Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Adding Circumscription to Decidable Fragments of First-Order Logic: A Complexity Rollercoaster

Carsten Lutz[1] , **Quentin Manière**[1,2]
[1]Department of Computer Science, Leipzig University, Germany
[2]Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI), Dresden/Leipzig, Germany
{carsten.lutz, quentin.maniere}@uni-leipzig.de

Abstract

We study extensions of expressive decidable fragments of first-order logic with circumscription, in particular the two-variable fragment FO^2, its extension C^2 with counting quantifiers, and the guarded fragment GF. We prove that if only unary predicates are minimized (or fixed) during circumscription, then decidability of logical consequence is preserved. For FO^2 the complexity increases from coNExp to $\mathrm{coNExp}^{\mathrm{NP}}$-complete, for GF it (remarkably!) increases from 2Exp to Tower-complete, and for C^2 the complexity remains open. We also consider querying circumscribed knowledge bases whose ontology is a GF sentence, showing that the problem is decidable for unions of conjunctive queries, Tower-complete in combined complexity, and elementary in data complexity. Already for atomic queries and ontologies that are sets of guarded existential rules, however, for every $k \geq 0$ there is an ontology and query that are k-Exp-hard in data complexity.

1 Introduction

There are various approaches to defining non-monotonic logics such as default rules, autoepistemic operators, and circumscription. Most of these are mainly used with propositional logic rather than with first-order logic (FO), for two reasons. First, many of the approaches such as default rules can yield non-intuitive results when used with first-order logics, interacting in unexpected ways with existential quantification; see for example (Baader and Hollunder 1995) for a discussion of this issue. And second, the undecidability of first-order logic of course carries over to its non-monotonic variants.

Description logics (DLs) are decidable fragments of FO for which non-monotonic variations have been studied extensively, see e.g. (Baader and Hollunder 1995; Donini, Nardi, and Rosati 2002; Bonatti, Lutz, and Wolter 2009; Giordano et al. 2013; Bonatti et al. 2015b). It turned out that circumscription provides one of the most well-behaved of such variations: it does not interact in dramatic ways with existential quantification, has a simple and appealing semantics that boils down to minimizing the interpretation of certain predicates, and comes with a clean way to preserve the decidability of the base logic. The latter is in fact achieved by permitting only unary predicates to be minimized or fixed during minimization while binary predicates must be allowed to vary (Bonatti, Lutz, and Wolter 2009). This still covers the main application of circumscription which is reasoning about typical properties of objects that belong to a certain class. To model the statement that KR papers are typically interesting, for example, one may write

$$\mathsf{KRPaper}(x) \land \neg\mathsf{abKRpaper}(x) \rightarrow \mathsf{Interesting}(x)$$

and then minimize the unary 'abnormality predicate' $\mathsf{abKRpaper}$. In this way, one may conclude that any concrete KR paper is interesting unless there is concrete evidence against that. For more information on DLs with circumscription, we refer to (Bonatti et al. 2015a; Stefano, Ortiz, and Simkus 2023; Lutz, Manière, and Nolte 2023)

It is well-known that DLs are generalized by various decidable and more expressive FO fragments, of which the two-variable fragment FO^2, the guarded fragment GF, and the extension C^2 of FO^2 with counting quantifiers are the most important ones. In this paper, we ask the following questions: *Do expressive decidable fragments of FO remain decidable when extended with circumscription (when only unary predicates are minimized or fixed)? And if so, what is the impact on computational complexity?* The answers are, in our opinion, somewhat surprising.

We study the reasoning problems of *circumscribed consequence* and *circumscribed querying*. In the former, two sentences ϕ and ψ are given along with a 'circumscription pattern' CP that specifies which predicates are minimized, fixed, and varying. We are then interested in deciding whether ψ holds in every model that is minimal in the sense specified by CP, written $\phi \models_{\mathsf{CP}} \psi$. Circumscribed querying is defined in the same way, but now ϕ is a knowledge base that consists of a sentence from the FO fragment under consideration (specifying an ontology) and a database, and ψ is a query. As query languages, we consider single-atom queries (AQs), conjunctive queries (CQs), and and unions thereof (UCQs).

We start with studying FO^2. Similarly to the case of description logic (Bonatti, Lutz, and Wolter 2009), a crucial step for proving decidability is to show that circumscribed FO^2 has the finite model property (FMP) in the sense that if $\phi \not\models_{\mathsf{CP}} \psi$, then there is a CP-minimal model $\mathfrak{A}$ of ϕ with $\mathfrak{A} \not\models \psi$ that is of bounded size. To prove this, we build on a well-known construction from (Grädel, Kolaitis, and Vardi 1997), used there to establish the FMP of non-circumscribed FO^2, which converts a potentially infinite model $\mathfrak{A}$ of an FO^2 sentence ϕ into a model $\mathfrak{B}$ of single exponential size. To apply this construction in the circumscribed case, however, we need an additional condition to be satisfied:

(♡) $\mathfrak{B}$ must not realize any 1-type more often than $\mathfrak{A}$ (for a suitable notion of 1-type).

The construction of (Grädel, Kolaitis, and Vardi 1997) does not satisfy this condition. We remark that this is in contrast to filtration, the (much simpler) finite model construction used for description logics such as $\mathcal{ALC}$.

We thus rework the construction of (Grädel, Kolaitis, and Vardi 1997) in a suitable way, obtaining a version that satisfies Condition (♡). This yields the FMP for circumscribed FO^2 and decidability as well as a $\text{coNExp}^{\text{NP}}$ upper complexity bound for circumscribed consequence. A matching lower bound is obtained from $\mathcal{ALC}$ and thus circumscribed consequence in FO^2 is of the same complexity as in (the much less expressive) $\mathcal{ALC}$. We obtain the same result for the combined complexity of circumscribed AQ-querying and also show Π_2^p-completeness for data complexity, again the same as in $\mathcal{ALC}$. Querying with UCQs is undecidable already for non-circumscribed FO^2, so we do not study it.

