{\phi_v : v \in V\}$ determines GNFO ϕ_Q and additionally both $\{\phi_v : v \in V\}$ and ϕ_Q are answer-guarded, then there is a first-order rewriting, and even a rewriting in GNFO. Recall from Section 2 that answer-guarded formulas are those of the form $\phi(\mathbf{x}) = R(\mathbf{x}) \wedge \phi'$ for some ϕ' and relation symbol R. Note that rewritings of determined queries, when they exist, can always be taken to be domain-independent queries, since $\phi_Q(I)$ is, by definition of determinacy, only dependent on $\phi_v(I)$ for $v \in V$. Note also that if answer-guarded GNFO views determine an GNFO ϕ_Q over finite structures, they determine it over all structures: this follows from the finite model property of GNFO because determinacy of ϕ_Q by $\{\phi_v : v \in V\}$ can be expressed as a GNFO sentence. Similarly, "$\{\phi_v : v \in V\}$ determine ϕ_Q" (when the ϕ_v and ϕ_Q are answer-guarded and in GNFO) can be decided in 2ExpTime using Theorem 2.2.

We can now state the consequence of the PBDP for determinacy-and-rewriting (relying again on the finite model property of GNFO).

COROLLARY 4.4. *Suppose a set of answer-guarded* GNFO *queries* $\{\phi_v : v \in V\}$ *determines an answer-guarded* GNFO *query* ϕ_Q *over finite structures. Then there is a* GNFO *query* ρ *that is a rewriting. Furthermore, there is an algorithm that, given* ϕ_v*'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, when restricting to finite structures. Using the finite model property of GNFO, we see that implicit definability hold over all structures. 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 [8]. ⊣

Work subsequent to this article has obtained tight bounds on the rewritings [12], via a constructive approach to GNFO interpolation.

Recall from our discussion above that rewritings are domain-independent, since they depend only on the facts produced by the view definitions. Thus, as discussed in Section 2, they can be converted to GN-RA. Note also that GNFO views V can check properties of a structure (e.g. linear TGDs) as well as return results. Using the above, we can get the following variant of Corollary 4.4 for sentences and queries:

Suppose a set of answer-guarded UCQ views $\{\phi_v : v \in V\}$ determine an answer-guarded UCQ ϕ_Q on finite structures satisfying a set of GNFO sentences Σ. Then there is a GNFO rewriting of Q using V that is valid over structures satisfying Σ.

4.3. 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 4.5. *The PBDP fails for* GFO.

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

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

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, we claim 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 $3 \leq k < \ell$. Here we will make use of the notion of guarded bisimulation between structures $\mathfrak{A}$ and $\mathfrak{B}$, due to Andréka, van Benthem, and Németi[2]. This is a non-empty family of partial isomorphisms from $\mathfrak{A}$ to $\mathfrak{B}$ satisfying the following back-and-forth conditions:

- For every partial isomorphism $f \in I$ with domain X and every guarded subset X' of the domain of $\mathfrak{A}$, there is a partial isomorphism $g \in I$ whose domain contains X' agreeing with f on $X \cap X'$
- for $f \in I$ with co-domain Y and every guarded subset Y' of the domain of $\mathfrak{B}$, there is a partial isomorphism $g \in I$ with domain containing Y' such that g^{-1} and f^{-1} agree on $Y \cap Y'$

It is known [2] that guarded bisimulation preserves expressibility in GFO.

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 $(c,d) \in E^{C_\ell}$. One can verify directly that Z is a guarded-bisimulation between C_k and C_ℓ. ⊣

It follows from Theorem 4.5 that GFO lacks CIP as well, which was already known [32]. 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 [37, 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 4.5, however, shows that PBDP fails for the Packed Fragment, because known results (cf. [37]) imply that no formula of the Packed Fragment can distinguish the cycle of length k from the cycle of length ℓ for $4 \leq k < \ell$. This can also be shown by appealing to the notion packed bisimulation [37], a variant of guarded bisimulation which characterizes expressibility in the packed fragment. In fact the relation Z defined in the proof of Theorem 4.5 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 sentence $\exists xyz(Rxy \wedge Ryz \wedge Rzx)$ distinguishes C_3 from C_4. By writing it as $\exists xyz(Rxy \wedge Ryz \wedge Rzx) \wedge \top$ we see that this sentence is in the packed fragment. Indeed, it turns out that there is a flaw in the proof of Lemma 4.4 in [37].

§5. Expressibility of certain answers for queries with respect to GNFO TGDs. A fundamental concept in the study of information integration and ontology-mediated data access is the notion of *certain answers* for a conjunctive query with respect to a database instance and a collection of sentences. For the sake of presentational consistency, we define certain answers here in terms of structures, rather than database instances. Note that the queries and sentences that we consider in this section are all domain independent. Hence, as pointed out in Section 2, their evaluation is determined by the underlying instance of a structure, and hence in this section we can make use of constructions taking instances to instances.

Given two structures $\mathfrak{A}, \mathfrak{B}$ over the same signature τ, recall the notation $\mathfrak{A} \subseteq^w \mathfrak{B}$, meaning that the two structures agree on the interpretation of the constant symbols, and, for every relation $R \in \tau$, $R^{\mathfrak{A}} \subseteq R^{\mathfrak{B}}$. Let $\mathfrak{A}$ be a finite structure, Σ a set of sentences in some logic, and $Q(x_1 \dots x_k)$ a formula in some logic. A tuple $(a_1 \dots a_k) \in dom(\mathfrak{A})^k$ is a *certain answer of Q with respect to $\mathfrak{A}$ and Σ* if $\mathfrak{B}, a_1 \dots a_k \models Q$ in every model $\mathfrak{B}$ of Σ such that $\mathfrak{A} \subseteq^w \mathfrak{B}$. Determining which tuples are certain answers is a central problem in information integration and ontology-mediated data access. Typically Σ is referred to as a set of *integrity constraints* (or just "constraints" below, for brevity), while Q is the *query*. The structure $\mathfrak{A}$ represents incomplete information about a structure, and the sentences Σ represent a constraint on the completion. A certain answer to query Q is a result which is already determined by Σ and the presence of the facts in $\mathfrak{A}$. In some cases one considers the "finite model analog" of the above definition: requiring that $\mathfrak{B}, a_1 \dots a_k \models Q$ in every *finite* model $\mathfrak{B}$ of Σ with $\mathfrak{A} \subseteq^w \mathfrak{B}$. For the constraints Σ we consider, there will be no distinction between the finite and unrestricted version of the problems.

