Query Answering with Transitive and Linear-Ordered Data

Antoine Amarilli, Michael Benedikt, Pierre Bourhis, Michael Vanden Boom

Query Answering with Transitive and Linear-Ordered Data

Antoine Amarilli
LTCI, CNRS, Télécom ParisTech, Université Paris-Saclay

Michael Benedikt
University of Oxford

Pierre Bourhis
CNRS CRIStAL, Université Lille 1, INRIA Lille

Michael Vanden Boom
University of Oxford

Abstract

We consider entailment problems involving powerful constraint languages such as *guarded existential rules*, in which additional semantic restrictions are put on a set of distinguished relations. We consider restricting a relation to be transitive, restricting a relation to be the transitive closure of another relation, and restricting a relation to be a linear order. We give some natural generalizations of guardedness that allow inference to be decidable in each case, and isolate the complexity of the corresponding decision problems. Finally we show that slight changes in our conditions lead to undecidability.

1 Introduction

The *query answering problem* (or certain answer problem), abbreviated here as QA, is a fundamental reasoning problem in both knowledge representation and databases. It asks whether a query (e.g. given by an existentially-quantified conjunction of atoms) is entailed by a set of constraints and a set of facts. A common class of constraints used for QA are the *existential rules*, also known as *tuple generating dependencies* (TGDs). Although query answering is known to be undecidable for general TGDs, there are a number of subclasses that admit decidable QA, such as those based on *guardedness*. For instance, *guarded* TGDs require all variables in the body of the dependency to appear in a single body atom (the *guard*). *Frontier-guarded* TGDs (FGTGDs) relax this condition and require only that some guard atom contains the variables that occur in both head and body [Baget *et al.*, 2011]. This includes standard SQL referential constraints as well as important constraint classes (e.g. role inclusions) arising in knowledge representation. Guarded existential rules can be generalized to *guarded logics* that allow disjunction and negation and still enjoy decidable QA, e.g. the guarded fragment of first-order logic (GF) [Andréka *et al.*, 1998] and the Guarded Negation Fragment (GNF) [Bárány *et al.*, 2011].

A key challenge is to extend these results to capture additional semantics of the relations. For example, the property that a binary relation is *transitive* cannot be expressed directly in guarded logics, and neither can the property that one relation is the *transitive closure* of another. Going beyond transitivity, one cannot express that a binary relation is a *linear order*. Since ordered data is common in applications, this means that a key part of data semantics is being lost.

There has been extensive work on decidability results for guarded logics thus extended with semantic restrictions. We first review known results for the *satisfiability problem*.

Ganzinger *et al.* [1999] showed that satisfiability is not decidable for GF when two relations are restricted to be transitive, even on *arity-two* signatures (i.e. with only unary and binary relations). For linear orders, [Kieronski, 2011] showed GF is undecidable when three relations are restricted to be (non-strict) linear orders, even with only two variables (so on arity-two signatures). Otto [2001] showed satisfiability is decidable for two-variable logic with one relation restricted to be a linear order. For transitive relations, one way to regain decidability for GF satisfiability was shown by Szwast and Tendera [2004]: allow transitive relations *only* in guards.

We now turn to the QA problem. Gottlob *et al.* [2013] showed that query answering for GF with transitive relations only in guards is undecidable, even on arity-two signatures. Baget *et al.* [2015] studied QA with respect to a collection of linear TGDs (those with only a single atom in the body and the head). They showed that the query answering problem is decidable with such TGDs and transitive relations, if the signature is binary or if other additional restrictions are obeyed.

The case of TGDs mentioning relations with a restricted interpretation has been studied in the database community mainly in the setting of acyclic schemas, such as those that map source data to target data. Transitivity restrictions have not been studied, but there has been work on inequalities [Abiteboul and Duschka, 1998] and TGDs with arithmetic [Afrati *et al.*, 2008]. Due to the acyclicity assumptions, QA is still decidable, and has data complexity in CoNP. The fact that the data complexity can be CoNP-hard is shown in [Abiteboul and Duschka, 1998], while polynomial cases have been isolated in [Abiteboul and Duschka, 1998] (in the presence of inequalities) and [Afrati *et al.*, 2008] (in the presence of arithmetic).

Transitivity has also been studied in description logics, where the signature contains unary relations (concepts) and binary relations (roles). In this arity-two context, QA is decidable for many description logics featuring expressive operators as well as transitivity, such as $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$ [Cal-

vanese *et al.*, 2009], Horn-$\mathcal{SROIQ}$ [Ortiz *et al.*, 2011], or regular-$\mathcal{EL}^{++}$ [Krötzsch and Rudolph, 2007], but they often restrict the interaction between transitivity and some features such as role inclusions and Boolean role combinations. QA becomes undecidable for more expressive description logics with transitivity such as $\mathcal{ALCOIF}^*$ [Ortiz *et al.*, 2010] and $\mathcal{ZOIQ}$ [Ortiz de la Fuente, 2010], and the problem is open for $\mathcal{SROIQ}$ and $\mathcal{SHOIQ}$ [Ortiz and Šimkus, 2012].

The main contribution of this work is to introduce a broad class of constraints over arbitrary-arity vocabularies where query answering is decidable when additional semantics are imposed on some *distinguished relations*. We show that transitivity restrictions can be handled in guarded and frontier-guarded constraints, as long as these distinguished relations are *not* used as guards — we call this new kind of restriction *base-guardedness* (and similarly, we extend frontier-guarded to "base-frontier-guarded" and so forth). The base-guarded restriction is orthogonal to the prior decidable cases such as transitive guards [Szwast and Tendera, 2004] or linear rules [Baget *et al.*, 2015].

On the one hand, we show that the condition allows us to define very expressive and flexible decidable logics, capable of expressing not only guarded existential rules, but guarded rules with negation and disjunction in the head. They can express not only integrity constraints but also conjunctive queries and their negations. On the other hand, a by-product of our results is new query answering schemes for some previously-studied classes of guarded existential rules with extra semantic restrictions. For example, our base-frontier-guarded constraints encompass all *frontier-one TGDs* (where at most one variable is shared between the body and head) [Baget *et al.*, 2009]. Hence, our results imply that QA is decidable with transitive closure and frontier-one constraints, which answers a question of [Baget *et al.*, 2015]. Our results even extend to frontier-one TGDs with distinguished relations that are required to be the transitive closure of other relations.

Our results are shown by arguing that it is enough to consider entailment over "tree-like" sets of facts. By representing the set of witness representations as a tree automaton, we derive upper bounds for the combined complexity of the problem. The sufficiency of tree-like examples also enable a refined analysis of *data complexity* (when the query and constraints are fixed). Further, we use a set of coding techniques to show matching lower bounds within our fragment. We also show that loosening our conditions leads to undecidability.

Finally, we solve the QA problem when the distinguished relations are *linear orders*. We show that it is undecidable even assuming base-guardedness, so we introduce a stronger condition called *base-coveredness*: not only are distinguished relations never used as guards, they are always *covered* by a non-distinguished atom. Our decidability technique works by "compiling away" linear order restrictions, obtaining an entailment problem without any special restrictions. The correctness proof for our reduction to classical QA again relies on the ability to restrict reasoning to sets of facts with tree-like representations. To our knowledge, these are the first decidability results for the QA problem with linear orders. Again, we give tight complexity bounds, and show that weakening the base-coveredness condition leads to undecidability.

2 Preliminaries

We work on a *relational signature* σ where each relation $R \in \sigma$ has an associated *arity* (written $\mathsf{arity}(R)$); we write $\mathsf{arity}(\sigma) := \max_{R\in\sigma} \mathsf{arity}(R)$. A *fact* $R(\vec{a})$ (or R-*fact*) consists of a relation $R \in \sigma$ and domain elements $\vec{a}$, with $|\vec{a}| = \mathsf{arity}(R)$. We denote a (finite or infinite) set of facts over σ by $\mathcal{F}$. We write $\mathsf{elems}(\mathcal{F})$ for the set of elements mentioned in the facts in $\mathcal{F}$.

We consider *constraints* and *queries* given in fragments of first-order logic (FO). For simplicity, we disallow constants in constraints and queries, although our results extend with them. Given a set of facts $\mathcal{F}$ and a sentence φ in FO, we talk of $\mathcal{F}$ *satisfying* φ in the usual way. The *size* of φ, written $|\varphi|$, is defined to be the number of symbols in φ.

The queries that we will use are *conjunctive queries* (CQ), namely, existentially quantified conjunction of atoms, which we restrict for simplicity to be Boolean. We also allow *unions of conjunctive queries* (UCQs), namely, disjunctions of CQs.

Problems considered. Given a *finite* set of facts $\mathcal{F}_0$, constraints Σ and query Q (given as FO sentences), we say that $\mathcal{F}_0$ and Σ *entail* Q if for every (possibly infinite) $\mathcal{F} \supseteq \mathcal{F}_0$ satisfying Σ, $\mathcal{F}$ satisfies Q. This amounts to asking whether $\mathcal{F}_0 \wedge \Sigma \wedge \neg Q$ is satisfiable (by a possibly infinite set of facts). We write $\mathsf{QA}(\mathcal{F}_0, \Sigma, Q)$ for this decision problem, called the *query answering* problem.

In this paper, we study the QA problem when imposing semantic constraints on some *distinguished* relations. We thus split the signature as $\sigma := \sigma_{\mathcal{B}} \sqcup \sigma_{\mathcal{D}}$, where $\sigma_{\mathcal{B}}$ is the *base signature* (its relations are the *base* relations), and $\sigma_{\mathcal{D}}$ is the *distinguished* signature. All distinguished relations are required to be binary, and they will be assigned special semantics. We study three kinds of special semantics.

We say $\mathcal{F}_0, \Sigma$ *entails* Q *over transitive relations*, and write $\mathsf{QAtr}(\mathcal{F}_0, \Sigma, Q)$ for the corresponding problem, if $\mathcal{F}_0 \wedge \Sigma \wedge \neg Q$ is satisfied by some set of facts $\mathcal{F}$ where each distinguished relation R_i^+ in $\sigma_{\mathcal{D}}$ is required to be *transitive*[1] in $\mathcal{F}$.

We say $\mathcal{F}_0, \Sigma$ *entails* Q *over transitive closure*, and write $\mathsf{QAtc}(\mathcal{F}_0, \Sigma, Q)$ for this problem, if the same holds on some $\mathcal{F}$ where each relation R_i^+ of $\sigma_{\mathcal{D}}$ is interpreted as the transitive closure of a corresponding binary base relation $R_i \in \sigma_{\mathcal{B}}$.

We say $\mathcal{F}_0, \Sigma$ *entails* Q *over linear orders*, and write $\mathsf{QAlin}(\mathcal{F}_0, \Sigma, Q)$, if the same holds on some $\mathcal{F}$ where each relation $<_i \in \sigma_{\mathcal{D}}$ is required to be a strict linear order on the elements of $\mathcal{F}$.

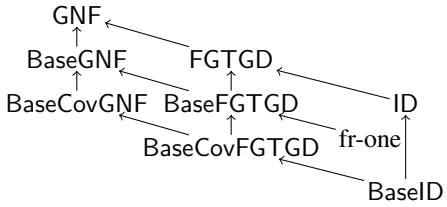
We now define the constraint languages (which are all fragments of FO) for which we study these QA problems.

Dependencies. The first constraint languages we study are restricted classes of *tuple-generating dependencies* (TGDs). A TGD is a first-order sentence τ of the form $\forall \vec{x}\, (\bigwedge_i \gamma_i(\vec{x}) \rightarrow$

[1] Note that we work with *transitive* relations, which may not be *reflexive*, unlike, e.g., R^* roles in $\mathcal{ZOIQ}$ description logics [Calvanese *et al.*, 2009]. However, our results can be adapted to the case of reflexive and transitive predicates (and reflexive transitive closure).

Fragment	QAtr		QAtc		QAlin	
	data	**combin.**	**data**	**combin.**	**data**	**combin.**
BaseGNF	coNP-c	2EXP-c	coNP-c	2EXP-c	undecidable	
BaseCovGNF	coNP-c	2EXP-c	coNP-c	2EXP-c	coNP-c	2EXP-c
BaseFGTGD	in coNP	2EXP-c	coNP-c	2EXP-c	undecidable	
BaseCovFGTGD	P-c	2EXP-c	coNP-c	2EXP-c	coNP-c	2EXP-c

(a) Summary of QA results (for base-covered fragments, queries are also base-covered)



(b) Taxonomy of fragments

$\exists \vec{y}\ \bigwedge_i \rho_i(\vec{x}, \vec{y})\ )$ where $\bigwedge_i \gamma_i$ and $\bigwedge_i \rho_i$ are conjunctions of atoms respectively called the *body* and *head* of τ.

We will be interested in TGDs that are *guarded* in various ways. A *guard* for $\vec{x}$ is an atom from σ using every variable in $\vec{x}$. We will be particularly interested in *base-guards*, which are guards coming from the base relations $\sigma_{\mathcal{B}}$.

A *frontier-guarded TGD* (FGTGD) is a TGD whose body contains a guard for the *frontier variables* — variables that occur in both head and body. It is a *base frontier-guarded TGD* (BaseFGTGD) if there is a base atom including all the frontier variables. We allow equality atoms $x = x$ to be guards, so BaseFGTGD subsumes *frontier-one TGDs*, which have one frontier variable. Frontier-guarded and frontier-one TGDs have been shown to have an attractive combination of expressivity and computational properties [Baget *et al.*, 2011].

We also introduce the more restricted class of *base-covered frontier-guarded TGDs* (BaseCovFGTGD): they are the BaseFGTGDs where, for every $\sigma_{\mathcal{D}}$-atom in the body, there is a base atom in the body containing its variables (a *base guard for the atom*). Note that each $\sigma_{\mathcal{D}}$-atom may have a different base guard.

An important special case of frontier-guarded TGDs for applications are *inclusion dependencies* (ID). An ID is a FGTGD where the body and head contain a single atom, and no variable occurs twice in the same atom. A *base inclusion dependency* BaseID is an ID where the body atom is in $\sigma_{\mathcal{B}}$, so the body atom serves as the base-guard for the frontier variables, while the constraint is trivially base-covered.

Guarded logics. Moving beyond TGDs, the second kind of constraint that we study are *guarded logics*.

The *guarded negation fragment* (GNF) is the fragment of FO which contains all atoms, and is closed under conjunction, disjunction, existential quantification, and the following form of negation: for any GNF formula $\varphi(\vec{x})$ and atom $A(\vec{x}, \vec{y})$ with free variables exactly as indicated, the formula $A(\vec{x}, \vec{y}) \wedge \neg\varphi(\vec{x})$ is in GNF. That is, existential quantification may be unguarded, but the free variables in any negated subformula must be guarded; universal quantification must be expressed with existential quantification and negation. GNF can express all FGTGDs, as well as non-TGD constraints and UCQs. For instance, as it allows disjunction, GNF can express *disjunctive inclusion dependencies*, DIDs, which generalize IDs: their body is a single atom with no repeated variables, and their head is a disjunction of atoms with no repeated variables.

We introduce the *base-guarded negation fragment* BaseGNF over σ: it is defined like GNF, but requires *base guards* instead of guards. The *base-covered guarded negation fragment* BaseCovGNF over σ consists of BaseGNF formulas such that every $\sigma_{\mathcal{D}}$-atom A that appears negatively (i.e., under the scope of an odd number of negations) appears conjoined with a base guard — i.e., a $\sigma_{\mathcal{B}}$-atom containing all variables of A. This technical condition is designed to generalize BaseCovFGTGDs. Indeed, a BaseCovFGTGD of the form $\forall\vec{x}(\bigwedge \gamma_i \rightarrow \exists\vec{y} \bigwedge \rho_i)$ can be written in BaseCovGNF as $\neg\exists\vec{x}(\bigwedge \gamma_i \wedge \neg\exists\vec{y} \bigwedge \rho_i)$.

We call a CQ Q *base-covered* if each $\sigma_{\mathcal{D}}$-atom in Q has a $\sigma_{\mathcal{B}}$-atom of Q containing its variables. This is the same as saying that $\neg Q$ is in BaseCovGNF. A UCQ is base-covered if each disjunct is.

Examples. We conclude the preliminaries by giving a few examples. Consider a signature with a binary base relation B, a ternary base relation C, and a distinguished relation R^+.

- $\forall xyz\big((R^+(x,y) \wedge R^+(y,z)) \rightarrow R^+(x,z)\big)$ is a TGD, but is not a FGTGD since the frontier variables x, z are not guarded. It cannot even be expressed in GNF.
- $\forall xy\big(R^+(x,y) \rightarrow B(x,y)\big)$ is an ID, hence a FGTGD. It is not a BaseID or even in BaseGNF, since the frontier variables are not base-guarded.
- $\forall xyz\big((B(z,x) \wedge R^+(x,y) \wedge R^+(y,z)) \rightarrow R^+(x,z)\big)$ is a BaseFGTGD. It is not a BaseCovFGTGD since there are no base atoms in the body to cover x, y and y, z.
- $\exists wxyz\big(R^+(w,x) \wedge R^+(x,y) \wedge R^+(y,z) \wedge R^+(z,w) \wedge C(w,x,y) \wedge C(y,z,w)\big)$ is a base-covered CQ.
- $\exists xy\big(B(x,y) \wedge \neg(R^+(x,y) \wedge R^+(y,x)) \wedge (R^+(x,y) \vee R^+(y,x))\big)$ cannot be rewritten as a TGD. But it is in BaseCovGNF.

Our main results are summarized in Table a, and the languages that we study are illustrated in Figure b. In particular, QAtr and QAtc are decidable for BaseGNF. This includes base-frontier-guarded rules, which allow one to use a transitive relation such as "part-of" (or even its transitive closure) whenever only one variable is to be exported to the head. This latter condition holds in the translations of many classical description logics. Our results also imply that QAlin is decidable for BaseCovGNF, which allows constraints that arise from data integration and data exchange over attributes with linear orders — e.g. views defined by selecting rows of a table where some inequality involving the attributes is satisfied.

3 Decidability results for transitivity

We first consider QAtc, where $\sigma_{\mathcal{B}}$ includes binary relations $R_1, \ldots, R_n$, and $\sigma_{\mathcal{D}}$ consists of binary relations $R_1^+, \ldots, R_n^+$ such that R_i^+ is the transitive closure of R_i.

Theorem 1. *We can decide* $\mathsf{QAtc}(\mathcal{F}_0, \Sigma, Q)$ *in* 2EXPTIME*, where* $\mathcal{F}_0$ *ranges over finite sets of facts,* Σ *over* BaseGNF *constraints, and* Q *over UCQs. In particular, this holds when* Σ *consists of* BaseFGTGD*s.*

In order to prove Theorem 1, we give a decision procedure to determine whether $\mathcal{F}_0 \wedge \Sigma \wedge \neg Q$ is satisfiable, when R_i^+ is interpreted as the transitive closure of R_i. When $\Sigma \in$ BaseGNF and Q is a Boolean UCQ, then $\Sigma \wedge \neg Q$ is in BaseGNF. So it suffices to show that BaseGNF satisfiability is decidable, when properly interpreting R_i^+.

As mentioned in the introduction, our proofs rely heavily on the fact that in query answering problems for these constraint languages, one can restrict to sets of facts that have a "tree-like" structure. We now make this notion precise. A *tree decomposition* of $\mathcal{F}$ consists of a tree (T, Child) and a labelling function λ associating each node of the tree T to a set of facts of $\mathcal{F}$, called the *bag* of that node, that satisfies the following conditions: (i) each fact of $\mathcal{F}$ must be in the image of λ; (ii) for each element $e \in \mathsf{elems}(\mathcal{F})$, the set of nodes whose bag uses e is a connected subset of T. It is $\mathcal{F}_0$*-rooted* if the root node is associated with $\mathcal{F}_0$. It has *width* $k-1$ if each bag other than the root mentions at most k elements.

For a number k, a σ sentence φ is said to have *transitive-closure friendly* k*-tree-like witnesses* if: for every finite set of facts $\mathcal{F}_0$, if there is an $\mathcal{F}$ extending $\mathcal{F}_0$ with additional $\sigma_{\mathcal{B}}$-facts such that $\mathcal{F}$ satisfies φ when each R^+ is interpreted as the transitive closure of R, then there is such an $\mathcal{F}$ that has an $\mathcal{F}_0$-rooted $(k-1)$-width tree decomposition. We can show that BaseGNF sentences have this kind of k-tree-like witness for an easily computable k. The proof uses a standard technique, involving an unravelling based on "guarded negation bisimulation" [Bárány *et al.*, 2011]:

Proposition 1. *Every sentence* φ *in* BaseGNF *has transitive-closure friendly* k*-tree-like witnesses, where* $k \leq |\varphi|$*.*

Here k can be taken to be the "width" of φ [Bárány *et al.*, 2011], which is roughly the maximum number of free variables in any subformula. Hence, it suffices to test satisfiability for BaseGNF restricted to sets of facts with tree decompositions of width $|\varphi| - 1$. It is well known that sets of facts of bounded tree-width can be encoded as trees over a finite alphabet that depends only on the signature and the tree-width. This makes the problem amenable to tree automata techniques, since we can design a tree automaton that runs on representations of these tree decompositions and checks whether some sentence holds in the corresponding set of facts.

Theorem 2. *Let* φ *be a sentence in* BaseGNF*, and let* $\mathcal{F}_0$ *be a finite set of facts. We can construct in* 2EXPTIME *a 2-way alternating parity tree automaton* $\mathcal{A}_{\varphi,\mathcal{F}_0}$ *such that*

$$\mathcal{F}_0 \wedge \varphi \text{ is satisfiable} \quad \text{iff} \quad L(\mathcal{A}_{\varphi,\mathcal{F}_0}) \neq \emptyset$$

when each $R_i^+ \in \sigma_{\mathcal{D}}$ *is interpreted as the transitive closure of* $R_i \in \sigma_{\mathcal{B}}$*. The number of states of* $\mathcal{A}_{\varphi,\mathcal{F}_0}$ *is exponential in* $|\varphi| \cdot |\mathcal{F}_0|$ *and the number of priorities is linear in* $|\varphi|$*.*

The construction can be viewed as an extension of [Calvanese *et al.*, 2005], and incorporates ideas from automata for guarded logics (see, e.g., [Grädel and Walukiewicz, 1999]).

Because 2-way tree automata emptiness is decidable in time exponential in the number of states and priorities [Vardi, 1998], this yields the 2EXPTIME bound for Theorem 1.

Consequences for QAtr**.** We can derive results for QAtr by observing that the QAtc problem subsumes it: to enforce that $R^+ \in \sigma_{\mathcal{D}}$ is transitive, simply interpret it as the transitive closure of a relation R that is never otherwise used. Hence:

Corollary 1. *We can decide* $\mathsf{QAtr}(\mathcal{F}_0, \Sigma, Q)$ *in* 2EXPTIME*, where* $\mathcal{F}_0$ *ranges over finite sets of facts,* Σ *over* BaseGNF *constraints (in particular,* BaseFGTGD*), and* Q *over UCQs.*

In particular, this result holds for *frontier-one TGDs* (those with a single frontier variable), as a single variable is always base-guarded. This answers a question of [Baget *et al.*, 2015].

Data complexity. Our results in Theorem 1 and Corollary 1 show upper bounds on the *combined complexity* of the QAtr and QAtc problems. We now turn to the complexity when the query and constraints are fixed but the initial set of facts varies — the *data complexity*.

We first show a CoNP data complexity upper bound for QAtc for BaseGNF constraints. The algorithm uses the fact that a counterexample to QAtc can be taken to have a $\mathcal{F}'$-rooted tree decomposition, for some $\mathcal{F}'$ that does not add new elements to $\mathcal{F}_0$, only new facts. While such a decomposition could be large, it suffices to guess $\mathcal{F}'$ and annotations describing, for each $|\varphi|$-tuple $\vec{c}$ in $\mathcal{F}'$, sufficiently many formulas holding in the subtree that interfaces with $\vec{c}$. The technique generalizes an analogous result in [Bárány *et al.*, 2012].

Theorem 3. *For any fixed* BaseGNF *constraints* Σ *and UCQ* Q*, given a finite set of facts* $\mathcal{F}_0$*, we can decide* $\mathsf{QAtc}(\mathcal{F}_0, \Sigma, Q)$ *in* CoNP *data complexity.*

For FGTGDs, the data complexity of QA is in PTIME [Baget *et al.*, 2011]. We can show that the same holds, but only for BaseCovFGTGDs, and for QAtr rather than QAtc:

Theorem 4. *For any fixed* BaseCovFGTGD *constraints* Σ *and base-covered UCQ* Q*, given a finite set of facts* $\mathcal{F}_0$*, we can decide* $\mathsf{QAtr}(\mathcal{F}_0, \Sigma, Q)$ *in* PTIME *data complexity.*

The proof uses a reduction to the standard QA problem for FGTGDs, and then applies the PTIME result of [Baget *et al.*, 2011]. The reduction again makes use of tree-likeness to show that we can replace the requirement that the R_i^+ are transitive by the weaker requirement of transitivity within small sets (intuitively, within bags of a decomposition). We will also use this idea for linear orders (see Proposition 3).

Restricting to QAtr is in fact essential to make data complexity tractable, as hardness holds otherwise.

Hardness. We now show complexity lower bounds. We already know that all our variants of QA are 2EXPTIME-hard in combined complexity, and CoNP-hard in data complexity, when GNF constraints are allowed: this follows from existing bounds on GNF reasoning [Bárány *et al.*, 2012] even without

distinguished predicates. However, in the case of the QAtc problem, we can show the same result for the much weaker language of BaseIDs.

We do this via a reduction from QA with *disjunctive inclusion dependencies*, which is known to be 2EXPTIME-hard in combined complexity [Bourhis *et al.*, 2013, Theorem 2] and CoNP-hard in data complexity [Calvanese *et al.*, 2006; Bourhis *et al.*, 2013], even without distinguished relations. We use the transitive closure to emulate disjunction (as was already suggested in the description logic context [Horrocks and Sattler, 1999]) by creating an R_i^+-fact and limiting the length of a witness R_i-path (this limit is imposed by Q'). The choice of the length of the witness path among two possibilities is used to mimic the disjunction. We thus show:

Theorem 5. *For any finite set of facts $\mathcal{F}_0$,* DIDs *Σ, and UCQ Q on a signature σ, we can compute in* PTIME *a set of facts $\mathcal{F}'_0$,* BaseIDs *Σ', and a base-covered CQ Q' on a signature σ' (with a single distinguished relation), such that* $\mathsf{QA}(\mathcal{F}_0, \Sigma, Q)$ *iff* $\mathsf{QAtc}(\mathcal{F}'_0, \Sigma', Q')$.

This implies the following, contrasting with Theorem 4:

Corollary 2. *The* QAtc *problem with* BaseIDs *and base-covered CQs is* CoNP*-hard in data complexity and* 2EXPTIME*-hard in combined complexity.*

In fact, the data complexity lower bound for QAtc even holds in the absence of constraints:

Proposition 2. *There is a base-covered CQ Q such that the data complexity of* $\mathsf{QAtc}(\mathcal{F}_0, \emptyset, Q)$ *is* CoNP*-hard.*

We prove this by reducing the problem of 3-coloring a directed graph, known to be NP-hard, to the complement of QAtc. It is well-known how to do this using TGDs that have disjunction in the head. As in the proof of Theorem 5, we simulate this disjunction by using a choice of the length of paths that realize transitive closure facts asserted in $\mathcal{F}_0$.

In all of these hardness results, we first prove them for UCQs, and then show how the use of disjunction can be eliminated, using a prior "trick" (see, e.g., [Gottlob and Papadimitriou, 2003]) to code the intermediate truth values of disjunctions within a CQ.