For GF, we follow the same general approach, with a remarkably different outcome. There are two constructions that show the finite model property of GF, both of them rather intricate. The historically first one was proposed by Grädel, based on a combinatorial construction due to Herwig (Grädel 1999). Later, Rosati introduced a different finite model construction while studying certain integrity constraints for databases (Rosati 2006), and this construction, now known as the *Rosati cover*, has been adapted to GF in (Bárány, Gottlob, and Otto 2014). Both constructions fail to yield Property (♡) and modifying them to achieve this property turns out to be much more difficult than in the case of FO^2. We give a modified version of the Rosati cover that yields finite models of non-elementary size, compared to single exponential size as for the original Rosati cover. This yields the FMP for circumscribed GF. We then show that the non-elementary size of finite models is unavoidable: circumscribed consequence in GF is TOWER-complete! To us, this huge difference to the FO^2 case came as a big surprise. We also show that circumscribed querying in GF is decidable, generalizing recent work on DLs (Lutz, Manière, and Nolte 2023). In combined complexity, it is TOWER-complete with the lower bound applying to AQs and the upper bound to UCQs. Regarding data complexity, it is elementary in the sense that for each GF ontology $\mathcal{O}$, circumscription pattern CP, and UCQ q, querying is in k-EXP for some k. We also show that there is no uniform bound on k: for each $k \geq 1$ we identify an ontology $\mathcal{O}$, circumscription pattern CP, and AQ q for which querying is k-EXP-hard. In fact, $\mathcal{O}$ is a set of existential rules, a 'positive' fragment of GF that is important for querying. We also show that with a single minimized predicate and all other predicates varying, the data complexity of AQ-querying is EXP-hard in GF. Note that since CQs are sandwiched beween AQs and UCQs, this also completely clarifies the (combined and data) complexity for this query language.

In addition, we provide first results on circumscribed consequence and AQ-querying in C^2. Using a reduction to Boolean algebra with Presburger arithmetic, we show that these problems are decidable. The complexity remains open.

An appendix with full proofs can be found in the long version of this paper, see (Lutz and Manière 2024).

2 Preliminaries

When speaking of first-order logic (FO), we generally mean the version with equality and constants (unless otherwise noted) and without function symbols. FO^2 is the two-variable fragment of FO, obtained by fixing two variables x and y and disallowing the use of any other variables (Scott 1962; Mortimer 1975; Grädel, Kolaitis, and Vardi 1997). C^2 is the extension of FO^2 with counting quantifiers of the form $\exists_{\leq n}$, $\exists_{\geq n}$, and $\exists_{=n}$ for every $n \geq 0$ (Grädel, Otto, and Rosen 1997; Pacholski, Szwast, and Tendera 1997; Pratt-Hartmann 2005). In FO^2 and C^2, we generally only admit predicates of arity at most two. In the guarded fragment of FO, denoted GF, quantification is restricted to the pattern

$$\forall \bar{y}(\alpha(\bar{x}, \bar{y}) \rightarrow \varphi(\bar{x}, \bar{y})) \qquad \exists \bar{y}(\alpha(\bar{x}, \bar{y}) \wedge \varphi(\bar{x}, \bar{y}))$$

where $\varphi(\bar{x}, \bar{y})$ is a GF formula with free variables among $\bar{x}, \bar{y}$ and $\alpha(\bar{x}, \bar{y})$ is a relational atom $R(\bar{x}, \bar{y})$ or an equality atom $x = y$ that in either case contains all variables in $\bar{x}, \bar{y}$ (Andréka, Németi, and van Benthem 1998; Grädel 1999). The formula α is called the *guard* of the quantified formula.

We use the standard notation of first-order logic, denoting structures with $\mathfrak{A}$ and $\mathfrak{B}$, their universes with A and B, and the interpretation of predicates R with $R^{\mathfrak{A}}$ and $R^{\mathfrak{B}}$. We reserve a countably infinite set of predicates of each arity. We use $|\phi|$ to denote the *length* of the formula ϕ, that is, the length of ϕ when encoded as a word over a suitable alphabet.

Circumscription. A *circumscription pattern* is a tuple $\mathsf{CP} = (\prec, \mathsf{M}, \mathsf{F}, \mathsf{V})$, where M, F and V partition the unary predicates into *minimized*, *fixed* and *varying* predicates, and $\prec$ is a strict partial order on M called the *preference relation*. The order $\prec$ also induces a preference relation $<_{\mathsf{CP}}$ on structures by setting $\mathfrak{B} <_{\mathsf{CP}} \mathfrak{A}$ if the following conditions hold:

1. $B = A$ and $c^{\mathfrak{A}} = c^{\mathfrak{B}}$ for all constants c,
2. for all $P \in \mathsf{F}$, $P^{\mathfrak{B}} = P^{\mathfrak{A}}$,
3. for all $P \in \mathsf{M}$ with $P^{\mathfrak{B}} \not\subseteq P^{\mathfrak{A}}$, there is a $Q \in \mathsf{M}$, $Q \prec P$, such that $Q^{\mathfrak{B}} \subsetneq Q^{\mathfrak{A}}$,
4. there exists a $P \in \mathsf{M}$ such that $P^{\mathfrak{B}} \subsetneq P^{\mathfrak{A}}$ and for all $Q \in \mathsf{M}$, $Q \prec P$ implies $Q^{\mathfrak{B}} = Q^{\mathfrak{A}}$.

A CP-*minimal model* of an FO sentence ϕ is a model $\mathfrak{A}$ of ϕ such that there is no $\mathfrak{B} <_{\mathsf{CP}} \mathfrak{A}$ that is a model of ϕ. Note that predicates of arity larger than one always vary to avoid undecidability (Bonatti, Lutz, and Wolter 2009). We also assume that nullary predicates always vary, which is w.l.o.g. as they can be simulated by unary predicates.

For FO sentences ϕ and ψ, we write $\phi \models_{\mathsf{CP}} \psi$ if every CP-minimal model $\mathfrak{A}$ of ϕ satisfies $\mathfrak{A} \models \psi$. Take any fragment F of FO such as FO^2. With *circumscribed consequence* in F we mean the problem to decide, given sentences ϕ and ψ from F and a circumscription pattern CP, whether $\phi \models_{\mathsf{CP}} \psi$.

Ontology-mediated querying. Ontology-mediated querying with circumscribed knowledge bases, as recently studied in (Lutz, Manière, and Nolte 2023), can be seen as a version of circumscribed consequence where ϕ encodes an ontology and a database and ψ is a query. We next make this precise.

A *database* is a finite set of ground atoms, in this context called *facts*. We use $\mathsf{adom}(D)$ to denote the set of constants that occur in D. A structure $\mathfrak{A}$ *satisfies* a database D if (1) it satisfies all facts in it and (2) interprets all constant symbols c in $\mathsf{adom}(D)$ as $c^{\mathfrak{A}} = c$ (and thus no two such c denote the same element of A). We then also say that $\mathfrak{A}$ is a *model* of D and write $\mathfrak{A} \models D$. Note that Point (2) is the *standard names assumption*, as usually made in the context of databases. A *knowledge base (KB)* $\mathcal{K}$ takes the form $\bigwedge \mathcal{O} \wedge D$ with $\mathcal{O}$ a finite set of FO sentences, called the *ontology*, and D a database. We usually denote $\mathcal{K}$ as a pair $(\mathcal{O}, D)$. We call $\mathcal{K}$ a GF-KB if all sentences in $\mathcal{O}$ fall into GF, and likewise for other FO fragments.

