Query Answering with Transitive and Linear-Ordered Data

Antoine Amarilli ANTOINE.AMARILLI@TELECOM-PARISTECH.FR
Télécom ParisTech, Université Paris-Saclay

Michael Benedikt MICHAEL.BENEDIKT@CS.OX.AC.UK
University of Oxford

Pierre Bourhis PIERRE.BOURHIS@UNIV-LILLE1.FR
CNRS CRIStAL, Université Lille 1, INRIA Lille

Michael Vanden Boom MICHAEL.VANDENBOOM@CS.OX.AC.UK
University of Oxford

Abstract

We consider entailment problems involving powerful constraint languages such as *guarded existential rules* in which we impose additional semantic restrictions 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 variants 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 these 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. That is, we generalize the standard querying problem in databases to take into account not only the explicit information (the facts) but additional "implicit information" given by constraints. 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, Mugnier, Rudolph, & Thomazo, 2011). These classes include 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) of Andréka, Németi, and van Benthem (1998) and the guarded negation fragment (GNF) of Bárány, ten Cate, and Segoufin (2011).

A key challenge is to extend these decidability results to capture additional semantics of the relations that are important in practice but cannot be expressed in these classes. For example, the property that a binary relation is *transitive* or is the *transitive closure* of another relation cannot be expressed directly in guarded logics. Yet, transitive relations, such as the "part-of" relationship among components, are common in data modelling. Hence, we would like to be able to capture these special semantics when reasoning.

A semantic restriction related to transitivity is the fact that that a binary relation is a strict *linear order*: a transitive, irreflexive and total relation. When the data stored in a relation is numerical, it may often happen that the data satisfies integrity constraints involving the standard linear order $<$ on integers; e.g., for every tuple in a ternary relation the value in the first position of a binary relation is less than the value in the third. Again, the ability to reason about the additional semantics of $<$ may be crucial in inference, but it is not possible to express this in guarded logics.

In this work we will look at conditions that enable decidable query answering for all three semantic restrictions above: transitivity, transitive closure, and linear ordering. We will show that there are common techniques that can be used to analyze all three cases, but also significant differences.

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

Ganzinger, Meyer, and Veanes (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 that 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, Pieris, and Tendera (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). 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 & Duschka, 1998) and TGDs with arithmetic (Afrati, Li, & Pavlaki, 2008). Due to the acyclicity assumptions, QA is decidable in these settings, and has data complexity in CoNP. The fact that the data complexity can be CoNP-hard was shown by Abiteboul and Duschka (1998), while polynomial cases were isolated by Abiteboul and Duschka (1998) with inequalities, and by Afrati et al. (2008) with arithmetic.

Query answering that features transitivity restrictions has also been studied for constraints expressed in description logics, i.e., in an arity-two setting where the signature contains unary relations (concepts) and binary relations (roles). QA is then decidable for many description logics featuring expressive logical operators as well as transitivity, such as $\mathcal{ZIQ}$, $\mathcal{ZOQ}$, $\mathcal{ZOI}$ (Calvanese, Eiter, & Ortiz, 2009), Horn-$\mathcal{SROIQ}$ (Ortiz, Rudolph, & Šimkus, 2011), or regular-$\mathcal{EL}^{++}$ (Krötzsch & Rudolph, 2007); however, these logics restrict the interaction between transitivity and the more expressive features such as role inclusions and Boolean role combinations. QA becomes undecidable for more expressive description logics with transitivity such as $\mathcal{ALCOIF}^*$ (Ortiz, Rudolph, & Šimkus, 2010) and $\mathcal{ZOIQ}$ (Ortiz de la Fuente, 2010). Decidability of QA is open for $\mathcal{SROIQ}$ and $\mathcal{SHOIQ}$ (Ortiz & Simkus, 2012).

Contributions. The main contribution of this work is to introduce a broad class of constraints over arbitrary-arity vocabularies where query answering is decidable when we impose three varieties of additional semantics on some distinguished relations: being transitive, being the transitive closure of another relation, or being a linear order.

- We provide new results on QA with transitivity and transitive closure restrictions. We show that query answering is decidable 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 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 & Tendera, 2004) for satisfiability, 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 also guarded rules with negation and disjunction in the head. These logics can express integrity constraints, as well as 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* (Baget, Leclère, Mugnier, & Salvat, 2009), where at most one variable is shared between the body and head. 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 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.

- We provide both upper and lower bounds on 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.

Both classes of results apply to the motivating scenarios for distinguished relations mentioned earlier. Our results on transitivity show that QAtr and QAtc are decidable for BaseGNF, a restriction of GNF. This includes base-frontier-guarded rules, where we can in particular 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 on QAlin show that the problem is decidable for BaseCovGNF, the base-covered version of GNF. This 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 order constraint involving the attributes is satisfied.

Organization. In Section 2, we formally define the query answering problems that we study, and the constraint languages that we use. We present our main decidability results on query answering with transitive data in Section 3, and with linear-ordered data in Section 4; we analyze both the combined complexity and data complexity of these decidable cases. We prove lower bounds for these problems in Section 5, and show that slight changes to the conditions lead to undecidability in Section 6.

Our main results are summarized in Figure 1, and the languages that we study are illustrated in Figure 2 (please see Section 2 for the definitions).

Some technical material that is not essential for understanding our main results can be found in the appendices.

Acknowledgements. This is an extended version of the conference paper by Amarilli, Benedikt, Bourhis, and Vanden Boom (2016).

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 ANR Aggreg Project ANR-14-CE25-0017, INRIA Northern European Associate Team Integrated Linked Data. Benedikt was sponsored by the Engineering and Physical Sciences Research Council of the United Kingdom (EPSRC), under grants EP/M005852/1 and EP/L012138/1. Vanden Boom was partially supported by EPSRC grant EP/L012138/1.

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 to be Boolean. We also allow *unions of conjunctive queries* (UCQs), namely, disjunctions of CQs.

2.1 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.

Fragment	QAtr		QAtc		QAlin	
	data	**combined**	**data**	**combined**	**data**	**combined**
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

Figure 1: Summary of QA results (for base-covered fragments, queries are also base-covered). Please refer to Sections 3 and 4 for upper bounds, Section 5 for lower bounds, and Section 6 for undecidability results.

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 (with *distinguished* relations). 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 $\mathcal{F}_0 \wedge \Sigma \wedge \neg Q$ is satisfied by some set of facts $\mathcal{F}$ where each relation $R_i^+ \in \sigma_{\mathcal{D}}$ 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 $\mathsf{QAlin}(\mathcal{F}_0, \Sigma, Q)$ for this problem, if $\mathcal{F}_0 \wedge \Sigma \wedge \neg Q$ is satisfied by some set of facts $\mathcal{F}$ 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 for which we study these QA problems.

2.2 Dependencies

The first constraint languages that we study are restricted classes of *tuple-generating dependencies* (TGDs). A TGD is an FO sentence τ of the form $\forall \vec{x} \left(\bigwedge_i \gamma_i(\vec{x}) \rightarrow \exists \vec{y} \ \bigwedge_i \rho_i(\vec{x}, \vec{y}) \right)$ where $\bigwedge_i \gamma_i$ and $\bigwedge_i \rho_i$ are non-empty 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 $\vec{x}$. In this work, we will be particularly interested in *base-guards*, which are guards coming from the base relations in $\sigma_{\mathcal{B}}$.

A *frontier-guarded TGD* or FGTGD (Baget et al., 2011) is a TGD τ whose body contains a guard for the *frontier variables*, i.e., the variables that occur in both head and body. We introduce the *base frontier-guarded TGDs* (BaseFGTGDs) as the TGDs with a *base frontier guard*, i.e., a

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). Please refer to Section 3.3 for more information.

$\sigma_{\mathcal{B}}$-atom including all the frontier variables. We allow equality atoms $x = x$ to be guards, so $\mathsf{BaseFGTGD}$ subsumes *frontier-one TGDs* (Baget et al., 2011), which have one frontier variable. We also introduce the more restrictive class of *base-covered frontier-guarded TGDs* ($\mathsf{BaseCovFGTGD}$): they are the $\mathsf{BaseFGTGD}$s where, for every $\sigma_{\mathcal{D}}$-atom in the body, there is a base atom in the body containing its variables, called a *base-guard for the atom*. Note that each $\sigma_{\mathcal{D}}$-atom may have a different base-guard.

Inclusion dependencies (ID) are an important special case of frontier-guarded TGDs used in many applications. An ID is a FGTGD of the form $\forall \vec{x}\ R(\vec{x}) \to \exists \vec{y}\ S(\vec{x}, \vec{y})$, i.e., where the body and head contain a single atom, and where we further impose that 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, and the constraint is trivially base-covered.

2.3 Guarded logics

Moving beyond TGDs, we also study constraints coming from *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 guarded 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 (Bourhis, Morak, & Pieris, 2013), which generalize IDs: a DID is a first-order sentence of the form $\forall \vec{x}\ R(\vec{x}) \to \bigvee_{1 \leq i \leq n} \exists \vec{y_i}\ S_i(\vec{x}, \vec{y_i})$ such that, for every $1 \leq i \leq n$, the sentence $\forall \vec{x}\ R(\vec{x}) \to \exists \vec{y_i}\ S_i(\vec{x}, \vec{y_i})$ is an ID. In particular, any ID is a DID, as is seen by taking $n = 1$ in the disjunction.

In this work, we introduce the *base-guarded negation fragment* $\mathsf{BaseGNF}$ over σ: it is defined like GNF, but requires *base-guards* instead of guards. The *base-covered guarded negation fragment* $\mathsf{BaseCovGNF}$ over σ consists of $\mathsf{BaseGNF}$ 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 $\mathsf{BaseCovFGTGD}$s. Indeed, a $\mathsf{BaseCovFGTGD}$ of the form $\forall \vec{x}(\bigwedge \gamma_i \to \exists \vec{y} \bigwedge \rho_i)$ can be written in $\mathsf{BaseCovGNF}$ 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 $\mathsf{BaseCovGNF}$. A UCQ is *base-covered* if each disjunct is.

2.4 Examples

We illustrate the different constraint languages and queries by giving a few examples. Consider a signature with a binary base relation B, a ternary base relation C, and a distinguished relation R^+.

- $\forall xyz\big((R^+(x, y) \wedge R^+(y, z)) \to R^+(x, z)\big)$ is a TGD, but is not a FGTGD since the frontier variables x, z are not guarded. It cannot even be expressed in GNF.

- $\forall xy\big(R^+(x, y) \to B(x, y)\big)$ is an ID, hence a FGTGD. It is not a BaseID or even in $\mathsf{BaseGNF}$, since the frontier variables are not base-guarded.

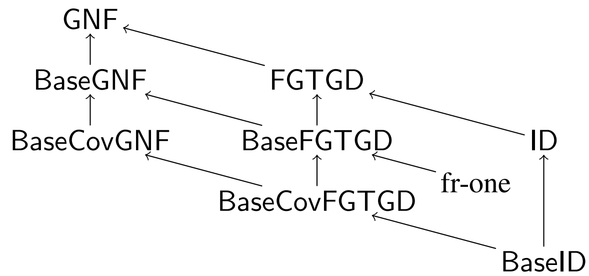

Figure 2: Taxonomy of fragments

- $\forall xyz\big((B(z,x) \wedge R^+(x,y) \wedge R^+(y,z)) \rightarrow R^+(x,z)\big)$ is a BaseFGTGD. However, 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.

2.5 Normal form

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 related notions that we will use in the proofs.

First, the *guardedness predicate* 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 is built from $\sigma_\mathcal{B}$-atoms with 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 $\mathsf{BaseGNF}$, 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 parameters will be important in later proofs.

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

Conversion into normal form. Observe that formulas in $\mathsf{BaseFGTGD}$ or $\mathsf{BaseCovFGTGD}$ 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 $\mathsf{BaseGNF}$ or $\mathsf{BaseCovGNF}$ 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, $\mathsf{BaseGNF}$ formulas can be converted into normal form, but with an exponential blow-up in size:

Proposition 1. *Let φ be a formula in* $\mathsf{BaseGNF}$. *We can construct an equivalent φ' in normal form in* $\mathsf{EXPTIME}$ *such that (i) $|\varphi'|$ is at most exponential in $|\varphi|$, (ii) the width of φ' is at most $|\varphi|$, (iii) the CQ-rank of φ' is at most $|\varphi|$, (iv) if φ is in* $\mathsf{BaseCovGNF}$, *then φ' is in normal form* $\mathsf{BaseCovGNF}$.

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

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

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. $\square$

2.6 Automata-related tools

In Section 3, we will make use of automata running on infinite binary trees, so we briefly recall some definitions and key properties; see (Thomas, 1997; Löding, 2011) for more background information. In particular, 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. For readers not interested in the details of the automaton construction in Section 3.2, this section can be skipped.

Trees. The input to the automata will be infinite full binary trees T over some finite set of propositions Γ; that is, each node v has exactly two children (one left child, and one right), and has a label $T(v) \in \mathcal{P}(\Gamma)$ that indicates the set of propositions that hold at v.

We can think of these trees as 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 will identify each node in a binary tree with a finite string over $\{0, 1\}$, with ϵ identifying the root, and $u0$ and $u1$ identifying the left child and right child of node u.

Tree automata. An *(2-way) 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, $\delta : Q \times \mathcal{P}(\Gamma) \to$

$\mathcal{B}^+(\mathrm{Dir} \times Q)$ is the transition function with directions $\mathrm{Dir} \subseteq \{\mathsf{left}, \mathsf{right}, \mathsf{up}, \mathsf{stay}\}$, 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 $\mathrm{Dir} \times Q$ (denoted $\mathcal{B}^+(\mathrm{Dir} \times Q)$) 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* or *membership game* (see Löding, 2011 for more information). The positions in the game are of the form $(q, v) \in Q \times T$. In position (q, v), Eve and Adam play a subgame based on $\delta(q, T(v))$, with Eve resolving disjunctions and Adam resolving conjunctions until an atom (d, q') in $\delta(q, T(v))$ is selected. Then the game continues from position (q', v') where v' is the node in direction d from v, and $v' = v$ if $d = \mathsf{stay}$. For example, if $\delta(q, T(v)) = (\mathsf{up}, s_1) \vee ((\mathsf{stay}, s_2) \wedge (\mathsf{right}, s_3))$, then the acceptance game starting from (q, v) would work as follows: Eve would select one of the disjuncts; if she selects the first disjunct then the game would continue from (s_1, u) where u is the parent of v, otherwise, Adam would choose one of the conjuncts and the game would continue from (s_2, v) or $(s_3, v1)$ depending on his choice.

A play $(q_0, v_0)(q_1, v_1)\dots$ is a sequence of positions in such a game. The play 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 processes the tree in a top-down fashion, using only directions left and right. A (1-way) nondeterministic automaton is a 1-way alternating automaton such that every transition function formula is of the form $\bigvee_j (\mathsf{left}, q_j) \wedge (\mathsf{right}, r_j)$.

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 1 (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 process takes a 1-way nondeterministic automaton that runs on trees with extra information about some predicate (annotated on the tree), and converts that automaton 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, Krötzsch, & Rudolph, 2015; Benedikt, Bourhis, & Vanden Boom, 2016).

Theorem 2. *Let $\Gamma' := \Gamma \cup \{P_1, \dots, P_j\}$. 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 the domain of T,

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

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 $\mathcal{A}'$, the number of priorities of $\mathcal{A}$ is linear in the number of priorities of $\mathcal{A}'$, and the overall size of $\mathcal{A}$ is linear in the size 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 node 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. □

3. Decidability results for transitivity

We are now ready to explore query answering for BaseGNF when the distinguished relations are transitively closed or are the transitive closure of certain base relations. We show that these query answering problems can actually be reduced to tree automata emptiness testing.

3.1 Deciding QAtc using automata

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 3. *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 3, 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 $\Sigma \in \mathsf{BaseGNF}$ 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 (specifically, k can be taken to be the maximum number of free variables in any subformula). The proof uses a standard technique, involving an unravelling based on "guarded negation bisimulation" (Bárány et al., 2011), so we defer the proof of this result to Appendix A.1.