4 Decidability results for linear orders

We now move to QAlin, the setting where the distinguished relations $<_i$ of $\sigma_{\mathcal{D}}$ are *linear* (total) strict orders, i.e., they are transitive, irreflexive, and total. We consider constraints and queries that are base-covered. We prove the following result.

Theorem 6. *We can decide* $\mathsf{QAlin}(\mathcal{F}_0, \Sigma, Q)$ *in* 2EXPTIME*, where $\mathcal{F}_0$ ranges over finite sets of facts, Σ over* BaseCovGNF*, and Q over base-covered UCQs. In particular, this holds when Σ consists of* BaseCovFGTGDs*.*

Our technique here is to reduce this to traditional QA where no additional restrictions (like being transitive or a linear order) are imposed. Starting with BaseCovGNF constraints, we reduce to a traditional QA problem with GNF constraints, and hence prove decidability in 2EXPTIME using [Bárány *et al.*, 2012]. However, the reduction is quite simple, and hence could be applicable to other constraint classes.

The idea behind the reduction is to include additional constraints that enforce the linear order conditions. However, we cannot express transitivity or totality in GNF. Hence, we will only add a weakening of these properties that is expressible in GNF, and then argue that this is sufficient for our purposes.

The reduction is described in the following proposition.

Proposition 3. *For any finite set of facts $\mathcal{F}_0$, constraints $\Sigma \in$* BaseCovGNF*, and base-covered UCQ Q, we can compute $\mathcal{F}'_0$ and $\Sigma' \in$* BaseGNF *in* PTIME *such that* $\mathsf{QAlin}(\mathcal{F}_0, \Sigma, Q)$ *iff* $\mathsf{QA}(\mathcal{F}'_0, \Sigma', Q)$.

In particular, $\mathcal{F}'_0$ is $\mathcal{F}_0$ together with facts $G(a, b)$ for every pair $a, b \in \mathsf{elems}(\mathcal{F}_0)$, where G is some fresh binary base relation. We define Σ' as Σ together with the *k-guardedly linear axioms* for each distinguished relation $<$, where k is $\max(|\Sigma \wedge \neg Q|, \mathsf{arity}(\sigma \cup \{G\}))$; namely:

- guardedly total:
  $\forall xy((\mathsf{guarded}_{\sigma_{\mathcal{B}} \cup \{G\}}(x, y) \wedge x \neq y) \rightarrow x < y \vee y < x)$
- irreflexive: $\neg \exists x(x < x)$
- k-guardedly transitive: for $1 \leq l \leq k-1$:
  $\neg \exists xy(\psi_l(x, y) \wedge \mathsf{guarded}_{\sigma_{\mathcal{B}} \cup \{G\}}(x, y) \wedge \neg(x < y))$
  and, for $1 \leq l \leq k$: $\neg \exists x(\psi_l(x, x) \wedge x = x \wedge \neg(x < x))$

where:

- $\mathsf{guarded}_{\sigma_{\mathcal{B}} \cup \{G\}}(x, y)$ is the formula expressing that x, y is base-guarded (an existentially-quantified disjunction over all possible base-guards containing x and y);
- $\psi_1(x, y)$ is just $x < y$; and
- $\psi_l(x, y)$ for $l \geq 2$ is: $\exists x_2 \ldots x_l(x < x_2 \wedge \cdots \wedge x_l < y)$.

Unlike the property of being a linear order, the k-guardedly linear axioms can be expressed in BaseGNF.

We now sketch the argument for the correctness of the reduction. The easy direction is where we assume $\mathsf{QA}(\mathcal{F}'_0, \Sigma', Q)$ holds, so any $\mathcal{F}' \supseteq \mathcal{F}'_0$ satisfying Σ' must satisfy Q. Now consider $\mathcal{F} \supseteq \mathcal{F}_0$ that satisfies Σ and where all $<$ in $\sigma_{\mathcal{D}}$ are strict linear orders. We must show that $\mathcal{F}$ satisfies Q. First, observe that $\mathcal{F}$ satisfies Σ' since the k-guardedly linear axioms for $<$ are clearly satisfied for all k when $<$ is a strict linear order. Now consider the extension of $\mathcal{F}$ to $\mathcal{F}'$ with facts $G(a, b)$ for all $a, b \in \mathsf{elems}(\mathcal{F}_0)$. This must still satisfy Σ': adding these facts means there are additional k-guardedly linear requirements on the elements from $\mathcal{F}_0$, but these requirements already hold since $<$ is a strict linear order. Hence, by our initial assumption, $\mathcal{F}'$ must satisfy Q. Since Q does not mention G, the restriction of $\mathcal{F}'$ back to $\mathcal{F}$ still satisfies Q as well. Therefore, $\mathsf{QAlin}(\mathcal{F}_0, \Sigma, Q)$ holds.

For the harder direction, suppose for the sake of contradiction that $\mathsf{QA}(\mathcal{F}'_0, \Sigma', Q)$ does not hold, but $\mathsf{QAlin}(\mathcal{F}_0, \Sigma, Q)$ does. Then there is some $\mathcal{F}' \supseteq \mathcal{F}'_0$ such that $\mathcal{F}'$ satisfies $\Sigma' \wedge \neg Q$. We will again rely on the ability to restrict to tree-like $\mathcal{F}'$, but with a slightly different notion of tree-likeness.

We say a set E of elements from $\mathsf{elems}(\mathcal{F})$ are *base-guarded* in $\mathcal{F}$ if there is some $\sigma_{\mathcal{B}}$-fact or G-fact in $\mathcal{F}$ that mentions all of the elements in E. A *base-guarded-interface tree decomposition* $(T, \mathsf{Child}, \lambda)$ for $\mathcal{F}$ is a tree decomposition satisfying the following additional property: for all nodes n_1 that are not the root of T, if n_2 is a child of n_1 and E is the set of elements mentioned in both n_1 and n_2, then E is base-guarded in $\mathcal{F}$. A sentence φ has *base-guarded-interface k-tree-like witnesses* if for any finite set of facts $\mathcal{F}_0$, if there

is some $\mathcal{F} \supseteq \mathcal{F}_0$ satisfying φ then there is such an $\mathcal{F}$ with an $\mathcal{F}_0$-rooted $(k-1)$-width base-guarded-interface tree decomposition.

Although the transformation from Σ to Σ' makes the formula larger, it does not increase the "width" that controls the bag size of tree-like witnesses. Hence, we can show:

Lemma 1. *The sentence $\Sigma' \wedge \neg Q$ has base-guarded-interface k-tree-like witnesses for $k = \max(|\Sigma \wedge \neg Q|, \mathsf{arity}(\sigma \cup \{G\}))$.*

Using this lemma, we can assume that we have some $\mathcal{F}' \supseteq \mathcal{F}'_0$ which has a $(k-1)$-width base-guarded-interface tree decomposition and witnesses $\Sigma' \wedge \neg Q$. If every $<$ in $\sigma_{\mathcal{D}}$ is a strict linear order in $\mathcal{F}'$, then restricting $\mathcal{F}'$ to the set of σ-facts yields some $\mathcal{F}$ that would satisfy $\Sigma \wedge \neg Q$, a contradiction. Hence, there are some distinguished relations $<$ that are not strict linear orders in $\mathcal{F}'$. We can show that such an $\mathcal{F}'$ can actually be extended to some $\mathcal{F}''$ that still satisfies $\Sigma' \wedge \neg Q$ but where all $<$ in $\sigma_{\mathcal{D}}$ are strict linear orders, which we already argued is impossible.

The crucial part of the argument is thus about extending k-guardedly linear counterexamples to genuine linear orders:

Lemma 2. *If there is $\mathcal{F}' \supseteq \mathcal{F}'_0$ that satisfies $\Sigma' \wedge \neg Q$ and has a $\mathcal{F}'_0$-rooted base-guarded-interface $(k-1)$-width tree decomposition, then there is $\mathcal{F}'' \supseteq \mathcal{F}'$ that satisfies $\Sigma' \wedge \neg Q$ where each distinguished relation is a strict linear order.*

The proof of Lemma 2 proceeds by showing that sets of facts that have $(k-1)$-width base-guarded-interface tree decompositions and satisfy k-guardedly linear axioms must already be cycle-free with respect to $<$. Hence, by taking the transitive closure of $<$ in $\mathcal{F}$, we get a new set of facts where every $<$ is a strict *partial* order. Any strict partial order can be further extended to a strict linear order using known techniques, so we can obtain $\mathcal{F}'' \supseteq \mathcal{F}'$ where $<$ is a strict partial order. This $\mathcal{F}''$ may have more $<$-facts than $\mathcal{F}'$, but the k-guardedly linear axioms ensure that these new $<$-facts are only about pairs of elements that are not base-guarded.

It remains to show that $\mathcal{F}''$ satisfies $\Sigma' \wedge \neg Q$. It is clear that $\mathcal{F}''$ still satisfies the k-guardedly linear axioms, but it could no longer satisfy $\Sigma \wedge \neg Q$. However, this is where the base-covered assumption on $\Sigma \wedge \neg Q$ is used: it can be shown that satisfiability of $\Sigma \wedge \neg Q$ in BaseCovGNF is not affected by adding new $<$-facts about pairs of elements that are not base-guarded.

Data complexity. Again, the result of Theorem 6 is a combined complexity upper bound. However, as it works by reducing to traditional QA in PTIME, data complexity upper bounds follow from [Bárány *et al.*, 2012].

Corollary 3. *For any* BaseCovGNF *constraints Σ and base-covered UCQ Q, given a finite set of facts $\mathcal{F}_0$, we can decide* $\mathsf{QAlin}(\mathcal{F}_0, \Sigma, Q)$ *in* CoNP *data complexity.*

This is similar to the way data complexity bounds were shown for QAtr (in Theorem 4). However, unlike for the QAtr problem, the constraint rewriting in this section introduces disjunction, so rewriting a QAlin problem for BaseCovFGTGDs does not produce a classical query answering problem for FGTGDs. Thus the rewriting does not imply a PTIME data complexity upper bound for BaseCovFGTGD. Indeed, we will see in Proposition 4 that this is CoNP-hard.

Hardness. QAlin for BaseCovGNF constraints is again immediately CoNP-hard in data complexity, and 2EXPTIME-hard in combined complexity, from the corresponding bounds on GNF [Bárány *et al.*, 2012]. However, we can show that hardness holds for the much weaker constraint language BaseID, by a reduction from DID reasoning, as in Section 3.

Theorem 7. *For any finite set of facts $\mathcal{F}_0$,* DID*s Σ, and UCQ Q on a signature σ, we can compute in* PTIME *a set of facts $\mathcal{F}'_0$,* BaseID*s Σ', and CQ Q' on a signature σ' (with a single distinguished relation), such that* $\mathsf{QA}(\mathcal{F}_0, \Sigma, Q)$ *iff* $\mathsf{QAlin}(\mathcal{F}'_0, \Sigma', Q')$.

The reduction allows us to transfer hardness results for DID from [Calvanese *et al.*, 2006; Bourhis *et al.*, 2013], exactly as was done in Theorem 5, to conclude:

Corollary 4. *The* QAlin *problem with* BaseID *and base-covered CQs is* CoNP*-hard in data complexity and* 2EXPTIME*-hard in combined complexity.*

Again, as in the previous section, the data complexity lower bound even holds in the absence of constraints:

Proposition 4. *There is a base-covered CQ Q such that the data complexity of* $\mathsf{QAlin}(\mathcal{F}, \emptyset, Q)$ *is* CoNP*-hard.*

5 Undecidability results for transitivity

We have shown in Section 3 that query answering is decidable with transitive relations (even with transitive closure), BaseFGTGDs, and UCQs (Theorem 1). Removing our base-guarded condition leads to undecidability of QAtc, even when constraints are inclusion dependencies:

Theorem 8. *There is a signature $\sigma = \sigma_{\mathcal{B}} \sqcup \sigma_{\mathcal{D}}$ with a single distinguished predicate S^+ in $\sigma_{\mathcal{D}}$, a set Σ of* ID*s on σ, and a CQ Q on $\sigma_{\mathcal{B}}$, such that the following problem is undecidable: given a finite set of facts $\mathcal{F}_0$, decide* $\mathsf{QAtc}(\mathcal{F}_0, \Sigma, Q)$.

The proof is by reduction from a tiling problem. The constraints use a transitive successor relation to define a grid of integer pairs. It then uses transitive closure to emulate disjunction, as in Theorem 5, and relies on Q to test for forbidden adjacent tile patterns.

An analogous result can be shown for QAtr, using (non-base-guarded) disjunctive inclusion dependencies:

Theorem 9. *There is an arity-two signature $\sigma = \sigma_{\mathcal{B}} \sqcup \sigma_{\mathcal{D}}$ with a single distinguished predicate S^+ in $\sigma_{\mathcal{D}}$, a set Σ of* DID*s on σ, a CQ Q on $\sigma_{\mathcal{B}}$, such that the following problem is undecidable: given a finite set of facts $\mathcal{F}_0$, decide* $\mathsf{QAtr}(\mathcal{F}_0, \Sigma, Q)$.

These results complement the undecidability results of [Gottlob *et al.*, 2013, Theorem 2], which showed that, on arity-two signatures, QAtr is undecidable with guarded TGDs and atomic CQs, even when transitive relations occur only in guards. Our results also contrast with the decidability results of [Baget *et al.*, 2015] which apply to QAtr: our Theorem 8 shows that their results cannot extend to QAtc.

6 Undecidability results for linear orders

Section 4 has shown that QAlin is decidable for base-covered CQs and BaseCovGNF constraints. Dropping the base-covered requirement on the query leads to undecidability:

Theorem 10. *There is a signature $\sigma = \sigma_{\mathcal{B}} \sqcup \sigma_{\mathcal{D}}$ where $\sigma_{\mathcal{D}}$ is a single strict linear order relation, a CQ Q on σ, and a set Σ of inclusion dependencies on $\sigma_{\mathcal{B}}$ (i.e., not mentioning the linear order, so in particular base-covered), such that the following problem is undecidable: given a finite set of facts $\mathcal{F}_0$, decide* $\mathsf{QAlin}(\mathcal{F}_0, \Sigma, Q)$.

This result is close to [Gutiérrez-Basulto *et al.*, 2013, Theorem 3], which deals not with a linear order, but inequalities in queries, which we can express with a linear order. However, this requires a UCQ. As in our prior hardness and undecidability results, we can adapt the technique to use a CQ.

By letting $\Sigma' := \Sigma \wedge \neg Q$ where Σ and Q are as in the previous theorem, we obtain base-guarded constraints for which QAlin is undecidable. In fact, Σ' can be expressed as a set of BaseFGTGDs. This implies that the base-covered requirement is necessary for the constraint language:

Corollary 5. *There is a signature $\sigma = \sigma_{\mathcal{B}} \sqcup \sigma_{\mathcal{D}}$ where $\sigma_{\mathcal{D}}$ is a single strict linear order relation, and a set Σ' of* BaseFGTGD *constraints, such that, letting $\top$ be the tautological query, the following problem is undecidable: given a finite set of facts $\mathcal{F}_0$, decide* $\mathsf{QAlin}(\mathcal{F}_0, \Sigma', \top)$.

7 Conclusion

We have given a detailed picture of the impact of transitivity, transitive closure, and linear order restrictions on query answering problems for a broad class of guarded constraints. We have shown that transitive relations and transitive closure restrictions can be handled in guarded constraints as long as they are not used in guards. For linear orders, the same is true if order atoms are covered by base atoms. This implies the analogous results for frontier-guarded TGDs, in particular frontier-one. But in the linear order case we show that PTIME data complexity cannot always be preserved.

We leave open the question of entailment over *finite* sets of facts. There are few techniques for deciding entailment over finite sets of facts for logics where it does not coincide with general entailment (and for the constraints considered here it does not coincide). An exception is [Bárány and Bojańczyk, 2012], but it is not clear if the techniques there can be extended to our constraint languages.

Acknowledgments

Amarilli was partly funded by the Télécom ParisTech Research Chair on Big Data and Market Insights. Bourhis was supported by CPER Nord-Pas de Calais/FEDER DATA Advanced Data Science and Technologies 2015-2020 and the ANR Aggreg Project ANR-14-CE25-0017, INRIA Northern European Associate Team Integrated Linked Data. Benedikt's work was sponsored by the Engineering and Physical Sciences Research Council of the United Kingdom (EPSRC), grants EP/M005852/1 and EP/L012138/1. Vanden Boom was partially supported by EPSRC grant EP/L012138/1.

References

[Abiteboul and Duschka, 1998] Serge Abiteboul and Oliver M. Duschka. Complexity of answering queries using materialized views. In *PODS*, 1998.

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

[Afrati *et al.*, 2008] Foto Afrati, Chen Li, and Vassia Pavlaki. Data exchange in the presence of arithmetic comparisons. In *EDBT*, 2008.

[Andréka *et al.*, 1998] Hajnal Andréka, István Németi, and Johan van Benthem. Modal languages and bounded fragments of predicate logic. *J. Philosophical Logic*, 27(3), 1998.

[Baget *et al.*, 2009] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. Extending decidable cases for rules with existential variables. In *IJCAI*, 2009.

[Baget *et al.*, 2011] Jean-François Baget, Marie-Laure Mugnier, Sebastian Rudolph, and Michaël Thomazo. Walking the complexity lines for generalized guarded existential rules. In *IJCAI*, 2011.

[Baget *et al.*, 2015] Jean-François Baget, Meghyn Bienvenu, Marie-Laure Mugnier, and Swan Rocher. Combining existential rules and transitivity: Next steps. In *IJCAI*, 2015.

[Bárány and Bojańczyk, 2012] Vince Bárány and Mikołaj Bojańczyk. Finite satisfiability for guarded fixpoint logic. *IPL*, 112(10):371–375, 2012.

[Bárány *et al.*, 2011] Vince Bárány, Balder ten Cate, and Luc Segoufin. Guarded negation. In *ICALP*, 2011.

[Bárány *et al.*, 2012] Vince Bárány, Balder ten Cate, and Martin Otto. Queries with guarded negation. *PVLDB*, 5(11):1328–1339, 2012.

[Benedikt *et al.*, 2016] Michael Benedikt, Pierre Bourhis, and Michael Vanden Boom. A step up in expressiveness of decidable fixpoint logics. In *LICS*, 2016.

[Bourhis *et al.*, 2013] Pierre Bourhis, Michael Morak, and Andreas Pieris. The impact of disjunction on query answering under guarded-based existential rules. In *IJCAI*, 2013.

[Bourhis *et al.*, 2015] Pierre Bourhis, Markus Krötzsch, and Sebastian Rudolph. Reasonable highly expressive query languages. In *IJCAI*, 2015.

[Calvanese *et al.*, 2005] Diego Calvanese, Giuseppe De Giacomo, and Moshe Y. Vardi. Decidable containment of recursive queries. *Theor. Comput. Sci.*, 336(1):33–56, 2005.

[Calvanese *et al.*, 2006] Diego Calvanese, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Data complexity of query answering in description logics. In *KR*, 2006.

[Calvanese *et al.*, 2009] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Regular path queries in expressive description logics with nominals. In *IJCAI*, 2009.

[Emerson and Jutla, 1988] E. Allen Emerson and Charanjit S. Jutla. The complexity of tree automata and logics of programs (Extended abstract). In *FOCS*, 1988.

[Ganzinger *et al.*, 1999] Harald Ganzinger, Christoph Meyer, and Margus Veanes. The two-variable guarded fragment with transitive relations. In *LICS*, 1999.

[Gottlob and Papadimitriou, 2003] Georg Gottlob and Christos Papadimitriou. On the complexity of single-rule datalog queries. *Inf. Comp.*, 183, 2003.

[Gottlob *et al.*, 2013] Georg Gottlob, Andreas Pieris, and Lidia Tendera. Querying the guarded fragment with transitivity. In *ICALP*, 2013.

[Grädel and Walukiewicz, 1999] Erich Grädel and Igor Walukiewicz. Guarded fixed point logic. In *LICS*, 1999.

[Gutiérrez-Basulto *et al.*, 2013] Victor Gutiérrez-Basulto, Yazmin Ibañez Garcia, Roman Kontchakov, and Egor V. Kostylev. Conjunctive queries with negation over DL-Lite: A closer look. In *Web Reasoning and Rule Systems*, 2013.

[Horrocks and Sattler, 1999] Ian Horrocks and Ulrike Sattler. A description logic with transitive and inverse roles and role hierarchies. *Journal of Logic and Computation*, 9(3):385–410, 1999.

[Kieronski, 2011] Emanuel Kieronski. Decidability issues for two-variable logics with several linear orders. In *CSL*, 2011.

[Krötzsch and Rudolph, 2007] Markus Krötzsch and Sebastian Rudolph. Conjunctive queries for $\mathcal{EL}$ with role composition. In *DL*, 2007.

[Löding, 2011] Christoph Löding. Automata on infinite trees. `http://www.automata.rwth-aachen.de/~loeding/inf-tree-automata.pdf`, 2011.

[Ortiz and Šimkus, 2012] Magdalena Ortiz and Mantas Šimkus. Reasoning and query answering in description logics. *Reasoning Web. Semantic Technologies for Advanced Query Answering*, pages 1–53, 2012.

[Ortiz *et al.*, 2010] Magdalena Ortiz, Sebastian Rudolph, and Mantas Šimkus. Query answering is undecidable in DLs with regular expressions, inverses, nominals, and counting. Technical report, Technische Universität Wien, 2010.

[Ortiz *et al.*, 2011] Magdalena Ortiz, Sebastian Rudolph, and Mantas Šimkus. Query answering in the Horn fragments of the description logics SHOIQ and SROIQ. In *IJCAI*, 2011.

[Ortiz de la Fuente, 2010] Maria Magdalena Ortiz de la Fuente. *Query answering in expressive description logics*. PhD thesis, Technischen Universität Wien, 2010.

[Otto, 2001] Martin Otto. Two variable first-order logic over ordered domains. *J. Symb. Log.*, 66(2):685–702, 2001.

[Szpilrajn, 1930] Edward Szpilrajn. Sur l'extension de l'ordre partiel. *Fundamenta Mathematicae*, 16(1):386–389, 1930.

[Szwast and Tendera, 2004] Wiesław Szwast and Lidia Tendera. The guarded fragment with transitive guards. *Annals of Pure and Applied Logic*, 128(13):227 – 276, 2004.

[Thomas, 1997] Wolfgang Thomas. Languages, Automata, and Logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*. Springer-Verlag, 1997.

[Vardi, 1998] Moshe Y. Vardi. Reasoning about the past with two-way automata. In *ICALP*, 1998.

The proofs for the results stated in the main paper are provided in this appendix.

A Normal form

The proofs make use of the fact that the fragments of GNF that we consider can be converted into a normal form that is related to the GN normal form introduced in the original paper on GNF [Bárány *et al.*, 2011]. The idea is that GNF formulas can be seen as being built up from atoms using guarded negation, disjunction, and CQs. We introduce this normal form here, and discuss some related notions we will use in the proofs.

First, the *guardedness predicate* $\mathsf{guarded}(\vec{x})$ asserts that $\vec{x}$ is guarded by some σ-atom; it can be seen as an abbreviation for the disjunction of existentially quantified relational atoms from σ involving all of the variables from $\vec{x}$. We write $\mathsf{guarded}_{\sigma_\mathcal{B}}(\vec{x})$ for the corresponding guardedness predicate restricted to $\sigma_\mathcal{B}$.

The *normal form for* BaseGNF over σ starts with $\sigma_\mathcal{B}$-atoms and builds up via the following rules:

- If $\varphi_1(\vec{x})$ and $\varphi_2(\vec{x})$ are in normal form BaseGNF, then $\varphi_1(\vec{x}) \vee \varphi_2(\vec{x})$ are in normal form BaseGNF.
- If $\varphi(\vec{x})$ is in normal form BaseGNF and $A(\vec{x})$ is a $\sigma_\mathcal{B}$-atom or the $\sigma_\mathcal{B}$-guardedness predicate, then $A(\vec{x}) \wedge \neg\varphi(\vec{x})$ is in normal form BaseGNF.
- If δ is a CQ over signature $\sigma \cup \{Y_1, \ldots, Y_n\}$, and $\varphi_1, \ldots, \varphi_n$ are in normal form BaseGNF, and for each Y_i atom in δ there is some $\sigma_\mathcal{B}$-atom or $\sigma_\mathcal{B}$-guardedness predicate in δ that contains its free variables, then $\delta[Y_1 := \varphi_1, \ldots, Y_n := \varphi_n]$ is in normal form BaseGNF. We call $\delta[Y_1 := \varphi_1, \ldots, Y_n := \varphi_n]$ a *CQ-shaped formula*.

Likewise, the *normal form for* BaseCovGNF over σ consists of normal form BaseGNF formulas such that for every CQ-shaped subformula δ that appears negatively (in the scope of an odd number of negations), and for every conjunct β in δ, there must be some $\sigma_\mathcal{B}$-atom or $\sigma_\mathcal{B}$-guardedness predicate in δ that contains the free variables of β.

Width and CQ-rank. For φ in normal form BaseGNF, we define the *width* of φ to be the maximum number of free variables in any subformula of φ. The *CQ rank* of φ is the maximum number of conjuncts in any CQ-shaped subformula $\exists\vec{x}(\bigwedge \gamma_i)$ where $\vec{x}$ is non-empty. These will be important parameters in later proofs.

We write $\mathsf{BaseGNF}^k$ to denote *normal form* BaseGNF *formulas of width* k, and similarly for $\mathsf{BaseCovGNF}^k$.

Conversion into normal form. Observe that formulas in BaseFGTGD or BaseCovFGTGD are of the form $\forall\vec{x}(\bigwedge \gamma_i \to \exists\vec{y} \bigwedge \rho_i)$ already and thus can be naturally written in normal form BaseGNF or BaseCovGNF as $\neg\exists\vec{x}(\bigwedge \gamma_i \wedge \neg\exists\vec{y} \bigwedge \rho_i)$, with no blow-up in the size or width.

In general, BaseGNF formulas can be converted into normal form, but with an exponential blow-up in size.

Proposition 5. *Let φ be a formula in* BaseGNF. *We can construct an equivalent φ' in normal form in* EXPTIME *such that*

- *$|\varphi'|$ is at most exponential in $|\varphi|$;*
- *the width of φ' is at most $|\varphi|$;*
- *the CQ-rank of φ' is at most $|\varphi|$;*
- *if φ is in* BaseCovGNF, *then φ' is in normal form* BaseCovGNF.

Proof sketch. The conversion works by using the same rewrite rules as in [Bárány *et al.*, 2011]:

$$\exists x(\theta \vee \psi) \to (\exists x\theta) \vee (\exists x\psi)$$
$$\theta \wedge (\psi \vee \chi) \to (\theta \wedge \psi) \vee (\theta \wedge \chi)$$
$$\exists x(\theta) \wedge \psi \to \exists x'(\theta[x'/x] \wedge \psi) \text{ where } x' \text{ is fresh}$$

The size, width, and CQ-rank bounds after performing this rewriting are straightforward to check.

The rewrite rules preserve the polarity of subformulas, which helps ensure that coveredness is preserved during this conversion. □

B Transitive-closure friendly tree decompositions for BaseGNF (Proof of Proposition 1)

Recall the statement of Proposition 1:

> **Proposition 1.** Every sentence φ in BaseGNF has transitive-closure friendly k-tree-like witnesses, where $k \leq |\varphi|$.

That is, for every φ in BaseGNF and for every finite set of facts $\mathcal{F}_0$, if there is any $\mathcal{F}$ extending $\mathcal{F}_0$ with $\sigma_{\mathcal{B}}$-facts and satisfying φ when each relation R^+ is interpreted as the transitive closure of R, then there is some $\mathcal{F}$ like this that has a $\mathcal{F}_0$-rooted $(k-1)$-width tree decomposition.