One of the benefits of GNFO is that one can effectively determine the certain answers whenever Q and Σ are expressed in GNFO, and thus in particular for every Σ in GNFO and conjunctive query Q [10]. But one can do better for GNFO formulas that are also TGDs. Recall from Section 3 that these are, up to equivalence, frontier-guarded TGDs: TGDs there is a guard on all exported variables. Baget et al. [5] proved that for every set of frontier-guarded dependencies Σ and conjunctive query Q, the certain answers can be computed in polynomial time in $\mathfrak{A}$. However, one could hope for more than just being able to compute the certain answers in polynomial time. A conjunctive query Q is *first-order rewritable* under sentences Σ if there is a first-order formula ϕ such that on any finite structure $\mathfrak{A}$, the tuples that satisfy ϕ in $\mathfrak{A}$ are exactly the certain answers to Q on $\mathfrak{A}$ under Σ. Thus a query is first-order rewritable with respect to Σ if we can reduce finding the certain answers to ordinary evaluation of a first-order formula (which can be done, for example, with a database management system). Unfortunately, it is known that there are guarded TGDs and conjunctive queries such that the certain answers can not be determined by evaluating a first-order query. We will now look at ways of "remedying" this situation.

We will 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. In this process, we will show that the certain answers can be expressed in a "nice" fragment of Datalog, where Datalog is the extension of conjunctive queries with a fixpoint mechanism (see Section 2).

EXAMPLE 5.1. *Consider a signature with binary relations $R(x, y)$ and $S(x, y)$ as well as unary relation $U(x)$.*

Consider the Guarded TGDs:

$$\forall xy\ [R(x, y) \wedge U(y) \rightarrow U(x)]$$
$$\forall x\ [U(x) \rightarrow \exists z\ S(x, z)]$$
$$\forall xy\ [S(x, y) \rightarrow T(x)]$$

and the query $Q(x) = T(x)$.

One can check that the certain answers of Q under Σ on any structure $\mathfrak{A}$ are that as the output of P on $\mathfrak{A}$, where P is the Datalog program with the following rules:

$$UReach(x) := U(x)$$
$$UReach(x) := \exists y\ R(x, y) \wedge UReach(y)$$
$$Goal(x) := UReach(x)$$

Notice that P is a Guarded Datalog program, since the body of each rule is guarded.

We will follow (and correct) the approach of Baget et al. [6], 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 [36]. The proof of Baget et al. [4] 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 [4]). However both the exact statement of that lemma and its proof are flawed. Our proof corrects the argument, making use of model-theoretic techniques to prove the bounded base lemma. It then follows the rest of the argument in [4] to show not only Datalog-rewritability, but rewritability into a Datalog program comprised of frontier-guarded rules (defined below).

The chase. To prove results about certain answers, we will need to make use of the standard "Chase construction" for TGDs (see, e.g. [23]): given a structure $\mathfrak{A}$ for signature σ and a finite set of TGDs Σ, the chase construction produces a structure $Chase_\Sigma(\mathfrak{A})$ with the following properties:

- $Chase_\Sigma(\mathfrak{A})$ satisfies Σ and $\mathfrak{A} \subseteq^w Chase_\Sigma(\mathfrak{A})$.
- for any boolean conjunctive query Q with constants from $\mathfrak{A}$, Q is satisfied in $\mathfrak{B}$ exactly when it is implied by Σ and the facts of $\mathfrak{A}$.

$Chase_\Sigma(\mathfrak{A})$ is formed just by repeatedly throwing in facts using fresh elements to witness the heads of unsatisfied TGDs. There are several variations of the chase [23, 40], but we describe a construction that will suffice for our purpose.

Inductively $Chase_\Sigma(\mathfrak{A})$ it is the union of structures $\mathfrak{B}_j$ with $\mathfrak{B}_0 = \mathfrak{A}$ and $\mathfrak{B}_{j+1}$ formed from $\mathfrak{B}_j$ by: for every $\sigma \in \Sigma$

$$\forall \mathbf{x} \left(\phi(\mathbf{x}) \to \exists \mathbf{y} \bigwedge_i A_i(\mathbf{x}, \mathbf{y}) \right)$$

for every homomorphism h of ϕ into $\mathfrak{B}_j$, add facts $A_i(h(\mathbf{x}), \mathbf{y}_0)$ to $\mathfrak{B}_j$, where $\mathbf{y}_0$ are values disjoint from $\mathrm{adom}(\mathfrak{B}_j)$, any constants of Σ, and the values used in any other σ, h for $\mathfrak{B}_j$.

Several of the arguments below will involve showing that Q is certain with respect to Σ and $\mathfrak{A}$ by arguing that Q must hold in $Chase_\Sigma(\mathfrak{A})$.

We will need an additional observation about the chase with Frontier-Guarded TGDs, which is that the chase has a tree-like structure. This is well-known [6], but it will be useful to state it in terms of our notion of squid extension from earlier in the paper.

LEMMA 5.2. *If Σ consists of frontier-guarded TGDs, then $Chase_\Sigma(\mathfrak{A})$ is a squid extension of $\mathfrak{A}$.*

PROOF. Letting $\mathfrak{B} = Chase_\Sigma(\mathfrak{A})$ recall that we must show that

(i) every set of elements from the active domain of $\mathfrak{A}$ that is guarded in $\mathfrak{B}$ is already guarded in $\mathfrak{A}$; and
(ii) $\mathfrak{B} \ominus \mathfrak{A}$ is a union of tentacles $\mathfrak{B}_X$ for X a guarded subset of $\mathfrak{A}$ such that for distinct X and X', $\mathfrak{B}_X$ and $\mathfrak{B}_{X'}$ overlap in their active domains only in $\mathrm{adom}(\mathfrak{A}) \cup C$, and finally $(\mathrm{adom}(\mathfrak{B}_X) \cap \mathrm{adom}(\mathfrak{A})) \setminus C \subseteq X$, where C is the set of elements of $\mathfrak{A}$ named by a constant symbol.

As we generate $\mathfrak{B} = Chase_\Sigma(\mathfrak{A})$ we build the set of tentacles $\mathfrak{B}_X$ for each guarded set X in $\mathfrak{A}$, inductively preserving the properties above. Initially $\mathfrak{B}_X$ contains every fact in $\mathfrak{A}$ that is guarded by X. Clearly, both properties hold.