Proposition 2. *Every sentence φ in* $\mathsf{BaseGNF}$ *has transitive-closure friendly k-tree-like witnesses, where $k \leq |\varphi|$.*

Hence, it suffices to test satisfiability for $\mathsf{BaseGNF}$ 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. 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 4. *Let φ be a sentence in* $\mathsf{BaseGNF}$ *over signature σ, and let $\mathcal{F}_0$ be a finite set of σ-facts. We can construct in* $\mathsf{2EXPTIME}$ *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 $|\varphi|$.

We present the details of this construction in the next section. It can be viewed as an extension of work by Calvanese, De Giacomo, and Vardi (2005), and incorporates ideas from automata for guarded logics by, e.g., Grädel and Walukiewicz (1999). It can also be viewed as an optimization of the construction in Benedikt et al. (2016), which we discuss in Section 3.4.

Because 2-way tree automata emptiness is decidable in time exponential in the number of states and priorities (Vardi, 1998), this yields the $\mathsf{2EXPTIME}$ bound for Theorem 3.

3.2 Automata for $\mathsf{BaseGNF}$ (Proof of Theorem 4)

Fix the signature $\sigma = \sigma_{\mathcal{B}} \sqcup \sigma_{\mathcal{D}}$. As described above, in order to test satisfiability of sentences in $\mathsf{BaseGNF}$ over σ, it suffices to consider only sets of facts with tree decompositions of some bounded width. We first describe how to encode sets of facts like this using trees over a finite alphabet, and then describe how to construct the automaton to prove Theorem 4.

Tree encodings. Consider a set of σ-facts $\mathcal{F}$, with an $\mathcal{F}_0$-rooted tree decomposition of width $k-1$ specified by tree (T, Child) and function λ. We must encode the information from λ in a finitary way, which we will do using a standard encoding technique for tree decompositions of bounded tree-width.

To achieve this, we specify a finite set U of *names* that can be used to describe the possibly infinite number of elements in $\mathcal{F}$; we will fix the size of this set momentarily. We include a name in U for each element in $\mathsf{elems}(\mathcal{F}_0)$; these are precisely the names used at the root of T, and in any other bag that mentions an element from $\mathcal{F}_0$. Then we map every other element in $\mathsf{elems}(\mathcal{F})$ to a name in U such that the following condition is satisfied: if u and v are neighboring nodes of T, then distinct elements of $\mathsf{elems}(\lambda(u)) \cup \mathsf{elems}(\lambda(v))$ are mapped to distinct names in u and v. This is possible if the size of U is $2k + l$, where $l = |\mathsf{elems}(\mathcal{F}_0)|$. Indeed, as each non-root bag uses at most k names, we know that $2k+l$ possible names are sufficient to be able to choose different names for distinct elements in neighboring nodes, and to choose names that do not conflict with the names for elements in the initial set of facts. This assignment of names is encoded using unary relations

D_a for each $a \in U$, so that $D_a(v)$ holds iff a is a name that was assigned to an element in v. The set of names used for $\mathsf{elems}(\mathcal{F}_0)$ is encoded using unary relations $V_{c/z}$ for each $z \in \mathsf{elems}(\mathcal{F}_0)$ and each $c \in U$, so that $V_{c/z}(v)$ holds iff v is the root and the element z from the initial set of facts is assigned name c. Facts in $\mathcal{F}$ are encoded using unary relations $R_{\vec{a}}$ for each $R \in \sigma$ of arity n and each n-tuple $\vec{a} \in U^n$, so that $R_{\vec{a}}(v)$ holds iff R holds of the tuple of elements named by $\vec{a}$ at v. This concludes the definition of our encoding scheme.

Tree decompositions and the corresponding encodings can generally have unbounded (possibly infinite) degree. For technical reasons, we want to use binary trees for our encodings, so we apply the well-known first-child, next-sibling transformation to the encoding described above (based on an arbitrary ordering of the children). We also make it a full binary tree by adding dummy nodes if necessary. After doing this, each node in a binary tree can be identified with a finite string over $\{0, 1\}$, with ϵ identifying the root, and $u0$ and $u1$ identifying the left child and right child of u. 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. Note that after this first-child, next-sibling transformation of the encoding described above, equality of elements is coded by the reuse of names across biological neighbors rather than actual neighbors in the tree. For convenience later, we use unary relations P_i for $i \in \{0, 1\}$ in the encoding such that $P_i(v)$ holds iff v is the i-th child of its parent.

We let $\tilde{\sigma}_{k,l}$ denote the *encoding signature* containing the relations described above, and we use the term *$\tilde{\sigma}_{k,l}$-tree* to refer to an infinite full binary tree over the signature $\tilde{\sigma}_{k,l}$.

Tree decodings. If a $\tilde{\sigma}_{k,l}$-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 $k - 1$.

Formally, let $\text{names}(v) := \{a \in U : D_a(v) \text{ holds}\}$ be the set of names used for elements in bag v in some tree; we will abuse notation and write $\vec{a} \subseteq \text{names}(v)$ to mean that $\vec{a}$ is a tuple over names from $\text{names}(v)$. Then a *consistent tree* T (with respect to $\tilde{\sigma}_{k,l}$ and $\mathcal{F}_0$) is a $\tilde{\sigma}_{k,l}$-tree such that every node v satisfies (i) $|\text{names}(v)| \leq k$, except for the root (where it has size l); (ii) for all $R_{\vec{a}} \in \tilde{\sigma}_{k,l}$, if $R_{\vec{a}}(v)$ holds then $\vec{a} \subseteq \text{names}(v)$; (iii) $P_i(v)$ holds iff v is the i-th child of its parent; (iv) for all $z \in \mathsf{elems}(\mathcal{F}_0)$, there is exactly one $c \in \text{names}(\epsilon)$ for the root ϵ such that $V_{c/z}(\epsilon)$ holds, and there is no $v \neq \epsilon$ with some $c \in \text{names}(v)$ such that $V_{c/z}(\epsilon)$ holds; (v) for every $c \in \text{names}(\epsilon)$, there is some $z \in \mathsf{elems}(\mathcal{F}_0)$ such that $V_{c/z}(\epsilon)$ holds; (vi) for each fact $R(z_1 \dots z_n) \in \mathcal{F}_0$, the fact $R_{c_1 \dots c_n}(\epsilon)$ holds in the tree, where each $c_i \in \text{names}(\epsilon)$ is the unique name such that $V_{c_i/z_i}(\epsilon)$ holds; (vii) 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 $a \in \text{names}(w_i)$ 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 $\{[v, a] : v \in T, a \in \text{names}(v)\}$, 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 $R \in \sigma$, 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 and local assignments. We will build an automaton for a sentence φ in $\mathsf{BaseGNF}$ inductively, constructing an automaton $\mathcal{A}_\psi$ for each subformula $\psi(\vec{x})$ of φ', the normal form formula equivalent to φ.

In order to deal with these formulas with free variables, the tree encodings can be extended with additional information about valuations for these 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). 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 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 $\vec{a} = a_1 \dots a_n \in U^n$ and $\vec{x} = x_1 \dots x_n$ 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. The point of localized automata is that they can test whether 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.

Main lemma. The construction is now described in the following lemma:

Lemma 1. *Given a normal form formula $\psi(\vec{x})$ in* $\mathsf{BaseGNF}$ *over signature σ and natural numbers k and l, we can construct a 2-way alternating parity tree automaton $\mathcal{A}_\psi$ such that for all consistent $\tilde{\sigma}_{k,l}$-trees T, for all local assignments $\vec{a}/\vec{x}$, and for all nodes v in T with $\vec{a} \subseteq$ names(v),*

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

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 $m_\psi = |\psi|$, r_ψ is the CQ-rank of ψ, and $K = 2k + l$. 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 $|\psi|$.

Proof. We proceed by induction on the normal form formula $\psi(\vec{x})$ in $\mathsf{BaseGNF}$. We will write m_ψ for $|\psi|$, write r_ψ for the CQ-rank of ψ, and write 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.

Atomic cases. For each of the atomic formulas $\psi(\vec{x})$, 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*:

Base atom. 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 (note that 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 force Eve to 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.

Guardedness predicate. 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.

Equality. 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.

Distinguished atom. 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 the label of v includes the fact $R(a, a')$, and 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 1). 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 2) 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 2, 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.

Guarded negation. 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 $\mathcal{A}_{\alpha}$, of 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).

Disjunction. 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).

CQ. Suppose $\psi(\vec{x})$ is a CQ $\exists y_1 \dots y_t(\alpha_1(\vec{z}_1) \wedge \cdots \wedge \alpha_s(\vec{z}_s))$ where each $\vec{z}_i$ is a tuple of variables coming from $\vec{x}$ and $y_1, \dots, y_t$, 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 $\mathcal{B}_{\alpha_1}, \dots, \mathcal{B}_{\alpha_s}$ with 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 1, 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 2) 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 2, 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.

CQ-shaped formulas. 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 variables Y_i marked on the tree. Specifically, for each Y_i of arity n and each $\vec{a} \in U^n$, there is a predicate $Y_{i,\vec{a}}$. If $Y_{i,\vec{a}}$ holds at some node v, then this indicates that the tuple of elements indexed by $\vec{a}$ at v is in the relation Y_i. These 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 automaton $\mathcal{A}_{\varphi_i}$ has at most $f(m_{\varphi_i}) \cdot 2^{f(Kr_{\varphi_i})}$ states, and the number of priorities is 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

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

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 2. *Given a finite set of σ-facts $\mathcal{F}_0$ and natural numbers k and l, we can construct a 2-way alternating parity tree automaton $\mathcal{A}_{\mathcal{F}_0}$ in time doubly exponential in $|\sigma| \cdot K$ (for $K = 2k + l$), such that for all $\tilde{\sigma}_{k,l}$-trees T,*

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

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 automata for each of the consistency conditions, so we omit the details. To check some fact $S(\vec{c})$ from $\mathcal{F}_0$, the automaton launches $\mathcal{A}^{\vec{c}/\vec{x}}_{S(\vec{x})}$ (obtained from Lemma 1) 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 $\mathcal{F}_0$. It can be checked that the automaton only needs two priorities, the number of states is exponential in $|\sigma| \cdot K$, and the overall size is at most doubly exponential in $|\sigma| \cdot K$. □

Concluding the proof. We can now conclude the proof of Theorem 4. We are given some sentence φ in $\mathsf{BaseGNF}$ over signature σ and some finite set of facts $\mathcal{F}_0$. Without loss of generality, we can assume that $|\varphi| \cdot |\mathcal{F}_0| \ge |\sigma|$. We construct the normal form φ' equivalent to φ in exponential time using Proposition 1. Although the size of φ' can be exponentially larger than φ, the CQ-rank is at most $|\varphi|$. By Proposition 2, we know that it suffices to consider only sets of facts with $\mathcal{F}_0$-rooted tree decompositions of width $|\varphi| - 1$, so we can restrict to considering automata on $\tilde{\sigma}_{k,l}$-trees for $k := |\varphi|$ and $l = |\mathsf{elems}(\mathcal{F}_0)|$.

Hence, we apply Lemma 1 to φ', k, and l, and construct a 2-way alternating parity tree automaton $\mathcal{A}_{\varphi'}$ for φ' (and hence φ) in time doubly exponential in $|\varphi| \cdot |\mathcal{F}_0|$. The number of states in this automaton is at most singly exponential in $|\varphi| \cdot |\mathcal{F}_0|$, and the number of priorities is linear in $|\varphi|$.

Next, we apply Lemma 2 to $\mathcal{F}_0$, k, and l, to get a 2-way alternating parity tree automaton $\mathcal{A}_{\mathcal{F}_0}$ that checks for consistency. This can be done in time doubly exponential in $|\sigma| \cdot (2k + l)$, which is at most doubly exponential in $|\varphi| \cdot |\mathcal{F}_0|$. The automaton has two priorities and the number of states at most singly exponential in $|\varphi| \cdot |\mathcal{F}_0|$.

Finally, we construct the desired 2-way alternating parity tree automaton $\mathcal{A}_{\varphi,\mathcal{F}_0}$ by taking the disjoint union of the automaton $\mathcal{A}_{\mathcal{F}_0}$ and $\mathcal{A}_{\varphi'}$, and giving Adam an initial choice of which of these automata to simulate. This automaton has a non-empty language iff φ is satisfiable. Moreover, the number of states in this automaton is still at most singly exponential in $|\varphi| \cdot |\mathcal{F}_0|$, and the number of priorities is linear in $|\varphi|$. This concludes the proof of Theorem 4.

3.3 Consequences for QAtr and other variants

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* 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).

As mentioned in the preliminaries, we have defined QAtr and QAtc based on transitive relations, not reflexive and transitive relations. However, the decidability and combined complexity results described in Theorem 3 and Corollary 1 (as well as the data complexity results that will be described later in Theorem 5 and Theorem 6) also apply to the corresponding query answering problems when the distinguished relations are reflexive transitive relations or the reflexive transitive closure of some base relation. Adapting the proofs to this case is a straightforward exercise: the only points that need to be changed are the precise handling of distinguished atoms in Proposition 7, and the game emulated by the automaton when dealing with distinguished atoms in the proof of Lemmas 1 and 2.

3.4 Relationship to GNFP$^{\mathsf{UP}}$

It is well-known that the transitive closure of a binary relation can be expressed in *least fixpoint logic* (LFP), the extension of FO with a least fixpoint operator. LFP can also express that a relation is transitively closed, or is the transitive closure of another relation. Unfortunately, satisfiability is undecidable for FO and hence LFP, so it is not possible to rely on this connection to prove decidability of QAtr or QAtc. On the other hand, the fixpoint extension of GNF (called GNFP) is decidable, but it is unable to express transitive closure (see Bárány et al., 2011; Benedikt et al., 2016), so it also cannot be used to decide QAtr or QAtc.

Recently, a new fixpoint logic called $\mathsf{GNFP}^{\mathsf{UP}}$ — *guarded negation fixpoint logic with unguarded parameters* — was introduced by Benedikt et al. (2016). This logic subsumes GNF and GNFP, and is expressive enough to define the transitive closure of a binary relation. It also subsumes a number of other highly expressive Datalog-like languages introduced by Bourhis et al. (2015). However, unlike LFP, satisfiability for $\mathsf{GNFP}^{\mathsf{UP}}$ is decidable (Benedikt et al., 2016). Hence, $\mathsf{QAtc}(\mathcal{F}_0, \Sigma, Q)$ for $\Sigma \in$ BaseGNF can be decided by converting $\mathcal{F}_0 \wedge \Sigma \wedge \neg Q$ to an equivalent $\mathsf{GNFP}^{\mathsf{UP}}$ sentence φ, and then testing for unsatisfiability of φ.

Each BaseGNF sentence with distinguished relations R_i^+ can be converted to an equivalent $\mathsf{GNFP}^{\mathsf{UP}}$ sentence with only base relations, since each occurrence of R_i^+ is replaced with a fixpoint formula using the base relation R_i that describes its transitive closure. The fixpoints in this formula are not nested in complicated ways; using the terminology of Benedikt et al., 2016, they have "parameter-depth" 1. Applying Theorem 20 in (Benedikt et al., 2016), this means that QAtc is decidable in 3EXPTIME. Thus, the approach using $\mathsf{GNFP}^{\mathsf{UP}}$ gives an alternative proof of the decidability of query answering with transitivity, but without the optimal 2EXPTIME complexity bound presented here. The automaton construction in this paper can be viewed as an optimization of the automaton construction for $\mathsf{GNFP}^{\mathsf{UP}}$ by Benedikt et al. (2016). The results on $\mathsf{GNFP}^{\mathsf{UP}}$, however, imply that query answering is decidable for BaseGNF not only when we have distinguished R_i^+, but when the distinguished relations are defined by regular expressions over base binary relations and their inverses, in the spirit of C2RPQs (see Example 4 in Benedikt et al., 2016).

Due to the syntactic restrictions in $\mathsf{GNFP}^{\mathsf{UP}}$, the translation described above would not produce a $\mathsf{GNFP}^{\mathsf{UP}}$ formula if distinguished relations were used as guards (i.e. if we started with a GNF formula, rather than a BaseGNF formula). This makes sense, since we will see in Section 6 that these query answering problems become undecidable when the distinguished relations are allowed as guards.