If $|\varphi| < 3$, then φ is necessarily a single 0-ary relation or its negation, in which case the result is trivial, with $k = 1$. Hence, in the rest of this section, we will assume that $|\varphi| \geq 3$ and k will be chosen such that $3 \leq k \leq |\varphi|$ (k will be an upper bound on the maximum number of free variables in any subformula of φ).

The proof uses a standard technique, involving an unravelling related to a variant of guarded negation bisimulation [Bárány *et al.*, 2011]. A related result and proof also appears in [Benedikt *et al.*, 2016].

Bisimulation game. The *GN^k bisimulation game* between sets of facts $\mathcal{F}$ and $\mathcal{G}$ is an infinite game played by two players, Spoiler and Duplicator. The game has two types of positions:

i) partial isomorphisms $f : \mathcal{F}\restriction_X \to \mathcal{G}\restriction_Y$ or $g : \mathcal{G}\restriction_Y \to \mathcal{F}\restriction_X$, where $X \subset \mathsf{elems}(\mathcal{F})$ and $Y \subset \mathsf{elems}(\mathcal{G})$ are both finite and are $\sigma_{\mathcal{B}}$-guarded;

ii) partial rigid homomorphisms $f : \mathcal{F}\restriction_X \to \mathcal{G}\restriction_Y$ or $g : \mathcal{G}\restriction_Y \to \mathcal{F}\restriction_X$, where $X \subset \mathsf{elems}(\mathcal{F})$ and $Y \subset \mathsf{elems}(\mathcal{G})$ are both finite and are of size at most k.

A *partial rigid homomorphism* is a partial homomorphism with respect to all σ-facts, such that the restriction to any $\sigma_{\mathcal{B}}$-guarded set of elements is a partial isomorphism.

From a type (i) position h, Spoiler must choose a finite subset $X \subset \mathsf{elems}(\mathcal{F})$ or a finite subset $Y \subset \mathsf{elems}(\mathcal{G})$, in either case of size at most k, upon which Duplicator must respond by a partial rigid homomorphism with domain X or Y accordingly, mapping it into the other set of facts in a manner consistent with h.

From a type (ii) position $h : X \to Y$ (respectively, $h : Y \to X$), Spoiler must choose a finite subset $X \subset \mathsf{elems}(\mathcal{F})$ (respectively, $Y \subset \mathsf{elems}(\mathcal{G})$) of size at most k, upon which Duplicator must respond by a partial rigid homomorphism with domain X (respectively, domain Y), mapping it into the other set of facts in a manner consistent with h.

Notice that a type (i) position is a special kind of type (ii) position where Spoiler has the option to *switch the domain* to the other set of facts, rather than just continuing to play in the current domain.

Spoiler wins if he can force the play into a position from which Duplicator cannot respond, and Duplicator wins if she can continue to play indefinitely.

A winning strategy for Duplicator in the GN^k bisimulation game implies agreement between $\mathcal{F}$ and $\mathcal{G}$ on certain BaseGNF formulas.

Proposition 6. *Let $\varphi(\vec{x})$ be a formula in* BaseGNF*, and let $k \geq 3$ be greater than or equal to the maximum number of free variables in any subformula of φ.*

If Duplicator has a winning strategy in the GN^k bisimulation game between $\mathcal{F}$ and $\mathcal{G}$ starting from a type (i) or (ii) position $\vec{a} \mapsto \vec{b}$ and $\mathcal{F}$ satisfies $\varphi(\vec{a})$ when interpreting each $R^+ \in \sigma_{\mathcal{D}}$ as the transitive closure of $R \in \sigma_{\mathcal{B}}$, then $\mathcal{G}$ satisfies $\varphi(\vec{b})$ when interpreting each $R^+ \in \sigma_{\mathcal{D}}$ as the transitive closure of $R \in \sigma_{\mathcal{B}}$.

Proof. For this proof, when we talk about sets of facts satisfying a formula, we mean satisfaction when interpreting $R^+ \in \sigma_{\mathcal{D}}$ as the transitive closure of $R \in \sigma_{\mathcal{B}}$. We will abuse terminology slightly and say that φ has width k if the maximum number of free variables in any subformula of φ is at most k (this is abusing the terminology since we are not assuming in this proof that φ is in normal form).

We proceed by induction on the number of $\sigma_{\mathcal{D}}$-atoms in φ and the size of φ.

Suppose Duplicator has a winning strategy in the GN^k bisimulation game between $\mathcal{F}$ and $\mathcal{G}$.

If φ is a $\sigma_{\mathcal{B}}$-atom $A(\vec{x})$, the result follows from the fact that the position $\vec{a} \mapsto \vec{b}$ is a partial homomorphism.

Suppose φ is a $\sigma_{\mathcal{D}}$-atom $R^+(x_1, x_2)$, and $\vec{a} = a_1a_2$ and $\vec{b} = b_1b_2$. If $\mathcal{F}, \vec{a}$ satisfies $R^+(x_1, x_2)$, there is some $n \in \mathbb{N}$ such that $n > 0$ and there is an R-path of length n between a_1 and a_2 in $\mathcal{F}$. We can write a formula $\psi_n(x_1, x_2)$ in BaseGNF (without any $\sigma_{\mathcal{D}}$-atoms) that is satisfied exactly when there is an R-path of length n. Since we do not need to write this in normal form, we can express ψ_n in BaseGNF with width 3 (maximum of 3 free variables in any subformula). Since $\mathcal{F}, \vec{a}$ satisfies ψ_n and ψ_n does not have any $\sigma_{\mathcal{D}}$-atoms and $k \geq 3$, we can apply the inductive hypothesis from the type (ii) position $\vec{a} \mapsto \vec{b}$ to ensure that $\mathcal{G}, \vec{b}$ satisfies ψ_n, and hence $\mathcal{G}, \vec{b}$ satisfies φ.

If φ is a disjunction, the result follows easily from the inductive hypothesis.

Suppose φ is a base-guarded negation $A(\vec{x}) \wedge \neg\varphi'(\vec{x}')$. By definition of BaseGNF, it must be the case that $A \in \sigma_{\mathcal{B}}$ and $\vec{x}'$ is a sub-tuple of $\vec{x}$. Since $\mathcal{F}, \vec{a}$ satisfies φ, we know that $\mathcal{F}, \vec{a}$ satisfies $A(\vec{x})$, and hence $\vec{a}$ is $\sigma_{\mathcal{B}}$-guarded. This means that $\vec{a} \mapsto \vec{b}$ is actually a partial isomorphism, so we can view it as a position of type (i). This ensures that $\mathcal{G}, \vec{b}$ also satisfies $A(\vec{x})$. It remains to show that it satisfies $\neg\varphi'(\vec{x}')$. Assume for the sake of contradiction that it satisfies $\varphi'(\vec{x}')$. Because $\vec{a} \mapsto \vec{b}$ is a type (i) position, we can consider the move in the game where Spoiler switches the domain to the other set of facts, and then restricts to the elements in the subtuple $\vec{b}'$ of $\vec{b}$ corresponding to $\vec{x}'$ in $\vec{x}$. Let $\vec{a}'$ be the corresponding subtuple of $\vec{a}$. Duplicator must have a winning strategy from the type (i) position $\vec{b}' \mapsto \vec{a}'$, so the inductive hypothesis ensures that $\mathcal{F}, \vec{a}'$ satisfies $\varphi'(\vec{x}')$, a contradiction.

Finally, suppose φ is an existentially quantified formula $\exists y(\varphi'(\vec{x}, y))$. We are assuming that $\mathcal{F}, \vec{a}$ satisfies φ. Hence,

there is some $c \in \mathsf{elems}(\mathcal{F})$ such that $\mathcal{F}, \vec{a}, c$ satisfies φ'. Because the width of φ is at most k, we know that the combined number of elements in $\vec{a}$ and c is at most k. Hence, we can consider the move in the game where Spoiler selects the elements in $\vec{a}$ and c. Duplicator must respond with $\vec{b}$ for $\vec{a}$, and some d for c. This is a valid move in the game, so Duplicator must still have a winning strategy from this position $\vec{a}c \mapsto \vec{b}d$, and the inductive hypothesis implies that $\mathcal{G}, \vec{b}, d$ satisfies φ'. Consequently, $\mathcal{G}, \vec{b}$ satisfies φ. □

Unravelling. The tree-like witnesses can be obtained using an unravelling construction related to the GN^k bisimulation game. This unravelling construction is adapted from [Benedikt *et al.*, 2016].

Fix a set of facts $\mathcal{F} \supseteq \mathcal{F}_0$. Consider the set Π of sequences of the form $X_0 X_1 \dots X_n$, where $X_0 = \mathsf{elems}(\mathcal{F}_0)$, and for all $i \geq 1$, $X_i \subseteq \mathsf{elems}(\mathcal{F})$ with $|X_i| \leq k$.

We can arrange these sequences in a tree based on the prefix order. Each sequence $\pi = X_0 X_1 \dots X_n$ identifies a unique node in the tree; we say a is *represented* at node π if $a \in X_n$. For $a \in \mathsf{elems}(\mathcal{F})$, we say π and π' are a-*equivalent* if a is represented at every node on the unique minimal path between π and π' in this tree. For a represented at π, we write $[\pi, a]$ for the a-equivalence class.

The *GN^k-unravelling of* $\mathcal{F}$ is a set of facts $\mathcal{F}^k$ over elements $\{[\pi, a] : \pi \in \Pi \text{ and } a \in \mathsf{elems}(\mathcal{F})\}$. The fact $R([\pi_1, a_1], \dots, [\pi_j, a_j]) \in \mathcal{F}^k$ iff $R(a_1, \dots, a_j) \in \mathcal{F}$ and there is some $\pi \in \Pi$ such that for all i, $[\pi, a_i] = [\pi_i, a_i]$. We can identify $[\epsilon, a]$ with the element $a \in \mathsf{elems}(\mathcal{F}_0)$, so there is a natural $\mathcal{F}_0$-rooted tree decomposition of width $k - 1$ for $\mathcal{F}^k$ induced by the tree of sequences from Π.

Because this unravelling is related so closely to the GN^k-bisimulation game, it is straightforward to show that Duplicator has a winning strategy in the bisimulation game between $\mathcal{F}$ and its unravelling.

Proposition 7. *Duplicator has a winning strategy in the GN^k bisimulation game between $\mathcal{F}$ and $\mathcal{F}^k$.*

Proof. Given a position f in the GN^k-bisimulation game, we say the *active set* is the set of facts containing the elements in the domain of f. In other words, the active set is either $\mathcal{F}$ or $\mathcal{F}^k$, depending on which set Spoiler is currently playing in. The *safe positions* f in the GN^k-bisimulation game between $\mathcal{F}$ and $\mathcal{F}^k$ are defined as follows: if the active set is $\mathcal{F}^k$, then f is safe if for all $[\pi, a] \in \mathsf{Dom}(f)$, $f([\pi, a]) = a$; if the active set is $\mathcal{F}$, then f is safe if there is some π such that $f(a) = [\pi, a]$ for all $a \in \mathsf{Dom}(f)$.

We now argue that starting from a safe position f, Duplicator has a strategy to move to a new safe position f'. This is enough to conclude that Duplicator has a winning strategy in the GN^k-bisimulation game between $\mathcal{F}$ and $\mathcal{F}^k$ starting from any safe position.

First, assume that the active set is $\mathcal{F}^k$.

- If f is a type (ii) position, then Spoiler can select some new set X' of elements from the active set. Each element in X' is of the form $[\pi', a']$. Duplicator must choose f' such that $[\pi', a']$ is mapped to a' in $\mathcal{F}$, in order to maintain safety. This new position f' is consistent with f on any elements in $X' \cap \mathsf{Dom}(f)$ since f is safe. This f' is still a partial homomorphism since any relation holding for a tuple of elements $[\pi_1, a_1], \dots, [\pi_n, a_n]$ from $\mathsf{Dom}(f')$ must hold for the tuple of elements $a_1, \dots, a_n$ in $\mathcal{F}$ by definition of $\mathcal{F}^k$. Consider some element $[\pi', a']$ in $\mathsf{Dom}(f')$. It is possible that there is some $[\pi, a']$ in $\mathsf{Dom}(f')$ with $[\pi, a'] \neq [\pi', a']$; however, $[\pi, a']$ and $[\pi', a']$ are not base-guarded in $\mathcal{F}^k$. Hence, any restriction f'' of f' to a base-guarded set of elements is a bijection. Moreover, such an f'' is a partial isomorphism: consider some $a_1, \dots, a_n$ in the range of f'' for which some relation S holds in $\mathcal{F}$; since $(f'')^{-1}(a_1), \dots, (f'')^{-1}(a_n)$ must be base-guarded, we know that there is some π such that $[\pi, a_1] = (f'')^{-1}(a_1)$,..., $[\pi, a_n] = (f'')^{-1}(a_n)$, so by definition of $\mathcal{F}^k$, S holds of $(f'')^{-1}(a_1), \dots, (f'')^{-1}(a_n)$ as desired. Hence, f' is a safe partial rigid homomorphism.
- If f is a type (i) position, then Spoiler can either choose elements in the active set and we can reason as we did for the type (ii) case, or Spoiler can select elements from the other set of facts.

  We first argue that if Spoiler changes the active set and chooses no new elements, then the game is still in a safe position. Since f is a type (i) position, we know that $\mathsf{Dom}(f)$ is guarded by some base relation S, so there is some π with $f(a) = [\pi, a]$ for all $a \in \mathsf{Dom}(f)$ by construction of $\mathcal{F}^k$. Hence, the new position $f' = f^{-1}$ is still safe.

  If Spoiler switches active sets and chooses new elements, then we can view this as two separate moves: in the first move, Spoiler switches active sets from $\mathcal{F}^k$ to $\mathcal{F}$ but chooses no new elements, and in the second move, Spoiler selects the desired new elements from $\mathcal{F}$. Because switching active sets leads to a safe position (by the argument in the previous paragraph), it remains to define Duplicator's safe strategy when the active set is $\mathcal{F}$, which we explain below.

Now assume that the active set is $\mathcal{F}$. Since f is safe, there is some π such that $f(a) = [\pi, a]$ for all $a \in \mathsf{Dom}(f)$.

- If f is a type (ii) position, then Spoiler can select some new set X' of elements from the active set. We define the new position f' chosen by Duplicator to map each element $a' \in X'$ to $[\pi', a']$ where $\pi' = \pi \cdot X'$. By construction of the unravelling, $\pi' \in \Pi$ and the resulting partial mapping f' still satisfies the safety property with π' as witness. Note that f' is consistent with f for elements of X' that are also in $\mathsf{Dom}(f)$, as we have $[\pi \cdot X', a'] = [\pi, a']$ for $a' \in X' \cap \mathsf{Dom}(f)$. Now consider some tuple $\vec{a} = a_1 \dots a_n$ of elements from $\mathsf{Dom}(f')$ that are in some relation S. We know that $f'(a_i) = [\pi', a_i]$, hence S must hold for $f'(\vec{a})$ in $\mathcal{F}^k$. Moreover, for any base-guarded set $\vec{a} = \{a_1, \dots, a_n\}$ of distinct elements from $\mathsf{Dom}(f')$, $f'(\vec{a})$ must yield a set of distinct elements $\{f'(a_1), \dots, f'(a_n)\}$, and these elements can only participate in some fact in $\mathcal{F}^k$ if the

underlying elements from $\vec{a}$ participate in the same fact in $\mathcal{F}$. Hence, f' is a safe partial rigid homomorphism.

- If f is a type (i) position, then Spoiler can either choose elements in the active set and we can reason as we did for the type (ii) case, or Spoiler can select elements from the other set of facts. It suffices to argue that if Spoiler changes the active set like this, and chooses no new elements, then the game is still in a safe position. But in this case $f' = f^{-1}$ is easily seen to still be safe.

This concludes the proof of Proposition 7. □

We can now conclude the proof of Proposition 1. Assume that $\mathcal{F} \supseteq \mathcal{F}_0$ is a set of facts that satisfies φ when interpreting R^+ as the transitive closure of R. Let $3 \leq k \leq |\varphi|$ be an upper bound on the maximum number of free variables in any subformula of φ. Since $\mathcal{F}$ satisfies φ, Propositions 7 and 6 imply that $\mathcal{F}^k$ also satisfies φ when properly interpreting R^+. Hence, we can conclude that the unravelling $\mathcal{F}^k$ is the transitive closure friendly k-tree-like witness for φ.

C Automata for BaseGNF (Proof of Theorem 2)

In this section, we prove Theorem 2, about constructing automata for sentences in BaseGNF and initial sets of facts $\mathcal{F}_0$:

> **Theorem 2.** Let φ be a sentence in BaseGNF, and let $\mathcal{F}_0$ be a finite set of facts. We can construct in 2EXPTIME a 2-way alternating parity tree automaton $\mathcal{A}_{\varphi,\mathcal{F}_0}$ such that
>
> $\mathcal{F}_0 \wedge \varphi$ is satisfiable iff $L(\mathcal{A}_{\varphi,\mathcal{F}_0}) \neq \emptyset$
>
> when $R^+ \in \sigma_{\mathcal{D}}$ are interpreted as the transitive closure of $R \in \sigma_{\mathcal{B}}$. The number of states of $\mathcal{A}_{\varphi,\mathcal{F}_0}$ is exponential in $|\varphi| \cdot |\mathcal{F}_0|$ and the number of priorities is linear in $|\varphi|$.

Before we prove the result, we need to specify the tree encodings/decodings and tree automata that we are using. For the remainder of the section, we fix some $\varphi \in$ BaseGNF and some finite set of facts $\mathcal{F}_0$.

C.1 Tree encodings/decodings

Tree encodings. By Proposition 1, we know that if $\varphi \in$ BaseGNF, then φ has transitive-closure friendly k-tree-like witnesses, for $k \leq |\varphi|$. A $\mathcal{F}_0$-rooted tree decomposition like this can be encoded as a tree with only a finite signature. Let U be a set of names of size $2k + l$ where l is the size of $\mathsf{elems}(\mathcal{F}_0)$. The signature $\tilde{\sigma}_k$ for the encodings is defined as follows.

- For all $a \in U$, there is a unary relation $D_a \in \tilde{\sigma}_k$ which indicates that a is a name for an element represented in the bag.
- For every relation $R \in \sigma$ of arity n and every n-tuple $\vec{a} \in U^n$, there is a unary relation $R_{\vec{a}} \in \tilde{\sigma}_k$, which indicates that R holds for the tuple of elements indexed by $\vec{a}$.
- For every $z \in \mathsf{elems}(\mathcal{F}_0)$ and $c \in U$, there is a unary relation $V_{c/z}$ which indicates the valuation for this element.

Tree decompositions and the corresponding encodings can generally have unbounded (possibly infinite) degree. We modify the standard encoding slightly so that we can use full binary trees: we apply the first-child, next-sibling transformation to the usual encoding, based on an arbitrary ordering of the children, and make it a full binary tree by adding dummy nodes if necessary.

Each node in a binary tree can be identified with a finite string over $\{0, 1\}$, with the root identified with ϵ. The *biological children* of a node u are the nodes $u01^+$ (these are the nodes that would have been children of u in the tree decomposition before the first-child next-sibling translation). The *biological parent* of $v \neq \epsilon$ is the unique u such that $v \in u01^+$. A *biological neighbor* is a biological child or biological parent. For these binary tree encodings, we add to $\tilde{\sigma}_k$ unary predicates P_i for $i \in \{0, 1\}$ which indicate the node is the i-th child of its parent.

From now on, we use the term *$\tilde{\sigma}_k$-tree* to refer to an infinite full binary tree over the signature $\tilde{\sigma}_k$.

Tree decodings. If a $\tilde{\sigma}_k$-tree satisfies certain consistency properties, then it can be decoded into a set of σ-facts with an $\mathcal{F}_0$-rooted tree decomposition of width $k - 1$. Let $\mathrm{names}(v) := \{a \in U : D_a(v)\}$ be the set of *names* used for elements in bag v in some tree. We will abuse notation and write $\vec{a} \subseteq \mathrm{names}(v)$ to mean that $\vec{a}$ is a tuple over names from $\mathrm{names}(v)$. A *consistent tree* T (with respect to $\tilde{\sigma}_k$ and $\mathcal{F}_0$) is a $\tilde{\sigma}_k$-tree such that every node v satisfies

- $|\mathrm{names}(v)| \leq k$, except for the root (which has size l);
- for all $R_{\vec{a}} \in \tilde{\sigma}_k$, if $R_{\vec{a}}(v)$ then $\vec{a} \subseteq \mathrm{names}(v)$;
- $P_i(v)$ holds iff v is the i-th child of its parent;
- for all $z \in \mathsf{elems}(\mathcal{F}_0)$, there is exactly one $c \in \mathrm{names}(\epsilon)$ for the root ϵ such that $V_{c/z}(\epsilon)$ holds, and there is no $v \neq \epsilon$ with some $c \in \mathrm{names}(v)$ such that $V_{c/z}(\epsilon)$ holds;
- for every $c \in \mathrm{names}(\epsilon)$, there is some $z \in \mathsf{elems}(\mathcal{F}_0)$ such that $V_{c/z}(\epsilon)$ holds;
- for each fact $R(z_1 \dots z_n) \in \mathcal{F}_0$, $R_{c_1 \dots c_n}(\epsilon)$ holds, where each $c_i \in \mathrm{names}(\epsilon)$ is the unique name such that $V_{c_i/z_i}(\epsilon)$ holds;
- for every $R_{c_1 \dots c_n}(\epsilon)$, the fact $R(z_1 \dots z_n)$ is in $\mathcal{F}_0$, where each $z_i \in \mathsf{elems}(\mathcal{F}_0)$ is the unique element such that $V_{c_i/z_i}(\epsilon)$ holds.

The last four conditions ensure that there is a bijection between the elements and facts represented at the root node and the elements and facts in $\mathcal{F}_0$.

Given a consistent tree T, we say nodes u and v are *a-connected* if there is a sequence of nodes $u = w_0, w_1, \dots, w_j = v$ such that w_{i+1} is a biological neighbor of w_i, and $a \in \mathrm{names}(w_i)$ for all $i \in \{0, \dots, j\}$. We write $[v, a]$ for the equivalence class of a-connected nodes of v. For $\vec{a} = a_1 \dots a_n$, we often abuse notation and write $[v, \vec{a}]$ for the tuple $[v, a_1], \dots, [v, a_n]$.

The *decoding* of T is the set of σ-facts $\mathsf{decode}(T)$ using elements $\{[v, a] : v \in T \text{ and } a \in \mathrm{names}(v)\}$, where we identify $z \in \mathsf{elems}(\mathcal{F}_0)$ with the unique $[\epsilon, c]$ such that $V_{c/z}(\epsilon)$ holds. For each relation R, we have $R([v_1, a_1], \dots, [v_j, a_j]) \in \mathsf{decode}(T)$ iff there is some $w \in T$ such that $R_{\vec{a}}(w)$ holds and $[w, a_i] = [v_i, a_i]$ for all i.

Free variables. The automaton construction will be an induction on the structure of the formula, so we will need to deal with formulas with free variables.

For this purpose, the tree encodings can be extended with additional information about valuations for free variables. Such trees use an extended signature.

Namely, for each free first-order variable z and each $c \in U$, we introduce a predicate $V_{c/z}$; if $V_{c/z}(v)$ holds, then this indicates that the valuation for z is the element named by c at v (we use notation similar to the valuations for $z \in \mathsf{elems}(\mathcal{F}_0)$, since these valuations all behave in a similar way).

At one point in what follows (specifically, in one case of the proof of Lemma 3), we will also use *second-order variables* to represent information about some additional relation Y. For each second-order variable Y of arity n and each $\vec{a} \in U^n$, the extended signature has a predicate $Y_{\vec{a}}$. If $Y_{\vec{a}}$ holds at some

node v, then this indicates that the tuple of elements indexed by $\vec{a}$ at v is in the relation Y.

We refer to these additional predicates that give a valuation for the free variables as *free variable markers*. In a consistent tree, the free variables markers for a first-order variable z must satisfy the condition that there is a unique v and unique $c \in \text{names}(v)$ such that $V_{c/z}(v)$ holds (i.e. for each z there is exactly one $V_{c/z}$-fact in the tree). The markers for a second-order variable Y must satisfy the condition that if $Y_{\vec{a}}(v)$ then $\vec{a} \subseteq \text{names}(v)$.

C.2 Automata tools

We will make use of automata running on infinite binary trees. We briefly recall some definitions and key properties. We will need to use 2-way automata that can move both up and down as they process the tree, so we highlight some less familiar properties about the relationship between 2-way and 1-way versions of these automata.

Trees. The input to the automata will be infinite full binary trees T over some finite set of propositions Γ. In other words, these are structures over a signature with binary relations for the left and right child relation, and unary relations for the propositions. We also assume there are propositions indicating whether each node is a left child, right child, or the root. We write $T(v)$ for the set of propositions that hold at node v.

Tree automata. An *alternating parity tree automaton* $\mathcal{A}$ is a tuple $\langle \Gamma, Q, q_0, \delta, \Omega \rangle$ where Γ is a finite set of propositions, Q is a finite set of states, $q_0 \in Q$ is the initial state, $\delta : Q \times \mathcal{P}(\Gamma) \to \mathcal{B}^+(\text{Dir} \times Q)$ is the transition function with directions $\text{Dir} \subseteq \{0, 1, -1\}$, and $\Omega : Q \to P$ is the priority function with a finite set of *priorities* $P \subseteq \mathbb{N}$.

The transition function δ maps a state and input letter to a positive boolean formula over $\text{Dir} \times Q$ (denoted $\mathcal{B}^+(\text{Dir} \times Q)$) that indicates possible next moves for the automaton.

Running the automaton $\mathcal{A}$ on some input tree T is best thought of in terms of an *acceptance game*. Positions in the game are of the form $(q, v) \in Q \times T$. In position (q, v), Eve chooses a disjunct θ in $\delta'(q, T(v))$, where δ' is the result of writing each of the transition function formulas in disjunctive normal form. Then Adam chooses a conjunct (d, q') in θ and the game continues from position (q', v'), where v' is the node in direction d from v (Adam loses if there is no such node v').

A play $(q_0, v_0)(q_1, v_1) \dots$ in the game is winning for Eve if it satisfies the *parity condition*: the maximum priority occurring infinitely often in $\Omega(q_0)\Omega(q_1) \dots$ is even. A *strategy* for Eve is a function that, given the history of the play and the current position in the game, determines Eve's choice in the game. Note that we allow the automaton to be started from arbitrary positions in the tree, rather than just the root. We say that $\mathcal{A}$ *accepts* T *starting from* v_0 if Eve has a strategy such that all plays consistent with the strategy starting from (q_0, v_0) are winning. $L(\mathcal{A})$ denotes the *language* of trees accepted by $\mathcal{A}$ starting from the root.

A 1-way alternating automaton is an automaton that uses only directions 0 and 1. A (1-way) nondeterministic automaton is a 1-way alternating automaton such that every transition function formula is of the form $\bigvee_j (0, q_0^j) \wedge (1, q_1^j)$.

Closure properties. We recall some closure properties of these automata, omitting the standard proofs; see [Thomas, 1997; Löding, 2011] for more information. Note that we state only the size of the automata for each property, but the running time of the procedures constructing these automata is polynomial in the output size.

First, the automata that we are using are closed under union and intersection (of their languages).

Proposition 8. *2-way alternating parity tree automata and 1-way nondeterministic parity tree automata are closed under union and intersection, with only a polynomial blow-up in the number of states, priorities, and overall size.*

For example, this means that if we are given 2-way alternating parity tree automata $\mathcal{A}_1$ and $\mathcal{A}_2$, then we can construct in PTIME a 2-way alternating parity tree automaton $\mathcal{A}$ such that $L(\mathcal{A}) = L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$.