A *conjunctive query (CQ)* is an FO formula of the form $q = \exists \bar{x}\, \varphi(\bar{x})$ where φ is a conjunction of relational atoms, possibly involving constants. An *atomic query (AQ)* is a CQ of the simple form $R(\bar{c})$ with $\bar{c}$ a tuple of constants. A *union of conjunctive queries (UCQ)* $q(\bar{x})$ is a disjunction of CQs. Let $\mathcal{K}$ be a KB and q a UCQ. We write $\mathcal{K} \models_{\mathsf{CP}} q$ if $\mathfrak{A} \models q$ for every CP-minimal model $\mathfrak{A}$ of $\mathcal{K}$. The notion $\mathcal{K} \models q$ is defined analogously, except that all models of $\mathcal{K}$ are considered, not only CP-minimal ones. Take a fragment F of FO such as GF and a query language Q such as UCQs. With *circumscribed Q-querying in F*, we mean the problem to decide, given a knowledge base $\mathcal{K} = (\mathcal{O}, D)$ with $\mathcal{O}$ a set of sentences from F and a query q from Q, whether $\mathcal{K} \models_{\mathsf{CP}} q$. When studying the combined complexity of this problem, all of $\mathcal{K}$, CP, and q are treated as inputs. For data complexity, we assume $\mathcal{O}$, CP, and q to be fixed and thus of constant size. We remark that our queries are Boolean, that is, they do not have answer variables. This is without loss of generality since constants are admitted in queries.

We shall also consider ontologies $\mathcal{O}$ that are sets of guarded existential rules. An *existential rule* is an FO sentence of the form

$$\forall \bar{x} \forall \bar{y}\, (\phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z}\, \psi(\bar{x}, \bar{z}))$$

where ϕ and ψ are conjunctions of relational atoms. We call ϕ the *body* of the rule and ψ the *head*. The rule is *guarded* if the body contains an atom that contains all variables in it. When writing existential rules, we usually omit the universal quantifiers. For every ontology $\mathcal{O}$ that is a set of guarded existential rules, there is a GF ontology $\mathcal{O}'$ such that for all databases D and UCQs q, we have $(\mathcal{O}, D) \models q$ iff $(\mathcal{O}', D) \models q$ (Calì, Gottlob, and Lukasiewicz 2009). To construct $\mathcal{O}'$, one simply adds a fresh predicate to the head of each rule in $\mathcal{O}$ that contains all variables in the head, and then translates the resulting set of rules into an equivalent GF sentence in a straightforward way. This proof also applies to circumscribed querying, letting the fresh predicates vary.

Example 1. *Consider the database*

$$D = \{W(w_1), W(w_2), E(w_1), E(w_2), \mathsf{offers}(s, p)\}.$$

where $\mathsf{offers}(s, p)$ *means that supplier s offers product p, W stands for warehouse, and E for express. Assuming that we have complete knowledge of all existing warehouses (e.g. in our company), we use a circumscription pattern* CP *that minimizes predicate W and lets all other predicates vary. Let the ontology $\mathcal{O}$ contain the guarded existential rules*

$$\mathsf{offers}(x, y) \rightarrow \exists z\, \mathsf{supplies}(x, y, z)$$
$$\mathsf{supplies}(x, y, z) \rightarrow W(z)$$

where $\mathsf{supplies}(x, y, z)$ *expresses that supplier x supplies product y to warehouse z. Note that since W is minimized, the existentially quantified variable z in the upper rule can only bind to w_1 and w_2. We thus obtain*

$$(\mathcal{O}, D) \models_{\mathsf{CP}} \exists z\, (\mathsf{supplies}(s, p, z) \wedge E(z)).$$

We now illustrate a basic trick that underlies the hardness proofs in Section 5. Extend the database with $\mathsf{mirror}(w_1, w_2)$ *and* $\mathsf{mirror}(w_2, w_1)$ *expressing that w_1 and w_2 are supplied with the same products by the same suppliers. We wish to extend $\mathcal{O}$ with*

$$\mathsf{supplies}(x, y, z_1) \wedge \mathsf{mirror}(z_1, z_2) \rightarrow \mathsf{supplies}(x, y, z_2)$$

which yields

$$(\mathcal{O}, D) \models_{\mathsf{CP}} \mathsf{supplies}(s, p, w_i) \textit{ for all } i \in \{1, 2\}. \quad (*)$$

However, the above rule is not guarded. We may work around this by using the guarded rules

$$\mathsf{supplies}(x, y, z_1) \rightarrow \exists z_2\, (\mathsf{mirror}(z_1, z_2) \wedge \mathsf{supplies}(x, y, z_2))$$
$$\mathsf{mirror}(x, y) \wedge \overline{\mathsf{mirror}}(x, y) \rightarrow \mathsf{false}$$

and extend the data with $\overline{\mathsf{mirror}}(w_i, w_i)$ *for $i \in \{1, 2\}$. Then if z_1 binds to w_1, the existentially quantified variable z_2 can only bind to w_2 and vice versa, and we again obtain $(*)$.*

Substitutions, Signatures, Types. For a tuple $\bar{a}$, we generally use a_i to denote the i-th element of $\bar{a}$, for $1 \leq i \leq |\bar{a}|$. A *substitution* σ is a function that maps variables to variables. We typically write σx in place of $\sigma(x)$. For a tuple $\bar{u}$ of variables and constants, we write $\sigma \bar{u}$ to denote the tuple obtained by applying σ componentwise, treating it as the identity on constants.

A *signature* is a set of constants and relation symbols. For an FO sentence ϕ, we use $\mathsf{sig}(\phi)$ to denote the set of such symbols in ϕ, $\mathsf{const}(\phi)$ to denote the set of constants in ϕ, and $\mathsf{const}_{=}(\phi)$ to denote the set of constants used in ϕ in an equality atom. A signature is *unary* if it only contains constants symbols and unary predicates.