3.5 Data complexity

Our results in Theorem 3 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, ten Cate, & Otto, 2012).

Theorem 5. *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.*

We now prove the theorem. Fix the signature σ. For a set of σ-facts $\mathcal{F}$, an $\mathcal{F}, k$*-rooted structure* consists of $\mathcal{F}$ unioned with sets of facts $T_{\vec{c}}$ for $\vec{c} \in \mathsf{elems}(\mathcal{F})^k$ 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 2:

Proposition 3. *For any set of σ-facts $\mathcal{F}$, if a* $\mathsf{BaseGNF}$ *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.*

Now, 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 the distinguished relations R_i^+ 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). We interpret $P_\tau(x_1 \dots x_k)$ 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 3. *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 relations R_i^+ 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 for 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 3 we easily obtain:

Corollary 2. *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 5:

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 2.

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 3 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. □

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 6. *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 4), so we defer the proof of this result to the appendix.

As we will see in Section 5, restricting to QAtr is in fact essential to make data complexity tractable, as hardness holds otherwise.

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.

Unlike the previous section which used base-frontier-guarded constraints, we restrict to *base-covered constraints and queries* in this section. We do this because we will see in Section 6.2 that QAlin is undecidable if we allow base-frontier-guarded constraints, in contrast to the decidability results for QAtc and QAtr with such constraints.

Our main result will again be decidability of QAlin with these additional conditions, but the proof techniques differ from the previous section: instead of using automata, we reduce QAlin to a traditional query answering problem (without distinguished relations), by approximating the linear order axioms in GNF.

4.1 Deciding QAlin by approximating linear order axioms

We prove the following result about the decidability and combined complexity of QAlin:

Theorem 7. *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* BaseCovFGTGD*s.*

Our technique here is to reduce QAlin with BaseCovGNF constraints to traditional QA for GNF constraints without imposing any additional restrictions (like being transitive or a linear order). This implies decidability in 2EXPTIME using (Bárány et al., 2012). The reduction is quite simple, and hence could be applicable to other constraint classes: it simply adds additional constraints that enforce the linear order conditions. However, as we cannot express transitivity or totality in GNF, we 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 4. *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. 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: $\forall xy\ ((\mathsf{guarded}_{\sigma_{\mathcal{B}} \cup \{G\}}(x,y) \wedge \neg(x = y)) \rightarrow x < y \vee y < x)$
- 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 $\mathsf{BaseGNF}$. 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.

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. In this case, 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, we prove the contrapositive of the implication, namely, we suppose that $\mathsf{QA}(\mathcal{F}_0', \Sigma', Q)$ does not hold and show that $\mathsf{QAlin}(\mathcal{F}_0, \Sigma, Q)$ does not hold either. From our assumption, there is some counterexample $\mathcal{F}' \supseteq \mathcal{F}_0'$ 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 4. *The sentence $\Sigma' \wedge \neg Q$ has base-guarded-interface k-tree-like witnesses when taking $k := \max(|\Sigma \wedge \neg Q|, \mathsf{arity}(\sigma \cup \{G\}))$.*

Proof. By Proposition 1 and Proposition 9, $\Sigma \wedge \neg Q$ 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 $\mathsf{BaseGNF}$ with width at most k.

The guardedly total axiom is written in normal form $\mathsf{BaseGNF}$ 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 $\mathsf{BaseGNF}$ 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 $\mathsf{BaseGNF}$, and can even be written in normal form $\mathsf{BaseGNF}$ 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 9. □

Using this lemma, we can assume that we have some counterexample $\mathcal{F}' \supseteq \mathcal{F}'_0$ that satisfies $\Sigma' \wedge \neg Q$ and has a $(k-1)$-width base-guarded-interface tree decomposition. 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 $\Sigma \wedge \neg Q$, i.e., a counterexample allowing us to conclude that $\mathsf{QAlin}(\mathcal{F}_0, \Sigma, Q)$ does not hold. The problem is that there may be distinguished relations $<$ that are not strict linear orders in $\mathcal{F}'$. We thus show that $\mathcal{F}'$ can 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 allows us to conclude the proof as we have argued.

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

Lemma 5. *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 5 is the main technical result in this section and is deferred to the next subsection. It 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 this $\mathcal{F}''$ 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 $\mathsf{BaseCovGNF}$ is not affected by adding new $<$-facts about pairs of elements that are not base-guarded.

4.2 Extending approximate linear orders to genuine linear orders (Proof of Lemma 5)

It remains to prove Lemma 5. 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 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 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, it is base-guarded by definition of base-guarded-interface tree decompositions.

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. $\square$

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

Lemma 7. *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

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

(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 6 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. $\square$

Base-coveredness lemma. Last, we note that adding only facts about unguarded sets of elements cannot impact $\mathsf{BaseCovGNF}$ constraints. This is where we use the base-coveredness assumption.

Lemma 8. *Let $\mathcal{F}'' \supseteq \mathcal{F}'$ where $\mathcal{F}''$ contains additional facts about distinguished relations, but no new facts about base-guarded tuples of elements, and where we have $\mathsf{elems}(\mathcal{F}'') = \mathsf{elems}(\mathcal{F}')$. 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 $\mathsf{BaseCovGNF}$.

Let $\mathsf{BaseCovGNF}^+$ (respectively, $\mathsf{BaseCovGNF}^-$) denote the normal form $\mathsf{BaseGNF}$ 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, as is the base case for equality atoms, using the fact that $\mathsf{elems}(\mathcal{F}'') = \mathsf{elems}(\mathcal{F}')$. 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 $\varphi := \exists \vec{y}\,(\beta_1(\vec{x}_1\vec{y}_1) \wedge \cdots \wedge \beta_j(\vec{x}_j\vec{y}_j))$, 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. $\square$

Final proof of Lemma 5. We are now ready to prove Lemma 5. 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 7 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 8 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 5.

4.3 Data complexity

The result of Theorem 7 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 6). 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; we will see in Proposition 6 that it is CoNP-hard.

5. Hardness results

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 even without distinguished predicates (Bárány et al., 2012). However, in some cases, we can show the same hardness results for weaker languages, using the distinguished predicates.

In this section, we first summarize our hardness results in Sections 5.1 and 5.2, and then present the proofs in Section 5.3.

5.1 Hardness for QAtc

In the setting where we have distinguished relations interpreted as the transitive closure of other relations, we can show 2EXPTIME-hardness in combined complexity, and CoNP-hardness in data complexity, for the much weaker language of BaseIDs. This is in contrast with Theorem 6, which showed PTIME data complexity for QAtr with the more expressive language of BaseCovFGTGDs.

We show hardness via a reduction from QA with *disjunctive inclusion dependencies* (DIDs): recall their definition in Section 2.3. DIDs are known to be 2EXPTIME-hard in combined complexity (Bourhis et al., 2013, Theorem 2) and CoNP-hard in data complexity (Calvanese, Lembo, Lenzerini, & Rosati, 2006; Bourhis et al., 2013), even without distinguished relations. We use transitive closure to emulate disjunction — as was already suggested in the description logic context by Horrocks and Sattler (1999) — by creating an R_i^+-fact and limiting the length of a witness R_i-path using Q'. The choice of the length of the witness path among the possible lengths is used to mimic the disjunction. We thus show:

Theorem 8. *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 a base-covered CQ Q' on a signature σ' (with a single distinguished relation), such that* $\mathsf{QA}(\mathcal{F}_0, \Sigma, Q)$ *iff* $\mathsf{QAtc}(\mathcal{F}_0', \Sigma', Q')$.

With the results of Calvanese et al. (2006) and Bourhis et al. (2013), this immediately implies our hardness result:

Corollary 4. *The* QAtc *problem with* BaseID*s 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 5. *There is a base-covered CQ Q such that* $\mathsf{QAtc}(\mathcal{F}_0, \emptyset, Q)$ *is* CoNP*-hard in data complexity.*

We prove this by reducing the problem of 3-coloring a directed graph, known to be NP-hard, to the complement of QAtc: we can easily do this using dependencies with disjunction in the head. Hence, as in the proof of Theorem 8, we simulate this disjunction by using a choice of the length of paths that realize transitive closure facts asserted in $\mathcal{F}_0$.

All of these hardness results are first proven using UCQs rather than CQs, and then strengthened by eliminating the disjunction in the query, using a prior trick (see, e.g., Gottlob & Papadimitriou, 2003) to code the intermediate truth values of disjunctions within a CQ. We state in Appendix C the general lemmas about this transformation, and explain why the proofs of this section still hold when using a CQ rather than a UCQ.

5.2 Hardness for QAlin

Our hardness results for BaseIDs and QAtc also apply to QAlin, using the same technique of translating from DIDs. What changes is the technique used to code disjunction: rather than the length of a path in the transitive closure, we use the totality of the order relation between elements to code disjunction in the relative ordering of elements. We can thus show the following analogue to Theorem 8:

Theorem 9. *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' (not mentioning the distinguished relations), and base-covered CQ Q' on a signature σ' (with a single distinguished relation), such that* $\mathsf{QA}(\mathcal{F}_0, \Sigma, Q)$ *iff* $\mathsf{QAlin}(\mathcal{F}_0', \Sigma', Q')$.

Hence, we conclude from (Calvanese et al., 2006; Bourhis et al., 2013) our hardness result:

Corollary 5. *The* QAlin *problem with* BaseID *and base-covered CQs is* CoNP*-hard in data complexity and* 2EXPTIME*-hard in combined complexity.*

We can also use a reduction from 3-coloring to show hardness in data complexity even without constraints:

Proposition 6. *There is a base-covered CQ Q such that* $\mathsf{QAlin}(\mathcal{F}, \emptyset, Q)$ *is* CoNP*-hard in data complexity.*

5.3 Proof of Theorems 8 and 9

We now start to prove the results of Sections 5.1 and 5.2. In this section, we first prove the results about the translation from DIDs to QAtc and QAlin, namely, Theorems 8 and 9. In the next section, we show the data complexity hardness results without constraints (Propositions 5 and 6). We start by proving Theorem 8, and we will adapt the proof afterwards to show Theorem 9. Recall the claim:

Theorem 8. *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 will establish a weaker form of the result where Q' is allowed to be a UCQ: the extension where we only use a CQ is shown in Appendix C.4.

Defining σ' from σ. We create the signature σ' (featuring both base and distinguished relations) from the signature σ of the DIDs and from the DIDs Σ themselves by:

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

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

  a base relation $\mathsf{Witness}_\tau$ in σ' of arity $|\vec{x}| + \sum_i |\vec{y_i}| + 2n + 2$. For simplicity, we will always use the same variables when writing $\mathsf{Witness}_\tau$-atoms, namely, we will write them $\mathsf{Witness}_\tau(\vec{x}, e, f, \vec{y}_1, e_1, f_1, \ldots, \vec{y}_n, e_n, f_n)$.

Defining Σ' from Σ and σ. We then create the BaseIDs Σ' from the DIDs Σ. First, for each relation R in σ, we create the following BaseID, asserting that the two additional positions of the base relation R' must be connected by an E-path.

$$\tau_R' : \forall \vec{x}\, e\, f\; R'(\vec{x}, e, f) \rightarrow E^+(e, f)$$

The intuition is that the failure of the query will impose that this E-path have length at most 2, so it has length either 1 or 2. Facts with a path of length 1 will be called *genuine facts*, which intuitively means that they really hold, and those with a path of length 2 will be called *pseudo-facts*, intuitively meaning that they will be ignored.

Then, for each DID $\tau : \forall \vec{x}\; R(\vec{x}) \rightarrow \bigvee_{1 \leq i \leq n} \exists \vec{y_i}\; R_i(\vec{x}, \vec{y_i})$, we create multiple BaseIDs. First, we create a BaseID τ' with a $\mathsf{Witness}_\tau$-fact in the head:

$$\tau' : \forall \vec{x}\, e\, f\; R'(\vec{x}, e, f) \rightarrow \exists \vec{y}_1\, e_1\, f_1\, \ldots\, \vec{y}_n\, e_n\, f_n\; \mathsf{Witness}_\tau(\vec{x}, e, f, \vec{y}_1, e_1, f_1, \ldots, \vec{y}_n, e_n, f_n)$$

Then, for $1 \leq i \leq n$, we create the following BaseID τ_i':

$$\tau_i' : \forall \vec{x}\, e\, f\, \vec{y_1}\, e_1\, f_1\, \ldots\, \vec{y_n}\, e_n\, f_n\; \mathsf{Witness}_\tau(\vec{x}, e, f, \vec{y}_1, e_1, f_1, \ldots, \vec{y}_n, e_n, f_n) \rightarrow R_i'(\vec{x}, \vec{y_i}, e_i, f_i)$$

In other words, whenever a DID τ would be applicable on a fact $R'(\vec{c}, e, f)$, we will create a fact $\mathsf{Witness}_\tau(\vec{c}, e, f, \vec{d}_1, e_1, f_1, \ldots, \vec{d}_n, e_n, f_n)$, which will cause *all* head atoms $R_i'(\vec{c}, \vec{d}_i, e_i, f_i)$ for the DID to be instantiated. However, thanks to the two additional positions, we will be free to choose which of these facts are pseudo-facts, and which are genuine. The query will then enforce the correct semantics for DIDs, by prohibiting $\mathsf{Witness}_\tau$-facts whose match was genuine but where all instantiated heads are pseudo-facts.

Defining Q' from Q, σ, and Σ. The UCQ Q' contains the following disjuncts (existentially closed):

- *Q-generated disjuncts*: For each disjunct ψ of the original UCQ Q, we create one disjunct ψ' in the UCQ Q' obtained by replacing each atom $R(\vec{x})$ of ψ by the conjunction $R'(\vec{x}, e, f) \wedge E(e, f)$, where e and f are fresh. That is, the query Q' matches whenever we have a witness for Q consisting of genuine facts.

- *E-path length restriction disjuncts*: For each predicate R in σ, we create the following disjunct in Q':
$$R'(\vec{x}, e, f) \wedge E(e, y_1) \wedge E(y_1, y_2) \wedge E(y_2, y_3).$$
This disjunct succeeds if the E-path annotating an R'-fact has length ≥ 3. Hence, for any fact $R'(\vec{a}, e, f)$, the E^+-fact from e to f enforced by the DID τ_R' in Σ must make $R'(\vec{a}, e, f)$ either a genuine fact or a pseudo-fact.

- DID *satisfaction disjuncts*: For every DID $\tau : \forall \vec{x}\ R(\vec{x}) \rightarrow \bigvee_i \exists \vec{y_i}\ R_i(\vec{x}, \vec{y_i})$ in Σ, we create the following disjunct in Q':
$$Q_\tau : \mathsf{Witness}_\tau(\vec{x}, e, f, \vec{y}_1, e_1, f_1, \ldots, \vec{y}_n, e_n, f_n) \wedge E(e, f) \wedge \bigwedge_i \left(E(e_i, w_i) \wedge E(w_i, f_i)\right).$$
Informally, the failure of Q_τ enforces that we cannot have the body of τ holding as a genuine fact and each head disjunct realized by a pseudo-fact.

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

Defining $\mathcal{F}_0'$ from $\mathcal{F}_0$. 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. Hence, all facts of $\mathcal{F}_0$ are created as genuine facts.

We have now defined σ', Σ', Q', and $\mathcal{F}_0'$. 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.

Forward direction of the correctness proof. 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, to make them genuine facts), yielding $\mathcal{F}_1$. The result of this process consists only of genuine facts, and satisfies all BaseIDs of the form τ_R'.

- Expand $\mathcal{F}_1$ to a superset of facts $\mathcal{F}_2$ by adding facts that solve violations of all dependencies in Σ' of the form τ'.