Another important language operation is projection. Let L' be a language of trees over propositions $\Gamma \cup \{P\}$. The *projection* of L' with respect to P is the language of trees T over Γ such that there is some $T' \in L'$ such that T and T' agree on all propositions in Γ. Projection is easy for nondeterministic automata since the valuation for the projected proposition can be guessed by Eve.

Proposition 9. *1-way nondeterministic parity tree automata are closed under projection, with no change in the number of states, priorities, and overall size.*

Finally, complementation is easy for alternating automata by taking the *dual* automaton, obtained by switching conjunctions and disjunctions in the transition function, and incrementing all of the priorities by one.

Proposition 10. *2-way alternating parity tree automata are closed under complementation, with no change in the number of states, priorities, and overall size.*

Connections between 2-way and 1-way automata. It was shown by Vardi [1998] that 2-way alternating parity tree automata can be converted to equivalent 1-way nondeterministic automata, with an exponential blow-up.

Theorem 11 ([Vardi, 1998])**.** *Let $\mathcal{A}$ be a 2-way alternating parity tree automaton. We can construct a 1-way nondeterministic parity tree automaton $\mathcal{A}'$ such that $L(\mathcal{A}) = L(\mathcal{A}')$. The number of states of $\mathcal{A}'$ is exponential in the number of states of $\mathcal{A}$, but the number of priorities of $\mathcal{A}'$ is linear in the number of priorities of $\mathcal{A}$.*

1-way nondeterministic tree automata can be seen as a special case of 2-way alternating automata, so the previous theorem shows that 1-way nondeterministic and 2-way alternating parity automata are equivalent, in terms of their ability to recognize trees starting from the root.

We need another conversion from 1-way nondeterministic to 2-way alternating automata that we call *localization*. This is the process by which a 1-way nondeterministic automaton that is running on trees with extra information about some predicate annotated on the tree is converted to an equivalent 2-way alternating automaton that operates on trees without these annotations under the assumption that these predicates hold only locally at the position the 2-way automaton is launched from. A similar localization theorem is present in prior work [Bourhis *et al.*, 2015; Benedikt *et al.*, 2016].

Theorem 12. *Let $\Gamma' := \Gamma \cup \{P_1, \ldots, P_j\}$. Let $\mathcal{A}'$ be a 1-way nondeterministic parity automaton on Γ'-trees. We can construct a 2-way alternating parity automaton $\mathcal{A}$ on Γ-trees such that for all Γ-trees T and nodes $v \in \mathsf{Dom}(T)$,*

$$\mathcal{A}' \text{ accepts } T' \text{ from the root } \textit{iff } \mathcal{A} \text{ accepts } T \text{ from } v$$

where T' is the Γ'-tree obtained from T by setting $P_1^{T'} = \cdots = P_j^{T'} = \{v\}$. The number of states of $\mathcal{A}$ is linear in the number of states of $\mathcal{A}'$, and the overall size of $\mathcal{A}$ is linear in the size of $\mathcal{A}'$. The number of priorities of $\mathcal{A}$ is linear in the number of priorities of $\mathcal{A}'$.

Proof sketch. $\mathcal{A}$ simulates $\mathcal{A}'$ by guessing in a backwards fashion an initial part of a run of $\mathcal{A}'$ on the path from v to the root and then processing the rest of the tree in a normal downwards fashion. The subtlety is that the automaton $\mathcal{A}$ is reading a tree without valuation for $P_1, \ldots, P_j$ so once the automaton leaves node v, if it were to cross this position again, it would be unable to correctly simulate $\mathcal{A}'$. To avoid this issue, we only send downwards copies of the automaton in directions that are not on the path from the root to v. □

Emptiness testing. Finally, we make use of the well-known fact that language emptiness of tree automata is decidable.

Theorem 13 ([Emerson and Jutla, 1988],[Vardi, 1998])**.** *For 1-way nondeterministic parity tree automata, emptiness is decidable in time polynomial in the number of states and exponential in the number of priorities. For 2-way alternating parity automata, it is decidable in time exponential in the number of states and priorities.*

C.3 Construction

We are almost ready to construct an automaton for $\varphi \in \mathsf{BaseGNF}$ and $\mathcal{F}_0$ to prove Theorem 2. It is convenient to work with normal form formulas, so let φ' be the normal form $\mathsf{BaseGNF}$ sentence that is equivalent to φ.

We build up this automaton inductively, so we must construct an automaton $\mathcal{A}_\psi$ for each subformula $\psi(\vec{x})$ of φ'. The automaton $\mathcal{A}_\psi$ will not specify a single initial state. Instead, there will be a designated initial state for each possible "local assignment" for the free variables $\vec{x}$. A *local assignment* $\vec{a}/\vec{x}$ for $\vec{a} = a_1 \ldots a_n \in U^n$ and $\vec{x} = x_1 \ldots x_n$ is a mapping such that $x_i \mapsto a_i$. A node v in a consistent tree T with $\vec{a} \subseteq \text{names}(v)$ and a local assignment $\vec{a}/\vec{x}$, specifies a valuation for $\vec{x}$. We say it is local since the free variable markers for $\vec{x}$ would all appear locally in v.

We will write $\mathcal{A}_\psi$ for the automaton for ψ (without specifying the initial state), and will write $\mathcal{A}_\psi^{\vec{a}/\vec{x}}$ for $\mathcal{A}_\psi$ with the designated initial state for $\vec{a}/\vec{x}$. We call $\mathcal{A}_\psi^{\vec{a}/\vec{x}}$ a *localized automaton*, since it is testing whether some tuple that is represented locally in the tree satisfies ψ. Localized automata are useful because they can be launched to test that a tuple of elements that appear together in a node satisfy some property — without having the markers for this tuple explicitly written on the tree.

The construction is described in the following lemma.

Lemma 3. *Let $\psi(\vec{x})$ be a subformula of φ' (the normal form version of φ). Let k be the width of φ', let l be the size of $\mathsf{elems}(\mathcal{F}_0)$, and let $K := 2k + l$.*

We can construct a 2-way alternating parity tree automaton $\mathcal{A}_\psi$ such that for all consistent trees T, for all local assignments $\vec{a}/\vec{x}$, and for all nodes v in T with $\vec{a} \subseteq \textit{names}(v)$,

$$\mathcal{A}_\psi^{\vec{a}/\vec{x}} \textit{ accepts } T \textit{ starting from } v$$
$$\textit{iff} \quad \mathsf{decode}(T), [v, \vec{a}] \textit{ satisfies } \psi$$

when each $R^+ \in \sigma_{\mathcal{D}}$ is interpreted as the transitive closure of $R \in \sigma_{\mathcal{B}}$.

Further, there is a polynomial function f independent of ψ such that the number of states of $\mathcal{A}_\psi$ is at most $N_\psi := f(m_\psi) \cdot 2^{f(Kr_\psi)}$ where $m_\psi = |\psi|$ and r_ψ is the CQ-rank of ψ. The overall size of the automaton and the running time of the construction is at most exponential in $|\sigma| \cdot N_\psi$. The number of priorities is linear in ψ.

Proof. We proceed by induction on normal form $\psi(\vec{x})$ in $\mathsf{BaseGNF}$. We will write m_ψ for $|\psi|$, r_ψ for the CQ-rank of ψ, and N_ψ for $f(m_\psi) \cdot 2^{f(Kr_\psi)}$ for some suitably chosen (in particular, non-constant) polynomial f independent of ψ (we will not define f explicitly).

During each case of the inductive construction, we will describe informally how to build the desired automaton, and we will analyze the number of priorities and the number of states required. We defer the analysis of the overall size of the automaton until the end of this proof.

Base cases. For each of the base cases $\psi(\vec{x})$, we first describe a 2-way alternating parity tree automaton $\mathcal{B}_\psi$ that runs on trees *with the free variable markers for $\vec{x}$ written on the tree*:

- Suppose ψ is a $\sigma_{\mathcal{B}}$-atom $\alpha(\vec{x})$. Eve tries to navigate to a node v whose label includes fact $\alpha(\vec{b})$. If she is able to do this, Adam can then challenge Eve to show that $\vec{x}$ corresponds to $\vec{b}$. Say he challenges her on $b_i \in \vec{b}$. Then Eve must navigate from v to the node carrying the marker b_i/x_i. However, she must do this by passing through a series of biological neighbors that also contain b_i (the intermediate nodes in between biological neighbors might not contain b_i). If she is able to do this, $\mathcal{B}_\psi$ enters an accepting sink state (with priority 0). The other states are non-accepting (with priority 1) to ensure that Eve actually witnesses $\alpha(\vec{x})$. The number of states of $\mathcal{B}_\psi$ is linear in K, since the automaton must remember the name b_i that Adam is challenging. There are two priorities.

- The case when ψ is the $\sigma_{\mathcal{B}}$-guardedness predicate $\mathsf{guarded}_{\sigma_{\mathcal{B}}}(\vec{x})$ is similar, except Eve can choose any atom α over $\sigma_{\mathcal{B}}$ that uses all of the variables $\vec{x}$, and then proceed as in the previous case.
- Suppose ψ is an equality $x_1 = x_2$. Eve navigates to the node v with the marker a/x_1. She is then required to navigate from v to the node carrying the marker for x_2. She must do so by passing through a series of biological neighbors that also contain a (again, the intermediate nodes in between biological neighbors might not contain a). If she is able to reach the marker a/x_2 in this way then x_1 and x_2 are marking the same element in the underlying set of facts, so $\mathcal{B}_\psi$ moves to a sink state with priority 0 and she wins. The other states have priority 1, so if Eve is not able to do this, then Adam wins. The state set is of size linear in K, in order to remember the name a. There are two priorities.
- Suppose ψ is a $\sigma_{\mathcal{D}}$-atom $R^+(x_1, x_2)$. Eve first tries to navigate to the node v_0 carrying the marker a_1/x_1 for x_1. The automaton $\mathcal{B}_\psi$ then simulates the following game. The initial position in the game is (v_0, a_1). In general, positions in the game are of the form (v, a) for a node v and a name a, and one round of the game consists of the following: Eve can either
  - choose a' in v such that label at v includes fact $R(a, a')$; she immediately wins if v includes marker a'/x_2, otherwise she proceeds to the next round in position (v, a'), or
  - choose some biological neighbor v' which includes the name a, and the game proceeds to the next round in position (v', a).

  This game can be implemented using a 2-way automaton. Winning corresponds to moving to a sink state with priority 0. All of the other states are assigned priority 1. This ensures that eventually Eve witnesses a path of R-facts from x_1 to x_2. The number of states in $\mathcal{B}_\psi$ is again linear in K, since it must remember the name a that is currently being processed along this path. There are only two priorities.

For each base case $\psi(\vec{x})$, we have constructed an automaton $\mathcal{B}_\psi$ with two priorities and a state set of size linear in K. However, this automaton runs on trees with the free variable markers for $\vec{x}$, so it remains to show that we can construct the automaton $\mathcal{A}_\psi$ required by the lemma that runs on trees without these markers.

First, we can convert $\mathcal{B}_\psi$ into an equivalent nondeterministic parity tree automaton with an exponential blow-up in the number of states and a linear blow-up in the number of priorities (using Theorem 11). After this step, the number of states is exponential in K.

For each local assignment $\vec{a}/\vec{x}$, we can then apply the localization theorem (Theorem 12) to the set of predicates of the form V_{a_i/x_i}, and eliminate the dependence on any other V_{c/x_i} for $c \neq a_i$ by always assuming these predicates do not hold. This results in a localized automaton $\mathcal{A}_\psi^{\vec{a}/\vec{x}}$ that no longer relies on free variable markers for $\vec{x}$. By Theorem 12, there is only a linear blow-up in the number of states and number of priorities, so after this step the number of states in each $\mathcal{A}_\psi^{\vec{a}/\vec{x}}$ is exponential in K.

Finally, we take $\mathcal{A}_\psi$ to be the disjoint union of $\mathcal{A}_\psi^{\vec{a}/\vec{x}}$ over all local assignments $\vec{a}/\vec{x}$; the designated initial state for each localization is the initial state for $\mathcal{A}_\psi^{\vec{a}/\vec{x}}$. Since there are at most K^k localizations, the number of states in $\mathcal{A}_\psi$ is still exponential in K, which can be assumed to be less than N_ψ by the choice of f. The number of priorities is a constant independent of ψ.

Inductive cases. We now proceed with the inductive cases. We build $\mathcal{A}_\psi$ with the help of inductively defined automata for its subformulas.

- Suppose ψ is a guarded negation of the form $\alpha(\vec{x}) \wedge \neg\psi'(\vec{x})$. Construct $\mathcal{A}_\psi$ by taking the disjoint union of $\mathcal{A}_\alpha$, the dual of $\mathcal{A}_{\psi'}$ (obtained by switching conjunctions and disjunctions in the transition function formulas in $\mathcal{A}_{\psi'}$, and incrementing each priority by one), and fresh states $q_{\vec{a}/\vec{x}}$ with priority 1 for each local assignment $\vec{a}/\vec{x}$. For each local assignment $\vec{a}/\vec{x}$, the designated initial state is $q_{\vec{a}/\vec{x}}$. From state $q_{\vec{a}/\vec{x}}$, Adam is given a choice whether to move to the initial state of $\mathcal{A}_\alpha^{\vec{a}/\vec{x}}$ or to the initial state of the dual of $\mathcal{A}_{\psi'}^{\vec{a}/\vec{x}}$. The idea is that Adam selects which of the conjuncts to challenge Eve on.

  The state set of $\mathcal{A}_\psi$ is of size at most
  $$\begin{aligned} &f(m_\alpha) \cdot 2^{f(Kr_\alpha)} + f(m_{\psi'}) \cdot 2^{f(Kr_{\psi'})} + K^k \\ &\leq 2^{f(Kr_\psi)}(f(m_\alpha) + f(m_{\psi'}) + 1) \\ &\leq 2^{f(Kr_\psi)} f(m_\alpha + m_{\psi'} + 1) \leq N_\psi. \end{aligned}$$
  The number of priorities is linear in the size of ψ, since it is at most the sum of the number the priorities in the subautomata for α and ψ' (which by the inductive hypothesis were linear in the size of these subformulas).
- Suppose ψ is a disjunction $\psi_1 \vee \cdots \vee \psi_s$. Construct $\mathcal{A}_\psi$ by taking the disjoint union of the $\mathcal{A}_{\psi_i}$ and fresh states $q_{\vec{a}/\vec{x}}$ with priority 1 for each local assignment $\vec{a}/\vec{x}$. For each local assignment $\vec{a}/\vec{x}$, the designated initial state is $q_{\vec{a}/\vec{x}}$. In state $q_{\vec{a}/\vec{x}}$, Eve chooses which $\mathcal{A}_{\psi_i}^{\vec{a}/\vec{x}}$ to simulate. The number of states of $\mathcal{A}_\psi$ is at most
  $$\begin{aligned} &f(m_{\psi_1}) \cdot 2^{f(Kr_{\psi_1})} + \cdots + f(m_{\psi_s}) \cdot 2^{f(Kr_{\psi_s})} + K^k \\ &\leq 2^{f(Kr_\psi)}(f(m_{\psi_1}) + \cdots + f(m_{\psi_s}) + 1) \\ &\leq 2^{f(Kr_\psi)} f(m_{\psi_1} + \cdots + m_{\psi_s} + 1) \leq N_\psi. \end{aligned}$$
  The number of priorities is linear in the size of ψ, since it is at most the sum of the number of priorities in the subautomata for ψ_1 to ψ_s (which by the inductive hypothesis were linear in the size of these subformulas).
- Suppose $\psi(\vec{x})$ is a CQ
  $$\exists y_1 \ldots y_t(\alpha_1(\vec{z}_1) \wedge \cdots \wedge \alpha_s(\vec{z}_s))$$
  where each $\vec{z}_i$ is a tuple of variables coming from $\vec{x}$ and $y_1, \ldots, y_t$, and each α_i is an atom over $\sigma_{\mathcal{B}} \cup \sigma_{\mathcal{D}}$. This is

a specific case, but it is helpful for handling the general CQ-shaped formulas in the next point.

We start by defining an automaton that runs on trees with free variable markers for $\vec{x}$ and $y_1 \dots y_t$. For $1 \leq i \leq s$, let $\mathcal{B}_{\alpha_i}$ be the automaton for α_i described in the base cases above that runs on trees with the free variable markers for $\vec{x}$ and $y_1 \dots y_t$. Let $\mathcal{C}$ be the automaton obtained by taking the disjoint union of $\mathcal{B}_{\alpha_1}, \dots, \mathcal{B}_{\alpha_s}$, and an automaton checking that there is precisely one free variable marker for $y_1 \dots y_t$, and adding a new initial state with priority 1 from which Adam can choose which of these subautomata to simulate. Thus, $\mathcal{C}$ is a 2-way alternating automaton with number of states linear in $Ks \leq Kr_\psi$, and two priorities; it checks that the body of the CQ holds in a tree with all of the free variable markers present.

We can then convert $\mathcal{C}$ to an equivalent nondeterministic parity tree automaton $\mathcal{C}'$ using Theorem 11, with an exponential blow-up in the number of states, and a linear blow-up in the number of priorities. After this step, the number of states is exponential in Kr_ψ.

Next, we take the projection of $\mathcal{C}'$ on the free variable markers for $y_1 \dots y_t$ to obtain $\mathcal{B}_\psi$: that is, $\mathcal{B}_\psi$ simulates $\mathcal{C}'$ while guessing the markers for the variables $y_1 \dots y_t$. This is an automaton for ψ, but it runs on trees with markers for the free variables $\vec{x}$.

For each local assignment $\vec{a}/\vec{x}$, we can then apply the localization theorem (Theorem 12) to the set of predicates of the form V_{a_i/x_i}, and eliminate the dependence on any other V_{c/x_i} for $c \neq a_i$ by always assuming these predicates do not hold. This results in a localized automaton $\mathcal{A}_\psi^{\vec{a}/\vec{x}}$ that no longer relies on free variable markers for $\vec{x}$. By Theorem 12, there is only a linear blow-up in the number of states and number of priorities, so after this step the number of states is exponential in Kr_ψ.

Finally, we take $\mathcal{A}_\psi$ to be the disjoint union of the $\mathcal{A}_\psi^{\vec{a}/\vec{x}}$ over all local assignments $\vec{a}/\vec{x}$; the designated initial state for each localization is the initial state for $\mathcal{A}_\psi^{\vec{a}/\vec{x}}$. Since there are at most K^k localizations, the number of states in $\mathcal{A}_\psi$ is still exponential in Kr_ψ, which can be assumed to be less than N_ψ by the choice of f. The number of priorities is a constant independent of ψ.

- Suppose ψ is a CQ-shaped formula of the form

$$\delta[Y_1 := \varphi_1, \dots, Y_n := \varphi_n]$$

where δ is a CQ over $\sigma \cup \{Y_1, \dots, Y_n\}$ and $\varphi_i \in \mathsf{BaseGNF}$. The inductive hypothesis yields $\mathcal{A}_{\varphi_i}$ for each of the φ_i. Let $\mathcal{N}$ be the automaton for the CQ δ obtained using a similar approach as the previous case. Note that this automaton runs on trees with a valuation for the free second-order variables Y_i marked on the tree. These free variables represent base-guarded relations (i.e. relations in which each tuple in the relation is base-guarded), since it is guaranteed that for each Y_i atom, there is a $\sigma_\mathcal{B}$-atom or $\sigma_\mathcal{B}$-guardedness predicate in δ that contains its free variables.

To construct $\mathcal{A}_\psi$, take the disjoint union of $\mathcal{N}$, $\mathcal{A}_{\varphi_1}, \dots \mathcal{A}_{\varphi_n}$. For each localization $\vec{a}/\vec{x}$, the designated initial state is the initial state for $\vec{a}/\vec{x}$ coming from $\mathcal{N}$. The idea is that $\mathcal{A}_\psi$ starts by simulating $\mathcal{N}$, but with Eve guessing valuations for Y_i. This is where it is important that the Y_i are $\sigma_\mathcal{B}$-guarded relations: since any Y_i-fact must be about a $\sigma_\mathcal{B}$-guarded set of elements, these elements must appear together in some node of the tree, so Eve can guess an annotation of the tree that indicates that Y_i holds of these elements. Adam can either accept Eve's guesses of the valuation and continue the simulation of $\mathcal{N}$, or can challenge one of Eve's assertions of Y_i by launching the appropriate localized version of φ_i. That is, if Eve guesses that $Y_i(\vec{z}_i)$ holds of $\vec{b}$ at v, then Adam could challenge this by launching $\mathcal{A}_{\varphi_i}^{\vec{b}/\vec{z}_i}$ starting from v. This is where it is crucial that we have localized automata for these subformulas and for all possible local assignments that can be launched from internal nodes when Adam challenges one of Eve's guesses: in particular, note that the same $\mathcal{A}_{\varphi_i}^{\vec{b}/\vec{z}_i}$ can be launched for different initial localizations $\vec{a}/\vec{x}$.

By the inductive hypothesis, each $\mathcal{A}_{\varphi_i}$ automaton has at most $f(m_{\varphi_i}) \cdot 2^{f(Kr_{\varphi_i})}$ states, and number of priorities linear in m_{φ_i}. Likewise, the automaton $\mathcal{N}$ for δ has two priorities and number of states exponential in Kr_δ, which we can assume to be at most at most $2^{f(Kr_\delta)}$.

Hence, the number of priorities in $\mathcal{A}_\psi$ is linear in m_ψ, and the number of states in $\mathcal{A}_\psi$ is at most

$$\begin{aligned} &2^{f(Kr_\delta)} + f(m_{\varphi_1}) \cdot 2^{f(Kr_{\varphi_1})} + \dots \\ &\qquad + f(m_{\varphi_n}) \cdot 2^{f(Kr_{\varphi_n})} \\ &\leq 2^{f(Kr_\psi)}(1 + f(m_{\varphi_1}) + \dots + f(m_{\varphi_n})) \leq N_\psi. \end{aligned}$$

This concludes the inductive cases.

Overall size. We have argued that each automaton has at most N_ψ states and the number of priorities at most linear in ψ. It remains to argue that the overall size of $\mathcal{A}_\psi$ is at most exponential in $|\sigma| \cdot N_\psi$. The size of the priority mapping is at most polynomial in N_ψ. The size of the alphabet is exponential in $|\sigma| \cdot K^k$, which is at most exponential in $|\sigma| \cdot N_\psi$. For each state and alphabet symbol, the size of the corresponding transition function formula can always be kept of size at most exponential in N_ψ. Hence, the overall size of the transition function is at most exponential in $|\sigma| \cdot N_\psi$. Thus, the overall size of $\mathcal{A}_\psi$ is at most exponential in $|\sigma| \cdot N_\psi$.

It can be checked that the running time of the construction is polynomial in the size of the constructed automaton, and hence is also exponential in $|\sigma| \cdot N_\psi$. □

We must also construct an automaton that checks that the input tree is consistent, and actually represents a set of facts $\mathcal{F}$ such that $\mathcal{F} \supseteq \mathcal{F}_0$ and where every R^+-fact in $\mathcal{F}_0$ is actually witnessed by some path of R-facts in $\mathcal{F}$. For notational simplicity in the statement of the lemma, we assume that the element names in $\mathcal{F}_0$ are used as the names in U for the root of the consistent trees (but this is only a technicality).

Lemma 4. *We can construct a 2-way alternating parity tree automaton $\mathcal{A}_{\mathcal{F}_0}$ in time doubly exponential in $|\sigma| \cdot K$, such that for all trees T,*

$$\begin{aligned} & \mathcal{A}_{\mathcal{F}_0} \textit{ accepts } T \\ \textit{iff} \quad & T \textit{ is consistent and for all facts } S(\vec{c}) \in \mathcal{F}_0, \\ & \mathsf{decode}(T), [\epsilon, \vec{c}] \textit{ satisfies } S(\vec{x}). \end{aligned}$$

when $R^+ \in \sigma_{\mathcal{D}}$ is interpreted as the transitive closure of $R \in \sigma_{\mathcal{B}}$. The number of states is at most exponential in $|\sigma| \cdot K$, the number of priorities is two, and the overall size is at most doubly exponential in $|\sigma| \cdot K$.

Proof. The automaton is designed to allow Adam to challenge some consistency condition or a particular fact $S(\vec{c})$ in $\mathcal{F}_0$.

It is straightforward to design suitable automata checking each consistency condition, so suppose Adam challenges some fact in $\mathcal{F}_0$. Then the automaton simply launches $\mathcal{A}^{\vec{c}/\vec{x}}_{S(\vec{x})}$ (obtained from Lemma 3) from the root. Note that in case $S(\vec{c})$ is some $R^+(c_1, c_2)$, this R-path witnessing this fact may require elements outside of $\mathsf{elems}(\mathcal{F}_0)$ even though c_1 and c_2 are names of elements in $\mathcal{F}_0$.

The number of states is exponential in $|\sigma| \cdot K$, and the overall size is at most doubly exponential in $|\sigma| \cdot K$. Only two priorities are needed. □

We can now conclude the proof of Theorem 2. Recall that φ is in $\mathsf{BaseGNF}$ and $\mathcal{F}_0$ is some finite set of facts. Without loss of generality, we can assume that $|\varphi| \cdot |\mathcal{F}_0| \geq |\sigma|$. We construct the normal form φ' equivalent to φ in exponential time using Proposition 5. Although the size of φ' can be exponentially larger than φ, the width and CQ-rank is at most $|\varphi|$, so we can apply Lemma 3 to construct an automaton for φ' (and hence φ) in time doubly exponential in $|\varphi| \cdot |\mathcal{F}_0|$. However, the number of states and priorities in this automaton is at most singly exponential in $|\varphi| \cdot |\mathcal{F}_0|$, and the number of priorities is linear in $|\varphi|$. By taking the intersection of this automaton from Lemma 3 with the automaton for $\mathcal{F}_0$ and consistency from Lemma 4, we have a 2-way alternating parity tree automaton $\mathcal{A}_{\varphi,\mathcal{F}_0}$ of the desired size that has a non-empty language iff φ is satisfiable. This concludes the proof of Theorem 2.

D Base-guarded-interface tree decompositions for BaseGNF

We prove the following result:

Proposition 11. *Every sentence φ in* BaseGNF *has base-guarded-interface k-tree-like witnesses for some $k \leq |\varphi|$.*

That is, for every sentence φ in BaseGNF and for every finite set of facts $\mathcal{F}_0$, if there is some $\mathcal{F} \supseteq \mathcal{F}_0$ satisfying φ then there is such an $\mathcal{F}$ that has a $\mathcal{F}_0$-rooted $(k-1)$-width base-guarded-interface tree decomposition.

The result and proof of Proposition 11 is very similar to Proposition 1. However, unlike Proposition 1, we do not interpret the distinguished relations in a special way here. This allows us to prove the stronger base-guarded-interface property about the corresponding tree decompositions, which will be important for later arguments (e.g., Proposition 3 and Theorem 4).

We first consider a variant of the GN^k bisimulation game defined earlier in Appendix B. The positions in the game are the same as before:

i) partial isomorphisms $f : \mathcal{F}\restriction_X \to \mathcal{G}\restriction_Y$ or $g : \mathcal{G}\restriction_Y \to \mathcal{F}\restriction_X$, where $X \subset \mathsf{elems}(\mathcal{F})$ and $Y \subset \mathsf{elems}(\mathcal{G})$ are both finite and are $\sigma_{\mathcal{B}}$-guarded;

ii) partial rigid homomorphisms $f : \mathcal{F}\restriction_X \to \mathcal{G}\restriction_Y$ or $g : \mathcal{G}\restriction_Y \to \mathcal{F}\restriction_X$, where $X \subset \mathsf{elems}(\mathcal{F})$ and $Y \subset \mathsf{elems}(\mathcal{G})$ are both finite and are of size at most k.

However, the rules of the game are different.