Fix a signature Σ. A *term* is a variable or a constant from Σ. An *atom* is of the form $R(\bar{u})$ or $v_1 = v_2$ with R a relation symbol from Σ, $\bar{u}$ a tuple of terms and v_1, v_2 terms. A *literal* is an atom or a negated atom. For every $n \geq 1$, fix a sequence of variables $x_1, \ldots, x_n$. An *n-type* on Σ is a maximal satisfiable set of literals that uses exactly the variables $x_1, \ldots, x_n$. Let $\mathfrak{A}$ be a structure. If $\bar{a} \in A^n$, then the n-type on Σ *realized at* $\bar{a}$ in $\mathfrak{A}$, denoted $\mathsf{tp}^n_{\mathfrak{A},\Sigma}(\bar{a})$, is the unique n-type t on Σ with $\mathfrak{A} \models t(\bar{a})$. We may drop superscript n as n is always identical to the length of $\bar{a}$. In this paper, 1-types will play a crucial role. For a set $S \subseteq A$, we use $\mathsf{tp}^1_{\mathfrak{A},\Sigma}(S)$ to denote the set of 1-types $\{\mathsf{tp}^1_{\mathfrak{A},\Sigma}(a) \mid a \in S\}$. As an abbreviation, we may write $\mathsf{tp}^1_{\Sigma}(\mathfrak{A})$ in place of $\mathsf{tp}^1_{\mathfrak{A},\Sigma}(A)$. Moreover, For every 1-type t on Σ, set

$$\#_{\mathfrak{A}}(t) := |\{a \in A \mid \mathsf{tp}^1_{\mathfrak{A},\Sigma}(a) = t\}|.$$

3 The Two-Variables Fragment FO^2

We show that circumscribed consequence is $\textsc{coNExp}^{\text{NP}}$-complete in FO^2 and so is circumscribed AQ-querying, in combined complexity. Note that querying with CQs or UCQs is undecidable already for non-circumscribed FO^2. We remark that this section showcases the general approach that we also use, in a more intricate form, for GF later on.

An FO^2 sentence is in *Scott normal form* if it has the form

$$\phi = \forall x \forall y\, \varphi \wedge \bigwedge_{i=1..n_\exists} \forall x \exists y\, \psi_i \qquad (*)$$

with φ and ψ_i quantifier-free. It has been shown in (Scott 1962; Grädel, Kolaitis, and Vardi 1997) that every FO^2 sentence ϕ_0 can be converted in polynomial time into an FO^2 sentence ϕ in Scott normal form that is a conservative extension of ϕ_0: every model of ϕ is a model of ϕ_0 and, conversely, every model of ϕ_0 can be extended to a model of ϕ by interpreting the fresh predicates in ϕ. It can be verified that the same holds for CP-minimal models, for any circumscription pattern CP, where the fresh predicates are set to varying. It follows that for deciding circumscribed consequence and circumscribed querying, we can w.l.o.g. assume the input sentence to be in Scott normal form. We thus assume throughout this section that all FO^2 sentences are in Scott normal form.

We now establish an improved finite model property for non-circumscribed FO^2 that satisfies Property ($\heartsuit$) from the introduction.

Proposition 1. *Let ϕ be an FO^2 sentence with $n_\exists$ existential quantifiers, $\Sigma \subseteq \mathsf{sig}(\phi)$ a unary signature that contains $\mathsf{const}_=(\phi)$, $\mathfrak{A}$ a model of ϕ, and $k = |(\mathsf{sig}(\phi) \setminus \mathsf{const}(\phi)|) \cup \mathsf{const}_=(\phi)$. Then there exists a model $\mathfrak{B}$ of ϕ such that*

1. $|B| \leq |\phi|^{n_\exists+1} \cdot 2^{n_\exists 4(k+6)}$;
2. $\mathsf{tp}^1_\Sigma(\mathfrak{A}) = \mathsf{tp}^1_\Sigma(\mathfrak{B})$;
3. $\#_\mathfrak{B}(t) \leq \#_\mathfrak{A}(t)$ *for each 1-type t on Σ;*
4. $\#_\mathfrak{B}(t) = \#_\mathfrak{A}(t)$ *for each 1-type t on Σ s.t. $\#_\mathfrak{A}(t) \leq 2^{|\Sigma|}$;*
5. $c^\mathfrak{A} = c^\mathfrak{B}$ *and* $\mathsf{tp}^1_{\mathfrak{A},\Sigma}(c) = \mathsf{tp}^1_{\mathfrak{B},\Sigma}(c)$ *for all constants c.*

The reader should think of Σ as containing all unary relations that are minimized and fixed in a circumscription pattern. We remark that the construction from (Grädel, Kolaitis, and Vardi 1997) only yields Proposition 1 without Points 3 and 4, that is, it may *increase* the number of instances of some of the 1-types realized in the original model. We next use Proposition 1 to establish the following.

Theorem 1. *Circumscribed FO^2 has the finite model property: if ϕ, ψ are FO^2-sentences with $\phi \not\models_{\mathsf{CP}} \psi$, then there is a CP-minimal model $\mathfrak{A}$ of $\vartheta = \phi \wedge \neg\psi$ with $|A| \leq |\vartheta|^{n_\exists+1} \cdot 4^{n_\exists(k+6)}$, where $n_\exists$ is the number of existential quantifiers in the Scott normal form of ϑ and $k = |\mathsf{sig}(\vartheta) \setminus \mathsf{const}(\vartheta)|$.*

Proof. Assume that $\phi \not\models_{\mathsf{CP}} \psi$ and let $\mathsf{CP} = (\prec, \mathsf{M}, \mathsf{F}, \mathsf{V})$. Then there is a CP-minimal model $\mathfrak{A}$ of ϕ with $\mathfrak{A} \not\models \psi$. Thus $\mathfrak{A}$ is a model of $\phi \wedge \neg\psi$. Let $\Sigma = \mathsf{M} \cup \mathsf{F} \cup \mathsf{const}_=(\phi)$. By Proposition 1, there is a model $\mathfrak{B}$ of $\phi \wedge \neg\psi$ that satisfies Points 1-4 of the proposition, with $\Sigma = \mathsf{sig}(\phi \wedge \neg\psi)$. We show that $\mathfrak{B}$ is a CP-minimal model of ϕ.

Assume to the contrary that there is a model $\mathfrak{B}'$ of ϕ such that $\mathfrak{B}' <_{\mathsf{CP}} \mathfrak{B}$. To obtain a contradiction, we construct a model $\mathfrak{A}'$ of ϕ such that $\mathfrak{A}' <_{\mathsf{CP}} \mathfrak{A}$.

Of course, $\mathfrak{A}'$ must have the same universe as $\mathfrak{A}$, thus we set $A' = A$. Due to Points 3 and 4 of Proposition 1, we find an injection $f : B \to A$ such that

(i) $\mathsf{tp}^1_{\mathfrak{B},\Sigma}(b) = \mathsf{tp}^1_{\mathfrak{A},\Sigma}(f(b))$ for all $b \in B$;

(ii) if t is a 1-type on Σ with $\#_\mathfrak{A}(t) \leq 2^{|\Sigma|}$, then $\mathsf{tp}^1_{\mathfrak{A},\Sigma}(a) = t$ implies that a is in the range of f, for all $a \in A$.