  Specifically, for every DID of the form τ', letting $\tau : \forall \vec{x}\ R(\vec{x}) \rightarrow \bigvee_{1 \leq i \leq n} \exists \vec{y_i}\ S_i(\vec{x}, \vec{y_i})$ be the corresponding DID in Σ, consider a fact $F' = R(\vec{c}, e, f)$ of $\mathcal{F}_1$ that matches the body of τ'. From the way we constructed $\mathcal{F}_1$, we know that it must contain $E(e, f)$, and that $\mathcal{F}$ must contain the fact $F = R(\vec{c})$. Now, as $\mathcal{F}$ satisfies τ, we know that there is $1 \leq i_0 \leq n$ such that $R_{i_0}(\vec{c}, \vec{d_{i_0}})$ holds in $\mathcal{F}$ for some choice of $\vec{d_{i_0}}$. Hence, by construction of $\mathcal{F}_1$ again, we know that it contains $F'_{i_0} = R'_{i_0}(\vec{c}, \vec{d_{i_0}}, e_{i_0}, f_{i_0})$ and $E(e_{i_0}, f_{i_0})$ for some e_{i_0} and f_{i_0}. For every $i \in \{1, \ldots, n\} \backslash \{i_0\}$, create fresh elements $\vec{d_i}, e_i, f_i, w_i$ in the domain of $\mathcal{F}_2$. Now, add to $\mathcal{F}_2$ the fact $F_{\mathrm{w}} = \mathsf{Witness}_\tau(\vec{c}, e, f, \vec{d_1}, e_1, f_1, \ldots, \vec{d_n}, e_n, f_n)$: in this fact, $\vec{c}$ is as in F', $\vec{d_{i_0}}, e_{i_0}, f_{i_0}$ are as in F'_{i_0}, and for the $\vec{d_i}, e_i, f_i$ for $i \neq i_0$ are the fresh elements that we just created.

  It is easy to see now that $\mathcal{F}_2$ now satisfies all $\mathsf{BaseIDs}$ of the form τ', and it still satisfies those of the form τ'_R. Further, it is easy to see that for any $\mathsf{Witness}$-fact F_{w} of $\mathcal{F}_2$ that violates a dependency of the form τ'_i in Σ', the value i must be different from the value i_0 used when creating F_{w} (as for $i = i_0$ the fact F'_{i_0} considered when creating F_{w} witnesses that F_{w} is not a violation of i_0). Hence, we have the following property: for any violation of a dependency of Σ' in $\mathcal{F}_2$, the elements to be exported are in $\mathsf{elems}(\mathcal{F}_2) \backslash \mathsf{elems}(\mathcal{F}_1)$, and they only occur in one fact and in one position of $\mathcal{F}_2$ and do not occur in $\mathcal{F}_1$.

- We now create $\mathcal{F}_3$ from $\mathcal{F}_2$ by taking care of the remaining violations by performing the *chase* with Σ' wherever applicable, always creating fresh elements.[2]

  Whenever we need to create a witness for some E^+ requirement, we always create an E-path of length 2 with a fresh element in the middle, that is, all facts created in $\mathcal{F}_3 \backslash \mathcal{F}_2$ are $\mathsf{Witness}_\tau$-facts and pseudo-facts.

Let $\mathcal{F}' := \mathcal{F}_3$. We now check that $\mathcal{F}'$ is a counterexample to $\mathsf{QAtc}(\mathcal{F}'_0, \Sigma', Q')$. As $\mathcal{F} \supseteq \mathcal{F}_0$, it is clear that $\mathcal{F}_1 \supseteq \mathcal{F}'_0$, so that $\mathcal{F}' \supseteq \mathcal{F}'_0$. Further, it is immediate by definition of the chase that $\mathcal{F}'$ satisfies Σ'. There remains to check that $\mathcal{F}'$ violates Q'. To this end, we will first observe that, by construction of $\mathcal{F}'$, the only E-facts that we create are paths of length 1 on fresh elements in the construction of $\mathcal{F}_1$ from $\mathcal{F}$, and paths of length 2 on fresh elements in the chase in $\mathcal{F}_3$ (these elements were either created as nulls in the chase in $\mathcal{F}_3$, or they were created in $\mathcal{F}_2$ where they only occurred in one fact and at one position). In particular, observe that, whenever we create an E-fact at any point, its endpoints are fresh (they have just been created), so that E-paths have length 1 or 2

2. The *chase* is a standard database construction (Abiteboul, Hull, & Vianu, 1995), which 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 $\vec{x} \cup \vec{y}$ to $\mathcal{F}$ such that the facts of $\psi(h(\vec{x}), h(\vec{y}))$ are 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.

   Note that the set of facts obtained from the chase of $\mathcal{F}$ by Σ satisfies Σ, and that, as all existentially quantified variables when applying rules are instantiated by fresh existential witnesses, no new facts are created on an element a of the initial set of facts $\mathcal{F}$ unless a occurs in a fact of $\mathcal{F}$ which is part of a violation.

and are on pairwise disjoint sets of elements. Hence, as $\mathcal{F}'$ satisfies the τ'_R, any fact $R'(\vec{c}, e, f)$ in $\mathcal{F}'$ is either a genuine fact (i.e., $E(e, f)$ holds in $\mathcal{F}'$) or a pseudo-fact (i.e., there is an E-path of length 2 from e to f in $\mathcal{F}'$), and *these two properties are mutually exclusive*.

We now check that Q' is violated, by considering each possible kind of disjuncts. For the *E-path length restriction disjuncts*, we just explained that the interpretation of E in $\mathcal{F}'$ consists of disjoint paths of length 1 or 2, so there is no E-path of length 3 at all in $\mathcal{F}'$.

For the DID *satisfaction disjuncts*, we will first observe that there are two kinds of $\mathsf{Witness}_\tau$-facts in $\mathcal{F}'$. Some $\mathsf{Witness}_\tau$-facts $\mathsf{Witness}_\tau(\vec{c}, e, f, \vec{d_1}, e_1, f_1, \ldots, \vec{d_n}, e_n, f_n)$ were created in $\mathcal{F}_2$, and for these we always have $E(e, f)$ in $\mathcal{F}_1$ (hence in $\mathcal{F}'$), and the same is true also of $E(e_{i_0}, f_{i_0})$ for the $1 \leq i_0 \leq n$ considered when creating them (using the fact that $\mathcal{F}$ satisfied Σ). All other $\mathsf{Witness}_\tau$-facts of $\mathcal{F}'$ are created in $\mathcal{F}_3$ and include only elements from $\mathsf{elems}(\mathcal{F}_3)\backslash\mathsf{elems}(\mathcal{F}_2)$ or elements occurring only in one position at one fact in $\mathcal{F}_2$ (and not occurring in $\mathcal{F}_1$): hence, for these $\mathsf{Witness}_\tau$-facts, neither $E(e, f)$ holds in $\mathcal{F}'$ nor does $E(e_i, f_i)$ hold for any $1 \leq i \leq n$.

This suffices to ensure that disjuncts of the form Q_τ in Q' cannot have a match in $\mathcal{F}'$, because their $\mathsf{Witness}_\tau$-atom can neither match $\mathsf{Witness}_\tau$-facts created in $\mathcal{F}_3$ (as $E(e, f)$ does not hold for them, unlike what Q_τ requires, remembering that paths of length 1 and 2 are mutually exclusive) nor $\mathsf{Witness}_\tau$-facts created in $\mathcal{F}_2$ (because, for $i = i_0$, the fact $E(e_{i_0}, f_{i_0})$ holds for them, violating again what Q_τ requires). Hence, the DID satisfaction disjuncts have no match in $\mathcal{F}'$.

Finally, for the *Q-generated disjuncts*, observe that any match of them must be on genuine facts, i.e., on facts of $\mathcal{F}'$ created for facts of $\mathcal{F}$, so we can conclude because $\mathcal{F}$ violates Q.

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

Backward direction of the correctness proof. 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 $\mathcal{F} \supseteq \mathcal{F}_0$. Further, as $\mathcal{F}'$ violates Q', it is clear that $\mathcal{F}$ violates Q, because 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.

Hence, assume by contradiction that there is a DID $\tau : \forall \vec{x}\ R(\vec{x}) \rightarrow \bigvee_{1 \leq i \leq n} \exists \vec{y_i}\ S_i(\vec{x}, \vec{y_i})$ of Σ and a fact $R(\vec{c})$ of $\mathcal{F}$ which violates it. Let $F' = R'(\vec{c}, e, f)$ be the fact in $\mathcal{F}'$ from which we created F; we know that $E(e, f)$ holds in $\mathcal{F}'$. Since $\mathcal{F}'$ satisfies τ' in Σ', we know that there are $\vec{d_1}, e_1, f_1, \ldots, \vec{d_n}, e_n, f_n$ such that $\mathsf{Witness}_\tau(\vec{c}, e, f, \vec{d_1}, e_1, f_1, \ldots, \vec{d_n}, e_n, f_n)$ holds. Further, as $\mathcal{F}'$ satisfies the τ'_i for $1 \leq i \leq n$, we know that $S_i(\vec{d_i}, e_i, f_i)$ hold in $\mathcal{F}'$ for all $1 \leq i \leq n$, and as $\mathcal{F}'$ satisfies the τ'_{S_i}, we know that $E^+(e_i, f_i)$ holds, so that there is at least one E-path connecting e_i and f_i. As the E-path length-restriction disjuncts are violated in $\mathcal{F}'$, these E-paths all have length in $\{1, 2\}$, and as the DID satisfaction disjunct Q_τ is violated in $\mathcal{F}'$, there is $1 \leq i_0 \leq n$ such that no path from e_{i_0} to f_{i_0} in $\mathcal{F}$ has length 2, so that some path must have length 1. Hence, $\mathcal{F}$ contains $S_{i_0}(\vec{d_{i_0}}, e_{i_0}, f_{i_0})$ and $E(e_{i_0}, f_{i_0})$, so $\mathcal{F}$ contains $S_{i_0}(\vec{d_{i_0}})$, which witnesses that τ is satisfied on $R(\vec{c})$ in $\mathcal{F}$, a contradiction. Hence, $\mathcal{F}$ satisfies Σ, which concludes the proof of Theorem 8.

We now prove Theorem 9, which states:

Theorem 9. *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' (not mentioning the distinguished relations), and base-*

covered 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 8. Again, we show the claim with a base-covered UCQ, and we show the result for a CQ in Appendix C.4

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.

Defining σ' from σ. The signature σ' is defined as in the proof of Theorem 8 except that we do not add the predicates E and E^+, but add a predicate $<$ as a distinguished relation instead.

Defining Σ' from Σ and σ. We also define Σ' as before except that we do not create the BaseIDs of the form $\forall \vec{x}\, e\, f\;\; R'(\vec{x}, e, f) \rightarrow E^+(e, f)$. Constraints like this are not necessary because the totality of $<$ already enforces the corresponding property. This means that Σ' does not mention the distinguished relations.

Defining Q' from Q, σ, and Σ. The UCQ Q' contains the following disjuncts (existentially closed), which are clearly base-covered:

- *Q-generated disjuncts*: For each disjunct ψ of the original UCQ Q, we create one disjunct ψ' in the UCQ Q' where each atom $R(\vec{x})$ is replaced by the conjunction $R'(\vec{x}, e, f) \wedge e < f$, where e and f are fresh. That is, the query Q' matches whenever we have a witness for Q consisting of genuine facts.

- *Order restriction disjuncts*: For each predicate R in σ, we create a disjunct $R'(\vec{x}, e, e)$. Intuitively, failure of this disjunct imposes that, for each relation $R' \in \sigma'$ that stands for a relation $R \in \sigma$, the elements in the two last positions must be different; so every fact must be either a genuine fact or a pseudo-fact.

- DID *satisfaction disjuncts*: For every DID $\tau : \forall \vec{x}\; R(\vec{x}) \rightarrow \bigvee_i \exists \vec{y_i} R_i(\vec{x}, \vec{y_i})$ in Σ, we create the following disjunct in Q':

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

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

Defining $\mathcal{F}'_0$ from $\mathcal{F}_0$. The process to define $\mathcal{F}'_0$ from $\mathcal{F}_0$ is defined like in the proof of Theorem 8 except that, instead of creating the facts $E(b_F, b'_F)$, we create facts $b_F < b'_F$.

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 8, so we sketch the proof and highlight the main differences.

Forward direction of the correctness proof. Let $\mathcal{F} \supseteq \mathcal{F}_0$ satisfy Σ and violate Q, and construct $\mathcal{F}'$ that satisfies Σ' and violates Q' and in which $<$ is an order relation. We do so as follows:

- Build $\mathcal{F}'$ from $\mathcal{F}$ by expanding each fact F with two fresh elements b_F and b'_F and adding the fact $b_F < b'_F$ to make it a genuine fact.

- Create $\mathcal{F}_2$ and $\mathcal{F}_3$ as before, except that pseudo-facts and genuine facts are annotated with $<$-facts rather than E^+-facts.
- Add one step where we construct $\mathcal{F}'$ from $\mathcal{F}_3$ by completing $<$ to be a total order. To do so, observe that $<$ in $\mathcal{F}_3$ must be a partial order, because all order facts that we have created are on disjoint elements (they are of the form $b_F < b'_F$ or $b'_F < b_F$ where b_F and b'_F are elements specific to a fact F). Hence, we define $\mathcal{F}'$ by simply completing $<$ 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' (note that the additional order facts created from $\mathcal{F}_3$ to $\mathcal{F}'$ cannot create a violation of Σ', as it does not mention $<$), 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. For the order-restriction disjuncts, we simply observe that for any R'-fact $R'(\vec{a}, e, f)$ in $\mathcal{F}'$, by construction we always have $e \neq f$.

Backward direction of the correctness proof. 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 8, but instead of the τ'_{S_i} we rely on totality of the order to deduce that either $e_i < f_i$, $e_i = f_i$, or $f_i < e_i$ for all i, and we rely on the order-restriction disjuncts (rather than the E-path length-restriction disjuncts) to deduce that either $e_i < f_i$ or $f_i < e_i$. We conclude as before by the DID satisfaction disjuncts that we must have $e_i < f_i$ for some i. Thus, we deduce from the satisfaction of Σ' by $\mathcal{F}'$ that $\mathcal{F}$ satisfies Σ, which concludes the backward direction of the correctness proof, and finishes the proof of Theorem 9.

5.4 Proof of Propositions 5 and 6

We now give data complexity lower bounds that show CoNP-hardness even in the absence of constraints. We first prove Proposition 5:

Proposition 5. *There is a base-covered CQ Q such that* $\mathsf{QAtc}(\mathcal{F}_0, \emptyset, Q)$ *is* CoNP*-hard in data complexity.*

Proof. We will show the result for a UCQ Q, and we extend it to a CQ in Appendix C.4. We show CoNP-hardness by reducing the 3-colorability problem in PTIME to the negation of the QAtc problem: this well-known NP-hard problem asks, given an undirected graph $\mathcal{G}$, whether it is 3-colorable, i.e., whether there is a mapping from the vertices of $\mathcal{G}$ to a set of 3 colors (without loss of generality the set $\{1, 2, 3\}$) such that no two adjacent vertices are assigned the same color. Observe that we can modify slightly the definition of this problem to allow vertices to carry multiple colors, i.e., be colored by *non-empty* subsets of $\{1, 2, 3\}$): the use of multiple colors on a vertex imposes more constraints on the vertex, so makes our life harder. In other words, we can restrict the search for solutions to colorings where each vertex has one single color, but when encoding the 3-colorability problem to QAtc we do not need to impose that vertices carry *exactly* one color (we must just impose that they carry *at least* one color).

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

- One binary relation G to code the edges of the graph provided as input to the reduction;
- One binary relation E and its transitive closure E^+ (playing a similar role as in the proof of Theorem 8);
- For each $\chi \in \{1, 2, 3\}$, a ternary relation C_χ. Intuitively, the first position of C_χ-facts will contain the element that codes a vertex (and occurs in the G-facts that describe the edges incident to that vertex), and the positions 2 and 3 will contain elements playing a similar role to elements e and f in R'-facts in the proof of Theorem 8. Namely, for a fact $C_\chi(a, e, f)$, if e and f are connected by a path of length 1, this will indicate that vertex a has color χ, while if they are connected by a path of length 2 this will indicate that a does not have color χ.