From a type (i) position h, Spoiler must choose a finite subset $X \subset \mathsf{elems}(\mathcal{F})$ or a finite subset $Y \subset \mathsf{elems}(\mathcal{G})$, in either case of size at most k, upon which Duplicator must respond by a partial rigid homomorphism with domain X or Y accordingly, mapping it into the other set of facts in a manner consistent with h. (This is the same as before).

In a type (ii) position h, Spoiler is only allowed to select some base-guarded subset X' of $\mathsf{Dom}(h)$, and then the game proceeds from the type (i) position obtained by restricting h to this base-guarded subset.

Thus, the game strictly alternates between type (ii) positions and base-guarded positions of type (i). We call this a *base-guarded-interface GN^k bisimulation game*, since the interfaces (i.e. shared elements) between the domains of consecutive positions must be base-guarded. We can then show:

Proposition 12. *Let $\varphi \in$ BaseGNF^k in normal form.*

If Duplicator has a winning strategy in the base-guarded-interface GN^k bisimulation game between $\mathcal{F}$ and $\mathcal{G}$ starting from a type (i) position $\vec{a} \mapsto \vec{b}$ and $\mathcal{F}$ satisfies $\varphi(\vec{a})$, then $\mathcal{G}$ satisfies $\varphi(\vec{b})$.

Proof. Suppose Duplicator has a winning strategy in the base-guarded-interface GN^k bisimulation game between $\mathcal{F}$ and $\mathcal{G}$.

If φ is a σ-atom $A(\vec{x})$, the result follows from the fact that the position $\vec{a} \mapsto \vec{b}$ is a partial homomorphism.

If φ is a disjunction, the result follows easily from the inductive hypothesis.

Suppose φ is a base-guarded negation $A(\vec{x}) \wedge \neg\varphi'(\vec{x}')$. By definition of BaseGNF, it must be the case that $A \in \sigma_{\mathcal{B}}$ and $\vec{x}'$ is a sub-tuple of $\vec{x}$. Since $\mathcal{F}, \vec{a}$ satisfies φ, we know that $\mathcal{F}, \vec{a}$ satisfies $A(\vec{x})$, which implies (by induction) that $\mathcal{G}, \vec{b}$ also satisfies $A(\vec{x})$. It remains to show that $\mathcal{G}$ satisfies $\neg\varphi'(\vec{x}')$. Assume for the sake of contradiction that it satisfies $\varphi'(\vec{x}')$. Because $\vec{a} \mapsto \vec{b}$ is a type (i) position, we can consider the move in the game where Spoiler switches the domain to the other set of facts, keeps the same set of elements, and then collapses to the base-guarded elements in the subtuple $\vec{b}'$ of $\vec{b}$ corresponding to $\vec{x}'$ in $\vec{x}$. Let $\vec{a}'$ be the corresponding subtuple of $\vec{a}$. Duplicator must still have a winning strategy from this new type (i) position $\vec{b}' \mapsto \vec{a}'$, so the inductive hypothesis ensures that $\mathcal{F}, \vec{a}'$ satisfies $\varphi'(\vec{x}')$, a contradiction.

Finally, suppose φ is a CQ-shaped formula

$$\delta[Y_1 := \varphi_1, \ldots, Y_n := \varphi_n]$$

where δ is a CQ $\exists\vec{y}(\alpha_1 \wedge \cdots \wedge \alpha_j)$ over $\sigma \cup \{Y_1, \ldots, Y_n\}$ and φ_i is in normal form BaseGNF^k. We are assuming that $\mathcal{F}, \vec{a}$ satisfies φ. Hence, there is some $\vec{c} \in \mathsf{elems}(\mathcal{F})$ such that $\mathcal{F}, \vec{a}, \vec{c}$ satisfies $(\alpha_1 \wedge \cdots \wedge \alpha_j)[Y_1 := \varphi_1, \ldots, Y_n := \varphi_n]$. Because the width of φ is at most k, we know that the combined number of elements in $\vec{a}$ and $\vec{c}$ is at most k. Hence, we can consider the move in the game where Spoiler selects the elements in $\vec{a}$ and $\vec{c}$. Duplicator must respond with some $\vec{d} \in \mathsf{elems}(\mathcal{G})$ such that $\vec{a}\vec{c} \mapsto \vec{b}\vec{d}$ is a partial rigid homomorphism, a type (ii) position. Now consider the possible conjuncts in this CQ-shaped formula. Conjuncts that are σ-atoms must be satisfied in $\mathcal{G}, \vec{b}\vec{d}$ since $\vec{a}\vec{c} \mapsto \vec{b}\vec{d}$ is a partial homomorphism with respect to σ. For the conjuncts φ_i corresponding to Y_i, we can consider Spoiler's restriction of $\vec{a}\vec{c}$ to the elements used by this conjunct, and the corresponding restriction of $\vec{b}\vec{d}$. This is a valid move to a type (i) position, since the definition of BaseGNF requires that these non-atomic conjuncts are base-guarded. Moreover, this new position witnesses the satisfaction of that conjunct in $\mathcal{F}$. Since Duplicator must still have a winning strategy from this new type (i) position, the inductive hypothesis implies that this conjunct is also satisfied in $\mathcal{G}$. Since this is true for all conjuncts in the CQ-shaped formula, $\mathcal{G}, \vec{b}$ satisfies φ as desired. □

We then use a variant of the unravelling based on this game. The *base-guarded-interface GN^k-unravelling* $\mathcal{F}^k_{\mathcal{B}}$ is defined in a similar fashion to the GN^k-unravelling, except it uses only sequences $\Pi \cap \{X_0 \ldots X_n \text{ : for all } i \geq 1, X_i \cap X_{i+1} \text{ is } \sigma_{\mathcal{B}}\text{-guarded}\}$. This unravelling has an $\mathcal{F}_0$-rooted base-guarded-interface tree decomposition of width $k-1$. Moreover:

Proposition 13. *Duplicator has a winning strategy in the base-guarded-interface GN^k bisimulation game between $\mathcal{F}$ and $\mathcal{F}^k_{\mathcal{B}}$.*

Proof. The proof is similar to Proposition 7. The delicate part of the argument is when Spoiler selects some new elements X' in $\mathcal{F}$ starting from a safe position f (for which there is some π such that $f(a) = [\pi, a]$ for all $a \in \mathsf{Dom}(f)$). We need to show that $[\pi', a']$ for $a' \in X'$ and $\pi' = \pi \cdot X'$

is well-defined in $\mathcal{F}^k_{\mathcal{B}}$. This is well-defined only if the overlap between the elements in π and π' is base-guarded. But because the base-guarded-interface GN^k bisimulation game strictly alternates between type (i) and (ii) positions, Spoiler can only select new elements X' in a type (i) position, so the overlap satisfies this requirement. The remainder of the proof is the same as in Proposition 7. □

We can conclude the proof of Proposition 11 as follows. Assume that $\mathcal{F}$ is a set of facts that satisfies φ. By Proposition 5, we can convert to an equivalent $\varphi' \in \mathsf{BaseGNF}^k$ in normal form with width $k \leq |\varphi|$. Since $\mathcal{F}$ satisfies φ', Propositions 13 and 12 imply that $\mathcal{F}^k_{\mathcal{B}}$ also satisfies $\varphi' \in \mathsf{BaseGNF}^k$. Hence, we can conclude that the unravelling $\mathcal{F}^k_{\mathcal{B}}$ is a base-guarded-interface k-tree-like witness for φ.

E Reduction of QAlin to QA (Proof of Lemmas 1 and 2 for Proposition 3)

Recall the statement of Proposition 3, which describes the reduction from QAlin to QA:

> **Proposition 3.** For any finite set of facts $\mathcal{F}_0$, constraints $\Sigma \in \mathsf{BaseCovGNF}$, and base-covered UCQ Q, we can compute $\mathcal{F}'_0$ and $\Sigma' \in \mathsf{BaseGNF}$ in PTIME such that $\mathsf{QAlin}(\mathcal{F}_0, \Sigma, Q)$ iff $\mathsf{QA}(\mathcal{F}'_0, \Sigma', Q)$.

Specifically, $\mathcal{F}'_0$ is $\mathcal{F}_0$ together with facts $G(a, b)$ for every pair $a, b \in \mathsf{elems}(\mathcal{F}_0)$, where G is some fresh binary base relation. Σ' consists of Σ together with the k-guardedly linear axioms for each distinguished relation, where k is $\max(|\Sigma \wedge \neg Q|, \mathsf{arity}(\sigma \cup \{G\}))$.

Recall that the *k-guardedly linear axioms* require that each binary relation $<$ is:

- guardedly total: $\forall xy((\mathsf{guarded}_{\sigma_{\mathcal{B}} \cup \{G\}}(x, y) \wedge x \neq y) \rightarrow x < y \vee y < x)$
- irreflexive: $\neg \exists x(x < x)$
- k-guardedly transitive: for $1 \leq l \leq k-1$: $\neg \exists xy(\psi_l(x, y) \wedge \mathsf{guarded}_{\sigma_{\mathcal{B}} \cup \{G\}}(x, y) \wedge \neg(x < y))$ and, for $1 \leq l \leq k$: $\neg \exists x(\psi_l(x, x) \wedge x = x \wedge \neg(x < x))$

where:

- $\mathsf{guarded}_{\sigma_{\mathcal{B}} \cup \{G\}}(x, y)$ is the formula expressing that x, y is base-guarded (an existentially-quantified disjunction over all possible base-guards containing x and y);
- $\psi_1(x, y)$ is just $x < y$; and
- $\psi_l(x, y)$ for $l \geq 2$ is: $\exists x_2 \ldots x_l(x < x_2 \wedge \cdots \wedge x_l < y)$.

The idea is that these axioms are strong enough to enforce conditions about transitivity and irreflexivity within "small" sets of elements — intuitively, within sets of at most k elements that appear together in some bag of a $(k-1)$-width tree decomposition.

The proof of the correctness of the reduction is described in the body of the paper, but relies on Lemmas 1 and 2, which we prove now.

E.1 Proof of Lemma 1

Recall the statement:

> **Lemma 1.** The sentence $\Sigma' \wedge \neg Q$ has base-guarded-interface k-tree-like witnesses for $k = \max(|\Sigma \wedge \neg Q|, \mathsf{arity}(\sigma \cup \{G\}))$.

By Proposition 5 and Proposition 11, $\Sigma \wedge \neg Q$ has a base-guarded-interface k-tree-like witness for $k = |\Sigma \wedge \neg Q|$.

To prove this lemma, then, it suffices to argue that the k-guardedly linear axioms can also be written in normal form BaseGNF with width at most k.

The guardedly total axiom is written in normal form BaseGNF as

$$\neg \exists xy(\mathsf{guarded}_{\sigma_{\mathcal{B}} \cup \{G\}}(x, y) \wedge \neg(x = y \vee x < y \vee y < x))$$

with width at most k. The irreflexive axiom is already written in normal form BaseGNF with width at most k. For the k-guardedly transitive axioms, note that $\psi_l(x, y)$ has width $l+1$ and $\psi_l(x, x)$ has width l, so that each of the k-guardedly transitive axioms has width at most k: this uses the fact that the width of the $\mathsf{guarded}_{\sigma_{\mathcal{B}} \cup \{G\}}$-atoms have arity at most $\mathsf{arity}(\sigma \cup \{G\})$, and we know that $k \geq \mathsf{arity}(\sigma \cup \{G\})$

Therefore, unlike the property of being a linear order, the k-guardedly linear restriction can be expressed in BaseGNF, and can even be written in normal form BaseGNF of width at most k. Overall, this means that if $\Sigma \wedge \neg Q$ has width at most k when converted into normal form then $\Sigma' \wedge \neg Q$ also has width at most k. Hence, the sentence $\Sigma' \wedge \neg Q$ has base-guarded-interface k-tree-like witnesses for $k = |\Sigma \wedge \neg Q|$, by Proposition 11.

E.2 Proof of Lemma 2

Recall the statement:

> **Lemma 2.** If there is $\mathcal{F}' \supseteq \mathcal{F}'_0$ that satisfies $\Sigma' \wedge \neg Q$ and has a $\mathcal{F}'_0$-rooted base-guarded-interface $(k-1)$-width tree decomposition, then there is $\mathcal{F}'' \supseteq \mathcal{F}'$ that satisfies $\Sigma' \wedge \neg Q$ where each distinguished relation is a strict linear order.

We start with some auxiliary lemmas about base-guarded-interface tree decompositions.

Transitivity lemma. We first prove a result about transitivity for sets of facts with base-guarded-interface tree decompositions.

Lemma 5. *Suppose $\mathcal{F}'$ is a set of facts with a $\mathcal{F}'_0$-rooted $(k-1)$-width base-guarded-interface tree decomposition $(T, \mathsf{Child}, \lambda)$. If $\mathcal{F}'$ is k-guardedly transitive with respect to binary relation $<$, and there is a $<$-path $a_1 \ldots a_n$ where the pair $\{a_1, a_n\}$ is base-guarded, then $a_1 < a_n \in \mathcal{F}'$.*

Proof. Suppose there is an $<$-path $a_1 \ldots a_n$ and that the pair $\{a_1, a_n\}$ is base-guarded, with v a node where a_1, a_n appear together. We can assume that $a_1 \ldots a_n$ is a minimal $<$-path between a_1 and a_n, so there are no repeated intermediate elements. Consider a minimal subtree T' of T containing v and containing all of the elements $a_1 \ldots a_n$. We proceed by induction on the length of the path and on the number of nodes of T' (with the lexicographic order on this pair) to show that $a_1 < a_n$ is in $\mathcal{F}'$.

If all elements $a_1 \ldots a_n$ are represented at v, then either (i) all elements are in the root or (ii) the elements are in some internal node. For (i), by construction of $\mathcal{F}'_0$, every pair of elements in $a_1 \ldots a_n$ is guarded (by G). Hence, repeated application of the axiom

$$\forall xyz((x < z \wedge z < y \wedge \mathsf{guarded}_{\sigma_{\mathcal{B}} \cup \{G\}}(x, y)) \rightarrow x < y)$$

(which is part of the k-guardedly transitive axioms) is enough to ensure that $a_1 < a_n$ holds. For (ii), since the bag size of an internal node is at most k, we must have $n \leq k$, in which case an application of the k-guardedly transitive axiom to the guarded pair $\{a_1, a_n\}$ ensures that $a_1 < a_n$ holds. This covers the base case of the induction.

Otherwise, there must be some $1 \leq i < j \leq n$ such that a_i and a_j are represented at v, but $a_{i'}$ is not represented at v for $i < i' < j$ (in particular a_{i+1} is not represented at v). We claim that a_i and a_j must be in an interface together.

We say a_{i+1} is *represented in the direction of* v' if v' is a child of v and a_{i+1} is represented in the subtree rooted at v', or v' is the parent of v and a_{i+1} is represented in the tree obtained from T' by removing the subtree rooted at v. Note that by definition of a tree decomposition, since a_{i+1} is not represented at v, it can only be represented in at most one direction.

Let v_{i+1} be the neighbor (child or parent) of v such that a_{i+1} is represented in the direction of v_{i+1}. It is straightforward to show that a_i and a_j must both be represented in the subtree in the direction of v_{i+1} in order to witness the facts $a_i < a_{i+1}$ and $a_{j-1} < a_j$. But a_i and a_j are both in v, so they must both be in v_{i+1}. Hence, a_i and a_j are in the interface between v and v_{i+1}.

If this is an interface with the root node, then the pair a_i, a_j is base-guarded (by definition of $\mathcal{F}'_0$). Otherwise, the definition of base-guarded-interface tree decompositions ensures that they are base-guarded.

Hence, we can apply the inductive hypothesis to the path $a_i \dots a_j$ and the subtree T'' of T' in the direction of v_{i+1} to conclude that $a_i < a_j$ holds (we can apply the inductive hypothesis because T'' is smaller than T' as we removed v, and $a_i \dots a_j$ is no longer than $a_1 \dots a_n$). If $i = 1$ and $j = n$, then we are done. If not, then we can apply the inductive hypothesis to the new, strictly shorter path $a_1 \dots a_i a_j \dots a_n$ in T' and conclude that $a_1 < a_n$ is in $\mathcal{F}'$ as desired. □

Cycles lemma. We next show that within base-guarded-interface tree decompositions, k-guarded transitivity and irreflexivity imply cycle-freeness.

Lemma 6. *Suppose $\mathcal{F}'$ is a set of facts with a $\mathcal{F}'_0$-rooted $(k-1)$-width base-guarded-interface tree decomposition $(T, \mathsf{Child}, \lambda)$. If $\mathcal{F}'$ is k-guardedly transitive and irreflexive with respect to $<$, then $<$ in $\mathcal{F}'$ cannot have a cycle.*

Proof. Suppose for the sake of contradiction that there is a cycle $a_1 \dots a_n a_1$ in $\mathcal{F}'$ using relation $<$. Take a minimal length cycle.

If elements $a_1 \dots a_n$ are all represented in a single node in T, then either (i) all elements are in the root or (ii) the elements are in some internal node. For (i), by construction of $\mathcal{F}'_0$, every pair of elements in $a_1 \dots a_n$ is guarded (by G). Hence, repeated application of the axiom

$$\forall xyz((x < z \wedge z < y \wedge \mathsf{guarded}_{\sigma_{\mathcal{B}} \cup \{G\}}(x, y)) \rightarrow x < y)$$

(which is part of the k-guardedly transitive axioms) would force $a_1 < a_1$ to be in $\mathcal{F}'$, which would contradict irreflexivity. Likewise, for (ii), since the bag size of an internal node is at most k, we must have $n \leq k$, so we can apply the k-guardedly transitive axioms to deduce $a_1 < a_1$, which contradicts irreflexivity.

Even if this is not the case, then since $a_n < a_1$ holds, there must be some node v in which both a_1 and a_n are represented. Since not all elements are represented at v, however, there is $1 \leq i < j \leq n$ such that a_i and a_j are represented at v, but $a_{i'}$ is not represented at v for $i < i' < j$. We claim that a_i and a_j must be in an interface together. Observe that a_{i+1} is not represented at v. Let v_{i+1} be the neighbor of v such that a_{i+1} is represented in the subtree in the direction of v_{i+1}. It is straightforward to show that a_i and a_j must both be represented in the subtree of T' in the direction of v_{i+1} in order to witness the facts $a_i < a_{i+1}$ and $a_{j-1} < a_j$. But a_i and a_j are both in v, so they must both be in v_{i+1}. Hence, a_i and a_j are in the interface between v and v_{i+1}. If this is an interface with the root node, then the pair a_i, a_j is base-guarded (by definition of $\mathcal{F}'_0$); otherwise, the definition of base-guarded-interface tree decomposition ensures that they are base-guarded. By Lemma 5 this means that $a_i < a_j$ holds. Hence, there is a strictly shorter cycle $a_1 \dots a_i a_j \dots a_n a_1$, contradicting the minimality of the original cycle. □

Base-coveredness lemma. Lastly, we note that adding only facts about unguarded sets of elements cannot impact BaseCovGNF constraints. This is where we are utilizing the base-coveredness assumption.

Lemma 7. *Let $\mathcal{F}'' \supseteq \mathcal{F}'$ with additional facts about distinguished relations, but no new facts about base-guarded tuples of elements. Let $\varphi(\vec{x}) \in \mathsf{BaseCovGNF}$. If $\mathcal{F}', \vec{a}$ satisfies $\varphi(\vec{x})$ then $\mathcal{F}'', \vec{a}$ satisfies $\varphi(\vec{x})$.*

Proof. We assume without loss of generality that φ is in normal form BaseCovGNF.

Let $\mathsf{BaseCovGNF}^+$ (respectively, $\mathsf{BaseCovGNF}^-$) denote the normal form BaseGNF formulas where the covering requirements (distinguished atoms in CQ-shaped subformulas are appropriately base-guarded) are required for positively occurring (respectively, negatively occurring) CQ-shaped formulas. Observe that $\mathsf{BaseCovGNF} = \mathsf{BaseCovGNF}^-$.

We prove a slightly stronger result:

For $\varphi(\vec{x}) \in \mathsf{BaseCovGNF}^-$:
$\mathcal{F}', \vec{a}$ satisfies $\varphi(\vec{x})$ implies $\mathcal{F}'', \vec{a}$ satisfies $\varphi(\vec{x})$.
For $\varphi(\vec{x}) \in \mathsf{BaseCovGNF}^+$:
$\mathcal{F}'', \vec{a}$ satisfies $\varphi(\vec{x})$ implies $\mathcal{F}', \vec{a}$ satisfies $\varphi(\vec{x})$.

We proceed by induction on the structure of φ. The base case for a $\sigma_{\mathcal{B}}$ atom is immediate. The inductive case for disjunction is also immediate.

Suppose $\varphi := A(\vec{x}) \wedge \neg\varphi'(\vec{x})$, and $\varphi \in \mathsf{BaseCovGNF}^-$. If $\mathcal{F}', \vec{a}$ satisfies $\varphi(\vec{x})$, then $\mathcal{F}'', \vec{a}$ satisfies $A(\vec{x})$ by the inductive hypothesis. We must also have $\mathcal{F}'', \vec{a}$ satisfies $\neg\varphi'(\vec{x})$, for if not, then $\mathcal{F}'', \vec{a}$ satisfies $\varphi'(\vec{x})$ (for $\varphi' \in \mathsf{BaseCovGNF}^+$), so the inductive hypothesis implies that $\mathcal{F}', \vec{a}$ satisfies $\varphi'(\vec{x})$, a contradiction. Hence, $\mathcal{F}'', \vec{a}$ satisfies $\varphi(\vec{x})$ as desired. The proof is similar starting from $\varphi \in \mathsf{BaseCovGNF}^+$.

That leaves only the general CQ-shaped formula case. Suppose $\varphi := \exists\vec{y}(\beta_1(\vec{x}_1\vec{y}_1) \wedge \dots \wedge \beta_j(\vec{x}_j\vec{y}_j))$, where $\vec{x}_i$ and $\vec{y}_i$ denote the tuple of variables from $\vec{x}$ and $\vec{y}$ used by β_i.

If φ is in $\mathsf{BaseCovGNF}^-$, then there are no covering restrictions for this CQ since it appears positively. If $\mathcal{F}', \vec{a}$ satisfies $\varphi(\vec{x})$, then there exists $\vec{b}$, such that $\mathcal{F}', \vec{a}_i\vec{b}_i$ satisfies β_i

for all $1 \leq i \leq j$. But $\mathcal{F}'' \supseteq \mathcal{F}'$, so this witness $\vec{b}$ and the corresponding facts also appear in $\mathcal{F}''$, and $\mathcal{F}'', \vec{a}$ satisfies φ.

If φ is in $\mathsf{BaseCovGNF}^+$ and $\mathcal{F}'', \vec{a}$ satisfies φ, then there is some $\vec{b}$ such that $\mathcal{F}'', \vec{a}_i\vec{b}_i$ satisfies β_i for all $1 \leq i \leq j$. It suffices to show that $\mathcal{F}', \vec{a}_i\vec{b}_i$ satisfies β_i for all $1 \leq i \leq j$. Consider the possible β_i. If β_i is a $\sigma_{\mathcal{B}}$-atom, then $\mathcal{F}', \vec{a}_i\vec{b}_i$ satisfies β_i, since $\mathcal{F}'$ has the same $\sigma_{\mathcal{B}}$-facts as $\mathcal{F}''$. If β_i is a $\sigma_{\mathcal{D}}$-atom, then the covering requirements ensure that there is some $\sigma_{\mathcal{B}}$-atom β_j in φ including at least the free variables $\vec{x}_i\vec{y}_i$ of β_i. This means $\vec{a}_i\vec{b}_i$ is base-guarded. Since $\mathcal{F}'$ and $\mathcal{F}''$ agree on facts about base-guarded tuples like this, $\mathcal{F}', \vec{a}_i\vec{b}_i$ satisfies β_i. Finally, if β_i is some structurally simpler $\mathsf{BaseCovGNF}$ formula, then the inductive hypothesis ensures that $\mathcal{F}', \vec{a}_i\vec{b}_i$ satisfies β_i. □

Final proof of Lemma 2. We are now ready to prove Lemma 2:

We start with some $\mathcal{F}' \subseteq \mathcal{F}'_0$ satisfying $\Sigma' \wedge \neg Q$ with a $\mathcal{F}'_0$-rooted $(k-1)$-width base-guarded-interface tree decomposition. We prove that there is an extension $\mathcal{F}''$ of $\mathcal{F}'$ satisfying $\Sigma' \wedge \neg Q$ in which each distinguished relation is a strict linear order. Note that because $\mathcal{F}'$ satisfies Σ', we know that $\mathcal{F}'$ is k-guardedly linear.

We present the argument when there is one $<$ in $\sigma_{\mathcal{D}}$ that is not a strict linear order in $\mathcal{F}'$, but the argument is similar if there are multiple distinguished relations like this, as we can handle each distinguished relation independently with the method that we will present. Let $\mathcal{G}$ be the extension of $\mathcal{F}'$ obtained by taking $<$ in $\mathcal{G}$ to be the transitive closure of $<$ in $\mathcal{F}'$. Suppose for the sake of contradiction that there is a $<$-cycle in $\mathcal{G}$. We proceed by induction on the number of facts from $\mathcal{G} \setminus \mathcal{F}'$ used in this cycle. If there are no facts from $\mathcal{G} \setminus \mathcal{F}'$ in the cycle, Lemma 6 yields the contradiction. Otherwise, suppose that there is a cycle involving (a_1, a_n), where (a_1, a_n) is a $<$-fact in $\mathcal{G} \setminus \mathcal{F}'$ coming from facts $(a_1, a_2), \ldots, (a_{n-1}, a_n)$ in $\mathcal{F}'$. By replacing (a_1, a_n) in this cycle with $(a_1, a_2), \ldots, (a_{n-1}, a_n)$, we get a (longer) cycle with fewer facts from $\mathcal{G} \setminus \mathcal{F}'$, which is a contradiction by the inductive hypothesis.

Since $<$ is transitive in $\mathcal{G}$, the relation $<$ in $\mathcal{G}$ must be a strict partial order. We now apply the "order extension principle" or "Szpilrajn extension theorem" [Szpilrajn, 1930]: any strict partial order can be extended to a strict total order. From this, we deduce that $\mathcal{G}$ can be further extended by additional $<$-facts to obtain some $\mathcal{F}''$ where $<$ is a strict total order.

We must prove that $\mathcal{F}'' \supseteq \mathcal{G} \supseteq \mathcal{F}' \supseteq \mathcal{F}'_0$ does not include any new $<$-facts about base-guarded tuples. Suppose for the sake of contradiction that there is a new fact $a < b$ in $\mathcal{F}'' \setminus \mathcal{F}'$, where $\{a, b\}$ is base-guarded in $\mathcal{F}'$. By the guardedly total axiom, it must be the case that there was already $b < a$ in $\mathcal{F}'$, and hence also in $\mathcal{F}''$. But $a < b$ and $b < a$ in $\mathcal{F}''$ would together imply $a < a$ in $\mathcal{F}''$, contradicting the fact that $\mathcal{F}''$ is a strict linear order.

Hence, $\mathcal{F}'$ and $\mathcal{F}''$ agree on all facts about base-guarded tuples. Since Q is base-covered and $\Sigma \in \mathsf{BaseCovGNF}$, $\Sigma \wedge \neg Q \in \mathsf{BaseCovGNF}$. Thus, Lemma 7 guarantees that $\Sigma \wedge \neg Q$ is still satisfied in $\mathcal{F}''$. Since $\mathcal{F}''$ also trivially satisfies all of the k-guardedly linear axioms, it satisfies $\Sigma' \wedge \neg Q$ as required.

This concludes the proof of Lemma 2, and hence the proof of Proposition 3.

F Data complexity upper bounds for transitivity

F.1 Proof of Theorem 3

We begin with the proof of Theorem 3. Recall the statement:

> **Theorem 3.** For any BaseGNF constraints Σ and CQ Q, given a finite set of facts $\mathcal{F}_0$, we can decide $\mathsf{QAtc}(\mathcal{F}_0, \Sigma, Q)$ in CoNP data complexity.