Recall that the chase is formed as the union of $\mathfrak{B}_j$, where $\mathfrak{B}_{j+1}$ is formed inductively from $\mathfrak{B}_j$ by firing rules $\sigma \in \Sigma$ based on a homomorphism h of the body of σ into the structure $\mathfrak{B}_j$ built so far, generating facts G that are added to $\mathfrak{B}_j$. Let F be the image of a guard atom for σ under h. Then by induction, F is associated with a unique $\mathfrak{B}_X$ for some X that is guarded in $\mathfrak{A}$. We add G to $\mathfrak{B}_X$.

We show that the inductive invariants are preserved. Clearly $\mathfrak{B}_{j+1} \ominus \mathfrak{A}$ is a union of tentacles, since we added G to exactly one tentacle. Let us consider the first property. Suppose a set $a_1 \dots a_k$ of elements of $\mathfrak{A}$ is guarded by G. Then $a_1 \dots a_k$ must correspond to exported variables of the rule; that is, none of them could have been generated as a fresh value in the creation of G. Thus they must be guarded by X.

For the second property, any new elements added to $\mathrm{adom}(\mathfrak{B}_X)$ must be disjoint from those in $\mathrm{adom}(\mathfrak{B}_{X'})$, and any fact is added to a unique $\mathfrak{B}_X$. Finally any element added to $[\mathrm{adom}(\mathfrak{B}_X) \cap \mathrm{adom}(\mathfrak{A})] \setminus C$ must be contained in the guard atom G, and by induction this is contained in X. ⊣

Rewriting the certain answers of atomic queries over guarded TGDs. We start with a result that gives the intuition for how this rewriting works:

THEOREM 5.3. *For every set Σ of Guarded TGDs, and for every atomic conjunctive query $Q(\boldsymbol{x})$, one can effectively find a Guarded Datalog program P such*

that the output of P on any structure $\mathfrak{A}$ is the same as the certain answers to Q on $\mathfrak{A}$.

Note that entailment here, and throughout the section, 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 holds in the finite.

A *Full Guarded TGD* is a TGD with no existentials in the head. The idea behind the proof the theorem will be that we take all full Guarded TGDs that are consequences of Σ, and turn them into Datalog rules. We will show that the full Guarded TGDs are sufficient to capture the certain answers.

We say that a structure $\mathfrak{A}$ is *fact-saturated* (with respect to Σ) if no new fact over the active domain of $\mathfrak{A}$ plus the elements named by constant symbols is entailed by the facts of $\mathfrak{A}$ together with Σ.

LEMMA 5.4. *For Σ a set of Guarded TGDs, if a structure $\mathfrak{A}$ is not fact-saturated with respect to Σ, then there is a guarded subset X of the domain of $\mathfrak{A}$ such that the induced substructure $\mathfrak{A}_X$ is not fact-saturated with respect to Σ.*

PROOF. We prove the contrapositive. Assume that every induced substructure $\mathfrak{A}_X$, for X a guarded subset, is fact-saturated with respect to Σ. Let $\mathfrak{B}$ be constructed from $\mathfrak{A}$ by chasing each $\mathfrak{A}_X$ with Σ independently and taking the union of the results: that is $\mathfrak{B} = \bigcup_{X \text{ guarded}} Chase_\Sigma(\mathfrak{A}_X)$. Recalling that the chase of $\mathfrak{A}_X$ only satisfies facts over $\mathfrak{A}_X$ that are entailed, we see that $\mathfrak{B}$ does not satisfy any new facts over the domain of $\mathfrak{A}$.

We claim that $\mathfrak{B}$ satisfies every sentence in Σ. Given a σ in Σ of the form

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

and a binding of variables $\mathbf{x}$ into $\mathbf{b} \in \mathfrak{B}$ such that the corresponding facts $\phi(\mathbf{b})$ hold in $\mathfrak{B}$. Note that since σ is a Guarded TGD, $\mathbf{b}$ is guarded. If $\mathbf{b}$ contains only constants and elements of $\mathbf{a}$, then each fact in $\phi(\mathbf{b})$ must be in $\mathfrak{A}$. Hence $\phi(\mathbf{b})$ is in $\mathfrak{A}_X$ and we are done, since $\mathfrak{A}_X$ satisfies Σ. Consider any non-constant element b_i outside of $\mathfrak{A}$. If any such element exists, then the guard fact for $\mathbf{b}$ must have been generated in the chase process for some $\mathfrak{A}_{X_0}$, hence every non-constant element a_i was generated in $\mathfrak{A}_{X_0}$, and every fact in $\phi_(\mathbf{b})$ involving such an element must be in $\mathfrak{A}_{X_0}$. Since each other fact is in $\mathfrak{A}$, hence in $\mathfrak{A}_{X_0}$, we have $\phi_(\mathbf{b})$ is contained in $\mathfrak{A}_{X_0}$ as before, and so we are done because Σ holds in $\mathfrak{A}_{X_0}$.

Thus we have a structure satisfying Σ, containing $\mathfrak{A}$, and containing no new facts over the elements of $\mathfrak{A}$ and the constants. Therefore $\mathfrak{A}$ must be fact-saturated. ⊣

We are now ready to give the proof of Theorem 5.3:

PROOF OF THEOREM 5.3. A *derived Full Guarded TGD* for Σ is a full guarded TGD with a single atom in the head which are entailed by Σ. We let $\Sigma_{FullGuarded}$ be all the derived full guarded TGDs. Note that there are only finitely many of these.

Lemma 5.4 implies that:

For every $\mathfrak{A}$ and atomic query $Q = Goal(\mathbf{x})$, the certain answers of Q over $\mathfrak{A}$ with respect to Σ are the same as the Q-facts entailed by $\mathfrak{A}$ and $\Sigma_{FullGuarded}$.

The Full TGDs of $\Sigma_{FullGuarded}$ are not quite Guarded Datalog: Guarded Datalog requires us to distinguish extensional and intensional relations, and requires that atoms over extensional relations do not occur as consequences within rules. We turn $\Sigma_{FullGuarded}$ into a Guarded Datalog program by replacing each relation R in $\Sigma_{FullGuarded}$ by a copy R'. Thus a Full TGD:

$$\forall \mathbf{xy}\ (R(\mathbf{x},\mathbf{y}) \ldots \rightarrow S(\mathbf{x}))$$

is transformed to the Datalog rule:

$$S'(\mathbf{x}) := \exists \mathbf{y}\ R'(\mathbf{x},\mathbf{y}) \ldots$$

In addition we add rules:

$$R'(\mathbf{x}) := R'(\mathbf{x})$$

Finally, we let $Goal'$ be the goal predicate. It is easy to see that this Datalog program computes a fact $Goal'(\mathbf{a})$ over $\mathfrak{A}$ exactly when $Goal(\mathbf{a})$ is entailed by $\Sigma_{FullGuarded}$ over $\mathfrak{A}$. ⊣

General conjunctive queries and Guarded TGDs. We now extend the result to general conjunctive queries. The conference paper [7] claimed that a the certain answers of an arbitrary answer-guarded CQ Q are expressible in Guarded Datalog. However this is easily seen to be false: indeed even with no constraints we still need to express that Q holds in $\mathfrak{A}$, which is expressible in Guarded Datalog only if Q is equivalent to a GFO formula.