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

- *E-path length restriction disjuncts*: For each $\chi \in \{1, 2, 3\}$, a disjunct that holds if the E-path for C_χ-facts has length ≥ 3:

$$C_\chi(x, e, f) \wedge E(e, y_1) \wedge E(y_1, y_2) \wedge E(y_2, y_3)$$

- *Adjacency disjuncts:* For $\chi \in \{1, 2, 3\}$, a disjunct Q_i that holds if two adjacent vertices were assigned the same color:

$$C_\chi(x, e, f) \wedge E(e, f) \wedge G(x, x') \wedge C_\chi(x', e', f') \wedge E(e', f')$$

- *Coloring disjunct:* A disjunct that holds if a vertex was not assigned any color:

$$\bigwedge_{\chi \in \{1,2,3\}} C_\chi(x, e_\chi, f_\chi) \wedge E(e_\chi, w_\chi) \wedge E(w_\chi, f_\chi)$$

Given an undirected graph $\mathcal{G}$, we then code it in PTIME as the set of facts $\mathcal{F}_0$ defined by having:

- One fact $G(x, y)$ and one fact $G(y, x)$ for each edge $\{x, y\}$ in $\mathcal{G}$
- One fact $C_\chi(x, e_{x,\chi}, f_{x,\chi})$ and one fact $E^+(e_{x,\chi}, f_{x,\chi})$ for each vertex x in $G(x, y)$ and for each $\chi \in \{1, 2, 3\}$, where all the $e_{x,\chi}$ and $f_{x,\chi}$ 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.

For the forward direction, consider a 3-coloring of $\mathcal{G}$. Construct $\mathcal{F} \supseteq \mathcal{F}_0$ as follows. For each vertex x of $\mathcal{G}$ (with facts $C_\chi(x, e_{x,\chi}, f_{x,\chi}) \in \mathcal{F}_0$ as defined above for all $\chi \in \{1, 2, 3\}$), create the facts $E(e_{x,\chi}, f_{x,\chi})$ where χ is the color assigned to x, and the facts $E(e_{x,\chi'}, w_{x,\chi'})$ and $E(w_{x,\chi'}, f_{x,\chi'})$ for the two other colors $\chi' \in \{1, 2, 3\} \backslash \{i\}$ (with the $w_{x,\chi'}$ 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 because each vertex has some color. Finally, the definition of a 3-coloring ensures that the adjacency disjuncts do not match either. Hence, $\mathcal{F}$ is a set of facts violating Q.

For the backward direction, consider some $\mathcal{F} \supseteq \mathcal{F}_0$ where E^+ is the transitive closure of E that violates Q. For any vertex x of $\mathcal{G}$ and for all $\chi \in \{1,2,3\}$, letting $C_\chi(x, e_{x,\chi}, f_{x,\chi})$ be the facts as defined in the construction, as the fact $E^+(e_{x,\chi}, f_{x,\chi})$ holds in $\mathcal{F}_0$ for each $\chi \in \{1,2,3\}$ and E^+ must be the transitive closure of E in $\mathcal{F}'$, there must be an E-path from $e_{x,\chi}$ to $f_{x,\chi}$. Further, as $\mathcal{F}$ violates the E-path length restriction disjuncts of Q, this path must be of length 1 or 2. Further, as $\mathcal{F}$ violates the coloring disjunct of Q, for every vertex x of $\mathcal{G}$, there must be one $\chi \in \{1,2,3\}$ such that the paths for x and χ has length 1. We define a coloring of $\mathcal{G}$ by choosing for each vertex x a color χ for which the E-path has length 1, i.e., the fact $E(e_{x,\chi}, f_{x,\chi})$ holds. (As pointed out in the beginning of the proof, for each x, there could be multiple such χ, but we can take any of them.) This indeed defines a 3-coloring, as any violation of the 3-coloring witnessed by two adjacent vertices of color χ would imply a match of the χ-th adjacency disjunct of Q in $\mathcal{F}$. This concludes the backward direction of the correctness proof of the reduction, and concludes the proof. □

We then modify the proof to show Proposition 6:

Proposition 6. *There is a base-covered CQ Q such that* $\mathsf{QAlin}(\mathcal{F}, \emptyset, Q)$ *is* CoNP*-hard in data complexity.*

Proof. Again we show the result for a UCQ Q, and extend it to a CQ in Appendix C.4. We define σ as in the proof of Proposition 5 but with an order relation $<$ instead of the two relations E and E^+ We define Q as in the proof of Proposition 5 but without the E-path length restriction disjunct, and replacing in the other disjuncts the length-1 paths $E(e, f)$ and $E(e', f')$ by $e < f$ and $e' < f'$ and the paths $E(e_\chi, w_\chi) \wedge E(w_\chi, f_\chi)$ by $f_\chi < e_\chi$: the resulting UCQ is clearly base-covered. Note that, unlike in the proof of Theorem 9, we need not worry about equalities (so we need not add order restriction disjuncts), as all the relevant elements are already created as distinct elements in $\mathcal{F}_0$. We define $\mathcal{F}_0$ in the same fashion as in the proof of Proposition 5 but without the E^+-facts.

We show the same equivalence as in that proof, but for QAlin. We do it by replacing E-paths of length 1 from an e to an f by $e < f$, and E-paths of length 2 by $f < e$. In the forward direction, we build a counterexample set of facts from a coloring as before, and extend $<$ to be an arbitrary total order (this is possible because all $<$-facts that we create are on disjoint pairs). In the backward direction, we use totality of $<$ to argue that a counterexample set of facts must choose some order between the $e_{x,\chi}$ and the $f_{x,\chi}$, and so must decide which colors are assigned to each vertex, in a way that yields a coloring (because the adjacency and coloring disjuncts are violated). □

6. Undecidability results

We now show how slight changes to the constraint languages and query languages used for the results in Sections 3 and 4 lead to undecidability of query answering.

The undecidability proofs in this section are by reduction from an *infinite tiling problem*, 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 we have $f((0,i)) = c_i$ 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 there are fixed $\mathbb{C}$, $\mathbb{V}$, $\mathbb{H}$ for which this problem is undecidable (Börger, Grädel, & Gurevich, 1997).

6.1 Undecidability results for QAtr and QAtc

We have shown in Section 3 that query answering is decidable with transitive relations (even with transitive closure), BaseFGTGDs, and UCQs (Theorem 3). Removing the base-frontier-guarded requirement makes QAtc undecidable, even when constraints are inclusion dependencies:

Theorem 10. *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* $\mathsf{QAtc}(\mathcal{F}_0, \Sigma, Q)$.

As mentioned earlier, 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 8, and relies on Q to test for forbidden adjacent tile patterns. Before we give the details of this proof, we prove an analogous (but easier) result for QAtr, which uses non-base-guarded disjunctive inclusion dependencies.

Theorem 11. *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. Fix $\mathbb{C}$, $\mathbb{V}$, $\mathbb{H}$ such that the infinite tiling problem is undecidable. We will give a reduction from this infinite tiling problem to $\mathsf{QAtr}(\mathcal{F}_0, \Sigma, Q)$. We prove the result with a UCQ instead of a CQ, and explain how we can use a CQ instead in Appendix C.5.

Definition of the reduction. The base relations of the signature are a binary relation S' (for "successor"), one binary relation K_i for each color C_i, and one unary relation K'_i for each color C_i. We also use one distinguished transitive relation, S^+. The idea is that we will create an infinite chain of S' included in S^+, giving us an infinite chain and its transitive closure. We can use it to define a grid structure to encode the tiling problem with grid positions represented by a pair of two elements of this chain.

Let Σ consist of the following DIDs (omitting universal quantifiers for brevity):

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

The $K_i(x,y)$ describe the assignment of colors to grid positions represented as pairs on the infinite chain as we explained. The point of $K'_i(x)$ is that it stands for $K_i(x,x)$: we need a different predicate because variable reuse is not allowed in inclusion dependencies. Note that some of these constraints are not base-guarded.

Let the UCQ Q be a disjunction of the following sentences (omitting existential quantifiers for brevity): for each forbidden horizontal pair $(C_i, C_j) \in \mathbb{H}$, with $1 \leq i, j \leq k$, the disjuncts

$$K_i(x,y) \wedge S'(y,y') \wedge K_j(x,y') \quad K'_i(y) \wedge S'(y,y') \wedge K_j(y,y') \quad K_i(y',y) \wedge S'(y,y') \wedge K'_j(y')$$

and for each forbidden vertical pair $(C_i, C_j) \in \mathbb{V}$, the analogous disjuncts

$$K_i(x,y) \wedge S'(x,x') \wedge K_j(x',y) \quad K'_i(x) \wedge S'(x,x') \wedge K_j(x',x) \quad K_i(x,x') \wedge S'(x,x') \wedge K'_j(x').$$

Given an initial instance $c_0, \ldots, c_n$ of the tiling problem, let the initial set of facts $\mathcal{F}_0$ consist of the fact $K'_j(a_0)$ such that C_j is the color of c_0, and for $0 \leq i < n$, the fact $S'(a_i, a_{i+1})$ and the fact $K_j(a_0, a_i)$ such that C_j is the color of initial element c_i.

Correctness proof for the reduction. We claim that the tiling problem has a solution iff there is a 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 extend the initial chain of S'-facts in $\mathcal{F}_0$ to an infinite chain $S'(a_0, a_1), \ldots, S'(a_m, a_{m+1}), \ldots$, 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 disjunct of Q in $\mathcal{F}$. This shows that the reduction is correct, and concludes the proof. □

We now prove the first statement, drawing inspiration from the previous proof, but using the transitive closure to emulate disjunction as we did in Theorem 8.

Proof of Theorem 10. 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^+. The extension to a CQ is explained in Appendix C.5.

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 cells on the diagonal of the grid, like K'_i in the previous proof), 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 (omitting universal quantifiers for brevity):

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

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):

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

Draft preview -- not a final published version

Generated Tue Nov 15 09:26:43 CET 2016

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

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

$$S'(x,w) \wedge G(x,y,z) \wedge T(z,z') \wedge C(z,z_1) \wedge C(z_1,z_2) \wedge \cdots \wedge C(z_{k-1},z_k), C(z_k,z_{k+1})$$

  and one disjunct defined similarly but with $G(x,y,z)$ replaced by $G'(x,z)$. Intuitively, these disjuncts impose that C-paths annotating T-facts must code colors between 1 and k (i.e., they cannot have length $k+1$ or greater), 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 11.

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

$$G(x,y,z) \wedge G(x,y',z') \wedge Q_i(z) \wedge Q_j(z') \wedge S'(y,y')$$
$$G'(y,z) \wedge G(y,y',z') \wedge Q_i(z) \wedge Q_j(z') \wedge S'(y,y')$$
$$G(y',y,z) \wedge G'(y',z') \wedge Q_i(z) \wedge Q_j(z') \wedge S'(y,y')$$

- *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 $G'(x,z) \wedge G(x,x',z')$ and $G(x,x',z) \wedge G'(x',z')$.

Given an initial instance of the tiling problem $c_0, \ldots, c_n$, the initial set of facts $\mathcal{F}_0$ consists of the following: (i) $S'(a_i, a_{i+1})$ for $0 \leq i < n$; (ii) $G(a_0, a_i, b_{0,i})$ for $0 < i \leq n$; (iii) $G'(a_0, b_{0,0})$ (iv) for all $0 \leq i \leq n$, letting l be such that c_i is the l-th color C_l, we create the *length-l gadget on* $b_{0,i}$: we create a path $C(b_{0,i}, d^1_{0,i}), C(d^1_{0,i}, d^2_{0,i}), \ldots C(d^{l-1}_{0,i}, d^l_{0,i})$, and the fact $T(b_{0,i}, d^l_{0,i})$, where the elements $b_{0,i}$ and $d^j_{0,i}$ are all fresh;

Correctness proof for the reduction. We claim that the tiling problem has a solution iff there is a 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 $S'(a_0, a_1), \ldots, S'(a_m, a_{m+1}), \ldots$ to complete the initial chain of S'-facts in $\mathcal{F}_0$, we create the implied S-facts, and make S^+ the transitive closure of S. 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 length restriction 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 length restriction 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 $\mathcal{F}_0$ whenever 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. □

Related work. The results that we have just shown in Theorem 10 and 11 complement the undecidability results of Gottlob et al. (2013). Their Theorem 2 shows that QAtr is undecidable for guarded TGDs, two transitive relations and atomic CQs, even with an empty set of initial facts. Their Corollary 1 shows that QAtr is undecidable with guarded disjunctive TGDs (TGDs with disjunction in the head, and with an atom in the body that guards all of the variables in the body) and UCQs, even when restricted to arity-two signatures with a single transitive relation that occurs only in guards, and an empty set of initial facts.

Our results contrast with the decidability results of Baget et al. (2015), which apply to QAtr with linear rules (under a safety condition, which they conjecture is not necessary for decidability): our Theorem 10 shows that QAtc with linear rules (without imposing their condition) is undecidable.

6.2 Undecidability results for QAlin

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

Theorem 12. *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* QAlin$(\mathcal{F}_0, \Sigma, Q)$.

Proof. We show the claim for a UCQ rather than a CQ, and explain in Appendix C.5 how the proof extends to a CQ. As in the proof of Theorem 10, 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 predicate could be rewritten away and is just used to make the inclusion dependencies shorter to write). We also introduce the following abbreviations: 1. we let $K'_1(x)$ stand for $\exists y\, x < y \wedge K_1(y)$; 2. we let $K'_k(x)$ stand for $\exists y\, x > y \wedge K_{k-1}(y)$; 3. for all $1 < i < k$, we let $K'_i(x)$ stand for $\exists y y'\, K_{i-1}(y) \wedge y < x \wedge x < y' \wedge K_i(y')$. Intuitively, the K'_i describe the color of elements, which is encoded in their order relation to elements labeled with the K_i.

We put the following inclusion dependencies in Σ:

$$\forall x\, S(x) \to \exists y\, R(x, y) \qquad \forall x\, S(x) \to \exists y\, D(x, y)$$
$$\forall xy\, R(x, y) \to S(y) \qquad \forall xy\, D(x, y) \to S(y)$$

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

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

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 other disjuncts enforce that there are no bad horizontal or vertical patterns.

Given an instance $c_0, \ldots, c_n$ of the tiling problem, we construct an initial set of facts $\mathcal{F}_0$ consisting of: (i) $S(a_0), \ldots, S(a_n)$ for fresh elements $a_0, \ldots, a_n$; (ii) $R(a_{i-1}, a_i)$ for $1 \leq i \leq n$ on these elements; (iii) $K_i(b_i)$ for $1 \leq i \leq k$ for fresh elements $b_1, \ldots, b_k$; (iv) for each i such that c_i is the color C_1, set $a_i < b_1$ (on the previously defined elements) ; (v) for each i such that c_i is the color C_k, set $a_i > b_{k-1}$ (on the previously defined elements); (vi) for each $1 < j < k$ and i such that c_i is C_j, set $b_{j-1} < a_i$ and $a_i < b_j$ (on the previously defined elements).

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 $\mathcal{F}_0$ 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 $a_0, \ldots, a_n$ the colors that they had in $\mathcal{F}_0$, 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. □

Theorem 12 implies that the base-covered requirement is also necessary for constraints:

Corollary 6. *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 on σ, such that, letting $\top$ be the tautological query, the following problem is undecidable: given a finite set of facts $\mathcal{F}_0$, decide* $\mathsf{QAlin}(\mathcal{F}_0, \Sigma', \top)$.

Proof. To prove Corollary 6 from Theorem 12, we take constraints Σ' that are equivalent to $\Sigma \wedge \neg Q$, where Σ and Q are as in the previous theorem (in particular, Q is a CQ). 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, if $Q = \exists \vec{x}\, \varphi(\vec{x})$ then 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, $\Sigma \wedge \neg Q$ can be written as a set of BaseFGTGD constraints as claimed. □

Related work. The result in Theorem 12 is related to prior work by Rosati (2007) and Gutiérrez-Basulto et al. (2015), which deals with query answering for UCQs and CQs with inequalities. These results are related because we can transform a query Q using an inequality $x \neq y$ into a new UCQ query $Q' \vee Q''$, where Q' and Q'' is the result of replacing $x \neq y$ in Q with $x < y$ and $y < x$, respectively. Further, we can also express constraints of the form $\forall xy(S_1(x,y) \wedge S_2(x,y) \to \bot)$, which are part of the description logics considered in those earlier papers, as inclusion dependencies $\forall xy(S_1(x,y) \wedge S_2(x,y) \to x < x)$. Hence, we could use these prior results to show the undecidability of QAlin for inclusion dependencies and a UCQ over $\sigma = \sigma_{\mathcal{B}} \sqcup \sigma_{\mathcal{D}}$ when $\sigma_{\mathcal{D}}$ is a single strict linear order. This is weaker than the result stated in Theorem 12, which uses a CQ, and in Corollary 6, which uses a tautological query.