Fix the signature σ.

For a set of σ-facts $\mathcal{F}$, an $\mathcal{F}, k$-*rooted structure* is one that consists of $\mathcal{F}$ unioned with sets of facts $T_{\vec{c}}$ for $\vec{c} \in \mathsf{Dom}(\mathcal{F})^k$ where the domain of $T_{\vec{c}}$ overlaps with the domain of $\mathcal{F}$ only in $\vec{c}$, the facts of $T_{\vec{c}}$ involving only elements of $\vec{c}$ are all present in $\mathcal{F}$, and for two k-tuples $\vec{c}$ and $\vec{c}'$, the domain of $T_{\vec{c}}$ overlaps with the domain of $T_{\vec{c}'}$ only within $\vec{c} \cap \vec{c}'$.

The following proposition follows from Proposition 1.

Proposition 14. *For any set of σ-facts $\mathcal{F}$, if a* BaseGNF *sentence Σ over σ is satisfiable by some set of facts containing $\mathcal{F}$ with relations R_i^+ interpreted as the transitive closure of R_i, then Σ is satisfied (with the same restriction) in an $\mathcal{F}', k$-rooted structure, where k is at most $|\sigma|$ and $\mathcal{F}'$ is a superset of $\mathcal{F}$ that has the same domain.*

Let $\mathsf{FO}(\sigma)$ denote first-order logic over the signature σ. Let $\mathsf{FO}(\sigma \cup \{d_1 \dots d_k\})$ denote first-order logic over the signature σ extended with k new constants, which will be used to represent the overlap elements. Note that formulas in both $\mathsf{FO}(\sigma)$ and $\mathsf{FO}(\sigma \cup \{d_1 \dots d_k\})$ can make use of distinguished R_i^+ relations that are part of σ.

Given an $\mathcal{F}, k$-rooted structure $\mathfrak{A}$, and number j, the *j-abstraction of* $\mathfrak{A}$ is the expansion of $\mathcal{F}$ with relations $P_\tau(x_1 \dots x_k)$ for each $\mathsf{FO}(\sigma \cup \{d_1 \dots d_k\})$ sentence τ of quantifier-rank j, up to logical equivalence (so there are finitely many such relations). $P_\tau(x_1 \dots x_k)$ is interpreted by the set of k-tuples $\vec{c}$ such that $T_{\vec{c}}$ satisfies τ when interpreting the constants in τ by $\vec{c}$. We let $\sigma_{j,k}$ be the signature of the j-abstraction of such structures.

Lemma 8. *For any sentence φ of $\mathsf{FO}(\sigma)$ and any k, there is j having the following property:*

Let $\mathfrak{A}_1$ be an $\mathcal{F}_1, k$-rooted structure for some set of σ-facts $\mathcal{F}_1$, and let $\mathfrak{A}_2$ be an $\mathcal{F}_2, k$-rooted structure for some set of σ-facts $\mathcal{F}_2$, where the interpretations of the R_i^+ relations in each structure are the transitive closure of the corresponding R_i relations. If the j-abstractions of $\mathfrak{A}_1$ and $\mathfrak{A}_2$ agree on all $\mathsf{FO}(\sigma_{j,k})$ sentences of quantifier-rank at most j, then $\mathfrak{A}_1$ and $\mathfrak{A}_2$ agree on φ.

Proof. Let j_φ be the quantifier-rank of φ. We choose $j := j_\varphi \cdot k$. We give a strategy for Duplicator in the j_φ-round standard pebble game for $\mathsf{FO}(\sigma)$ over $\mathfrak{A}_1$ and $\mathfrak{A}_2$. With i moves left to play, we will ensure the following invariants on a game position consisting of a sequence $\vec{p}_1 \in \mathfrak{A}_1$ and $\vec{p}_2 \in \mathfrak{A}_2$:

- Let $\vec{p_1}'$ be the subsequence of $\vec{p}_1$ that comes from $\mathcal{F}_1$ and let $\vec{p_2}'$ be defined similarly for $\vec{p}_2$ and $\mathcal{F}_2$. Then $\vec{p_1}'$ and $\vec{p_2}'$ should form a winning position for Duplicator in the $i \cdot k$ round $\mathsf{FO}(\sigma_{j,k})$ game on the j-abstractions.
- Fix any k-tuple $\vec{c}_1 \in \mathcal{F}_1$ and let $P^1_{\vec{c}_1}$ be the subsequence of $\vec{p}_1$ that lies in $T_{\vec{c}_1}$ within $\mathfrak{A}_1$. Then if $P^1_{\vec{c}}$ is non-empty, $\vec{c}_1$ also lies in $\vec{p}_1$. Let $\vec{c}_2$ be the corresponding k-tuple to $\vec{c}_1$ in $\vec{p}_2$, and let $P^2_{\vec{c}_2}$ be the subsequence of $\vec{p}_2$ that lies in $T_{\vec{c}_2}$ within $\mathfrak{A}_2$. Then $P^1_{\vec{c}_1}$ and $P^2_{\vec{c}_2}$ form a winning position in the i-round pebble game on $T_{\vec{c}_1}$ and $T_{\vec{c}_2}$.

  The analogous property holds fixing any k-tuple $\vec{c}_2 \in \mathcal{F}_2$.

We now explain the strategy of the Duplicator, focusing for simplicity on moves of Spoiler within $\mathfrak{A}_1$, with the strategy on $\mathfrak{A}_2$ being similar. If Spoiler plays within $\mathcal{F}_1$, Duplicator responds using her strategy for the games on the j-abstractions of $\mathcal{F}_1$ and $\mathcal{F}_2$. It is easy to see that the invariant is preserved.

If Spoiler plays an element within a substructure $T_{\vec{c}_1}$ within $\mathfrak{A}_1$ that is already inhabited, then by the invariant $\vec{c}_1$ is pebbled and there is a corresponding $\vec{c}_2$ in $\mathfrak{A}_2$ with substructure $T_{\vec{c}_2}$ of $\mathfrak{A}_2$ such that the pebbles within $T_{\vec{c}_2}$ are winning positions in the game on $T_{\vec{c}_1}$ and $T_{\vec{c}_2}$ with i moves left to play. Thus Duplicator can respond using the strategy in this game from those positions.

Now suppose Spoiler plays an element e_1 within a substructure $T_{\vec{c}_1}$ within $\mathfrak{A}_1$ that is not already inhabited. We first use $\vec{c}_1$ as a sequence of plays for Spoiler in the game on the j-abstractions of $\mathfrak{A}_1$ and $\mathfrak{A}_2$, extending the positions given by $\vec{p_1}'$ and $\vec{p_2}'$. By the inductive invariant, responses of Duplicator exist, and we collect them to get a tuple $\vec{c}_2$. Since a winning strategy in a game preserves atoms, and we have a fact in the j-abstraction corresponding to the j-type of $\vec{c}_1$ in $T_{\vec{c}_1}$, we know that $\vec{c}_2$ must satisfy the same j-type in $T_{\vec{c}_2}$ that $\vec{c}_1$ does in $T_{\vec{c}_1}$. Therefore $\vec{c}_1$ must satisfy the same $\mathsf{FO}(\sigma \cup \{d_1 \dots d_k\})$ sentences of quantifier-rank at most j in $T_{\vec{c}_1}$ as $\vec{c}_2$ does in $T_{\vec{c}_2}$. Thus Duplicator can use the corresponding strategy to respond to e_1 with an e_2 in $T_{\vec{c}_2}$ such that $\{e_1\}$ and $\{e_2\}$ are a winning position in the $i-1$ round pebble game on $T_{\vec{c}_1}$ and $T_{\vec{c}_2}$.

Since the response of Duplicator corresponds to k moves in the game within the j-abstractions, one can verify that the invariant is preserved.

We must verify that this strategy gives a partial isomorphism. Consider a fact F that holds of a tuple $\vec{t}_1$ within $\mathfrak{A}_1$, and let $\vec{t}_2$ be the tuple obtained using this strategy in $\mathfrak{A}_2$. We first consider the case where F is a $\sigma_{\mathcal{B}}$-fact.

- If $\vec{t}_1$ lies completely within some $T_{\vec{c}_1}$, then the last invariant guarantees that $\vec{t}_2$ lies in some $T_{\vec{c}_2}$. The last invariant also guarantees that $\sigma_{\mathcal{B}}$-facts of $\mathfrak{A}_1$ are preserved since such facts must lie in $T_{\vec{c}_1}$, and the corresponding positions are winning in the game between $T_{\vec{c}_1}$ and $T_{\vec{c}_2}$.
- If $\vec{t}_1$ lies completely within $\mathcal{F}_1$, then the first invariant guarantees that the fact is preserved.

By the definition of a rooted structure, the above two cases are exhaustive.

We now consider the case where F is of the form $R_i^+(t_1, t_2)$.

- If t_1 and t_2 both lie in some $T_{\vec{c}_1}$, then we reason as in the first case above.

- If t_1 and t_2 are both in $\mathcal{F}_1$, we reason as in the second case above.
- If t_1 lies in $T_{\vec{c}_1}$, t_2 lies in $T_{\vec{c}_2}$, then t_1 reaches some c_i, $c_i \in \vec{c}_1$, c_i reaches some $c_j \in \vec{c}_2$, and c_j reaches t_2 within $T_{\vec{c}_2}$. Then we use a combination of the first two cases above to conclude that F is preserved. □

From Lemma 8 we easily obtain:

Corollary 6. *Given φ and k there is a number j and a sentence φ' in the language of j-abstractions over σ such that for all sets of facts $\mathcal{F}$, an $\mathcal{F}, k$-rooted structure satisfies φ iff its j-abstraction satisfies φ'.*

We can now put these results together to prove Theorem 3:

Proof. Fixing Q and Σ, we give an NP algorithm for the complement. Let $\varphi = \Sigma \wedge \neg Q$, and $k = |\varphi|$. Let j and φ' be the number and formula guaranteed for φ by Corollary 6.

Let $\mathsf{FO}(\sigma \cup \{d_1 \dots d_k\})$ denote first-order logic over the signature σ of $\Sigma \wedge \neg Q$, together with k constants.

Let Types_j be the collection of assignments of truth values to all $\mathsf{FO}(\sigma \cup \{d_1 \dots d_k\})$ sentences with quantifier-rank at most j such that the conjunction of the corresponding sentences is consistent. Note that the set is finite since j and the signature are fixed.

Given $\mathcal{F}$, guess an extension $\mathcal{F}'$ with additional facts but the same domain. Guess a function f mapping each k-tuple over $\mathcal{F}$ to a $\rho \in \mathsf{Types}_j$, and then for each $\tau \in \mathsf{FO}(\sigma \cup \{d_1 \dots d_k\})$ of quantifier rank at most j, interpret P_τ by the set of tuples $\vec{c}$ such that $\tau \in f(\vec{c})$. Check whether $\mathcal{F}'$ satisfies φ' with these interpretations, and if so return true.

We argue for correctness. If the algorithm returns true with $\mathcal{F}'$ the witness, then create an $\mathcal{F}', k$-rooted structure $\mathfrak{A}$ by picking for each $\vec{c}$ a structure satisfying the sentences in $f(\vec{c})$ with distinguished elements interpreted by $\vec{c}$ (such a structure exists by consistency of $f(\vec{c})$), and letting the remaining domain elements be disjoint from the domain of $\mathcal{F}'$. Note that by construction, $\mathfrak{A}$ has $\mathcal{F}'$ as its j-abstraction. By the choice of j and φ', and the observation above, $\mathfrak{A}$ satisfies $\Sigma \wedge \neg Q$. Thus this structure witnesses that $\mathsf{QAtc}(\mathcal{F}, \Sigma, Q)$ is false.

On the other hand, if $\mathsf{QAtc}(\mathcal{F}, \Sigma, Q)$ is false, then by Proposition 14 we have an extension $\mathcal{F}'$ without adding values to the domain, and an $\mathcal{F}', k$-rooted structure $\mathfrak{A}$ that satisfies $\Sigma \wedge \neg Q$. By the choice of j and φ', the j-abstraction of $\mathfrak{A}$ satisfies φ'. For each $\vec{c}$ in the j-abstraction of $\mathfrak{A}$, the type of $\vec{c}$ must be in Types_j. Hence we can guess collections such that the algorithm returns true. □

F.2 Proof of Theorem 4: PTIME data complexity bound for QAtr

We now turn to the case where our constraints are restricted to BaseCovFGTGDs and deal with QAtr, not QAtc. Recall that Theorem 4 states a PTIME data complexity bound for this case:

> **Theorem 4.** For any fixed BaseCovFGTGD constraints Σ and base-covered UCQ Q, given a finite set of facts $\mathcal{F}_0$, we can decide $\mathsf{QAtr}(\mathcal{F}_0, \Sigma, Q)$ in PTIME data complexity.

The proof will follow from a reduction to traditional QA, similar to the proof of Proposition 3:

Proposition 15. *For any finite set of facts $\mathcal{F}_0$, constraints $\Sigma \in$ BaseCovGNF, and base-covered UCQ Q, we can compute $\mathcal{F}_0'$ and $\Sigma' \in$ GNF in PTIME such that $\mathsf{QAtr}(\mathcal{F}_0, \Sigma, Q)$ iff $\mathsf{QA}(\mathcal{F}_0', \Sigma', Q)$. Furthermore, if Σ is in BaseCovFGTGD then Σ' is in FGTGD.*

Proof. We define $\mathcal{F}_0'$ and Σ' as follows:

- $\mathcal{F}_0'$ is $\mathcal{F}_0$ together with facts $G(a, b)$ for every pair $a, b \in \mathsf{elems}(\mathcal{F}_0)$ for some fresh binary base relation G, and
- Σ' is Σ together with the k-guardedly-transitive axioms for each distinguished relation, where k is $|\Sigma \wedge \neg Q|$.

These can be constructed in time polynomial in the size of the input.

As discussed in the proof of Lemma 1, the k-guardedly transitive axioms (see Appendix E) can be written in normal form BaseGNF with width at most k, and hence in GNF.

Now we prove the correctness of the reduction. Suppose $\mathsf{QA}(\mathcal{F}_0', \Sigma', Q)$ holds, so any $\mathcal{F}' \supseteq \mathcal{F}_0'$ satisfying Σ' must satisfy Q. Now consider $\mathcal{F} \supseteq \mathcal{F}_0$ that satisfies Σ and where all R^+ in $\sigma_{\mathcal{D}}$ are transitive. We must show that $\mathcal{F}$ satisfies Q. First, observe that $\mathcal{F}$ satisfies Σ' since the k-guardedly-transitive axioms for R^+ are clearly satisfied for all k when R^+ is transitively closed. Now consider the extension of $\mathcal{F}$ to $\mathcal{F}'$ with additional facts $G(a, b)$ for all $a, b \in \mathsf{elems}(\mathcal{F}_0)$. This must still satisfy Σ': adding these guards means there are additional k-guardedly-transitive requirements on the elements from $\mathcal{F}_0$, but these requirements already hold since R^+ is transitively closed on all elements. Hence, by our initial assumption, $\mathcal{F}'$ must satisfy Q. Since Q does not mention G, the restriction of $\mathcal{F}'$ back to $\mathcal{F}$ still satisfies Q as well. Therefore, $\mathsf{QA}(\mathcal{F}_0, \Sigma, Q)$ holds.

On the other hand, suppose for the sake of contradiction that $\mathsf{QA}(\mathcal{F}_0', \Sigma', Q)$ does not hold, but $\mathsf{QAtr}(\mathcal{F}_0, \Sigma, Q)$ does. Then there is some $\mathcal{F}' \supseteq \mathcal{F}_0'$ such that $\mathcal{F}'$ satisfies $\Sigma' \wedge \neg Q$, and hence also satisfies $\Sigma \wedge \neg Q$. Since $\Sigma \wedge \neg Q$ is in BaseGNF, Proposition 11 implies that we can take $\mathcal{F}'$ to be a set of facts that has an $\mathcal{F}_0'$-rooted $(k-1)$-width base-guarded-interface tree decomposition. Let $\mathcal{F}''$ be the result of taking the transitive closure of the distinguished relations in $\mathcal{F}'$. By Lemma 5, transitively closing like this can only add R^+-facts about pairs of elements that are not base-guarded. Moreover, Lemma 7 ensures that adding R^+-facts about these non-base-guarded pairs of elements does not affect satisfaction of BaseCovGNF sentences, so $\mathcal{F}''$ must still satisfy $\Sigma \wedge \neg Q$. Restricting $\mathcal{F}''$ to its σ-facts results in an $\mathcal{F}$ where every distinguished relation is transitively closed and where $\Sigma \wedge \neg Q$ is still satisfied, since Σ and Q do not mention relation G. But this contradicts the assumption that $\mathsf{QAtr}(\mathcal{F}_0, \Sigma, Q)$ holds.

This concludes the proof of correctness.

Finally, observe that the k-guardedly-transitive axioms can be written as FGTGDs (in fact, BaseFGTGDs): they are equivalent to the conjunction of FGTGDs of the form

$$\forall x\, y\, x_1 \dots x_{l+1} \left[\big(x = x_1 \wedge x_{l+1} = y \wedge R^+(x_1, x_2) \wedge \dots \wedge R^+(x_l, x_{l+1}) \wedge S(x, y)\big) \rightarrow R^+(x, y) \right]$$

for all $S \in \sigma_{\mathcal{B}} \cup \{G\}$, $1 \leq l \leq k$, and $R^+ \in \sigma_{\mathcal{D}}$. Therefore, if Σ is in BaseCovFGTGD then Σ' is in FGTGD as claimed. □

Theorem 4 easily follows from this.

Proof of Theorem 4. Recall that we have fixed constraints Σ in BaseCovFGTGD and a base-covered UCQ Q. We must show PTIME data complexity of $\mathsf{QAtr}(\mathcal{F}_0, \Sigma, Q)$ for any finite initial set of facts $\mathcal{F}_0$. Use Proposition 15 to construct Σ' from Σ (in constant time, since Σ is fixed) and $\mathcal{F}'_0$ from $\mathcal{F}_0$ (in time polynomial in $|\mathcal{F}_0|$). Since Σ is in BaseCovFGTGD, Σ' is in FGTGD. Therefore, the PTIME data complexity upper bound for QAtr with BaseCovFGTGDs follows from the PTIME data complexity upper bound for QA with FGTGDs [Baget *et al.*, 2011]. □

G Hardness results

Chase. In the proofs of this section and of subsequent sections, we will need the standard database construction of the *chase* [Abiteboul *et al.*, 1995] by TGDs:

Definition 1. *The chase applies to a set of facts $\mathcal{F}$ and to a set Σ of TGDs, and constructs a set of facts $\mathcal{F}' \supseteq \mathcal{F}$, possibly infinite, which satisfies Σ, in the following manner.*

We first define a chase round *as follows: for each TGD $\tau : \forall \vec{x}\, \varphi(\vec{x}) \rightarrow \exists \vec{y}\, \psi(\vec{x}, \vec{y})$, for each homomorphism h from $\vec{x}$ to the elements of $\mathcal{F}$ such that the facts of $\varphi(h(\vec{x}))$ are in $\mathcal{F}$, if h does not extend to a homomorphism from $\vec{x} \cup \vec{y}$ such that the facts of $\psi(h(\vec{x}), h(\vec{y}))$ hold in $\mathcal{F}$, then we call $\varphi(h(\vec{x}))$ a* violation *of τ in $\mathcal{F}$: we repair it by creating fresh elements (called* existential witnesses*) $\vec{b}$ for each variable of $\vec{y}$, and add to $\mathcal{F}$ the facts $\psi(h(\vec{x}), \vec{b})$.*

Applying a chase round means performing this process in parallel for all TGDs and violations, creating fresh existential witnesses for each TGD and violation. The chase *of $\mathcal{F}$ by Σ is the (potentially infinite) set of facts obtained by repeated applications of chase rounds.*

When we use the chase, we will often use the fact that the result satisfies Σ, and that all existentially quantified variables when applying rules are instantiated by fresh existential witnesses (so no new facts are created on an element unless it occurs on a fact which is part of a violation).

G.1 Proof of Theorems 5 and 7

The hardness results for QAtc and QAlin mentioned in the body depend on the reductions described in Theorems 5 and 7. We start by proving Theorem 5, and we will adapt the proof afterwards to show Theorem 7.

Recall the result statement:

> **Theorem 5.** For any finite set of facts $\mathcal{F}_0$, DIDs Σ, and UCQ Q on a signature σ, we can compute in PTIME a set of facts $\mathcal{F}'_0$, BaseIDs Σ', and a base-covered CQ Q' on a signature σ' (with a single distinguished relation), such that $\mathsf{QA}(\mathcal{F}_0, \Sigma, Q)$ iff $\mathsf{QAtc}(\mathcal{F}'_0, \Sigma', Q')$.

We start by creating a UCQ Q', and then modify the proof to make Q' a CQ. Throughout the proof, whenever we talk of the DIDs in Σ, we mean all dependencies of Σ, including those where the head is a trivial disjunction with only one disjunct.

Definition of the reduction. Create the signature σ' from σ and Σ by:

- adding a fresh binary base predicate E and taking the transitive closure E^+ of E as the one distinguished relation of σ';
- replacing each predicate R in σ with a base predicate R' in σ' of arity $\mathsf{arity}(R) + 2$;
- adding to σ', for each DID of the form

$$\tau : \forall \vec{x}\, R(\vec{x}) \rightarrow \bigvee_{1 \leq i \leq n} \exists \vec{y_i}\, R_i(\vec{x}, \vec{y_i}),$$

  a base predicate $\mathsf{witness}_\tau(\vec{x}, \vec{y}_1, e_1, f_1, \ldots, \vec{y}_n, e_n, f_n)$.

We create Σ' from Σ by replacing each DID $\tau : \forall \vec{x}\, R(\vec{x}) \rightarrow \bigvee_{1 \leq i \leq n} \exists \vec{y_i}\, R_i(\vec{x}, \vec{y_i})$ by BaseIDs equivalent to:

$$\forall \vec{x}\, e\, f\, R'(\vec{x}, e, f) \rightarrow \exists \vec{y_1} e_1 f_1, \ldots, \vec{y_n} e_n f_n \; \mathsf{witness}_\tau(\vec{x}, \vec{y}_1, e_1, f_1, \ldots, \vec{y}_n, e_n, f_n) \wedge \bigwedge_i R'_i(\vec{x}, \vec{y_i}, e_i, f_i) \wedge E^+(e_i, f_i)$$

Note that we have written the BaseIDs for a given τ as a single TGD above with multiple conjuncts in the head, but we can easily rewrite them as multiple BaseIDs for τ: the first one has the same body and the $\mathsf{witness}_\tau$-fact as head atom, and the others have the $\mathsf{witness}_\tau$-fact as body atom and each one of the other facts as head atom.

The intuition for the proof is that a fact $R(\vec{c})$ over the original schema will correspond to facts $R'(\vec{c}, e, f)$ in the new schema with fresh elements e and f. The fresh elements will always be connected by an E-path (as required by the E^+-fact), which will be imposed (via failure of the query) to have length 1 or 2. Facts of this type with a path of length 1 will be called *genuine facts*, which intuitively hold, and those with a path of length 2 will be called *pseudo-facts* and will be ignored by the query.

This mechanism allows us to eliminate disjunction from DIDs as follows: we require that, when the body atom holds, there are witness facts $R'_i(\vec{c}, \vec{d_i}, e_i, f_i)$ for *all* of the disjuncts. However, we will use the query to require that, when the match of the body atom is a genuine fact, not all disjuncts can be pseudo-facts, so one of them must be a genuine fact; the others can be made pseudo-facts. Note that Σ' still requires matches for all of the disjuncts even when the body is matched to a pseudo-fact; however, the query will only require that one of the head atoms is matched to a genuine fact when the body is itself matched to a genuine fact. To this end, we will call the sequence $\vec{c}, \vec{d_1}, e_1, f_1, \ldots, \vec{d_n}, e_n, f_n$ will be called a *witness vector* for $R(\vec{c})$ and τ, and we capture such witness vectors in the predicate $\mathsf{witness}_\tau$.

The UCQ Q' contains the following disjuncts:

- *Q-generated disjuncts*: One disjunct for each disjunct of the original UCQ Q, where each atom $R(\vec{x})$ is replaced by the conjunction $R'(\vec{x}, e, f) \wedge E(e, f)$, where e and f are fresh. That is, we have a witness for Q consisting of genuine facts.
- *E-path length restriction disjuncts*: For each predicate R in σ, we have a disjunct that succeeds if the E-path for an R'-fact has length ≥ 3, i.e., $R'(\mathbf{x}, e, f) \wedge E(e, y_1) \wedge E(y_1, y_2) \wedge E(y_2, y_3)$. Intuitively, for every R'-fact, the E^+-fact on its two last elements must make it either a genuine fact or a pseudo-fact.
- DID *satisfaction disjuncts*: For every DID τ :

$\forall \vec{x}\ R(\vec{x}) \rightarrow \bigvee_i \exists \vec{y_i} R_i(\vec{x}, \vec{y_i})$ in Σ, we have a disjunct

$$Q_\tau : R'(\vec{x}, e, f) \wedge E(e, f) \wedge \mathsf{witness}_\tau(\vec{x}, \vec{y}_1, e_1, f_1, \ldots, \vec{y}_n, e_n, f_n) \wedge \bigwedge_{1 \leq i \leq n} R'_i(\vec{x}, \vec{y}_i, e_i, f_i) \wedge (E(e_i, w_i) \wedge E(w_i, f_i))$$

Informally, the failure of Q_τ enforces that we cannot have the body of τ holding as a genuine fact and each of the components of the witness vector realized by a pseudo-fact.

Observe that all of these disjuncts are trivially base-covered (since they do not use E^+).

We now explain how to rewrite the facts of an initial fact set $\mathcal{F}_0$ on σ to a fact set $\mathcal{F}'_0$ on σ'. Create $\mathcal{F}'_0$ by replacing each fact $F = R(\vec{a})$ of $\mathcal{F}_0$ by the facts $R'(\vec{a}, b_F, b'_F)$, and $E(b_F, b'_F)$, where b_F and b'_F are fresh, so that they are genuine facts.

Correctness proof for the reduction. We now show that the claimed equivalence holds: $\mathsf{QA}(\mathcal{F}_0, \Sigma, Q)$ holds iff $\mathsf{QAtc}(\mathcal{F}'_0, \Sigma', Q')$ holds.

First, let $\mathcal{F} \supseteq \mathcal{F}_0$ satisfy Σ and violate Q. We must construct $\mathcal{F}'$ that satisfies Σ' and violates Q' (when interpreting E^+ as the transitive closure of E).

We construct $\mathcal{F}'$ using the following steps:

- Modify $\mathcal{F}$ in the same way that we used to build $\mathcal{F}'_0$ from $\mathcal{F}_0$ (i.e., expand each fact with two fresh elements with an E-edge between them), yielding $\mathcal{F}_1$;
- We now need to ensure that witnesses exist as required by Σ', which we will create as pseudo-facts.

  For every DID τ of Σ and fact $F = R(\vec{c})$ of $\mathcal{F}$ that matches the body of τ, as $\mathcal{F}$ satisfies Σ, there is at least one i_0 such that some fact $R_{i_0}(\vec{c}, \vec{d}_{i_0})$ witnesses that τ is not violated in $\mathcal{F}$. Call I the set of such indices for which a witness exists in $\mathcal{F}$. We then know by construction that, for all $i \in I$, the set of facts $\mathcal{F}_1$ contains $F_i := R'_i(\vec{c}, \vec{d}_i, e_i, f_i)$ and $F'_i := E(e_i, f_i)$ for some e_i and f_i. For every $i \in \{1, \ldots, n\} \backslash I$, create a fresh $\vec{d}_i, e_i, f_i, w_i$ and add a fact $R'_i(\vec{c}, \vec{d}_i, e_i, f_i)$ along with the facts $E(e_i, w_i)$ and $E(w_i, g_i)$ indicating that this is a pseudo-fact. We also add a fact $\mathsf{witness}_\tau(\vec{c}, \vec{d}_1, e_1, f_1 \ldots \vec{d}_n, e_n, f_n)$ containing the witness vector consisting of the elements of the F_i above (those that we created, for $i \in \{1, \ldots, n\} \backslash I$, and those that already existed, for $i \in I$).