We define $\mathfrak{A}'$ so that its restriction to the range of f is isomorphic to $\mathfrak{B}'$, with f being an isomorphism. In particular, this restriction interprets all constants $c \in \mathsf{const}_=(\phi)$. It can be verified that $c^{\mathfrak{A}'} = c^\mathfrak{A}$ for all such c, see appendix.

To define the remaining part of $\mathfrak{A}'$, we use cloning. For every $a \in A$ that is not in the range of f we identify an $\widehat{a} \in A$ that is in the range of f and such that $\mathsf{tp}^1_{\mathfrak{A},\Sigma}(\widehat{a}) = \mathsf{tp}^1_{\mathfrak{A},\Sigma}(a)$. Take any such a and let $\mathsf{tp}^1_{\mathfrak{A},\Sigma}(a) = t$. From (ii), we obtain $\#_\mathfrak{A}(t) > 2^{|\Sigma|}$. Point 4 yields $\#_\mathfrak{B}(t) > 2^{|\Sigma|}$. Thus, we find distinct $b, b' \in B$ such that $\mathsf{tp}^1_{\mathfrak{B},\Sigma}(b) = \mathsf{tp}^1_{\mathfrak{B},\Sigma}(b') = t$ and $\mathsf{tp}^1_{\mathfrak{B}',\Sigma}(b) = \mathsf{tp}^1_{\mathfrak{B}',\Sigma}(b')$. Set $\widehat{a} = f(b)$.

We then make each $a \in A$ that is not in the range of f a clone of $\widehat{a}$ in $\mathfrak{A}'$, that is, for $\Gamma = \mathsf{sig}(\phi')$ we set

- $\mathsf{tp}^1_{\mathfrak{A}',\Gamma}(a) = \mathsf{tp}^1_{\mathfrak{A}',\Gamma}(\widehat{a})$;
- $\mathsf{tp}^2_{\mathfrak{A}',\Gamma}(a, b) = \mathsf{tp}^2_{\mathfrak{A}',\Gamma}(\widehat{a}, b)$ for all b in the range of f;
- $\mathsf{tp}^2_{\mathfrak{A}',\Gamma}(a, b) = \mathsf{tp}^2_{\mathfrak{A}',\Gamma}(\widehat{a}, \widehat{b})$ for all b not in the range of f.

It remains to interpret the constants $c \in \Gamma \setminus \mathsf{const}_=(\phi)$: set $c^{\mathfrak{A}'} = c^\mathfrak{A}$.

We prove in the appendix that $\mathfrak{A}'$ is a model of ϕ, since $\mathfrak{B}'$ is and it can be verified that $\mathfrak{A}' <_{\mathsf{CP}} \mathfrak{A}$, since $\mathfrak{B}' <_{\mathsf{CP}} \mathfrak{B}$. ☐

It is now easy to derive the main result of this section.

Theorem 2. *Circumscribed consequence in FO^2 is $\textsc{coNExp}^{\text{NP}}$-complete.*

Proof. The lower bound is inherited from the description logic $\mathcal{ALC}$ (Bonatti, Lutz, and Wolter 2009). The upper bound is based on Theorem 1, as follows.

It is not hard to see that there exists an NP algorithm that takes as input an FO^2 sentence ϕ, a circumscription pattern CP, and a finite structure $\mathfrak{A}$ and checks whether $\mathfrak{A}$ is *not* a CP-minimal model of ϕ: the algorithm first checks in polynomial time whether $\mathfrak{A}$ is a model of ϕ, answering "yes" if this is not the case. Otherwise, it guesses a structure $\mathfrak{A}'$ with $A' = A$ and checks whether $\mathfrak{A}'$ is a model of ϕ and $\mathfrak{A}' <_{\mathsf{CP}} \mathfrak{A}$. It answers "yes" if both checks succeed, and "no" otherwise. Clearly, checking whether $\mathfrak{A}' <_{\mathsf{CP}} \mathfrak{A}$ can be done in time polynomial in the size of $\mathfrak{A}$.

We use this NP algorithm as an oracle in a $\textsc{NExp}$-algorithm for deciding $\phi \not\models_{\mathsf{CP}} \psi$: by Theorem 1, it suffices to guess a structure $\mathfrak{A}$ with $|A| \leq |\vartheta|^{n_\exists+1} \cdot 2^{n_\exists 4(k+6)}$, check that it is not a model of ψ, and then use the NP algorithm from above to check that $\mathfrak{A}$ is a CP-minimal model of ϕ. ☐

From the above, we also obtain results on circumscribed AQ-querying.

Theorem 3. *Circumscribed AQ-querying in FO^2 is* coNEXP$^{\text{NP}}$*-complete in combined complexity and* Π_2^p*-complete in data complexity.*

Proof. For combined complexity, it suffices to show that circumscribed consequence and circumscribed AQ-querying mutually reduce to one another in polynomial time. First, $(\mathcal{O}, D) \models_{\mathsf{CP}} R(\bar{c})$ is equivalent to $\phi \models_{\mathsf{CP}'} R(\bar{c})$ where

$$\phi = \bigwedge \mathcal{O} \wedge \bigwedge D \wedge \bigwedge_{c \in \mathsf{adom}(D)} \big(P_c(c) \wedge \bigwedge_{c' \in \mathsf{adom}(D) \setminus \{c\}} \neg P_{c'}(c)\big).$$

with each P_c a fresh unary predicate and CP' identical to CP except that the fresh predicates are varying. And second, $\phi \models_{\mathsf{CP}} \psi$ is equivalent to $\phi' \models_{\mathsf{CP}'} P(c)$ where $\phi' = \phi \wedge (\psi \to P(c))$, P is a fresh unary predicate that is varying in CP', and c a fresh constant.

The lower bound for data complexity is inherited from $\mathcal{ALC}$ (Lutz, Manière, and Nolte 2023). For the upper bound, we may argue exactly as in the proof of Theorem 2, where the structure $\mathfrak{A}$ to be guessed is now of polynomial size since $n_\exists$ and k are now constants in Propositions 1 and Theorem 1. For k, this depends on the assumption, which we may make w.l.o.g., that the database contains only predicates that occur also in the ontology or query. □

4 Upper Bounds for the Guarded Fragment

We show that circumscribed consequence in GF is in TOWER and so is circumscribed UCQ-querying, in combined complexity. We also show that UCQ-querying is in ELEMENTARY in data complexity, that is, for every GF ontology $\mathcal{O}$, circumscription pattern CP, and AQ $A(\bar{x})$, there is a $k \geq 1$ such that given a database D, it is in k-EXP to decide whether $(\mathcal{O}, D) \models_{\mathsf{CP}} A(\bar{x})$.