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 the transitive closure relation is not needed as a guard. 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. We build upon some known polynomial data complexity upper bounds for classes of guarded constraints without distinguished relations, and show how to extend them to the setting of distinguished relations that are required to be transitive or a transitive closure. However, in the case of distinguished relations required to be linear orders, we show that PTIME data complexity does not always carry over.

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). One exception is due to Kieroński and Tendera (2007), which establishes decidability for the guarded fragment with transitivity, under the two-variable restriction and assuming that transitive relations appear only in guards. Another exception is due to Bárány and Bojańczyk (2012) in the context of guarded logics, but it is not clear if the techniques there can be extended to our constraint languages.

References

Abiteboul, S., & Duschka, O. M. (1998). Complexity of Answering Queries Using Materialized Views. In *PODS*.

Abiteboul, S., Hull, R., & Vianu, V. (1995). *Foundations of Databases*. Addison-Wesley.

Afrati, F., Li, C., & Pavlaki, V. (2008). Data Exchange in the Presence of Arithmetic Comparisons. In *EDBT*.

Amarilli, A., Benedikt, M., Bourhis, P., & Vanden Boom, M. (2016). Query Answering with Transitive and Linear-Ordered Data. In *IJCAI*.

Andréka, H., Németi, I., & van Benthem, J. (1998). Modal Languages and Bounded Fragments of Predicate Logic. *J. Philosophical Logic*, *27*(3).

Baget, J.-F., Leclère, M., Mugnier, M.-L., & Salvat, E. (2009). Extending Decidable Cases for Rules with Existential Variables. In *IJCAI*.

Baget, J.-F., Mugnier, M.-L., Rudolph, S., & Thomazo, M. (2011). Walking the Complexity Lines for Generalized Guarded Existential Rules. In *IJCAI*.

Baget, J., Bienvenu, M., Mugnier, M., & Rocher, S. (2015). Combining Existential Rules and Transitivity: Next Steps. In *IJCAI*.

Bárány, V., & Bojańczyk, M. (2012). Finite Satisfiability for Guarded Fixpoint Logic. *IPL*, *112*(10), 371–375.

Bárány, V., ten Cate, B., & Otto, M. (2012). Queries with Guarded Negation. *PVLDB*, *5*(11), 1328–1339.

Bárány, V., ten Cate, B., & Segoufin, L. (2011). Guarded Negation. In *ICALP*.

Benedikt, M., Bourhis, P., & Vanden Boom, M. (2016). A Step Up in Expressiveness of Decidable Fixpoint Logics. In *LICS*.

Börger, E., Grädel, E., & Gurevich, Y. (1997). *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer.

Bourhis, P., Krötzsch, M., & Rudolph, S. (2015). Reasonable Highly Expressive Query Languages. In *IJCAI*.

Bourhis, P., Morak, M., & Pieris, A. (2013). The Impact of Disjunction on Query Answering Under Guarded-Based Existential Rules. In *IJCAI*.

Calvanese, D., De Giacomo, G., & Vardi, M. Y. (2005). Decidable Containment of Recursive Queries. *Theor. Comput. Sci.*, *336*(1), 33–56.

Calvanese, D., Eiter, T., & Ortiz, M. (2009). Regular Path Queries in Expressive Description Logics with Nominals.. In *IJCAI*.

Calvanese, D., Lembo, D., Lenzerini, M., & Rosati, R. (2006). Data Complexity of Query Answering in Description Logics. In *KR*.

Ganzinger, H., Meyer, C., & Veanes, M. (1999). The Two-Variable Guarded Fragment with Transitive Relations. In *LICS*.

Gottlob, G., & Papadimitriou, C. (2003). On the Complexity of Single-Rule Datalog Queries. *Inf. Comp.*, *183*.

Gottlob, G., Pieris, A., & Tendera, L. (2013). Querying the Guarded Fragment with Transitivity. In *ICALP*.

Grädel, E., & Walukiewicz, I. (1999). Guarded Fixed Point Logic. In *LICS*.

Gutiérrez-Basulto, V., Ibáñez-García, Y. A., Kontchakov, R., & Kostylev, E. V. (2015). Queries with Negation and Inequalities over Lightweight Ontologies. *J. Web Sem.*, *35*, 184–202.

Horrocks, I., & Sattler, U. (1999). A Description Logic with Transitive and Inverse Roles and Role Hierarchies. *Journal of Logic and Computation*, *9*(3), 385–410.

Kieronski, E. (2011). Decidability Issues for Two-Variable Logics with Several Linear Orders. In *CSL*.

Kieroński, E., & Tendera, L. (2007). On finite satisfiability of the guarded fragment with equivalence or transitive guards. In *LPAR*.

Krötzsch, M., & Rudolph, S. (2007). Conjunctive Queries for $\mathcal{EL}$ with Role Composition. In *DL*.

Löding, C. (2011). Automata on Infinite Trees. `http://www.automata.rwth-aachen.de/~loeding/inf-tree-automata.pdf`.

Ortiz, M., Rudolph, S., & Šimkus, M. (2011). Query Answering in the Horn Fragments of the Description Logics SHOIQ and SROIQ. In *IJCAI*.

Ortiz, M., Rudolph, S., & Šimkus, M. (2010). Query Answering is Undecidable in DLs with Regular Expressions, Inverses, Nominals and Counting. Tech. rep., Technische Universität Wien.

Ortiz, M., & Simkus, M. (2012). Reasoning and Query Answering in Description Logics. In *Reasoning Web*, pp. 1–53.

Ortiz de la Fuente, M. (2010). *Query Answering in Expressive Description Logics*. Ph.D. thesis, Technischen Universität Wien.

Otto, M. (2001). Two Variable First-Order Logic over Ordered Domains. *J. Symb. Log.*, *66*(2), 685–702.

Rosati, R. (2007). The Limits of Querying Ontologies. In *ICDT*.

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

Szwast, W., & Tendera, L. (2004). The Guarded Fragment with Transitive Guards. *Annals of Pure and Applied Logic*, *128*(13), 227 – 276.

Thomas, W. (1997). Languages, Automata, and Logic. In Rozenberg, G., & Salomaa, A. (Eds.), *Handbook of Formal Languages*. Springer-Verlag.

Vardi, M. Y. (1998). Reasoning about the Past with Two-Way Automata. In *ICALP*.

Appendix

A. Tree decompositions

Recall that a *tree decomposition* of a set of facts $\mathcal{F}$ 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.

In this appendix, we present two special types of tree decompositions for BaseGNF that were utilized in the body of the paper. 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).

A.1 Transitive-closure friendly tree decompositions for BaseGNF

Recall the statement of Proposition 2:

> **Proposition 2.** 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 $R^+ \in \sigma_{\mathcal{D}}$ is interpreted as the transitive closure of the corresponding $R \in \sigma$, then there is some $\mathcal{F}$ like this that has a $\mathcal{F}_0$-rooted $(k-1)$-width tree decomposition. Note that in such an $\mathcal{F}$, the only $\sigma_{\mathcal{D}}$-facts appearing in $\mathcal{F}$ are from $\mathcal{F}_0$, and such distinguished facts must actually be part of the transitive closure of the corresponding base relation in $\mathcal{F}$. Other $\sigma_{\mathcal{D}}$-facts may be implied by $\sigma_{\mathcal{B}}$-facts, and both the explicit and implicit $\sigma_{\mathcal{D}}$-facts must be considered when reasoning about φ. However, we emphasize that besides the $\sigma_{\mathcal{D}}$-facts in $\mathcal{F}_0$, there are no $\sigma_{\mathcal{D}}$-facts appearing in the tree decomposition — the explicit inclusion of such $\sigma_{\mathcal{D}}$-facts could make it impossible to find a k-tree-like witness.

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).

Bisimulation game. Let $\mathcal{F}$ and $\mathcal{G}$ be sets of facts extending $\mathcal{F}_0$. The *GN^k bisimulation game* between $\mathcal{F}$ 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 $\mathsf{BaseGNF}$ formulas.

Proposition 7. *Let $\mathcal{F}$ and $\mathcal{G}$ be sets of facts extending $\mathcal{F}_0$. Let $\varphi(\vec{x})$ be a formula in* $\mathsf{BaseGNF}$*, 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 $\vec{a} = a_1 a_2$ and $\vec{b} = b_1 b_2$. 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 $\mathsf{BaseGNF}$ (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 $\mathsf{BaseGNF}$ 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 $\mathsf{BaseGNF}$, 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 $\mathcal{F}$ that extends $\mathcal{F}_0$ with additional $\sigma_{\mathcal{B}}$-facts. 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 $\{[\pi, a] : \pi \in \Pi \text{ and } a \in \mathsf{elems}(\mathcal{F})\}$ with $S([\pi_1, a_1], \dots, [\pi_j, a_j]) \in \mathcal{F}^k$ iff $S(a_1, \dots, a_j) \in \mathcal{F}$ and there is some $\pi \in \Pi$ such that $[\pi, a_i] = [\pi_i, a_i]$ for $i \in \{1, \dots, j\}$. We can identify $[\epsilon, a]$ with the element $a \in \mathsf{elems}(\mathcal{F}_0)$, so $\mathcal{F}^k$ extends $\mathcal{F}_0$. Hence, 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 8. *Let $\mathcal{F}$ be a set of facts extending $\mathcal{F}_0$ with additional $\sigma_{\mathcal{B}}$-facts, and let $\mathcal{F}^k$ be the GN^k-unravelling of $\mathcal{F}$. Then 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 $[\pi_1, a_1], \ldots, [\pi_n, a_n]$ from $\mathsf{Dom}(f')$ must hold for the tuple of elements $a_1, \ldots, a_n$ 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 $a_1, \ldots, a_n$ in the range of f'' for which some relation S holds in $\mathcal{F}$; since $(f'')^{-1}(a_1), \ldots, (f'')^{-1}(a_n)$ 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 $(f'')^{-1}(a_1), \ldots, (f'')^{-1}(a_n)$ 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 $\vec{a} = a_1 \ldots a_n$ 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 $\vec{a} = \{a_1, \ldots, a_n\}$ of distinct elements from $\mathsf{Dom}(f')$, $f'(\vec{a})$ must yield a set of distinct elements $\{f'(a_1), \ldots, f'(a_n)\}$, 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 8. □

We can now conclude the proof of Proposition 2. Assume that $\mathcal{F}$ is a set of facts extending $\mathcal{F}_0$ with additional $\sigma_{\mathcal{B}}$-facts such that $\mathcal{F}$ 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 8 and 7 imply that $\mathcal{F}^k$ also satisfies φ when properly interpreting R^+. Moreover, for each $\sigma_{\mathcal{D}}$-fact $R^+(c,d) \in \mathcal{F}_0$, there is some $n \in \mathbb{N}$ such that $n > 0$ and there is an R-path of length n between c and d in $\mathcal{F}$. We can write a formula $\psi_n(c,d)$ in $\mathsf{BaseGNF}$ (without any $\sigma_{\mathcal{D}}$-atoms) that is satisfied exactly when there is an R-path of length n. Since ψ_n can be expressed in $\mathsf{BaseGNF}$ with at most 3 free variables in any subformula and $\mathcal{F} \models \psi_n(c,d)$, Propositions 8 and 7 imply that $\mathcal{F}^k$ also satisfies ψ_n, and hence $\mathcal{F}^k \models R^+(c,d)$. Thus, we can conclude that the unravelling $\mathcal{F}^k$ is a transitive-closure friendly k-tree-like witness for φ.

A.2 Base-guarded-interface tree decompositions for $\mathsf{BaseGNF}$

Recall that 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.

We prove the following result:

Proposition 9. *Every sentence φ in* $\mathsf{BaseGNF}$ *has base-guarded-interface k-tree-like witnesses for some $k \leq |\varphi|$.*

The result and proof of Proposition 9 is very similar to Proposition 2. However, unlike Proposition 2, 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 is important in some arguments (e.g., Proposition 4 and Theorem 6).

We first consider a variant of the GN^k bisimulation game defined earlier in Section A.1. 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 10. *Let $\mathcal{F}$ and $\mathcal{G}$ be sets of facts extending $\mathcal{F}_0$. Let $\varphi \in \mathsf{BaseGNF}^k$ 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 $\mathsf{BaseGNF}$, 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 $\mathsf{BaseGNF}^k$. 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 $\mathsf{BaseGNF}$ 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 11. *Let $\mathcal{F} \supseteq \mathcal{F}_0$ and let $\mathcal{F}^k_{\mathcal{B}}$ be the base-guarded-interface GN^k-unravelling of $\mathcal{F}$. Then 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 8. 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 8. $\square$

We can conclude the proof of Proposition 9 as follows. Assume that $\mathcal{F}$ is a set of facts that satisfies φ. By Proposition 1, 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 11 and 10 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.

B. Data complexity upper bounds for transitivity

B.1 Proof of Theorem 6: PTIME data complexity bound for QAtr

We now turn to the case where our constraints are restricted to $\mathsf{BaseCovFGTGDs}$ and deal with QAtr, not QAtc. Recall that Theorem 6 states a PTIME data complexity bound for this case:

> **Theorem 6.** For any fixed $\mathsf{BaseCovFGTGD}$ 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 4:

Proposition 12. *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 $\Sigma' \in \mathsf{GNF}$ in PTIME such that $\mathsf{QAtr}(\mathcal{F}_0, \Sigma, Q)$ iff $\mathsf{QA}(\mathcal{F}'_0, \Sigma', Q)$. Furthermore, if Σ is in $\mathsf{BaseCovFGTGD}$ 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 4, the k-guardedly transitive axioms (see Section 4) can be written in normal form $\mathsf{BaseGNF}$ 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 9 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 6, transitively closing like this can only add R^+-facts about pairs of elements that are not base-guarded. Moreover, Lemma 8 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

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

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. $\square$

Theorem 6 easily follows from this.

Proof of Theorem 6. 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 12 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). $\square$

C. From UCQ to CQ

In this section, we first prove general auxiliary lemmas about reducing from QA problems with UCQs to QA problems with CQs. We first give such a lemma for regular QA, inspired by Gottlob and Papadimitriou (2003) and known in the folklore, and then modify it to the various QA notions that we study for distinguished relations. We then revisit the results of Sections 5 and 6 and explain how the proofs in the main text using UCQs can be extended to use only CQs.

The general idea to replace UCQs by CQs is to extend the arity of the relations to include a flag that indicates whether a fact is a "real fact" or a "pseudo-fact": the flag is propagated by the TGDs. We then add pseudo-facts to the instances to ensure that each UCQ disjunct has a match that involves pseudo-facts. This ensures that we can replace the UCQ by a *conjunction* of the original disjuncts, with an OR on the flag of the match of each disjunct: this OR can be performed using a suitable relation which we add to the instances.

This idea must then be tweaked to work for QA with distinguished relations, as we cannot increase the arity of these relations. Fortunately, we will show that it suffices to add the flag to a subset of the relations (which we will call the *flagged relations*), without changing the arity of the others (which we call the *special relations*), under some conditions on the TGDs and queries.