We thus need to move to a slight extension of Guarded Datalog that allows one non-guarded rule at top-level. We consider Datalog programs where the special relation *Goal* does not occur in the body of any rule. Every Datalog program can be rewritten this way. A *goal rule* in such a Datalog program is one that has the relation *Goal* in the head. A Datalog program is *internally-guarded* if every rule that is not a goal rule is Guarded.

Recall that a conjunctive query is answer-guarded if it includes an atomic formula that guards all free variables. In particular all Boolean conjunctive queries are answer-guarded.

Our goal is the following result.

THEOREM 5.5. *For every set Σ of guarded TGDs, and for every conjunctive query Q, one can effectively find an internally-guarded Datalog program P such that on any structure $\mathfrak{A}$ and binding $\mathbf{c}$ for the free variables of Q in $\mathfrak{A}$, $\mathbf{c}$ belongs to the output of P on $\mathfrak{A}$ exactly when $\mathfrak{A} \wedge \Sigma \models Q(\mathbf{c})$.*

In the proof we will make use of the same construction as for the atomic case: given $\mathfrak{A}$, we take each guarded set X of $\mathfrak{A}$, and let $\mathfrak{B} = \bigcup_X Chase_\Sigma(\mathfrak{A}_X)$. In the previous proof we showed that $\mathfrak{B}$ satisfies the constraints Σ. We note further:

The sets $Chase_\Sigma(\mathfrak{A}_X) \ominus \mathfrak{A}$ as X ranges over guarded subset of $\mathfrak{A}$, form tentacles witnessing that $\mathfrak{B}$ is a squid-extension of $\mathfrak{A}$.

Clearly the active domains of these sets overlap only in $\mathfrak{A}_X$, and $\mathfrak{B} \ominus \mathfrak{A}$ is their union. That each $Chase_\Sigma(\mathfrak{A}_X)$ has no new guarded sets which contain only elements in $\mathfrak{A}$ follows from Lemma 5.2.

Since $\mathfrak{B}$ satisfies Σ and $\mathfrak{A} \subseteq^w \mathfrak{B}$, we know that the certain answers of Q over $\mathfrak{A}$ are contained in $Q(\mathfrak{B})$. Thus we can use the properties of $\mathfrak{B}$ to characterize the certain answers of a conjunctive query Q in terms of the entailment of atomic

facts. Let $\mathfrak{A}^+$ be the restriction of $\mathfrak{B}$ to facts whose elements are either in adom($\mathfrak{A}$) or are named by constants.

Suppose we have a homomorphism h of Q to $Chase_\Sigma\mathfrak{A}$, and let h_Q be the image. We can break up the image according to where it fits into a squid decomposition. The observation above implies that $h_Q = F_0 \cup \bigcup_{i \leq n} F_i$ where:

- F_0 is contained in $\mathfrak{A}^+$; that is, F_0 is the part of the image in $\mathfrak{A}^+$.
- For $1 \leq i \leq n$ there is a set G_i of facts in $\mathfrak{B}$ such that F_i is contained in the chase of G_i under Σ. That is, these are the portions in $Chase_\Sigma(\mathfrak{A}_X)$.

The idea behind the proof of Theorem 5.5 will be to add new relations for subqueries of Q, along with Full guarded rules that capture their semantics. For each subquery q of Q, let R_q be a new relation symbol.

For any subquery q of Q, a *derived subquery rule* for q is a full guarded TGD of the form:

$$\forall \mathbf{xy}\ (\bigwedge_i A_i(\mathbf{x}, \mathbf{y}) \rightarrow R_q(\mathbf{x}))$$

such that the corresponding guarded TGD

$$\forall \mathbf{xy}\ (\bigwedge_i A_i \rightarrow q(\mathbf{x}))$$

is a consequence of Σ.

Proof of Theorem 5.5. Consider the signature with intensional relations R_q for every answer-guarded subquery q of Q. Consider the set of full TGDs D_Q consisting of:

- All derived full guarded TGDs (over the original signature)
- All derived subquery rules as defined above
- All TGDs

$$\bigwedge_i A_i \wedge \bigwedge_j R_{q_j} \rightarrow \mathit{Goal}(\mathbf{x})$$

such that $\bigwedge A_i \wedge \bigwedge_j q_j$ entails $Q(\mathbf{x})$

We claim that for any $\mathbf{c} \in \mathfrak{A}$, $\mathit{Goal}(\mathbf{c})$ is entailed by $\mathfrak{A} \wedge D_Q$ if and only if Q is entailed by $\mathfrak{A} \wedge \Sigma$. In one direction, if $\mathit{Goal}(\mathbf{c})$ is entailed by $\mathfrak{A} \wedge D_Q$, then there is a single goal rule that derives this. Thus we have derived facts $A_i(\mathbf{c}_i)$ using derived full guarded TGDs and we have derived facts $R_{q_j}(\mathbf{c}_j)$ using derived subquery rules. We know that $A_i(\mathbf{c}_i)$ must be entailed by $\mathfrak{A} \wedge \Sigma$ by definition of the derived full guarded TGDs, and that $q_j(\mathbf{c}_j)$ is entailed by definition of the derived subquery rules. Thus by the definition of the goal rules $Q(\mathbf{c})$ is entailed.

In the other direction, if $Q(\mathbf{c})$ is entailed by $\mathfrak{A} \wedge \Sigma$, then $Q(\mathbf{c})$ holds in $Chase_\Sigma(\mathfrak{A})$. We thus have a homomorphism h and witness $h_Q = A_0 \cup \bigcup_{i \leq n} A_i$, and sets G_i above for this.