  We call $\mathcal{F}_2$ the result of performing this process simultaneously in all places where it is applicable. Observe that, in $\mathcal{F}_2$, we have ensured that no rule of Σ' has a violation whose body matches a genuine fact.
- The above process creates new pseudo-facts, and we also have to satisfy the rules of Σ' for these. We create $\mathcal{F}_3$ from $\mathcal{F}_2$ by simply chasing with Σ' wherever applicable (see Definition 1), always creating fresh elements; whenever we need a witness for some E^+ requirement, we always create an E-path of length 2 with a fresh element in the middle, that is, we always create pseudo-facts.

Let $\mathcal{F}' := \mathcal{F}_3$. It is clear that $\mathcal{F}' \supseteq \mathcal{F}'_0$ and that E^+ is indeed the transitive closure of E, and it is immediate by definition of the chase that $\mathcal{F}'$ satisfies Σ', so we must check that $\mathcal{F}'$ violates Q', which we do by considering each kind of disjunct.

For the *E-path length restriction disjuncts* observe that we only create paths of length 1 or 2 of E (of length 1 when creating $\mathcal{F}_1$, and of length 2 when creating $\mathcal{F}_2$ and $\mathcal{F}_3$). We always create these paths on fresh elements, so these paths of length 1 and 2 are never connected; hence, there is no E-path of length 3 at all in $\mathcal{F}'$.

For the DID *satisfaction disjuncts*, assume by contradiction that there is a match for a disjunct Q_τ of Q' in $\mathcal{F}'$. Fix $\vec{c}, e, f$ such that $R'(\vec{c}, e, f) \wedge E(e, f)$ holds, a fact $\mathsf{witness}_\tau(\vec{c}, \vec{d}_1, e_1, f_1 \ldots \vec{d}_n, e_n, f_n)$, and pseudo-facts $R'_i(\vec{c}, \vec{d}_i, e_i, f_i)$ with e_i, f_i connected by paths of length 2. The genuine fact $R'(\vec{c}, e, f)$ could not have been generated within either of the second or third steps in the creation of $\mathcal{F}'$ above, since all the R'-facts generated there have paths only of length 2 between the last two components (that is, they are pseudo-facts). Thus $R'(\vec{c}, e, f)$ must have been generated in the first step, coming from fact $R(\vec{c})$ in $\mathcal{F}$. But then, as $\mathcal{F}$ satisfies τ, there must be $i_0 \leq n$ such that $\mathcal{F}$ contains $R_{i_0}(\vec{c}, \vec{d}_{i_0})$ for some $\vec{d}_{i_0}$, and thus $\mathcal{F}_1$ must contain $R'_{i_0}(\vec{c}, \vec{d}_{i_0}, e_{i_0}, f_{i_0})$ and $E(e_{i_0}, f_{i_0})$ for some e_{i_0} and f_{i_0}. Further, the $\mathsf{witness}_\tau$-fact must have been created in the second step above, as the other $\mathsf{witness}_\tau$-facts are created during the third step, where they only cover pseudo-facts rather than genuine facts. Hence, in creating the witness vector corresponding to $R(\vec{c}, e, f)$ in the second step, we would not have generated a path of length 2 for e_{i_0}, f_{i_0} (as we would have had $i_0 \in I$), a contradiction.

Finally, for the *Q-generated disjuncts*, observe that any match of them must be on facts of $\mathcal{F}'$ created for facts of $\mathcal{F}$ (as they are annotated by E-paths of length 1), so we can conclude because $\mathcal{F}$ violates Q.

Hence, $\mathcal{F}'$ satisfies Σ' and violates Q', which concludes the first direction.

In the other direction, let $\mathcal{F}' \supseteq \mathcal{F}'_0$ be a counterexample to $\mathsf{QAtc}(\mathcal{F}'_0, \Sigma', Q')$. Consider the set of R'-facts from $\mathcal{F}'$ such that $R' \in \sigma'$ corresponds to some $R \in \sigma$ and the elements in the last two positions of this R'-fact are connected by an E-fact, i.e., the genuine facts. Construct a set of facts $\mathcal{F}$ on σ by projecting away the last two positions from these R'-facts, and discarding all of the other facts.

It is clear by construction of $\mathcal{F}'_0$ that $\mathcal{F} \supseteq \mathcal{F}_0$ and that E^+ is indeed the transitive closure of E in $\mathcal{F}$. Further, as $\mathcal{F}'$ violates Q', it is clear that $\mathcal{F}$ violates Q, as any match of a disjunct of Q on $\mathcal{F}$ implies a match of the corresponding Q-generated disjunct Q' in $\mathcal{F}'$. So it suffices to show that $\mathcal{F}$ satisfies Σ.

Assume by contradiction that some DID τ of Σ is violated on a fact $F = R(\vec{x})$ of $\mathcal{F}$ (that is F matches the body of τ). Let $F' = R'(\vec{c}, e, f)$ be the fact in $\mathcal{F}'$ from

which we created F; we know that the last two elements of F' are connected by an E-fact. Since $\mathcal{F}'$ satisfies Σ', we know that there are $\vec{d}_1, e_1, f_1, \ldots, \vec{d}_n, e_n, f_n$ such that $\mathsf{witness}_\tau(\vec{c}, \vec{d}_1, e_1, f_1, \ldots, \vec{d}_n, e_n, f_n)$ and $\bigwedge_i R'_i(\vec{c}, \vec{d}_i, e_i, f_i) \wedge E^+(e_i, f_i)$. Moreover, since E^+ is the transitive closure of E in $\mathcal{F}'$, we know that for each i, there is some E-path connecting e_i and f_i. By the E-path length-restriction disjuncts and DID satisfaction disjuncts, it must be the case that there is an E-path of length at most 2 between each e_i and f_i, and for some j there cannot be a path of length 2 between e_j and f_j (otherwise, $\mathcal{F}'$ would satisfy the corresponding DID satisfaction disjunct in Q'), so then the path must have length 1. But this means that $\mathcal{F}'$ contains $R'_j(\vec{c}, \vec{d}_j, e_j, f_j)$ and $E(e_j, f_j)$, so $R_j(\vec{c}, \vec{d}_j)$ is a fact in $\mathcal{F}$ witnessing the satisfaction of DID τ, a contradiction. Hence, $\mathcal{F}$ satisfies Σ, which concludes the proof.

From UCQ to CQ. Last, we explain how to replace the UCQ Q' by a CQ. We do this by a general process that we will reuse in several upcoming proofs: intuitively, we increase the arity to annotate facts with an additional Boolean value carried over in dependencies and add an Or-relation to combine such values.

Formally, define a signature σ_{Or} with a ternary relation Or and a unary relation True. Define a set of facts $\mathcal{F}_{\mathrm{Or}}$ with two domain elements $\mathfrak{t}$ and $\mathfrak{f}$ that contains the fact $\mathrm{True}(\mathfrak{t})$ and the facts $\mathrm{Or}(b, b', b'')$ for all $\{(b, b', b \vee b') \mid b, b' \in \{\mathfrak{f}, \mathfrak{t}\}\}$.

Define σ'' from σ' by increasing the arity of each relation in σ' *except* E *and* E^+ and adding the relations from σ_{Or}.

Define Σ'' from Σ' by adding a new variable b which is universally quantified and is put in the head and body facts.

Define Q'' from Q' as follows:

- Add to the atoms of each disjunct of Q' (except E-atoms) one common variable which is shared between all atoms and left free: we call each resulting CQ $Q_i(w_i)$, where w_i is the new variable.
- Define the Boolean CQ Q'' as the following (existentially closed), where m is the number of disjuncts of Q':

$$\begin{aligned}&\mathrm{Or}(w_1, w_2, w'_1) \wedge \mathrm{Or}(w'_1, w_3, w'_2) \wedge \cdots \\ &\wedge \mathrm{Or}(w'_{m-2}, w_m, w'_{m-1}) \wedge \mathrm{True}(w'_{m-1}) \\ &\wedge \bigwedge_{1 \leq i \leq m} Q_i(w_i)\end{aligned}$$

Note that Q'' is trivially covered.

Define the set of facts $\mathcal{F}''_0$ from $\mathcal{F}'_0$ by:

- Adding the facts of $\mathcal{F}_{\mathrm{Or}}$;
- Putting $\mathfrak{t}$ as the last element of all other facts except E-facts;
- Adding *vacuous matches*: for each $Q_i(w_i)$, we add a set of facts that satisfy $Q_i(w_i)$, with $\mathfrak{f}$ as the common last element of all facts, but the domains being otherwise disjoint.

  Intuitively, the purpose of the vacuous matches is to ensure that the $Q_i(w_i)$ always have a match but with w_i set to false, and otherwise they have no purpose and they simply do not interact with the other facts.

  We accordingly call an element *vacuous* in a set of facts if it occurs in no fact with $\mathfrak{t}$ as the last element, and call a fact *vacuous* if it is an E- or E^+-fact on vacuous elements, or it is a fact for another relation than E but its last element is not $\mathfrak{t}$.

We will now show the following equivalence: $\mathsf{QAtc}(\mathcal{F}'_0, \Sigma', Q')$ iff $\mathsf{QAtc}(\mathcal{F}''_0, \Sigma'', Q'')$, which concludes the proof.

In one direction, we assume we have a counterexample $\mathcal{F}'$ to $\mathsf{QAtc}(\mathcal{F}'_0, \Sigma', Q')$. We construct $\mathcal{F}''$ from $\mathcal{F}'$ by extending it according to the process above (to define $\mathcal{F}''_0$ from $\mathcal{F}'_0$), and chasing by Σ'' on all facts from the vacuous matches (see Definition 1), with $\mathfrak{f}$ being propagated as the last element, so the resulting elements and facts are all vacuous. We claim that $\mathcal{F}''$ witnesses the failure of $\mathsf{QAtc}(\mathcal{F}''_0, \Sigma'', Q'')$.

It is clear that $\mathcal{F}''$ is a superset of $\mathcal{F}''_0$ and that E^+ is indeed interpreted as the transitive closure of E. We argue that Σ'' is satisfied by $\mathcal{F}''$, by looking whether the required witness facts exist for each type of fact. Vacuous facts have the required witnesses because we chased them in constructing $\mathcal{F}''$. No constraints of Σ'' hold about the facts from $\mathcal{F}_{\mathrm{Or}}$. Finally, for the facts of $\mathcal{F}''$ created from facts of $\mathcal{F}'$, they have the required witnesses because Σ' was satisfied by $\mathcal{F}'$ and the last position of such a fact is always $\mathfrak{t}$ so the last variable was correctly exported.

We now explain why $\mathcal{F}''$ violates Q''. Assuming by contradiction that $\mathcal{F}''$ satisfies Q'', by definition of the Or- and True-facts that $\mathcal{F}''$ contains by construction, it must be the case that $\mathcal{F}''$ satisfies $Q_i(\mathfrak{t})$ for some Q_i. But it is then clear that $\mathcal{F}'$ satisfies the corresponding disjunct of Q', as this match cannot involve any vacuous facts. This proves one direction.

For the other direction, we assume we have a counterexample $\mathcal{F}''$ for $\mathsf{QAtc}(\mathcal{F}''_0, \Sigma'', Q'')$. We construct $\mathcal{F}'$ from $\mathcal{F}''$ by keeping only the facts in the base signature with last element $\mathfrak{t}$ and keeping precisely the E- and E^+-facts that are connected to them. It is clear that, as $\mathcal{F}'' \supseteq \mathcal{F}''_0$, we have $\mathcal{F}' \supseteq \mathcal{F}'_0$. To see that $\mathcal{F}'$ satisfies Σ', assume by contradiction that F' witnesses a violation of an ID τ' of Σ' in $\mathcal{F}'$, and let F'' be the corresponding fact in $\mathcal{F}''$. By definition of Σ'', there is a corresponding ID τ'' in Σ'' that asserts the existence of a fact F''_2. So the only way F' can violate τ' is that F''_2 is an E-fact or that the last element of F''_2 is not $\mathfrak{t}$, but as the last element of F'' is $\mathfrak{t}$, this is impossible. Hence, we have a contradiction, and F' satisfies Σ'.

The only thing left to show is that $\mathcal{F}'$ violates Q'. Assuming to the contrary that $\mathcal{F}'$ satisfies some disjunct of Q', we know that, considering the corresponding $Q_i(w_i)$, $\mathcal{F}''$ satisfies $Q_i(\mathfrak{t})$. Now, from the facts in $\mathcal{F}_{\mathrm{Or}} \subseteq \mathcal{F}''$, and from the vacuous matches and their connected E -and E^+-facts, we know that we can construct a match of the entire CQ Q'' in $\mathcal{F}''$, a contradiction as $\mathcal{F}''$ violates Q'. This concludes the correctness proof, and concludes the proof of Theorem 5.

We now prove Theorem 7, which states:

Theorem 7. For any finite set of facts $\mathcal{F}_0$, DIDs Σ, and UCQ Q on a signature σ, we can compute in PTIME a set of facts $\mathcal{F}'_0$, BaseIDs Σ', and CQ Q' on a signature σ' (with a single distinguished relation), such that $\mathsf{QA}(\mathcal{F}_0, \Sigma, Q)$ iff $\mathsf{QAlin}(\mathcal{F}'_0, \Sigma', Q')$.

The entire proof is shown by adapting the proof of Theorem 5. We start by showing the claim with a UCQ. Intuitively, instead of using E^+ to emulate a disjunction on the length of the path to encode genuine facts and pseudo facts, we will use the order relation to emulate disjunction on the same elements: $e < f$ will indicate a genuine fact, whereas $f < e$ will indicate a pseudo-fact, and $e = f$ will be prohibited by the query.

Definition of the reduction. We define σ' as in the proof of Theorem 5, except that we do not add the predicates E and E^+, but add a predicate $<$ as a distinguished relation instead. We also define Σ' as before except that we drop all mention of E^+.

The UCQ Q' contains the following disjuncts (existentially closed):

- *Order restriction disjuncts*: For each $R \in \sigma$, we have a disjunct $R'(\vec{x}, e, e)$ to enforce disjunction between genuine facts and pseudo-facts.
- *Q-generated disjuncts*: Each disjunct of the original UCQ Q, where each atom $R(\vec{x})$ is replaced by the conjunction $R'(\vec{x}, z, z') \wedge z < z'$, where z and z' are fresh. That is, we have a witness for Q consisting of genuine facts.
- DID *satisfaction disjuncts*: For every DID τ : $\forall \vec{x}\, R(\vec{x}) \rightarrow \bigvee_i \exists \vec{y_i} R_i(\vec{x}, \vec{y_i})$ in Σ, we have a disjunct

$$Q_\tau : R'(\vec{x}, e, f) \wedge e < f \\ \wedge \mathsf{witness}_\tau(\vec{x}, \vec{y}_1, e_1, f_1 \ldots \vec{y}_n, e_n, f_n) \\ \wedge \bigwedge_{1 \leq i \leq n} R'_i(\vec{x}, \vec{y}_i, e_i, f_i) \wedge f_i < e_i$$

  Intuitively, Q_τ is satisfied if the body of τ is matched to a genuine fact but each of the components of the witness vector is matched to a pseudo-fact.

Observe that all CQs of the resulting UCQ are base-covered, as required.

The process to define $\mathcal{F}'_0$ from $\mathcal{F}_0$ is defined like in the proof of Theorem 5 except that we remove the E-facts and replace them by $b_F < b'_F$.

Correctness proof for the reduction. The proof that $\mathsf{QA}(\mathcal{F}_0, \Sigma, Q)$ holds iff $\mathsf{QAlin}(\mathcal{F}'_0, \Sigma', Q')$ holds is similar to the proof for Theorem 5, so we sketch the proof and highlight the main differences.

For one direction, let $\mathcal{F} \supseteq \mathcal{F}_0$ satisfy Σ and violate Q. We construct $\mathcal{F}'$ from $\mathcal{F}$ as follows:

- Construct $\mathcal{F}_1$ from $\mathcal{F}$ as we constructed $\mathcal{F}'_0$ from $\mathcal{F}_0$ above.
- The construction of $\mathcal{F}_2$ and $\mathcal{F}_3$ is as before, except that we create a $<$-fact to indicate a pseudo-fact.
- The new step is that $\mathcal{F}'$ is constructed from $\mathcal{F}_3$ by completing $<$ to be a total order. To do so, however, we must ensure that our definition of $<$ in $\mathcal{F}_3$ does not contain any cycles. This is easy to see, however: we only imposed an order relation between *disjoint pairs* of elements. Hence, it is clear that $<$ cannot contain any loop, so we can simply complete this partial order to a total order using the order extension principle [Szpilrajn, 1930].

As before it is clear that $\mathcal{F}' \supseteq \mathcal{F}'_0$ and that $\mathcal{F}'$ satisfies Σ', and we have made sure that $<$ is a total order. To see why Q' is not satisfied in $\mathcal{F}'$, we proceed exactly as before for the DID satisfaction disjuncts and Q-generated disjuncts, but replacing "having an E-fact between e and f" by "having $e < f$", and replacing "having an E-path of length 2 between e and f" by "having $e > f$", and likewise for e_i and f_i. By construction, we never have $e = f$ or $e_i = f_i$ in any fact within $\mathcal{F}'$, so we also do not match the order-restriction disjuncts in Q'.

For the other direction, suppose we have some counterexample $\mathcal{F}'$ to $\mathsf{QAtc}(\mathcal{F}'_0, \Sigma', Q')$. We construct $\mathcal{F}$ from $\mathcal{F}'$ by keeping all facts whose last two elements e and f are such that $e < f$. The result still clearly satisfies $\mathcal{F} \supseteq \mathcal{F}_0$, and the proof of why it violates Q is unchanged. To show that $\mathcal{F}$ satisfies Σ, we adapt the argument of the proof of Theorem 5, but instead of relying on the E-path length disjuncts we rely on totality of the order and the order-restriction disjuncts. Totality of the order ensures that for fact F_τ, and for all i, we have either $e_i < f_i$, $e_i = f_i$ and $f_i < e_i$. But the order-restriction disjuncts are violated, so it must be either $e_i < f_i$ or $f_i < e_i$, and the DID satisfaction disjuncts of Q are violated, so we must have $e_i < f_i$ for some i. Hence, we can argue as before that the satisfaction of Σ' by $\mathcal{F}'$ ensures that Σ is satisfied in $\mathcal{F}$.

From UCQ to CQ. The proof from UCQ to CQ works exactly like before, except that we do not increase the arity of $<$ (recall that we did not increase the arity of E^+ and E), and we use QAlin instead of QAtc. When showing that we can construct a counterexample to $\mathcal{F}''$ to $\mathsf{QAlin}(\mathcal{F}''_0, \Sigma'', Q'')$ from a counterexample to $\mathsf{QAlin}(\mathcal{F}'_0, \Sigma', Q')$, we make $<$ a total order in $\mathcal{F}''$ using again the order extension principle (the order on vacuous matches, and on the domain elements of $\mathcal{F}_{\mathrm{Or}}$, is arbitrary). Observe that the resulting CQ is clearly base-covered, as all disjuncts of the UCQ Q' were base-covered.

G.2 Proof of Propositions 2 and 4

We now give data complexity lower bounds that show CoNP-hardness even in the absence of constraints.

We first prove Proposition 2:

Proposition 2. There is a base-covered CQ Q such that the data complexity of $\mathsf{QAtc}(\mathcal{F}_0, \emptyset, Q)$ is CoNP-hard.

Proof. We first prove the result for a UCQ Q, and then for a CQ Q'.

Definition of the reduction. We define the signature σ as containing:

- one binary predicate E and its transitive closure E^+ (again playing a similar role as in the proof Theorem 5);
- one binary relation G to code the edges of a graph which will be provided as input to the reduction;
- one 7-ary relation V to code vertices and their color. The idea is that one position is for the vertex and then for each of the 3 colors we will have two positions that will encode whether or not the vertex has that color. If the positions associated with a color C are connected by an E-edge, this will indicate coloring the vertex with color C, while if they are connecting by a path of length 2 this will indicate not being colored with color C.

We then define the UCQ Q to contain the following disjuncts (existentially closed):

- *E-path length restriction disjuncts*: For each predicate R in σ, we enforce that the E-path for the R'-fact has length ≥ 3: $R'(\mathbf{x}, e, f) \wedge E(e, y_1) \wedge E(y_1, y_2) \wedge E(y_2, y_3)$.
- *Adjacency disjuncts:* For $i \in \{1, 2, 3\}$, the disjunct Q_i that succeeds if two adjacent vertices were assigned the same color:

$$V(x, e_1, f_1, e_2, f_2, e_3, f_3) \wedge G(x, x')$$
$$\wedge V(x', e'_1, f'_1, e'_2, f'_2, e'_3, f'_3) \wedge E(e_i, f_i) \wedge E(e'_i, f'_i)$$

- *Coloring disjunct:* A disjunct that succeeds if a vertex was not assigned any color: $V(x, e_1, f_1, e_2, f_2, e_3, f_3) \wedge \bigwedge_{i \in \{1,2,3\}} E(e_i, w_i) \wedge E(w_i, f_i)$

Given a directed graph $\mathcal{G}$, we code it in PTIME as the instance $\mathcal{F}_0$ defined by having:

- One fact $G(x, y)$ for each edge (x, y) in $\mathcal{G}$
- The facts $V(x, e_{x,1}, f_{x,1}, e_{x,2}, f_{x,2}, e_{x,3}, f_{x,3})$ and $E^+(e_{x,i}, f_{x,i})$ for $i \in \{1, 2, 3\}$ for each vertex x in $G(x, y)$, where all the $e_{x,i}$ and $f_{x,i}$ are fresh.

Correctness proof for the reduction. We now show that $\mathcal{G}$ is 3-colorable iff $\mathsf{QAtc}(\mathcal{F}_0, \emptyset, Q)$ is false, completing the reduction.

First, consider a 3-coloring of $\mathcal{G}$. Construct $\mathcal{F} \supseteq \mathcal{F}_0$ as follows. For each vertex x of $\mathcal{G}$ (with corresponding V-fact $V(x, e_{x,1}, f_{x,1}, e_{x,2}, f_{x,2}, e_{x,3}, f_{x,3})$ as defined above) create the facts $E(e_{x,i}, f_{x,i})$ where i is the color assigned to x, and the facts $E(e_{x,j}, w_{x,j})$ and $E(w_{x,j}, f_{x,j})$ for the other colors $j \in \{1, 2, 3\} \backslash \{i\}$ (with the two $w_{x,j}$ being fresh). It is clear that $\mathcal{F}$ thus defined is such that $\mathcal{F} \supseteq \mathcal{F}_0$, and that E^+ is the transitive closure of E in $\mathcal{F}$. The E-path length restriction disjuncts of Q do not match in $\mathcal{F}$ (note that we only create E-paths whose endpoints are pairwise distinct), and the coloring disjunct does not match either. Finally, the fact that we have a 3-coloring ensures that the adjacency disjuncts do not match either. Hence, we have a set of facts violating Q.

For the other direction, consider some $\mathcal{F} \supseteq \mathcal{F}_0$ that violates Q. Since $\mathcal{F}$ violates the first and last disjunct of Q and E^+ is the transitive closure of E, any vertex x of $\mathcal{G}$ (with corresponding V-fact $V(x, e_{x,1}, f_{x,1}, e_{x,2}, f_{x,2}, e_{x,3}, f_{x,3})$ defined above) there must be an E-path of length 1 or 2 from $e_{x,i}$ to $f_{x,i}$ for all $i \in \{1, 2, 3\}$. Further, as $\mathcal{F}$ violates the last disjunct of Q, at least one of these paths must have length 1. Define a coloring of $\mathcal{G}$ by giving each vertex x a color i such that $E(e_{x,i}, f_{x,i})$ holds in the V-fact for x. This indeed defines a 3-coloring, as any violation of the 3-coloring witnessed by two adjacent vertices of color i would imply a match of Q_i in $\mathcal{F}$.

From UCQ to CQ. We replace the UCQ Q by a CQ Q' in the same manner as in the proof of Theorem 5: we increase the arity of all predicates and add the relations of σ_{Or}, add to $\mathcal{F}_0$ the facts of $\mathcal{F}_{\mathrm{Or}}$ and the vacuous matches, and rewrite the query as in the proof of Theorem 5. We can then adapt the argument of that proof to show that the resulting QAtc problem with the CQ is equivalent to the previously defined problem with a UCQ. □

We then modify the proof to show Proposition 4:

> **Proposition 4.** There is a base-covered CQ Q such that the data complexity of $\mathsf{QAlin}(\mathcal{F}, \emptyset, Q)$ is CoNP-hard.

Proof. We define σ as in the previous proof but with an order relation $<$ and without E, E^+. We define Q as in the proof of Proposition 2 but without its first disjunct, and replacing in the other disjuncts $E(e_i, f_i)$ by $e_i < f_i$, and $E(e_i, w_i) \wedge E(w_i, f_i)$ by $f_i < e_i$. Unlike in the proof of Theorem 7, we need not worry about equalities (and we need not add order restriction disjuncts), as all the elements of relevant V-facts are created already in $\mathcal{F}_0$, where they are created as distinct elements. We define $\mathcal{F}_0$ in the same fashion as in the proof of Proposition 2 but without the E^+-facts.

We prove the same equivalence as in that proof but for QAlin. We do it by replacing E-paths of length 1 from an e-element to an f-element by $e < f$, and E-paths of length 2 by $f < e$.

We replace the UCQ by a CQ exactly as in the other proof. As in the proof of Theorem 7, the order on the vacuous matches is arbitrary. □

H Undecidability results related to transitivity (from Section 5)

We first prove the second result as it is simpler to understand.

Theorem 9. There is an arity-two signature $\sigma = \sigma_{\mathcal{B}} \sqcup \sigma_{\mathcal{D}}$ with a single distinguished predicate S^+ in $\sigma_{\mathcal{D}}$, a set Σ of DIDs on σ, a CQ Q on $\sigma_{\mathcal{B}}$, such that the following problem is undecidable: given a finite set of facts $\mathcal{F}_0$, decide $\mathsf{QAtr}(\mathcal{F}_0, \Sigma, Q)$.

Proof. As in Proposition 4, we first prove the result with a UCQ and then modify the proof to use a CQ.

An *infinite tiling problem* is specified by a set of colors $\mathbb{C} = C_1, \ldots, C_k$, a set of forbidden *horizontal* patterns $\mathbb{H} \subseteq \mathbb{C}^2$ and a set of forbidden *vertical* patterns $\mathbb{V} \subseteq \mathbb{C}^2$. It asks, given a sequence $c_0, \ldots, c_n$ of colors of $\mathbb{C}$, whether there exists a function $f : \mathbb{N}^2 \to \mathbb{C}$ such that $f((0,i)) = c_i$ for all $0 \leq i \leq n$, and for all $i, j \in \mathbb{N}$, we have $(f(i,j), f(i+1,j)) \notin \mathbb{H}$ and $(f(i,j), f(i,j+1)) \notin \mathbb{V}$.

It is well-known that we can take $\mathbb{C}$, $\mathbb{V}$, $\mathbb{H}$ such that the corresponding tiling problem is undecidable; we fix such a problem.

Definition of the reduction. We define a binary relation S' (for "successor"), a transitive relation S^+, one binary relation K_i for each color C_i, and one unary relation K'_i for each color C_i.