C.1 UCQ to CQ for general QA

We first show a translation result for UCQ to CQ:

Lemma 9. *For any signature σ, TGDs Σ and UCQ Q, one can compute in* PTIME *a signature σ', TGDs Σ' and CQ Q' such that the* QA *problem for Σ and Q reduces in* PTIME *to the* QA *problem for Σ' and Q' (for combined complexity and for data complexity): namely, given a set of facts $\mathcal{F}_0$ on σ, we can compute in* PTIME *in $\mathcal{F}_0$, Σ and Q a set of facts $\mathcal{F}'_0$ on σ' such that $\mathsf{QA}(\mathcal{F}_0, \Sigma, Q)$ holds iff $\mathsf{QA}(\mathcal{F}'_0, \Sigma', Q')$ holds.*

Proof. Let σ_{Or} be a constant signature consisting of a ternary relation Or and a unary relation True. We let σ' be the signature obtained from σ by creating one relation R' in σ' for every R in σ with $\mathsf{arity}(R') := \mathsf{arity}(R) + 1$, and further adding the relations of σ_{Or}.

We define Σ' from Σ by considering each TGD $\tau : \forall \vec{x}\ \varphi(\vec{x}) \to \exists \vec{y}\ \psi(\vec{x}, \vec{y})$, and, letting z be a fresh variable, replacing τ by the TGD $\tau' : \forall \vec{x}\, z\ \varphi'(\vec{x}, z) \to \exists \vec{y}\ \psi'(\vec{x}, z, \vec{y})$, where φ' and ψ' are obtained from φ and ψ respectively by replacing each σ-atom $R(\vec{w})$ by the σ'-atom $R'(\vec{w}, z)$. Since TGD bodies are not empty, this ensures that the new variable z indeed occurs in the new body φ'.

We now describe the construction of Q' from Q. Suppose the UCQ Q is $\bigvee_{1 \leq i \leq m} \exists \vec{x_i}\ Q_i(\vec{x_i})$, where each Q_i is a conjunction of atoms over σ. Let $z_1, \ldots, z_m$ be fresh variables. For each $1 \leq i \leq m$, we define a conjunction of atoms $Q'_i(\vec{x}_i, z_i)$ on σ' which is obtained from $Q_i(\vec{x_i})$ by replacing each σ-atom $R(\vec{w})$ by the σ'-atom $R'(\vec{w}, z_i)$. We now define Q' as:

$$\exists z_1 \ldots z_m\, z'_1 \ldots z'_{m-1}\, \vec{x}_1 \ldots \vec{x}_m$$
$$\mathrm{Or}(z_1, z_2, z'_1) \wedge \mathrm{Or}(z'_1, z_3, z'_2) \wedge \cdots \wedge \mathrm{Or}(z'_{m-2}, z_m, z'_{m-1}) \wedge \mathrm{True}(z'_{m-1}) \wedge \bigwedge_{1 \leq i \leq m} Q'_i(\vec{x}_i, z_i)$$

It is clear that the computation of σ', Σ', and Q' from σ, Σ, and Q is in PTIME.

We now describe the PTIME transformation on input sets of facts. Let $\mathcal{F}_0$ be a set of facts. Letting $\mathfrak{t}$ and $\mathfrak{f}$ be two fresh elements, let $\mathcal{F}_{\mathrm{Or}}$ be the set of facts that contains the fact $\mathrm{True}(\mathfrak{t})$ and the facts $\mathrm{Or}(b, b', b'')$ for all $\{(b, b', b \vee b') \mid b, b' \in \{\mathfrak{f}, \mathfrak{t}\}\}$. Let $\mathcal{F}_{\mathfrak{f}}$ be the set of facts $\{R'(\mathfrak{f}, \ldots, \mathfrak{f}) \mid R' \in \sigma'\}$, and let $(\mathcal{F}_0)_{+\mathfrak{t}}$ be $\{R'(\vec{a}, \mathfrak{t}) \mid R(\vec{a}) \in \mathcal{F}_0\}$. We define $\mathcal{F}'_0 := \mathcal{F}_{\mathrm{Or}} \sqcup \mathcal{F}_{\mathfrak{f}} \sqcup (\mathcal{F}_0)_{+\mathfrak{t}}$ which is clearly computable in PTIME.

We now show correctness of the reduction. In the forward direction, consider a counterexample set of facts $\mathcal{F}$ on σ which is a superset of $\mathcal{F}_0$, satisfies Σ, and violates Q: up to renaming we can ensure that $\mathfrak{t}, \mathfrak{f} \notin \mathsf{elems}(\mathcal{F})$. Let us construct a counterexample $\mathcal{F}'$ for $\mathcal{F}'_0$, Σ', and Q', by setting $\mathcal{F}' := \mathcal{F}_{\mathrm{Or}} \sqcup \mathcal{F}_{+\mathfrak{t}} \sqcup \mathcal{F}_{\mathfrak{f}}$ where $\mathcal{F}_{\mathfrak{f}}$ is as above and where $\mathcal{F}_{+\mathfrak{t}} := \{R'(\vec{a}, \mathfrak{t}) \mid R(\vec{a}) \in \mathcal{F}\}$.

It is clear that $\mathcal{F}'$ is a superset of $\mathcal{F}'_0$ because $\mathcal{F}_{+\mathfrak{t}}$ is. To see why $\mathcal{F}'$ satisfies Σ', consider a match $M' \subseteq \mathcal{F}'$ of the body of a TGD τ' of Σ' in $\mathcal{F}'$. As Σ' does not mention the relations of σ_{Or}, no fact of $\mathcal{F}_{\mathrm{Or}}$ can occur in M'. Now, the facts of $\mathcal{F}_{\mathfrak{f}}$ have $\mathfrak{f}$ as their last element, and those of $\mathcal{F}_{+\mathfrak{t}}$ have $\mathfrak{t}$ as their last element, so, as all atoms of the body of τ' have the same variable at their last element, either $M' \subseteq \mathcal{F}_{\mathfrak{f}}$, or $M' \subseteq \mathcal{F}_{+\mathfrak{t}}$. In the first case, we can find a match of the head of τ in $\mathcal{F}_{\mathfrak{f}}$ (where all variables are mapped to $\mathfrak{f}$), so we conclude that M' is not a violation. In the second case, considering the preimage M of M' in $\mathcal{F}$, it is clear that M is a match of the TGD τ of Σ, so, as $\mathcal{F}$ satisfies Σ, we can extend M to a match of the head of τ in $\mathcal{F}$, yielding a match of the head of τ' in $\mathcal{F}'$, so that again M' cannot be a violation. Hence, $\mathcal{F}'$ satisfies Σ.

Last, to see why $\mathcal{F}'$ violates Q', assume by contradiction that there is a homomorphism from Q' to $\mathcal{F}'$. Notice that, in our construction of $\mathcal{F}'$, the only element $a \in \mathsf{elems}(\mathcal{F}')$ such that $\mathrm{True}(a)$ holds is $a = \mathfrak{t}$. Hence, necessarily, h must map z'_{m-1} to $\mathfrak{t}$. However, as the only Or-facts in $\mathcal{F}'$ are those of $\mathcal{F}_{\mathrm{Or}}$, it is clear that h must map some z_{i_0} to $\mathfrak{t}$. Thus, a suitable restriction of h is a match of $\exists \vec{x}_{i_0}\, Q'_{i_0}(\vec{x}_{i_0}, \mathfrak{t})$ in $\mathcal{F}'$. Now, as all facts in the image of h have $\mathfrak{t}$ as their last element, the image of h must be contained in $\mathcal{F}_{+\mathfrak{t}}$, so we deduce that $\exists \vec{x}_{i_0}\, Q_{i_0}(\vec{x}_{i_0})$ has a match in $\mathcal{F}$, contradicting the fact that $\mathcal{F}$ violates Q. This concludes the forward direction of the correctness proof.

In the backward direction, consider a counterexample set of facts $\mathcal{F}' \supseteq \mathcal{F}'_0$ that satisfies Σ' and violates Q'. Construct the set of facts $\mathcal{F} := \{R(\vec{a}) \mid R'(\vec{a}, \mathfrak{t}) \in \mathcal{F}'\}$. As $(\mathcal{F}_0)_{+\mathfrak{t}} \subseteq \mathcal{F}'_0$, clearly $\mathcal{F}_0 \subseteq \mathcal{F}$. To see why $\mathcal{F}$ satisfies Σ, consider a match $M \subseteq \mathcal{F}$ of the body of some TGD τ of Σ in $\mathcal{F}$, and consider its preimage M' in $\mathcal{F}'$, where all facts have $\mathfrak{t}$ as their last element: M' is a match of the body of $\tau' \in \Sigma'$. Hence, as $\mathcal{F}'$ satisfies Σ', M' extends to a match of the head of τ', and the last elements of all its facts is $\mathfrak{t}$, so we can find a suitable extension in $\mathcal{F}$ as well. Hence, M is not a violation of τ in $\mathcal{F}$, so $\mathcal{F}$ satisfies Σ.

Last, to see why $\mathcal{F}$ violates the UCQ Q, assume by contradiction that the Q has a match in $\mathcal{F}$. This means that there is $1 \leq i_0 \leq m$ such that the disjunct $\exists \vec{x}_{i_0}\, Q_{i_0}(\vec{x}_{i_0})$ has a match M_{i_0} in $\mathcal{F}$. By construction of $\mathcal{F}$, this means that $\exists \vec{x}_{i_0}\, Q'_{i_0}(\vec{x}_{i_0}, \mathfrak{t})$ has a match M'_{i_0} in $\mathcal{F}'$. Now, observe that, for all $1 \leq i \leq m$, there is a match M''_i of $\exists \vec{x}_i\, Q_i(\vec{x}_i, \mathfrak{f})$ in $\mathcal{F}_{\mathfrak{f}}$ obtained by mapping all variables to $\mathfrak{f}$. As $\mathcal{F}_{\mathfrak{f}} \subseteq \mathcal{F}'_0 \subseteq \mathcal{F}'$, the same is true of $\mathcal{F}'$. Now, as $\mathcal{F}_{\mathrm{Or}} \subseteq \mathcal{F}'_0 \subseteq \mathcal{F}'$, we can extend M'_{i_0} and the M''_i for $i \neq i_0$ to a match of Q' in $\mathcal{F}'$ by matching z_{i_0} to $\mathfrak{t}$, every z_i to $\mathfrak{f}$ for $i \neq i_0$, every z'_i for $i' < i_0 - 1$ to $\mathfrak{f}$, and the z'_i for $i' \geq i_0 - 1$ to $\mathfrak{t}$. This contradicts the fact that $\mathcal{F}'$ violates Q'. Hence, $\mathcal{F}$ violates Q and is a counterexample to QA. This concludes the correctness proof. □

C.2 UCQ to CQ for QAlin

We first adapt the general QA result in Lemma 9 to QAlin, which avoids increasing the arity of the distinguished relations that are interpreted as linear orders.

In order to avoid increasing the arity of the distinguished relations, we will ban distinguished relations in the dependencies, and require *base-domain-coveredness* of the query (which is a weakening of base-coveredness):

Definition 1. *A CQ Q is* base-domain-covered *if every variable x occurring in a distinguished atom in Q also occurs in a base atom in Q.*

We can now state the following variant of Lemma 9:

Lemma 10. *For any signature σ (partitioned in base relations $\sigma_{\mathcal{B}}$ and distinguished relations $\sigma_{\mathcal{D}}$), TGDs Σ on $\sigma_{\mathcal{B}}$, and base-domain-covered UCQ Q, one can compute in* PTIME *a signature σ' partitioned as $\sigma'_{\mathcal{B}} \sqcup \sigma_{\mathcal{D}}$, TGDs Σ' on $\sigma'_{\mathcal{B}}$, and a base-domain-covered CQ Q' such that the* QAlin *problem for Σ and Q reduces in* PTIME *to the same problem for Σ' and Q', in the sense of Lemma 9.*

Further, if Σ is BaseIDs *then Σ' also is; if Σ is empty then Σ' also is; if Q is base-covered then Q' also is.*

Proof. We first preprocess the input UCQ Q without loss of generality to remove any disjuncts where some distinguished relation $<_i$ is not a partial order (i.e., it has a cycle): the rewritten Q is equivalent to the original one for QAlin because the removed disjuncts can never be entailed because of the semantics of distinguished relations.

We now adapt the proof of Lemma 9. The definition of σ_{Or} is unchanged, and we define $\sigma' := \sigma''_{\mathcal{B}} \sqcup \sigma_{\mathrm{Or}} \sqcup \sigma_{\mathcal{D}}$ where $\sigma''_{\mathcal{B}}$ is defined by increasing the arity of the relations from $\sigma_{\mathcal{B}}$ as before. The definition of Σ' is unchanged. It easy to see that if Σ is empty then so is Σ', and if Σ consists of BaseIDs (i.e., there are no repetitions of variables in the body and in the head, and only one body fact) then this is still the case of Σ'.

The definition of Q' is unchanged except that we do not rewrite distinguished relations: as the query is base-domain-covered, each fresh variable z_i that we add occurs in Q'_i, and the base-domain-coveredness (resp. base-coveredness) of Q' is easy to see from that of Q.

We modify the definition of $\mathcal{F}_{\mathrm{Or}}$ to complete each distinguished relation to a total order on $\mathcal{F}_{\mathrm{Or}}$ (e.g., create $\mathfrak{f} <_i \mathfrak{t}$ for all distinguished relations $<_i$). We define $\mathcal{F}_{\mathfrak{f}}$ in a new fashion. First, letting m be the maximal number of variables of a disjunct of Q, for each distinguished relation $<_i$, we create fresh values $\mathfrak{f}^i_1, \ldots, \mathfrak{f}^i_m$, and create facts $\mathfrak{f}^i_1 <_i \cdots <_i \mathfrak{f}^i_n$ in $\mathcal{F}_f$. Second, we create all facts $R'(\vec{f}, \mathfrak{f})$ where $R' \in \sigma_{\mathcal{B}}$ and $\vec{f}$ is any subsequence of the $\mathfrak{f}^i_j$. We then define $(\mathcal{F}_0)_{+\mathfrak{t}}$ as the union of $\{R'(\vec{a}, \mathfrak{t}) \mid R(\vec{a}) \in \mathcal{F}_0 \wedge R \in \sigma_{\mathcal{B}}\}$ and of the $\sigma_{\mathcal{D}}$-facts of $\mathcal{F}_0$ kept as-is; then we define $\mathcal{F}'_0 := \mathcal{F}_{\mathrm{Or}} \sqcup (\mathcal{F}_0)_{+\mathfrak{t}} \sqcup \mathcal{F}_{\mathfrak{f}}$.

To adapt the forward direction of the correctness proof, let us consider a counterexample $\mathcal{F}$ to $\mathsf{QAlin}(\mathcal{F}_0, \Sigma, Q)$ with suitably interpreted distinguished relations. We define $\mathcal{F}'' := \mathcal{F}'_0 \sqcup \mathcal{F}_{+\mathfrak{t}} \sqcup \mathcal{F}_{\mathfrak{f}}$, with $\mathcal{F}_{\mathfrak{f}}$ as above and $\mathcal{F}_{+\mathfrak{t}}$ defined as above by adding $\mathfrak{t}$ to base facts and keeping distinguished facts as-is, and define $\mathcal{F}'$ from $\mathcal{F}''$ by completing each distinguished relation to be a total order: this is possible, because the order constraints on $\mathcal{F}_{\mathfrak{f}}$ and on $\mathcal{F}_{\mathrm{Or}}$ overlap only on $\mathfrak{f}$, and with $\mathcal{F}_{+\mathfrak{t}}$ they do not overlap at all.

We now explain how the correctness argument of the forward direction is adapted. Clearly $\mathcal{F}' \supseteq \mathcal{F}'_0$. To see why $\mathcal{F}'$ satisfies Σ', as Σ' does not involve the distinguished relations, we reason as in Lemma 9 to deduce that a match is either included in $\mathcal{F}_{\mathfrak{f}}$ or in $\mathcal{F}_{+\mathfrak{t}}$: the first case is similar as a head match can be found in $\mathcal{F}_{\mathfrak{f}}$ by definition, and the second case is unchanged. To see why $\mathcal{F}'$ violates Q', we show as in Lemma 9 that there is a match h of some Q'_{i_0} to $\mathcal{F}'$ that maps all *base* facts of Q'_{i_0} to facts with $\mathfrak{t}$ as their last element. Now, the only distinguished facts where each individual elements occurs in $\mathcal{F}'$ in base facts of this form are the ones from $\mathcal{F}_{+\mathfrak{t}}$, that we constructed from $\mathcal{F}$ which violated Q; hence, we can conclude because Q is base-domain-covered.