By our prior results, A_0 is entailed by the full guarded rules and $\mathfrak{A}$. Letting $q_j(\mathbf{c}_j)$ be obtained from F_j by turning each element outside of $\mathfrak{A}$ into an existentially quantified variable. By the definition of $Chase_\Sigma$, we have that q_j is entailed by G_j using Σ. Thus we have a corresponding derived subquery rule with R_{q_j} in the head generated by this. By our prior results, each fact in G_j is entailed by $\mathfrak{A}$ and the derived guarded Full TGDs. Combining these last two statements we see that $R_{q_j}(\mathbf{c}_j)$ is entailed from $\mathfrak{A}$ using the first two sets of rules. Since the

homomorphism of Q generates q_j and A_0, we have a goal rule $\bigwedge A_i \wedge q_j \to Q$. This completes the argument. ⊣

Frontier-guarded TGDs. We now generalize the result about rewriting certain answers to frontier-guarded TGDs. By a frontier-guarded rule in a Datalog program we mean a rule whose body contains an atomic formula that guards all variables that appear also in the head. A *Frontier-guarded Datalog program* is a Datalog program in which each rule is frontier-guarded.

THEOREM 5.6. *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 P such that the output of P on any structure $\mathfrak{A}$ is the same as the certain answers to Q on $\mathfrak{A}$.*

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 atomic formula is connected. This can be ensured by introducing new zero-ary predicates when needed.

Let k be the maximal number of variables in a TGD of Σ. For an answer-guarded conjunctive query $q(x_1 \dots x_j)$, let $R_q(x_1 \dots x_j)$ be a new relation symbol. For any number k, let FGTGD_k be all the frontier-guarded TGDs in the signature extending Σ with R_q for each answer-guarded q with at most k-variables. such that both the bodies and the heads have at most k atoms. Let Σ'_k 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 $\mathfrak{A}$, let $C_{\mathfrak{A}}$ be the set of elements of $\mathfrak{A}$ named by constant symbols.

We say that $\mathfrak{A}$ is *guardedly fact-saturated* (with respect to a set of TGDs Σ) if every possible fact over $\mathrm{adom}(\mathfrak{A}) \cup C_{\mathfrak{A}}$ entailed by the facts of $\mathfrak{A}$ together with Σ, *such that the values occurring in the fact form a guarded set in* $\mathfrak{A}$, belongs to $\mathfrak{A}$. In the absence of constants, guardedly fact-saturated means that no new fact over $\mathrm{adom}(\mathfrak{A})$ guarded by an existing ground atomic formula of $\mathfrak{A}$ is entailed.

We start with a lemma that is analogous to Lemma 5.4 but for frontier-guarded TGDs:

LEMMA 5.7. *A structure is fact-saturated with respect to a set of frontier-guarded TGDs Σ if and only if it is guardedly fact-saturated with respect to Σ.*

Note the difference from Lemma 5.4. There the sufficient condition for $\mathfrak{A}$ to be saturated was that $\mathfrak{A}$ was closed under *applying a saturation procedure to each guarded set in isolation.* Here our sufficient condition is that saturating $\mathfrak{A}$ in its entirety does not miss any fact guarded over $\mathfrak{A}$.

PROOF. Clearly, a guardedly fact-saturated structure is fact-saturated. In the other direction, if $\mathfrak{A}$ is guardedly-saturated, we show by induction on the formation of $Chase_\Sigma(\mathfrak{A})$ that each fact in $Chase_\Sigma(\mathfrak{A})$ over the elements in $\mathfrak{A}$ and the elements named by the constants must already be in $\mathfrak{A}$: in the inductions step we consider applying a rule σ that produced a fact F using only elements in $\mathfrak{A}$ and the elements named by the constants; the hypotheses must have included a guarded fact that contained all of these elements, and by induction all such

facts are already in $\mathfrak{A}$; now applying guarded-saturation we conclude that F is in $\mathfrak{A}$ as well. Since we have a structure extending $\mathfrak{A}$ that satisfies Σ and contains no new facts over $\mathfrak{A}$ and the elements named by constants, we can conclude that $\mathfrak{A}$ is saturated with respect to Σ. ⊣

We now claim the following "bounded base lemma" which differs from Lemma 5.4 and Lemma 5.7 by considering small subsets, but not guarded ones:

LEMMA 5.8. *Letting k be the maximal number of variables in a TGD of Σ, whenever a structure $\mathfrak{A}$ is not fact-saturated with respect to Σ'_k, then there is a subset X of the domain of $\mathfrak{A}$, with $|X| \leq k$, and such that the induced substructure $\mathfrak{A}_X$ is not fact-saturated with respect to Σ'_k*

A lemma similar to Lemma 5.8 occurs in Marnette's unpublished work [36] (Marnette's "bounded depth property").

We now prove Lemma 5.8. The lemma is proven by contraposition. Suppose that every substructure $\mathfrak{A}_X$ of $\mathfrak{A}$ with $|X| \leq k$ is fact-saturated. Let $Chase_{\Sigma'_k}(\mathfrak{A}_X)$ be the result of the chase with Σ'_k on $\mathfrak{A}_X$. Note that by the second property of the chase mentioned at the beginning of the section, all the facts over $\mathfrak{A}_X$ in $Chase_{\Sigma'_k}(\mathfrak{A}_X)$ are entailed by Σ and $\mathfrak{A}_X$, Since $\mathfrak{A}_X$ is fact-saturated, we deduce that $Chase_{\Sigma_k}(\mathfrak{A}_X)$ does not contain any additional facts over the set X plus the set of elements named by constant symbols. We now define $\mathfrak{B}$ to be the union of all these $Chase_{\Sigma'_k}(\mathfrak{A}_X)$. By construction, $\mathfrak{B}$ extends $\mathfrak{A}$ and contains no new guarded facts over $\text{adom}(\mathfrak{A})$ and the elements named by constant symbols. Further, note that $\text{adom}(Chase_{\Sigma_k}(\mathfrak{A}_X))$ for different X's overlap only on $\text{adom}(\mathfrak{A})$ and the elements named by constant symbols. Using Lemma 5.2 we can see that $\mathfrak{B}$ represents a squid-extension of $\mathfrak{A}$, with each tentacle contained in one of the $Chase_{\Sigma_k}(\mathfrak{A}_X)$.

We will now show that $\mathfrak{B} \models \Sigma'_k$. If we can show this, we have a contradiction of the fact that $\mathfrak{A}$ was not fact-saturated.

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 minimal in terms of number of atoms among dependencies in Σ'_k that are not satisfied in $\mathfrak{B}$. Now, consider any map $h : \{\mathbf{x}\} \rightarrow \text{adom}(\mathfrak{B})$ witnessing the fact that the TGD is false in $\mathfrak{B}$.