We remind the reader of the relevant complexity classes, namely ELEMENTARY $= \bigcup_{k \geq 1} k$-EXP and

$$\text{TOWER} = \bigcup_{f \in \text{FELEM}} \text{SPACE}(\text{tower}(f(n)))$$

where FELEM is the class of all elementary functions and tower(x) denotes a tower of twos of height x.

4.1 Circumscribed Consequence

A GF sentence is in *Scott normal form* if it takes the form

$$\bigwedge_{1 \leq i \leq n_\forall} \forall \bar{x}\, (\alpha_i \to \varphi_i) \wedge \bigwedge_{1 \leq i \leq n_\exists} \forall \bar{x}\, (\beta_i \to \exists \bar{y}\, (\gamma_i \wedge \psi_i)) \quad (*)$$

where the α_i, β_i and γ_i are atoms and the φ_i and ψ_i are quantifier-free. It has been shown in (Grädel 1999) that every GF sentence ϕ can be converted in polynomial time into a GF sentence in Scott normal form that is a conservative extension of ϕ. As in the case of FO^2, we may thus assume that all sentences are in Scott normal form.

We now state the improved finite model property for GF that satisfies Property (♡) from the introduction. Let $\mathsf{tower}(0, n) := n$ and, for every $k \geq 1$, define $\mathsf{tower}(k, n) := 2^{\mathsf{tower}(k-1,n)}$, so that $\mathsf{tower}(k, n)$ refers to an exponentiation tower that consists of k twos followed by an n.

Proposition 2. *Let ϕ be a GF sentence in Scott normal form, $\Sigma \subseteq \mathsf{sig}(\phi)$ a unary signature that contains $\mathsf{const}_=(\phi)$, and $\mathfrak{A}$ a model of ϕ. Then there exists a model $\mathfrak{B}$ of ϕ that satisfies the following properties:*

1. $|B| \leq \mathsf{tower}(4^{|\Sigma|+4}, |\phi|)$*;*
2. $\mathsf{tp}^1_\Sigma(\mathfrak{A}) = \mathsf{tp}^1_\Sigma(\mathfrak{B})$*;*
3. $|\{a \in B \mid \mathsf{tp}^1_{\mathfrak{B},\Sigma}(a) = t\}| \leq |\{a \in A \mid \mathsf{tp}^1_{\mathfrak{A},\Sigma}(a) = t\}|$ *for every 1-type t on Σ;*
4. $c^{\mathfrak{A}} = c^{\mathfrak{B}}$ *for all constants c in ϕ.*

To establish Proposition 2, we use a modified version of the Rosati cover that leaves untouched a selected part Δ from the original model. In addition, if Δ contains all the instances of some unary type in the original model, then so does Δ in the modified Rosati cover. The precise formulation follows.

Lemma 1. *Let ϕ be a GF sentence in Scott normal form and $\Sigma \subseteq \mathsf{sig}(\phi)$ a unary signature that contains $\mathsf{const}_=(\phi)$. For all models $\mathfrak{A}$ of ϕ and all $\Delta \subseteq A$ that contain $c^{\mathfrak{A}}$ for every constant c in ϕ, there exists a model $\mathfrak{B}$ of ϕ that satisfies the following properties:*

1. $|B| \leq 2^{(|\Delta|+|\phi|)^{|\phi|+11}}$*;*
2. $\mathsf{tp}^1_{\mathfrak{A},\Sigma}(a) = \mathsf{tp}^1_{\mathfrak{B},\Sigma}(a)$ *for all $a \in \Delta$;*
3. $\mathsf{tp}^1_{\mathfrak{A},\Sigma}(A \setminus \Delta) = \mathsf{tp}^1_{\mathfrak{B},\Sigma}(B \setminus \Delta)$*;*
4. $\Delta \subseteq B$ *and $c^{\mathfrak{A}} = c^{\mathfrak{B}}$ for all constants c in ϕ.*

We now prove Proposition 2 by using Lemma 1 and choosing an appropriate Δ. For any structure $\mathfrak{A}$ and $\Delta \subseteq A$ that contains $c^{\mathfrak{A}}$ for all constants c in ϕ, we use $\mathsf{rc}(\mathfrak{A}, \Delta)$ to denote the finite model of ϕ produced by Lemma 1 (where 'rc' stands for 'Rosati cover').

Let ϕ, Σ, and $\mathfrak{A}$ be as in Proposition 2. The challenge is to choose Δ so that Point 3 of Proposition 2 is satisfied. Call a 1-type t *stable* w.r.t. $\Delta \subseteq A$ if $\#_{\mathsf{rc}(\mathfrak{A},\Delta)}(t) \leq \#_{\mathfrak{A}}(t)$ and call Δ *stabilizing* if all 1-types are stable w.r.t. Δ. To attain Point 3, it clearly suffices to choose a stabilizing Δ.

We use a set Δ that contains *all* instances of 1-types realized only a certain number of times: for $m \geq 1$, set

$$\Delta_m := \{a \in A \mid \#_{\mathfrak{A}}(\mathsf{tp}^1_{\mathfrak{A},\Sigma}(a)) \leq m\} \cup \{c^{\mathfrak{A}} \mid c \in \mathsf{const}(\phi)\}.$$

Now consider Δ_m, for some $m \geq 1$. For those 1-types t that are realized in $\mathfrak{A}$ at most m times, it is clear from Points 2 and 3 of Lemma 1 that $\mathsf{rc}(\mathfrak{A}, \Delta_m)$ has the very same instances of t, and thus t is stable w.r.t. Δ_m. Other types, however, may not be stable.

So can we find a value for m to make Δ_m stabilizing? This is trivially the case for

$$m := \max(\{\#_{\mathfrak{A}}(t) \mid t \in \mathsf{tp}^1_\Sigma(\mathfrak{A})uc, \#_{\mathfrak{A}}(t) < +\infty\}),$$

but we would like to have an m that is bounded from above to comply with Point 1 in Proposition 2.

Lemma 2. *There exists a stabilizing set Δ_m such that $2 \leq m \leq \mathsf{tower}(2^{2|\Sigma|+4}, |\phi|)$.*

Proof. We start with $m_0 = 2$ (starting with 1 would also work but using 2 simplifies calculations as we are dealing with towers of 2s). If Δ_{m_0} is stabilizing, we are done.