We now explain how the backward direction of the correctness proof is adapted. We construct $\mathcal{F}$ as the disjoint union of $\{R(\vec{a}) \mid R'(\vec{a}, \mathfrak{t}) \in \mathcal{F} \wedge R' \in \sigma'_{\mathcal{B}}\}$ and of the distinguished facts of $\mathcal{F}'$ kept as-is. It is clear that $\mathcal{F}'$ suitably interprets the distinguished relations, because its restriction to $\sigma_{\mathcal{D}}$ is the same as $\mathcal{F}$, which does. Again we have $\mathcal{F} \supseteq \mathcal{F}_0$. The fact that $\mathcal{F}$ satisfies Σ is as before, except that the arity of distinguished facts is not changed in M', and the witness head facts of M' in $\mathcal{F}'$ may include distinguished facts, in which case they are found as-is in $\mathcal{F}$.

To see that $\mathcal{F}$ violates Q, we reuse the argument of Lemma 9. The only new point that is needed is that the new $\mathcal{F}_{\mathfrak{f}}$ can still be used to find matches of any Q'_{i_0} with the last variable mapped to $\mathfrak{f}$, but this is easy to see from that construction. This concludes the proof. $\square$

C.3 UCQ to CQ for QA with special relations

We next adapt Lemma 9 so it can be applied to both QAtr and QAtc.

We partition the signature into two sets of relations: *flagged relations*, whose arity will be increased as in the proof of Lemma 9 above, and *special relations*, whose arity we will not increase. This partition is different from that of the rest of the paper, where we had base and distinguished relations. We prove a generalization of Lemma 9 to the setting with flagged relations and special relations, and where we also allow logical constraints on special relations that go beyond the TGDs allowed in Lemma 9: in particular this will allow us to impose transitivity and transitive closure requirements. The tradeoff is that we will need to impose a restriction on the TGDs that feature the flagged relations.

We first define the constraints that we will allow on the special relations:

Definition 2. *On a signature $\sigma := \sigma_{\mathcal{F}} \sqcup \sigma_{\mathcal{S}}$ partitioned into flagged and special relations, a* special constraint set Θ *is a set of logical constraints on $\sigma_{\mathcal{S}}$ involving any of the following:*

- *Disjunctive inclusion dependencies on $\sigma_{\mathcal{S}}$;*
- Transitivity assertions*, i.e., assertions that some binary relation in $\sigma_{\mathcal{S}}$ is transitive (i.e., a special kind of TGD);*
- Transitive closure assertions*, i.e., assertions that some binary relation in $\sigma_{\mathcal{S}}$ is the transitive closure of another binary relation in $\sigma_{\mathcal{S}}$.*

Hence, special constraint sets can be used to express the semantics of distinguished relations in the QAtr and QAtc problems, so distinguished relations will be special relations in this section. However, in the case of QAtc, the relations of which we are taking the transitive closure must themselves be special (indeed, we cannot increase their arities, as they must remain binary). This is why the partition into base and distinguished relations may be different than the partition into flagged and special relations.

In addition to the special constraint set Θ, our result will allow us to write TGDs Σ, and the negation of a CQ, like in Lemma 9. However, in exchange for the freedom of keeping special relations binary, we need to impose a condition on the TGDs and on the CQ, which we call *flagged-reachability*. Intuitively, the goal of this condition is to ensure that we can discriminate between matches of special relations in the query or in dependency bodies that use facts annotated by $\mathfrak{t}$, versus the matches whose facts are annotated by $\mathfrak{f}$. Indeed, this information cannot be seen on the special relations which do not carry the flag.

Definition 3. *Let G be the graph over the atoms of Q where two atoms are connected iff they share a variable. A CQ Q is* flagged-reachable *if any special atom $A(x,y)$ in Q has a path to some flagged atom $B(\vec{z})$ in G. A UCQ is flagged-reachable if all of its disjuncts are. A TGD is* flagged-reachable *if its body is.*

The flagged-reachable restriction suffices to ensure that matches of special relations in queries and rule bodies must correspond to $\mathfrak{f}$ or to non-$\mathfrak{f}$ elements, by looking at the flagged facts to which the special relations must be connected. We can thus show:

Lemma 11. *For any signature $\sigma := \sigma_{\mathcal{F}} \sqcup \sigma_{\mathcal{S}}$, flagged-reachable TGDs Σ, special constraint set Θ on $\sigma_{\mathcal{S}}$, and flagged-reachable CQ Q. one can compute in* PTIME *a signature $\sigma' := \sigma'_{\mathcal{F}} \sqcup \sigma_{\mathcal{S}}$, TGDs Σ' and CQ Q' such that the* QA *problem for $\Sigma \sqcup \Theta$ and Q reduces in* PTIME *to the* QA *problem for $\Sigma' \sqcup \Theta$ and Q' (for combined complexity and for data complexity): namely, given a set of facts $\mathcal{F}_0$ on σ, we can compute in* PTIME *in $\mathcal{F}_0$, Σ, Θ and Q a set of facts $\mathcal{F}'_0$ on σ' such that* $\mathsf{QA}(\mathcal{F}_0, \Sigma \sqcup \Theta, Q)$ *holds iff* $\mathsf{QA}(\mathcal{F}'_0, \Sigma' \sqcup \Theta, Q')$ *holds.*

Further, the following properties transfer from Σ to Σ': being IDs*; being* BaseIDs*; being empty. Further, if some relation in σ is not mentioned in Q then Q' does not mention it either.*

Proof. We amend the proof of Lemma 9. We first explain the change in the construction. The definition of σ_{Or} is unchanged, and we define $\sigma' := \sigma''_{\mathcal{F}} \sqcup \sigma_{\mathrm{Or}} \sqcup \sigma_{\mathcal{S}}$ where $\sigma''_{\mathcal{F}}$ is defined by increasing the arity of the relations from $\sigma_{\mathcal{F}}$ (like σ' from σ in Lemma 9). The definition of Σ' is unchanged except that the special relations are not rewritten; note that flagged-reachability and non-emptiness of the bodies ensure that they always contain a flagged relation, so that the variable that we add indeed occurs in the body (but it may not occur in the head). Clearly Σ' is still flagged-reachable. Further, it is clear that, if Σ consists of IDs or BaseIDs, then Σ' also does, as there is still only one fact in the body (and in the case of BaseID it is still a flagged fact), and there are no repetitions of variables. It is further clear that if Σ is empty then Σ' also is.

The definition of Q' is unchanged except that special relations are not rewritten; again, flagged-reachability of the query ensures that each fresh variable z_i indeed occurs in Q'_i, and the flagged-reachability of Q' is easy to see from that of Q. We define $\mathcal{F}_{\mathfrak{f}} := \{R(\mathfrak{f}, \ldots, \mathfrak{f}) \mid R \in \sigma'_{\mathcal{F}} \sqcup \sigma_{\mathcal{S}}\}$, and we define $(\mathcal{F}_0)_{+\mathfrak{t}}$ as the union of $\{R'(\vec{a}, \mathfrak{t}) \mid R(\vec{a}) \in \mathcal{F}_0 \wedge R \in \sigma_{\mathcal{F}}\}$ and of the $\sigma_{\mathcal{S}}$-facts of $\mathcal{F}_0$ kept as-is; then we define $\mathcal{F}'_0 := \mathcal{F}_{\mathrm{Or}} \sqcup (\mathcal{F}_0)_{+\mathfrak{t}} \sqcup \mathcal{F}_{\mathfrak{f}}$ as before.

We now explain how to modify the correctness proof. For the forward direction, consider a counterexample $\mathcal{F}$ to $\mathsf{QA}(\mathcal{F}_0, \Sigma \sqcup \Theta, Q)$, assuming without loss of generality that $\mathfrak{t}, \mathfrak{f} \notin \mathcal{F}$. We define $\mathcal{F}' := \mathcal{F}'_0 \sqcup \mathcal{F}_{+\mathfrak{t}} \sqcup \mathcal{F}_{\mathfrak{f}}$, with $\mathcal{F}_{\mathfrak{f}}$ as above and $\mathcal{F}_{+\mathfrak{t}}$ defined as above by adding $\mathfrak{t}$ to flagged facts and keeping special facts as-is.

To verify that this construction for the forward direction is correct, we must first show that $\mathcal{F}'$ satisfies the constraints Θ on $\sigma_{\mathcal{S}}$. For disjunctive inclusion dependencies τ, letting M be a match of the body, as Θ does not mention the facts of $\mathcal{F}_{\mathrm{Or}}$, we must have $M \in \mathcal{F}_{\mathfrak{f}}$ or $M \in \mathcal{F}_{+\mathfrak{t}}$. Now, in the first case, $\mathcal{F}_{\mathfrak{f}}$ must contain a match of some head disjunct of τ, and in the second case, by considering M in $\mathcal{F}$ which satisfies τ, we can also extend the match in $\mathcal{F}_{+\mathfrak{t}}$ hence in $\mathcal{F}'$. For transitivity assertions, we reason in the same way: seeing them as a TGD with a connected body that does not mention the relations of σ_{Or}, we deduce again that any match of them must be within $\mathcal{F}_{\mathfrak{f}}$ or within $\mathcal{F}_{+\mathfrak{t}}$, which allows us to conclude. The same reasoning works for transitive closure assertions. Hence, $\mathcal{F}'$ satisfies Θ.

Now, we must verify the other conditions on $\mathcal{F}'$, for which we adapt the proof of Lemma 9. In particular, observe that $\mathcal{F}'$ is still a superset of $\mathcal{F}'_0$. To check that there are no violations of Σ', consider a match M' of a TGD τ' of Σ'; as before M' includes no fact of $\mathcal{F}_{\mathrm{Or}}$, and the *flagged facts* $M'_{\mathcal{B}}$ of M' are either included in $\mathcal{F}_{+\mathfrak{t}}$ or in $\mathcal{F}_{\mathfrak{f}}$ depending on their last element. Now, as τ' is flagged-reachable, we observe that the *special facts* $M'_{\mathcal{D}}$ of M' must be connected to the flagged facts, so that, as $\mathcal{F}_{+\mathfrak{t}}$ and $\mathcal{F}_{\mathfrak{f}}$ have disjoint domains and no facts connect them except σ_{Or} facts which do not occur in τ', it must again be the case that the *entire match* M' is either included in $\mathcal{F}_{\mathfrak{f}}$ or in $\mathcal{F}_{+\mathfrak{t}}$. So we can conclude as before (in the second case, the preimage M of M' in $\mathcal{F}$ is defined without changing the special facts, but the same reasoning applies).

To check that $\mathcal{F}'$ violates Q', as before we reason by contradiction and deduce that $\exists \vec{x}_{i_0} Q'_{i_0}(\vec{x}_{i_0}, \mathfrak{t})$ holds in $\mathcal{F}'$. Now, its match M' in $\mathcal{F}'$ must consist of a match $M'_{\mathcal{B}}$ of flagged facts of M' with $\mathfrak{t}$ as their last position (so they are in $\mathcal{F}_{+\mathfrak{t}}$) and a match $M'_{\mathcal{D}}$ of special facts. As in the previous paragraph, we now use the fact that Q, hence Q'_{i_0}, is flagged-reachable, so we must also have $\mathsf{elems}(M'_{\mathcal{D}}) \subseteq \mathsf{elems}(M'_{\mathcal{B}})$. Hence, we have $M' \subseteq \mathcal{F}''$, and we deduce as before (except that we do not increase the arity of special facts) that the preimage M of M' is a match of Q in $\mathcal{F}$ and conclude by contradiction. This proves the correctness of the forward direction.

For the backward direction, we construct $\mathcal{F}$ as the disjoint union of $\{R(\vec{a}) \mid R'(\vec{a}, \mathfrak{t}) \in \mathcal{F} \wedge R' \in \sigma'_{\mathcal{F}}\}$ and of the special facts of $\mathcal{F}'$ kept as-is. It is clear that $\mathcal{F}'$ satisfies Θ, because its restriction to $\sigma_{\mathcal{S}}$ is the same as $\mathcal{F}$, which satisfies Θ. Again we have $\mathcal{F} \supseteq \mathcal{F}_0$. The fact that $\mathcal{F}$ satisfies Σ is as before, except that the arity of special facts is not changed in M', and the witness head facts of M' in $\mathcal{F}'$ may include special facts, in which case they are found as-is in $\mathcal{F}$. The fact that $\mathcal{F}$ violates Q is exactly as before, which concludes the correctness proof. $\square$

C.4 Revisiting the results of Section 5

Theorem 8. We apply Lemma 11 by picking as special relations E and E^+, and taking all others to be flagged relations. The special constraint set Θ asserts that E^+ is the transitive closure of E. We observe that the BaseIDs Σ' created in the proof are flagged-reachable, because their bodies always consist of base facts. Now, we observe that the UCQ Q' is also flagged-reachable: the E-atoms in Q-generated disjuncts are connected to the corresponding atom $R'(\vec{x}, e, f)$ (note that this is why we included this atom in the disjunct, rather than disallowing length-3 paths globally), the E-facts in E-path length restriction disjuncts are connected to an R'-fact, and the E-facts in DID satisfaction disjuncts are connected to the $\mathsf{Witness}_\tau$ fact.

Hence, we can deduce from Lemma 11 that Theorem 8 extends to QAtc with CQs, both for data complexity and combined complexity.

Theorem 9. It is immediate to observe that the BaseIDs Σ' do not mention the distinguished relations $<$, and we have already observed in the proof that the UCQ that we define is base-covered, hence base-domain-covered, so we deduce from Lemma 10 that Theorem 9 also extends to QAlin with CQs, for data and combined complexity.

Proposition 5. We let E and E^+ be the special relations, let all others be flagged relations, and let the special constraint set Θ assert that E^+ is the transitive closure of E. It is easy to observe that the query defined in the proof is flagged-reachable (this is the reason why we included the C_χ-atom in the E-path length restriction disjuncts). As the constraints are empty, their image by Lemma 11 also is, so we deduce that the data complexity lower bound of Proposition 5 still applies to QAtc with CQs.

Proposition 6. As the UCQ defined in the proof is base-covered, hence base-domain-covered, and the constraints are empty, we deduce from Lemma 10 that the lower bound still applies to QAlin with CQs.

C.5 Revisiting the results of Section 6

Theorem 11. We use Lemma 11. The special relations are S^+, K_i, and K_i', and the other relations are flagged. We let the special constraint set Θ consist of the three DIDs used in the proof, and of the assertion that S^+ is transitive. We let Σ consist of the two other dependencies used in the proof, which are inclusion dependencies, and are flagged-reachable. Applying Lemma 11, we can reduce our undecidable QA problem to QA on signature with S^+ as its only distinguished predicate, with a CQ which does not mention the distinguished predicate S^+ (because the original CQ did not), and with constraints comprising Θ (so a transitivity assertion for S^+, as well as DIDs), the translation of Σ (which are inclusion dependencies, so also DIDs), which allows us to conclude that the QA problem studied in Theorem 11 is indeed undecidable.

Theorem 10. As before, we use Lemma 11, and pick S' as the only flagged relation, and let all other relations be special relations. We let Σ consist of the one ID applying only to S'; it is flagged-reachable. According to these definitions, the disjuncts of the CQ that we write are always flagged-reachable, as they are connected and always include a S'-fact. The special constraint set Θ consists of all other IDs used in the proof, and of the assertion that S^+ is the transitive closure of S. Applying Lemma 11, we reduce the QA problem with a UCQ to QA for Θ (so IDs plus the transitive closure assertion on the one distinguished relation S^+), for the translation of Σ' (which are IDs), and a CQ which still does not use the one distinguished relation S^+ of the new signature. This establishes the result of Theorem 10.

Theorem 12. We use Lemma 10. We check that, indeed, the constraints do not mention the distinguished relation, and that the two UCQ disjuncts which do are base-domain-covered. Hence, the translation shows undecidability of QAlin for BaseIDs and a CQ, concluding the proof of Theorem 12.