We claim that there are only two possibilities for the image of the body of the TGD:

CLAIM 5.9. *The h-image of the atoms of ϕ must be entirely contained in either $\mathfrak{A}$ or in one of the tentacles $\mathfrak{B}'$ of the squid decomposition of $\mathfrak{B}$, if we disregard facts that consist entirely of constants.*

PROOF OF CLAIM. Suppose this were not the case. For $J \subseteq^w \mathfrak{B}$, let $\text{edom}(J)$, the *extended domain of* J be the active domain of J unioned with the elements of $\mathfrak{B}$ named by constants.

The h-image of the frontier variables of ϕ is contained in the extended domain of some tentacle $\mathfrak{B}'$, since the image is guarded, and by assumption the guard cannot be in $\mathfrak{A}$. Consider the subquery ϕ_1 of ϕ formed by removing all atoms that are mapped by h into $\mathfrak{B}'$ and not into $\mathfrak{A}$. Note that at least one atom A_1 must map outside of $\mathfrak{B}'$ and outside of $\mathfrak{A}$, by the assumptions, hence A_1 must map into some other tentacle $\mathfrak{B}''$. Also, there is at least one atom A_2 in ϕ mapped out of A_1, even if the frontier is empty, by assumption. By connectedness of ϕ,

if we follow the path from A_1 to A_2 we will reach an atom that is not mapped to either $\mathfrak{B}''$ or $\mathfrak{B}'$. Hence ϕ_1 contains at least two atoms.

We modify ϕ_1 so that its free variables are those variables mapping onto the intersection of $\mathrm{adom}(\mathfrak{A})$ and $\mathrm{adom}(\mathfrak{B}')$, resulting in another answer-guarded query. Now consider the FGTGD

$$\forall \mathbf{x}\ \phi_1 \rightarrow R_\phi$$

Since ϕ_1 is smaller than ϕ, this FGTGD is smaller than σ. Further it is in Σ'_k. Thus by minimality of σ it must hold in $\mathfrak{B}$.

Now let ϕ_2 be formed from ϕ by replacing the subquery ϕ_1 by R_ϕ, and consider the FGTGD

$$\sigma' = \forall \mathbf{x}\ (\phi_2(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi(\mathbf{x},\mathbf{y}))$$

It is easy to see that $\sigma' \in \Sigma'_k$, since the use of definitions is conservative. The fact that ϕ_1 had at least two atoms means that σ' is also smaller than σ. Thus, again applying minimality of σ this dependency holds in $\mathfrak{B}$. But then σ must hold in $\mathfrak{B}$ as well, a contradiction. This completes the proof of the claim. ⊣

Using the claim, we complete the proof of Lemma 5.8. If the h-image of the atoms of ϕ is entirely contained in some tentacle, then this tentacle belongs to some $Chase_{\Sigma'_k}(\mathfrak{A}_X)$ which we know by construction satisfies Σ. 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 the atoms of ϕ is entirely contained in $\mathfrak{A}$, then we take the subset $\mathfrak{A}'$ of $\mathfrak{A}$ of size at most k that contains all elements in the h-image of the atoms of ϕ, and use the fact that $\mathfrak{A}' \subseteq^w \mathfrak{B} \models \Sigma$.

Either way, we reach a contradiction.

This completes the proof of Lemma 5.8.

We are now ready to prove Theorem 5.6.

Proof of Theorem 5.6. Let $\Sigma'_{Full,k}$ be all the full frontier-guarded rules in Σ'_k. We first claim that for any $\mathfrak{A}$, atomic conjunctive query $A(\mathbf{x})$ and $\mathbf{c} \in \mathfrak{A}$ if $\mathfrak{A}$ and Σ entail $A(\mathbf{c})$, then $\mathfrak{A}$ and $\Sigma'_{Full,k}$ proves $A(\mathbf{c})$.

To see this, let $\mathfrak{A}^+$ be formed by closing $\mathfrak{A}$ under $\Sigma'_{Full,k}$. We claim $\mathfrak{A}^+$ is fact-saturated for Σ. If not, then by Lemma 5.8 there is a subset $B = \{B_1(\mathbf{c}_1) \dots B_j(\mathbf{c}_j)\}$ of size at most k that is not fact-saturated for Σ'_k. But by Lemma 5.7 (which holds for all frontier-guarded TGDs, and hence in particular to Σ'_k), B is not guardedly fact-saturated with respect to Σ'_k. Hence there is a fact $F(\mathbf{c})$ with $\mathbf{c}$ contained in a guarded subset of B such that $F(\mathbf{c})$ is entailed by B under Σ'_k but is not in B. But then the rule $B_1(\mathbf{x}_1) \dots B_j(\mathbf{x}_j) \rightarrow F(\mathbf{x})$ is in $\Sigma'_{Full,k}$, contradicting the assumption of $\mathfrak{A}^+$.

We define the Datalog program by converting $\Sigma'_{Full,k}$ into Datalog rules in the same way as we did for $\Sigma_{FullGuarded}$ in Theorem 5.3. ⊣

Consequences for deciding FO-rewritability. In [10], 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 [10], 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 [10] with the algorithm in Theorem 5.6 to obtain decidability. In fact, we can obtain the result for general conjunctive queries, not just answer-guarded ones:

COROLLARY 5.10. *FO-rewritability of conjunctive queries Q under sets of frontier-guarded TGDs Σ is decidable.*

PROOF. In the case where Q is a boolean conjunctive query, we use the technique above: obtain a frontier-guarded Datalog rewriting and then checking whether it is equivalent to a first-order formula using the result of [10].

Now consider the case where Q is a general conjunctive query. We can form a boolean CQ Q^* by changing the free-variables $x_1 \dots x_n$ of Q to constants $c_1 \dots c_n$. Theorem 5.6 implies that we can decided whether the certain answers to Q^* with respect to Σ are first-order definable. But the certain answers of Q^* with respect to Σ are first-order definable if and only if the certain answers to Q with respect to Σ are first-order definable: we can change a first-order definition of one to a first-order definition of the other by just replacing constants with free variables or vice versa. ⊣

§6. Related Work and Conclusions. We have investigated various problems that involve rewriting of GNFO formulas in different contexts, building on the decidability results for GNFO established in [8], and the complexity results for open- and closed-world querying established in [10].