We write the following DIDs Σ (note that they are not base-guarded), dropping universal quantification for brevity:

$$S'(x,y) \to \exists z\ S'(y,z)$$
$$S'(x,y) \to S^+(x,y)$$
$$S^+(x,y) \to \bigvee_i K_i(x,y)$$
$$S^+(x,y) \to \bigvee_i K_i(y,x)$$
$$S^+(x,y) \to \bigvee_i K'_i(x)$$

Intuitively, $K'_i(x)$ stands for $K_i(x,x)$, but we need a different predicate because variable reuse is not allowed in inclusion dependencies.

The UCQ Q is a disjunction of the following disjuncts (existentially closed):

- For each forbidden horizontal pair $(C_i, C_j) \in \mathbb{H}$, with $1 \leq i, j \leq k$, the disjuncts:
$$K_i(x,y) \wedge S'(y,y') \wedge K_j(x,y')$$
$$K'_i(y) \wedge S'(y,y') \wedge K_j(y,y')$$
$$K_i(y',y) \wedge S'(y,y') \wedge K'_j(y')$$
- For each forbidden vertical pair $(C_i, C_j) \in \mathbb{V}$, the analogous disjuncts:
$$K_i(x,y) \wedge S'(x,x') \wedge K_j(x',y)$$
$$K'_i(x) \wedge S'(x,x') \wedge K_j(x',x)$$
$$K_i(x,x') \wedge S'(x,x') \wedge K'_j(x')$$

Given an initial instance of the tiling problem $c_0, \ldots, c_n$, we encode it in the initial set of facts $\mathcal{F}_0$:

- $S'(a_i, a_{i+1})$ for $0 \leq i < n$;
- for $0 < i \leq n$, the fact $K_j(a_0, a_i)$ such that C_j is the color of initial element c_i;
- the fact $K'_j(a_0)$ such that C_j is the color of c_0.

Correctness proof for the reduction. We claim that the tiling problem has a solution iff there is a (generally infinite) superset of $\mathcal{F}_0$ that satisfies Σ and violates Q and where S^+ is transitive. From this we conclude the reduction and deduce the undecidability of QAtr as stated.

For the forward direction, from a solution f to the tiling problem for input $\vec{c}$, we construct the counterexample $\mathcal{F} \supseteq \mathcal{F}_0$ as follows. We first create an infinite chain $S'(a_0, a_1), \ldots, S'(a_m, a_{m+1}), \ldots$ to complete the initial chain of S'-facts in $\mathcal{F}_0$, and fix S^+ to be the transitive closure of this S'-chain (so it is indeed transitive). For all $i, j \in \mathbb{N}$ such that $i \neq j$, we create the fact $K_l(a_i, a_j)$ where $l = f(i,j)$. For all $i \in \mathbb{N}$, we create the fact $K'_l(a_i)$ where $l = f(i,i)$. This clearly satisfies the constraints in Σ, and does not satisfy the query because f is a tiling.

For the backward direction, consider a $\mathcal{F} \supseteq \mathcal{F}_0$ that satisfies Σ and violates Q. Starting at the chain of S'-facts of $\mathcal{F}_0$, we can deduce, using the constraints, the existence of an infinite chain $a_0, \ldots, a_n, \ldots$ of S'-facts (whose elements may be distinct or not, this does not matter). Define a tiling f matching the initial tiling problem instance as follows. For all $i < j$ in $\mathbb{N}$, as there is a path of S'-facts from a_i to a_j, we infer that $S^+(a_i, a_j)$ holds, so that $K_l(a_i, a_j)$ holds for some $1 \leq l \leq k$; pick one such fact, taking the fact of $\mathcal{F}_0$ if $i = 0$ and $j \leq n$, and fix $f(i,j) := l$. For $i > j$ we can likewise see that $S^+(a_j, a_i)$ holds whence $K_l(a_i, a_j)$ holds for some l, and we continue as before. For $i \in \mathbb{N}$, as $S'(a_i, a_{i+1})$ holds, we know that $K_l(a_i)$ holds for some $1 \leq l \leq k$ (again we take the fact of $\mathcal{F}_0$ if $i = 0$), and fix accordingly $f(i,i) := l$. The resulting f clearly satisfies the initial tiling problem instance $c_0, \ldots, c_n$, and it is clearly a solution to the tiling problem, as any forbidden pattern in f would witness a match of a CQ of Q in $\mathcal{F}$. This shows that the reduction is correct, and concludes the proof with the UCQ Q.

From UCQ to CQ. We now adapt the proof to use a CQ, similarly to the proof of Theorem 5: we use the signature σ'_{Or} constructed by extending S' (and *only* S') to have a third position, and adding the relations of σ_{Or} (see the proof of Theorem 5). We modify Σ by changing the first dependency to propagate also the third element of the S'-atoms, i.e., $\forall x\, y\, b\ S'(x,y,b) \to \exists z\ S'(y,z,b)$, and replace the second dependency similarly by $\forall x\, y\, b\ S'(x,y,b) \to S^+(x,y)$. We construct the CQ Q' from the UCQ Q by adding one variable b to each S'-fact of each disjunct, and connecting these disjuncts on their free variables with Or-facts and adding a True-fact as in the proof of Theorem 5.

We then define the initial set of facts $\mathcal{F}'_0$ to include the facts of $\mathcal{F}_{\mathrm{Or}}$, the facts of $\mathcal{F}_0$ where each S' is extended by adding

$\mathfrak{t}$ as its third element, and vacuous matches for each Q_i (defined as in the proof of Theorem 5, with the element $\mathfrak{f}$ at the third position of each S'-fact, all the vacuous matches having pairwise disjoint domains except for $\mathfrak{f}$).

To show the forward direction, we construct $\mathcal{F}$ from $\mathcal{F}_0$ as before but with $\mathfrak{t}$ at the third position of all created S'-facts, plus a completion of the vacuous matches obtained by chasing (see Definition 1). Like in the proof of Theorem 5, any match of the CQ Q' must imply a match of one of the disjuncts of the UCQ Q, and as this match must be on an S'-fact with $\mathfrak{t}$ as third position, it cannot involve a vacuous match, so we conclude as before.

Conversely, for the backward direction, we define the tiling analogously to what we did before, and we observe that any violation of the tiling property would imply a match of one disjunct of the UCQ Q, which we can extend thanks to the vacuous matches to a match of Q'. This concludes the proof. □

We now prove the first statement, drawing inspiration from the previous proof, but using the transitive closure to emulate disjunction as in Theorem 5. Recall the statement:

> **Theorem 8.** There is a signature $\sigma = \sigma_{\mathcal{B}} \sqcup \sigma_{\mathcal{D}}$ with a single distinguished predicate S^+ in $\sigma_{\mathcal{D}}$, a set Σ of IDs on σ, and a CQ Q on $\sigma_{\mathcal{B}}$, such that the following problem is undecidable: given a finite set of facts $\mathcal{F}_0$, decide $\mathsf{QAtc}(\mathcal{F}_0, \Sigma, Q)$.

Proof. We reuse the notations for tiling problems from the previous proof. We first prove the result with two distinguished relations S^+ and C^+ and with a UCQ, and then explain how the proof is modified to use only a single transitive relation S^+, and finally explain how to adapt the proof to use a CQ rather than a UCQ.

Definition of the reduction. We define a binary relation S (for "successor") of which S^+ is interpreted as the transitive closure, one binary relation S', one 3-ary relation G (for "grid"), one binary relation G' (standing for the diagonal cells of the grid), one binary relation T (a terminal for gadgets that we will define to indicate colors) and one binary relation C of which C^+ is interpreted as the transitive closure. The distinction between S and S' is not important for now but will be important when we adapt the proof later to use a single distinguished relation.

We write the following inclusion dependencies Σ (with universal quantification dropped for brevity):

$$\begin{aligned}
S'(x,y) &\to \exists z\, S'(y,z)\\
S'(x,y) &\to S(x,y)\\
S^+(x,y) &\to \exists z\, G(x,y,z)\\
S^+(y,x) &\to \exists z\, G(x,y,z)\\
S^+(x,y) &\to \exists z\, G'(x,z)\\
G(x,y,z) &\to \exists w\, T(z,w)\\
G'(x,z) &\to \exists w\, T(z,w)\\
T(z,w) &\to C^+(z,w)
\end{aligned}$$

In preparation for defining the query Q, we define $Q_i(z)$ for all $i > 0$ to match the left endpoint of T-facts covered by a C-path of length i (intuitively coding color i):

$$\exists z_1 \dots z_i\, w\, C(z,z_1) \wedge C(z_1,z_2) \wedge \dots, C(z_{i-1},z_i) \wedge T(z,z_i),$$

The query Q is a disjunction of the following disjuncts (existentially closed):

- *C-path sanity disjuncts:* One disjunct written as follows, where k is the number of colors

$$\begin{aligned}
&G(x,y,z) \wedge S'(x,w) \wedge\ T(z,z) \wedge C(z,z_1)\\
&\wedge C(z_1,z_2) \wedge \dots \wedge C(z_{k-1},z_k) \wedge C^+(z_k,z')
\end{aligned}$$

  and one disjunct defined similarly but with $G(x,y,z)$ replaced by $G'(x,z)$. Intuitively, these disjuncts impose that C-paths that cover T-facts must code colors between 1 and k, and the distinction between G and G' is for reasons similar to the distinction between the K_i and K'_i in the proof of Theorem 9.

- *Horizontal adjacency disjuncts:* For each forbidden horizontal pair $(C_i, C_j) \in \mathbb{H}$, with $1 \leq i,j \leq k$, the disjuncts:

$$\begin{aligned}
G(x,y,z) \wedge G(x,y',z') \wedge Q_i(z) \wedge Q_j(z') \wedge S'(y,y')\\
G'(y,z) \wedge G(y,y',z') \wedge Q_i(z) \wedge Q_j(z') \wedge S'(y,y')\\
G(y',y,z) \wedge G'(y',z') \wedge Q_i(z) \wedge Q_j(z') \wedge S'(y,y')
\end{aligned}$$

- *Vertical adjacency disjuncts:* For each $(C_i, C_j) \in \mathbb{V}$, the same queries but replacing atoms $S'(y,y')$ by $S'(x,x')$ and the two first atoms of the last two subqueries by:
  - $G'(x,z) \wedge G(x,x',z')$
  - $G(x,x',z) \wedge G'(x',z')$

Given an initial instance of the tiling problem $c_0, \dots, c_n$, we encode it in the initial set of facts $\mathcal{F}_0$:

- $S'(a_i, a_{i+1})$ for $0 \leq i < n$;
- $G(a_0, a_i, b_{0,i})$ for $0 < i \leq n$;
- $G'(a_0, b_{0,0})$
- for all $0 \leq i \leq n$, letting j be such that c_i is the j-th color C_j, we create the *length-j gadget on $b_{0,i}$*: we create a path $C(b_{0,i}, d^1_{0,i}), C(d^1_{0,i}, d^2_{0,i}), \dots C(d^{j-1}_{0,i}, d^j_{0,i})$, and the fact $T(b_{0,i}, d^j_{0,i})$;

Correctness proof for the reduction. We claim that the tiling problem has a solution iff there is a (generally infinite) superset of $\mathcal{F}_0$ that satisfies Σ and violates Q, where the S^+ and C^+ predicates are interpreted as the transitive closure of S and C, from which we conclude the reduction and deduce the undecidability of QAtc as stated.

For the forward direction, from a solution f to the tiling problem for input $\vec{c}$, we construct $\mathcal{F} \supseteq \mathcal{F}_0$ as follows. We first create an infinite chain $S'(a_0,a_1), \dots, S'(a_m, a_{m+1}), \dots$ to complete the initial chain of S'-facts in $\mathcal{F}_0$, we create the implied S-facts, and make S^+ the transitive closure. We then create one fact $G(a_i, a_j, b_{i,j})$ for all $i \neq j$ in $\mathbb{N}$ and one

fact $G'(a_i, b_{i,i})$ for all $i \in \mathbb{N}$. Last, for all $i, j \in \mathbb{N}$, letting $l := f(i, j)$, we create the length-l gadget on $b_{i,j}$ with fresh elements.

It is clear that $\mathcal{F}$ contains the facts of $\mathcal{F}_0$. It is easy to verify that it satisfies Σ. To see that we do not satisfy the query, observe that:

- The C-path sanity disjuncts have no match because all C-paths created have length $\leq k$ and are on disjoint sets of elements;
- For the horizontal adjacency disjuncts, it is clear that, in any match, z must be of the form $b_{i,j}$ and z' of the form $b_{i,j+1}$; the reason for the three different forms is that the case where $i = j$ and $i \neq j$ are managed differently. Then, as f respects $\mathbb{H}$, we know that the Q_i and Q_j subqueries cannot be satisfied, because for any $l \in \mathbb{N}$ and $i', j' \in \mathbb{N}$, we have $Q_l(b_{i',j'})$ iff $f(i', j') = l$ by construction;
- The reasoning for the vertical adjacency disjuncts is analogous.

Hence, $\mathcal{F} \supseteq \mathcal{F}_0$, satisfies Σ, and violates Q, which concludes the proof of the forward direction of the implication.

For the backward direction, consider a $\mathcal{F} \supseteq \mathcal{F}_0$ that satisfies Σ and violates Q. Starting at the chain of S'-facts of $\mathcal{F}_0$, we can see that there is an infinite chain $a_0, \ldots, a_n, \ldots$ of S'-facts (whose elements may be distinct or not, this does not matter), and hence we infer the existence of the corresponding S-facts. We can also infer the existence of elements $b_{i,j}$ for all $i, j \in \mathbb{N}$ (again, these elements may be distinct or not) such that $G'(a_i, b_{i,i})$ holds and $G(a_i, a_j, b_{i,j})$ holds if $i \neq j$. From this we conclude that there is a fact $T(b_{i,j}, c_{i,j})$ for all $i, j \in \mathbb{N}$, with a C-path from $b_{i,j}$ to $c_{i,j}$. As the C-path sanity disjuncts are violated, there cannot be such a C-path of length $\geq k$, so we can define a function f from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{C}$ by setting $f(i, j)$ to be c_l where l is the length of one such path, for all $i, j \in \mathbb{N}$; this can be performed in a way that matches $\mathcal{F}_0$ (by choosing the path that appears in $\mathcal{F}_0$ if there is one).

Now, assume by contradiction that f is not a valid tiling. If there are $i, j \in \mathbb{N}$ such that $(f(i, j), f(i, j + 1)) \in \mathbb{H}$, then consider the match $x := a_i$, $y := a_j$, $y' := a_{j+1}$, $z := b_{i,j}$, and $z' := b_{i,j+1}$. If $i \neq j$ and $i \neq j + 1$, we know that $G(a_i, a_j, b_{i,j})$ and $G(a_i, a_{j+1}, b_{i,j+1})$ hold, and taking the witnessing paths used to define $f(i, j)$ and $f(i, j + 1)$, we obtain matches of $Q_{f(i,j)}(b_{i,j})$ and $Q_{f(i,j+1)}(b_{i,j+1})$, so that we obtain a match of one of the disjuncts of Q (one of the first horizontal adjacency disjuncts), a contradiction. The cases where $i = j$ and where $i = j + 1$ are similar and correspond to the second and third kinds of horizontal adjacency disjuncts. The case of $\mathbb{V}$ is handled similarly with the vertical adjacency disjuncts. Hence, f is a valid tiling, which concludes the proof of the backward direction of the implication, shows the equivalence, and concludes the reduction and the undecidability proof.

Adapting to a single distinguished relation. To prove the result with a single distinguished relation S^+, simply replace all occurrences of C and C^+ in the query and constraints by S and S^+. The rest of the construction is unchanged. The proof of the backwards direction is unchanged, using S in place of C; what must be changed is the proof of the forward direction.

Let f be the solution to the tiling problem. We start by constructing a set of facts $\mathcal{F}_1$ as before from f to complete $\mathcal{F}_0$, replacing the C-facts in the gadgets by S-facts. Now, we complete S^+ to add the transitive closure of these paths (note that they are disjoint from any other S-fact), and complete this to a set of facts to satisfy Σ: create G- and G'-facts, and create gadgets, this time taking all of them to have length $k + 1$: this yields $\mathcal{F}_2$. We repeat this last process indefinitely on the path of S-facts created in the gadgets of the previous iteration, and let $\mathcal{F}$ be the result of this infinite process, which satisfies Σ.

We justify as before that Q has no matches: as we create no S'-facts in $\mathcal{F}_i$ for all $i > 1$, it suffices to observe that no new matches of Q can include any of the new facts, because each disjunct includes an S'-fact. Hence, we can conclude as before.

From UCQ to CQ. To prove the result with a CQ rather than a UCQ, we proceed as for the proof of Theorem 5: we extend S' to be a ternary relation with a propagated value, add the relations of σ_{Or} (see the definition in the proof of Theorem 5), modify the S'-atoms in all disjuncts of the UCQ Q to add the variable, connect them as before yielding the CQ Q', and modify the initial instance to add dummy matches, to add the element $\mathtt{t}$ to the S'-facts that we create, and to add the facts of $\mathcal{F}_{\mathrm{Or}}$. As before, the proof of the forward direction is unchanged except that we add the value $\mathtt{t}$ to all S'-facts, and chase on the vacuous matches to satisfy the constraints (recall Definition 1). The query is violated because, thanks to the S' contained in each disjunct of Q, any match of the query ensures that we have a match of a disjunct of Q on the part that corresponds to $\mathcal{F}_0$ (not on the vacuous matches). For the backwards direction, we extract the tiling as before, and argue thanks to the vacuous matches that any violation of the tiling property would violate a UCQ of Q, and hence violate Q'. This concludes the proof. □

I Undecidability results related to linear orders (from Section 6)

We first prove Theorem 10. Recall the statement:

> **Theorem 10.** There is a signature $\sigma = \sigma_{\mathcal{B}} \sqcup \sigma_{\mathcal{D}}$ where $\sigma_{\mathcal{D}}$ is a single strict linear order relation, a CQ Q on σ, and a set Σ of inclusion dependencies on $\sigma_{\mathcal{B}}$ (i.e., not mentioning the linear order, so in particular base-covered), such that the following problem is undecidable: given a finite set of facts $\mathcal{F}_0$, decide $\mathsf{QAlin}(\mathcal{F}_0, \Sigma, Q)$.

Proof. We first show the claim for a UCQ rather than a CQ. As in the proof of Theorem 8, we fix an undecidable infinite tiling problem $\mathbb{C}$, $\mathbb{V}$, $\mathbb{H}$, and will reduce that problem to the QAlin problem.

Definition of the reduction. We consider the signature consisting of two binary relations R and D (for "right" and "down"), $k-1$ unary relations $K_1, \ldots, K_{k-1}$ (representing the colors), and one unary relation S (representing the fact of being a vertex of the grid – this is just to simplify things).

We put the following inclusion dependencies in Σ:

- $\forall x\, S(x) \rightarrow \exists y\, R(x,y)$
- $\forall x\, S(x) \rightarrow \exists y\, D(x,y)$
- $\forall xy\, R(x,y) \rightarrow S(y)$
- $\forall xy\, D(x,y) \rightarrow S(y)$

We will use the following abbreviations:

- $K'_1(x)$ stands for $\exists y\, x < y \wedge K_1(y)$
- $K'_k(x)$ stands for $\exists y\, x > y \wedge K_{k-1}(y)$
- for all $1 < i < k$, $K'_i(x)$ stands for $\exists yy'\, K_{i-1}(y) \wedge y < x \wedge x < y' \wedge K_i(y')$.

Intuitively, the K'_i describe the color of elements, which is encoded in their order relation to elements labeled with the K_i.

We consider a UCQ formed of the following disjuncts (existentially closed):

- $R(x,y) \wedge D(x,z) \wedge R(z,w) \wedge D(y,w') \wedge w < w'$
- $R(x,y) \wedge D(x,z) \wedge R(z,w) \wedge D(y,w') \wedge w' < w$
- for each $(c,c') \in \mathbb{H}$, $R(x,y) \wedge K'_c(x) \wedge K'_{c'}(y)$
- for each $(c,c') \in \mathbb{V}$, $D(x,y) \wedge K'_c(x) \wedge K'_{c'}(y)$

Intuitively, the first two disjuncts enforce a grid structure, by saying that going right and then down must be the same as going down and then right. The two other disjuncts enforce that there are no bad horizontal or vertical patterns.

Let us now present the reduction. Consider an instance $c_0, \ldots, c_n$ of the tiling problem. We construct a set of facts $\mathcal{F}_0$ as follows:

- $S(a_0), \ldots, S(a_n)$
- $R(a_{i-1}, a_i)$ for $1 \leq i \leq n$
- $K_i(b_i)$ for $1 \leq i \leq k$
- for each i such that c_i is the color C_1, $a_i < b_1$
- for each i such that c_i is the color C_k, $a_i > b_{k-1}$
- for each $1 < j < k$ and i such that c_i is C_j, $b_{j-1} < a_i$ and $a_i < b_j$

Correctness proof for the reduction. Let us show that the reduction is sound. Let us first assume that the tiling problem has a solution f. We construct a counterexample $\mathcal{F} \supseteq \mathcal{F}_0$ as a grid of the R and D relations, with the first elements of the first row being the $a_0, \ldots, a_n$, and with the color of elements being coded as their order relations to the b_j like when constructing I above. Complete the interpretation of $<$ to a total order by choosing one arbitrary total order among the elements labeled with the same color, for each color. The resulting interpretation is indeed a total order relation, formed of the following: some total order on the elements of color 1, the element b_1, some total order on the elements of color 2, the element b_2, …, the element b_{k-2}, some total order on the elements of color $k-1$, the element b_{k-1}, some total order on the elements of color k.

It is immediate that the result satisfies Σ. To see why it does not satisfy the first two disjuncts of the UCQ, observe that any match of $R(x,y) \wedge D(x,z) \wedge R(z,w) \wedge D(y,w')$ must have $w = w'$, by construction of the grid in $\mathcal{F}$. To see why it does not satisfy the other disjuncts, notice that any such match must be a pair of two vertical or two horizontal elements; since the elements can match only one K'_c which reflects their assigned color, the absence of matches follows by definition of f being a tiling.

Conversely, let us assume that there is a counterexample $\mathcal{F} \supseteq \mathcal{F}_0$ which satisfies Σ and violates Q. Clearly, if the first two disjuncts of Q are violated, then, for any element where S holds, considering its R and D successors that exist by Σ, and respectively their D and R successors, we reach the same element. Hence, from $a_0, \ldots, a_n$, we can consider the part of $\mathcal{F}$ defined as a grid of the R and D relations, and it is indeed a full grid (R and D edges occur everywhere they should). Now, we observe that any element except the b_j must be inserted at some position in the total suborder $b_1 < \cdots < b_{k-1}$, so that at least one predicate K'_j holds for each element of the grid (several K'_j may hold in case $\mathcal{F}$ has more elements than the b_i that are labeled with the K_i). Choose one of them, in a way that assigns to $a_0, \ldots, a_n$ their correct colors, and use this to define a function f that extends $a_0, \ldots, a_n$. We claim that this f indeed describes a tiling.

Assume by contradiction that it does not. If there are two horizontally adjacent values (i,j) and $(i+1,j)$ realizing a configuration (c,c') from $\mathbb{H}$, by completeness of the grid there is an R-edge between the corresponding elements u, v in $\mathcal{F}$. Further, by the fact that (i,j) and $(i+1,j)$ were given the color that they have in f, we must have $K'_c(u)$ and $K'_c(v)$ in $\mathcal{F}$, so that we must have had a match of a disjunct of Q, a contradiction. The absence of forbidden vertical patterns is proven in the same manner.

From UCQ to CQ. We now adapt the previous proof to use a CQ rather than a UCQ. Define the new signature σ'_{Or} as in the proof of Theorem 5 by adding the relations of σ_{Or}, and

otherwise increasing the arity of each relation of $\sigma_{\mathcal{B}}$ by one. Rewrite the IDs Σ as in the proof of Theorem 5, yielding:

- $\forall x\ S(x, b) \rightarrow \exists y\ R(x, y, b)$
- $\forall x\ S(x, b) \rightarrow \exists y\ D(x, y, b)$
- $\forall xy\ R(x, y, b) \rightarrow S(y, b)$
- $\forall xy\ D(x, y, b) \rightarrow S(y, b)$

We add to our initial set of facts $\mathcal{F}_0$ the facts of $\mathcal{F}_{\mathrm{Or}}$ as in the proof of Theorem 5.

We construct the CQ from the original UCQ by the same process as in the proof of Theorem 5, and construct the initial sets of facts as in that proof as well.

We then argue that this QAlin problem with the UCQ is equivalent to the one with the CQ. For the forward direction, from a solution to the initial instance $a_0, \ldots, a_n$ of the tiling problem, we build a suitable $\mathcal{F} \supseteq \mathcal{F}_0$ from the previously defined $\mathcal{F}$ by putting $\mathfrak{t}$ as the last element of R- and D-facts. We complete the vacuous matches by chasing on them with the dependencies of Σ (as in the proof of Theorem 5; recall Definition 1), we define $<$ arbitrarily on each vacuous match, arbitrarily between them, arbitrarily with $\mathfrak{f}$ and $\mathfrak{t}$, and then as before on the true grid. To show that the query has no match, we first claim that any match of the query must be a match of one of the disjuncts where the free variable is bound to $\mathfrak{t}$. Indeed, this is clear by definition of the Or and True relations. Now, we claim that none of the disjuncts have such a match. Indeed, if one disjunct has such a match, it implies that all facts of the match (except the order facts) have $\mathfrak{t}$ in the last position, and, as these facts are the same as in the original proof (up to the last element), the absence of match is for the same reason as in the original proof.

Conversely, let us consider a $\mathcal{F} \supseteq \mathcal{F}_0$ satisfying Σ and violating the query. We first observe that, for any disjunct of the query, it has a match where the free variable is bound to $\mathfrak{f}$, as witnessed by the vacuous matches. Hence, if the query has no match, it must mean that none of the disjuncts has a match with the free variable bound to $\mathfrak{t}$. Indeed, if there were one, then, from this match, from the vacuous matches, and using the facts which we know are present in the table of Or and True, we would obtain a match of the entire query. Hence, restricting our attention to the facts of $\mathcal{F}'$ with $\mathfrak{t}$ in their last position, using the fact that none of the disjuncts has a match there, we conclude as in the original proof. □

Now recall the statement of Corollary 5:

> **Corollary 5.** There is a signature $\sigma = \sigma_{\mathcal{B}} \sqcup \sigma_{\mathcal{D}}$ where $\sigma_{\mathcal{D}}$ is a single strict linear order relation, and a set Σ' of BaseFGTGD constraints, such that, letting $\top$ be the tautological query, the following problem is undecidable: given a finite set of facts $\mathcal{F}_0$, decide $\mathsf{QAlin}(\mathcal{F}_0, \Sigma', \top)$.

To prove Corollary 5 from Theorem 10, we take constraints Σ' that are equivalent to $\Sigma \wedge \neg Q$, where Σ and Q are as in the previous theorem. Recall that Σ is a set of inclusion dependencies on $\sigma_{\mathcal{B}}$, and therefore are BaseFGTGDs. Hence, it only remains to argue that $\neg Q$ can be written as a BaseFGTGD. Indeed, write Q as $\exists \vec{x}\ \varphi(\vec{x})$ and consider the constraint

$$\forall \vec{x}(\varphi(\vec{x}) \rightarrow \exists y(y < y))$$

where $<$ is the distinguished relation. Since $<$ must be a strict linear order in QAlin, $\exists y(y < y)$ is equivalent to $\bot$ and this new constraint is logically equivalent to $\neg Q$. Moreover, this constraint is trivially in BaseFGTGD since there are no frontier variables. Hence, $\neg Q$ can be written as an BaseFGTGD as claimed.