Otherwise there must be a 1-type t on Σ that is not stable w.r.t. Δ_{m_0}, i.e. $\#_{\mathsf{rc}(\mathfrak{A},\Delta_{m_0})}(t) > \#_{\mathfrak{A}}(t)$. This implies in particular that $\#_{\mathfrak{A}}(t)$ is no larger than the size of the universe of $\mathsf{rc}(\mathfrak{A},\Delta_{m_0})$. Using the bound from Point 1 of Lemma 1, we set $m_1 = 2^{(|\Delta_{m_0}|+|\phi|)^{|\phi|+11}}$. Now all instances of t in $\mathfrak{A}$ are contained in Δ_{m_1} and by Points 2 and 3 of Lemma 1 we have $\#_{\mathsf{rc}(\mathfrak{A},\Delta_{m_1})}(t) = \#_{\mathfrak{A}}(t)$, thus t is stable w.r.t. Δ_{m_1}, and in fact w.r.t. every Δ_m with $m \geq m_1$. We proceed in this way, with m_1 in place of m_0, etc. This yields a sequence $m_0, m_1, m_2, \dots$ and for each $i \geq 0$ the set of 1-types on Σ that is stable w.r.t. $\Delta_{m_{i+1}}$ is a strict superset of the set of 1-types on Σ that is stable w.r.t. Δ_{m_i}. Since Σ is unary and the 1-types of interest all come from the fixed interpretation $\mathfrak{A}$, thus agreeing on the constant symbols, there are at most $2^{|\Sigma|}$ many 1-types to consider. Therefore, after at most $2^{|\Sigma|}$ iterations we have found an i such that Δ_{m_i} is stabilizing. Let us argue that we have achieved the claimed bound on m. Take any $i \geq 0$. Then $|\Delta_{m_i}| \leq |\phi| + 2^{|\Sigma|}m_i$. Moreover, using the bound from Point 1 of Lemma 1 and $m_i \geq 2$, we can show that $m_{i+1} \leq 2^{m_i^{|\phi|^7}}$. From this, an easy induction on i yields $m_i \leq \mathsf{tower}(i+1, |\phi|^{7i})$. Since we stop at the latest at $i = 2^{|\Sigma|}$, from this in turn we can show that $m_i \leq \mathsf{tower}(2^{|\Sigma|}+1, |\phi|^{7\cdot 2^{|\Sigma|}})$, which implies $m_i \leq \mathsf{tower}(2^{2|\Sigma|+4}, |\phi|)$. □

To conclude the proof of Proposition 2, it then suffices to apply Lemma 1 with $\Delta = \Delta_m$, where m is in Lemma 2, and taking $\mathfrak{B}$ to be the resulting Rosati cover $\mathsf{rc}(\mathfrak{A},\Delta_m)$. Since Δ_m is stabilizing, Point 3 of Proposition 2 is satisfied. For Point 1, we may use Point 1 of Lemma 1 and the fact that $|\Delta_m| \leq 2^{|\Sigma|}m + |\phi|$.

We now lift the finite model property from Proposition 2 to circumscribed consequence. To apply Proposition 2, we choose a unary signature Σ that contains the minimized and fixed predicates from the circumscription pattern used. The rest of the proof is similar to that of Theorem 1.

Theorem 4. *Circumscribed GF has the finite model property. More precisely, every satisfiable GF sentence ϕ circumscribed by $\mathsf{CP} = (\prec, \mathsf{M}, \mathsf{F}, \mathsf{V})$ has a model $\mathfrak{A}$ with $|A| \leq \mathsf{tower}(4^{|\Sigma|+4}, |\phi|)$, where $\Sigma = \mathsf{const}_=(\phi) \cup \mathsf{M} \cup \mathsf{F}$.*

Building on Theorem 4, we now obtain the following using a brute-force enumeration procedure.

Theorem 5. *Circumscribed consequence in GF is decidable and in* TOWER.

4.2 Circumscribed Querying

We prove that UCQ-querying (and thus also CQ- and AQ-querying) in GF is decidable.

Theorem 6. *Circumscribed UCQ-querying in GF is in* TOWER *w.r.t. combined complexity and in* ELEMENTARY *w.r.t. data complexity.*

Without circumscription, decidability of UCQ-querying in GF is almost immediate as one can replace the UCQ q with the disjunction q' of all acyclic CQs that imply a CQ in q (up to a certain size) and then express q' as a GF sentence, obtaining a reduction to unsatisfiability (Bárány, Gottlob, and Otto 2014). This does not work with circumscription.

Example 2. *Take the ontology $\mathcal{O}$ that consists of the sentence*

$$\forall x \left(A(x) \rightarrow \exists y (R(x,y) \wedge \exists z (R(y,z) \wedge \exists u\, (R(z,u) \wedge A(u))))\right),$$

the database $D = \{A(a)\}$, and let CP minimize A and vary all other predicates. Then $(\mathcal{O}, D) \models_{\mathsf{CP}} q$ where:

$$q := \exists x\, \exists y\, \exists z\;\; R(x,y) \wedge R(y,z) \wedge R(z,x),$$

but there is no acyclic CQ q' that implies q and satisfies $(\mathcal{O}, D) \models_{\mathsf{CP}} q'$.

We thus use a more sophisticated approach which exploits the fact that if $(\mathcal{O}, D) \not\models_{\mathsf{CP}} q$, then this is witnessed by a model $\mathfrak{B}$ that, in a certain loose sense, has the shape of a forest. More precisely, $\mathfrak{B}$ can be obtained from any model $\mathfrak{A}$ that witnesses $(\mathcal{O}, D) \not\models_{\mathsf{CP}} q$ by a version of guarded unraveling (see e.g. (Grädel and Otto 2014)) that leaves untouched a 'core' of $\mathfrak{A}$ defined as

$$\mathsf{core}_\Sigma(\mathfrak{A}) := \{a \in A \mid \#_{\mathfrak{A}}(\mathsf{tp}^1_{\mathfrak{A},\Sigma}(a)) \leq \mathsf{tower}(4^{|\Sigma|+4}, |\phi|)\}$$

for a suitable signature Σ. With 'leaving untouched', we mean that elements from this core are not duplicated during unraveling, but 'reused' whenever needed. This serves the purpose of guaranteeing minimality w.r.t. the circumscription pattern. We do not explicitly define this unraveling as this is not needed for the proofs, but we hope that this discussion guides the reader's intuition.

To prepare for the subsequent development, we give a central lemma that establishes a sufficient condition for a model $\mathfrak{B}$ to be CP-minimal, based on comparing it to a CP-minimal reference model $\mathfrak{A}$. This is a version of the 'core lemma' of (Lutz, Manière, and Nolte 2023).