Although we did not discuss the exact complexity of the decision problem for FO-rewritability of certain answers under frontier-guarded TGDs, we believe that an elementary bound can be extracted from analysis of [10]. Prior to that work, we know of no result on deciding first-order rewritability in the setting of general relational languages. However, for description logics, some positive results were obtained by Bienvenue, Lutz, and Wolter [14]. In [15], it was shown that certain answers w.r.t. a GNFO sentence can be expressed in frontier-guarded *disjunctive* Datalog. Unlike our result for frontier-guarded TGDs, however, this characterization is not known to imply decidability of first-order rewritability or even Datalog-rewritability. Weakly-guarded TGDs [16] are another member of the Datalog$^{\pm}$ family that has been shown to have attractive properties for the complexity of open-world query answering. One can show, however, that they do not share with FGTGD's the decidability of FO-rewritability.

Here we have considered syntactically capturing restrictions of GNFO, and show that the corresponding target classes for rewritings are natural. For description logics, some characterizations with a similar flavor have been proven by Lutz, Piro, and Wolter [34]. The Unary Negation Fragment is another fragment of FO containing many modal and description logics which possesses CIP and (hence) PBDP [48]. Interpolation and implicit definability have also been heavily studied within the description logic community [35, 47]. Unfortunately, having BDP or CIP for a stronger logic does not imply it for a weaker logic, or vice versa.

Finally, recently, in follow-up work [12], tight bounds on the complexity were found for a number of problems considered here, including interpolation and preservation results.

Acknowledgements. This paper is an expanded version of the conference abstract [7]. Benedikt was supported by EPSRC grant EP/H017690/1, and ten Cate was supported by NSF Grants IIS-0905276 IIS-1217869. Bárány's work was done while affiliated with TU Darmstadt.

The authors gratefully acknowledge their debt to Martin Otto for enlightening discussions. We want to thank Maarten Marx for helpful discussions and help in verifying the counterexamples of Section 4.

REFERENCES

[1] Serge Abiteboul, Richard Hull, and Victor Vianu, ***Foundations of Databases***, Addison-Wesley, 1995.

[2] Hajnal Andréka, Johan van Benthem, and István Németi, *Modal languages and bounded fragments of predicate logic*, ***Journal of Philosophical Logic***, vol. 27 (1998), pp. 217–274.

[3] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider (editors), ***The description logic handbook***, Cambridge University Press, 2003.

[4] Jean-François Baget, Marie-Laure Mugnier, Sebastian Rudolph, and Michaël Thomazo, *Complexity Boundaries for Generalized Guarded Existential Rules*, 2011, Research Report LIRMM 11006.

[5] Jean-François Baget, Michel Leclère, and Marie-Laure Mugnier, *Walking the Decidability Line for Rules with Existential Variables*, ***Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*** (Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors), AAAI Press, 2010.

[6] Jean-François Baget, Marie-Laure Mugnier, Sebastian Rudolph, and Michaël Thomazo, *Walking the Complexity Lines for Generalized Guarded Existential Rules*, ***IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*** (Toby Walsh, editor), IJCAI/AAAI, 2011, pp. 712–717.

[7] Vince Bárány, Michael Benedikt, and Balder ten Cate, *Rewriting guarded negation queries*, ***Mathematical Foundations of Computer Science 2013 - 38th International Symposium, MFCS 2013, Klosterneuburg, Austria, August 26-30, 2013. Proceedings*** (Krishnendu Chatterjee and Jirí Sgall, editors), Lecture Notes in Computer Science, vol. 8087, Springer, 2013, pp. 98–110.

[8] Vince Bárány, Balder Ten Cate, and Luc Segoufin, *Guarded negation*, ***Journal of the ACM***, vol. 62 (2015), no. 3, pp. 22:1–22:26.

[9] Vince Bárány, Georg Gottlob, and Martin Otto, *Querying the guarded fragment*, ***Logical Methods in Computer Science***, vol. 10 (2014), no. 2.

[10] Vince Bárány, Balder ten Cate, and Martin Otto, *Queries with guarded negation*, ***Procedings of the VLDB Endowment***, vol. 5 (2012), no. 11, pp. 1328–1339.

[11] Vince Bárány, Balder ten Cate, and Luc Segoufin, *Guarded negation*, ***Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, proceedings, part II*** (Luca Aceto, Monika Henzinger, and Jirí Sgall, editors), Lecture Notes in Computer Science, vol. 6756, Springer, 2011, pp. 356–367.

[12] Michael Benedikt, Balder ten Cate, and Michael Vanden Boom, *Effective interpolation and preservation in guarded logics*, ***Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*** (Thomas A. Henzinger and Dale Miller, editors), ACM, 2014, pp. 13:1–13:10.

[13] E. W. Beth, *On Padoa's method in the theory of definitions*, ***Indagationes Mathematicae***, vol. 15 (1953), pp. 330 – 339.

[14] Meghyn Bienvenu, Carsten Lutz, and Frank Wolter, *Deciding fo-rewritability in EL*, ***Proceedings of the 2012 International Workshop on Description Logics, DL-2012, Rome, Italy, June 7-10, 2012*** (Yevgeny Kazakov, Domenico Lembo, and Frank Wolter, editors), CEUR Workshop Proceedings, vol. 846, CEUR-WS.org, 2012.

[15] Meghyn Bienvenu, Balder ten Cate, Carsten Lutz, and Frank Wolter, *Ontology-based Data Access: A Study Through Disjunctive Datalog, CSP, and MMSNP*, ***Proceedings of the 32nd Symposium on Principles of Database Systems*** (New York, NY, USA), PODS '13, ACM, 2013, pp. 213–224.

[16] Andrea Calì, Georg Gottlob, and Michael Kifer, *Taming the infinite chase: Query answering under expressive relational constraints*, ***Principles of knowledge representation and reasoning: Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008*** (Gerhard Brewka and Jérôme Lang, editors), AAAI Press, 2008, pp. 70–80.

[17] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz, *A general datalog-based framework for tractable query answering over ontologies*, ***Proceedings of the Twenty-Eigth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2009, June 19 - July 1, 2009, Providence, Rhode Island, USA*** (Jan Paredaens and Jianwen Su, editors), ACM, 2009, pp. 77–86.

[18] A.K. Chandra and P.M. Merlin, *Optimal implementation of conjunctive queries in relational databases*, ***9th ACM Symposium on Theory of Computing***, 1977, pp. 77–90.

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

[20] William Craig, *Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory*, this Journal, vol. 22 (1957), no. 3, pp. 269–285.

[21] Heinz-Dieter Ebbinghaus and Jörg Flum, ***Finite Model Theory***, Springer-Verlag, 1999.

[22] Ronald Fagin, *Horn clauses and database dependencies*, ***Journal of the ACM***, vol. 29 (1982), no. 4, pp. 952–985.

[23] Ronald Fagin, Phokion G. Kolaitis, Renee J. Miller, and Lucian Popa, *Data Exchange: Semantics and Query Answering*, ***Theoretical Computer Science***, vol. 336 (2005), no. 1, pp. 89–124.

[24] Jörg Flum, Markus Frick, and Martin Grohe, *Query evaluation via tree-decompositions*, ***Journal of the ACM***, vol. 49 (2002), no. 6, pp. 716–752.

[25] Harvey Friedman, *The complexity of explicit definitions*, ***Advances in Mathematics***, vol. 20 (1976), no. 1, pp. 18–29.

[26] Tomasz Gogacz and Jerzy Marcinkowski, *The hunt for a red spider: Conjunctive query determinacy is undecidable*, ***Proceedings of the 2015 30th annual acm/ieee symposium on logic in computer science (lics)*** (Washington, DC, USA), IEEE Computer Society, 2015, pp. 281–292.

[27] ———, *Red spider meets a rainworm: Conjunctive query finite determinacy is undecidable*, ***Proceedings of the 35th acm sigmod-sigact-sigai symposium on principles of database systems*** (New York, NY, USA), PODS '16, ACM, 2016, pp. 121–134.

[28] Georg Gottlob, Nicole Leone, and Francesco Scarcello, *Robbers, marshals, and guards: game theoretic and logical characterizations of hypertree width*, ***Journal of Computer and Systems Sciences***, vol. 66 (2003), no. 4, pp. 775–808.

[29] Erich Grädel, *On the restraining power of guards*, ***Journal of Symbolic Logic***, vol. 64 (1999), no. 4, pp. 1719–1742.

[30] Erich Grädel and Martin Otto, *The freedoms of (guarded) bisimulation*, ***Johan van Benthem on Logic and Information Dynamics*** (Alexandru Baltag and Sonja Smets, editors), Outstanding Contributions to Logic, vol. 5, Springer, 2014, pp. 3–31.

[31] Eva Hoogland, *Definability and interpolation: model-theoretic investigations*, ***Ph.D. thesis***, University of Amsterdam, 2000.

[32] Eva Hoogland, Maarten Marx, and Martin Otto, *Beth definability for the guarded fragment*, ***Logic Programming and Automated Reasoning, 6th International Conference, LPAR'99, Tbilisi, Georgia, September 6-10, 1999, Proceedings*** (Harald Ganzinger, David A. McAllester, and Andrei Voronkov, editors), Lecture Notes in Computer Science, vol. 1705, Springer, 1999, pp. 273–285.

[33] Maurizio Lenzerini, *Data Integration: A Theoretical Perspective*, ***Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*** (New York, NY, USA), PODS '02, ACM, 2002, pp. 233–246.

[34] Carsten Lutz, Robert Piro, and Frank Wolter, *Description Logic TBoxes: Model-Theoretic Characterizations and Rewritability*, ***IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*** (Toby Walsh, editor), IJCAI/AAAI, 2011, pp. 983–988.

[35] Carsten Lutz and Frank Wolter, *Foundations for uniform interpolation and forgetting in expressive description logics*, ***IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*** (Toby Walsh, editor), IJCAI/AAAI, 2011, pp. 989–995.

[36] Bruno Marnette, *Resolution and Datalog Rewriting Under Value Invention and Equality Constraints*, ***Technical report***, 2011, http://arxiv.org/abs/1212.0254.

[37] Maarten Marx, *Queries determined by views: pack your views*, ***Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 11-13, 2007, Beijing, China*** (Leonid Libkin, editor), ACM, 2007, pp. 23–30.

[38] Maarten Marx and Yde Venema, ***Multidimensional Modal Logic***, Kluwer, 1997.

[39] Alan Nash, Luc Segoufin, and Victor Vianu, *Views and queries: Determinacy and rewriting*, ***ACM Transactions on Database Systems***, vol. 35 (2010), no. 3, pp. 21:1–21:41.

[40] A. Onet, *The chase procedure and its applications in data exchange*, ***Deis***, 2013, pp. 1–37.

[41] M. Otto, *Expressive completeness through logically tractable models*, ***Annals of Pure and Applied Logic***, (2013), pp. 1418–1453.

[42] Martin Otto, *Modal and guarded characterisation theorems over finite transition systems*, ***Annals of Pure and Applied Logic***, vol. 130 (2004), pp. 173–205.

[43] ———, *Highly acyclic groups, hypergraph covers and the guarded fragment*, ***Journal of the ACM***, vol. 59 (2012), no. 1, pp. 5:1–5:40.

[44] Eric Rosen, *Modal logic over finite structures*, ***Journal of Logic Language and Information***, vol. 6 (1997), no. 4, pp. 427–439.

[45] Benjamin Rossman, *Homomorphism preservation theorems*, ***Journal of the ACM***, vol. 55 (2008), no. 3, pp. 15:1–15:53.

[46] Balder ten Cate, *Interpolation for extended modal languages*, ***Journal of Symbolic Logic***, vol. 70 (2005), no. 1, pp. 223–234.

[47] Balder ten Cate, Enrico Franconi, and Inanç Seylan, *Beth definability in expressive description logics*, ***IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*** (Toby Walsh, editor), IJCAI/AAAI, 2011, pp. 1099–1106.

[48] Balder ten Cate and Luc Segoufin, *Unary negation*, ***28th International Symposium on Theoretical Aspects of Computer Science, STACS 2011, March 10-12, 2011, Dortmund, Germany*** (Thomas Schwentick and Christoph Dürr, editors), LIPIcs, vol. 9, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011, pp. 344–355.

[49] Johan van Benthem, ***Modal logic and classical logic***, Bibliopolis, Napoli, 1985.

[50] Michalis Yannakakis, *Algorithms for Acyclic Database Schemes*, ***Proceedings of the Seventh International Conference on Very Large Data Bases - Volume 7***, VLDB '81, VLDB Endowment, 1981, pp. 82–94.

GOOGLE INC., MOUNTAIN VIEW, CA

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF OXFORD

GOOGLE INC., MOUNTAIN VIEW, CA
and
DEPARTMENT OF COMPUTER SCIENCE, UC-SANTA CRUZ