Lemma 3. *Let ϕ be a GF sentence, $\mathsf{CP} = (\prec, \mathsf{M}, \mathsf{F}, \mathsf{V})$, and $\Sigma = \mathsf{const}_=(\phi) \cup \mathsf{M} \cup \mathsf{F}$. Further let $\mathfrak{A}$ be a CP-minimal model of ϕ and let $\mathfrak{B}$ be a model of ϕ such that*

1. $\mathsf{core}_\Sigma(\mathfrak{A}) \subseteq B$ *and* $c^{\mathfrak{B}} = c^{\mathfrak{A}}$ *for all* $c \in \mathsf{const}(\phi)$*;*
2. $\mathsf{tp}^1_{\mathfrak{A},\Sigma}(a) = \mathsf{tp}^1_{\mathfrak{B},\Sigma}(a)$ *for all* $a \in \mathsf{core}_\Sigma(\mathfrak{A})$*;*
3. $\mathsf{tp}^1_{\mathfrak{B},\Sigma}(B \setminus \mathsf{core}_\Sigma(\mathfrak{A})) = \mathsf{tp}^1_{\mathfrak{A},\Sigma}(A \setminus \mathsf{core}_\Sigma(\mathfrak{A}))$,

Then $\mathfrak{B}$ is a CP-minimal model of ϕ.

Intuitively, Lemma 3 says that the exact multiplicity of types realized in $\mathfrak{A}$ outside of $\mathsf{core}_\Sigma(\mathfrak{A})$ is irrelevant for CP-minimality.

Assume that we are given as an input a GF knowledge base $(\mathcal{O}, D)$, a circumscription pattern CP, and a Boolean UCQ q. We want to decide whether there is a countermodel $\mathcal{I}$ against $(\mathcal{O}, D) \models_{\mathsf{CP}} q(\bar{a})$. This may be rephrased as $\phi \models_{\mathsf{CP}} q(\bar{a})$ for

$$\phi = \bigwedge \mathcal{O} \wedge \bigwedge D \wedge \bigwedge_{c \in \mathsf{adom}(D)} \big(P_c(c) \wedge \bigwedge_{c' \in \mathsf{adom}(D)\setminus\{c\}} \neg P_{c'}(c)\big).$$

We shall use the latter formulation. We may assume that ϕ is in Scott normal form, that is, of the form $(*)$. Let $\Sigma = \mathsf{const}_=(\phi) \cup \mathsf{M} \cup \mathsf{F}$. Set $M = |\mathsf{const}(\phi)| + 2^{|\Sigma|} \cdot \mathsf{tower}(4^{|\Sigma|+4}, |\phi|) + 2^{|\Sigma|}$, and fix a set U of size M. In an outer loop, our algorithm iterates over all pairs $(\mathfrak{A}_0, T_{\overline{\mathsf{core}}})$ with $\mathfrak{A}_0$ a finite structure that interprets all constants from ϕ and $T_{\overline{\mathsf{core}}}$ a set of 1-types such that the following conditions are satisfied:

- $A_0 \subseteq U$;
- $T_{\overline{\mathsf{core}}} \subseteq \mathsf{tp}^1_\Sigma(\mathfrak{A}_0)$.

We define $\Delta := \{a \in A_0 \mid \mathsf{tp}^1_{\mathfrak{A}_0,\Sigma}(a) \notin T_{\overline{\mathsf{core}}}\}$.

For each pair $(\mathfrak{A}_0, T_{\overline{\mathsf{core}}})$, we then check whether the following additional conditions are satisfied:

(I) $\mathfrak{A}_0$ can be extended to a model $\mathfrak{A}$ of ϕ such that

(a) $\mathfrak{A}|_{A_0} = \mathfrak{A}_0$,

(b) $\mathsf{tp}^1_{\mathfrak{A},\Sigma}(A \setminus \Delta) = T_{\overline{\mathsf{core}}}$,

(c) $\mathfrak{A} \not\models q$;

(II) there exists a CP-minimal model $\mathfrak{B}$ of ϕ such that

(d) $\mathsf{core}_\Sigma(\mathfrak{B}) = \Delta$;

(e) $\mathsf{tp}^1_{\mathfrak{B},\Sigma}(a) = \mathsf{tp}^1_{\mathfrak{A}_0,\Sigma}(a)$ for all $a \in \Delta$ and

(f) $\mathsf{tp}^1_{\mathfrak{B},\Sigma}(B \setminus \Delta) = T_{\overline{\mathsf{core}}}$

We return 'yes' if all pairs fail the check and 'no' otherwise.

If the checks succeed, then the model $\mathfrak{A}$ of ϕ from Condition (I) is a countermodel against $(\mathcal{O}, D) \models_{\mathsf{CP}} q$. In particular, we may apply Lemma 3, using the model $\mathfrak{B}$ from Condition (II) as the reference model, to show that $\mathfrak{A}$ is CP-minimal. Conversely, from any countermodel $\mathfrak{A}$ against $(\mathcal{O}, D) \models_{\mathsf{CP}} q$, we can read off a pair $(\mathfrak{A}_0, T_{\overline{\mathsf{core}}})$ by choosing $T_{\overline{\mathsf{core}}} := \mathsf{tp}^1_{\mathfrak{A},\Sigma}(A \setminus \mathsf{core}_\Sigma(\mathfrak{A}))$ and $\mathfrak{A}_0$ to be the restriction of $\mathfrak{A}$ to universe

$$U_{\mathfrak{A}} := \{c^{\mathfrak{A}} \mid c \in \mathsf{const}(\phi)\} \cup \mathsf{core}_\Sigma(\mathfrak{A}) \cup \{w_t \mid t \in T_{\overline{\mathsf{core}}}\}$$

where $w_t \in A$ is chosen arbitrarily such that $\mathsf{tp}^1_{\mathfrak{A},\Sigma}(w_t) = t$. Then $\mathfrak{A}$ witnesses Condition (I) and choosing $\mathfrak{B} = \mathfrak{A}$ witnesses Condition (II).

Of course, we have to prove that Conditions (I) and (II) are decidable. For Condition (II), we prove that the following is a consequence of Theorem 1 and Lemma 3.

Lemma 4. *Let ϕ be a GF sentence, $\mathsf{CP} = (\prec, \mathsf{M}, \mathsf{F}, \mathsf{V})$, and $\Sigma = \mathsf{const}_=(\phi) \cup \mathsf{M} \cup \mathsf{F}$. Let $\mathfrak{A}$ be a CP-minimal model of ϕ. Then there exists a CP-minimal model $\mathfrak{B}$ of ϕ such that*

1. $\mathsf{core}_\Sigma(\mathfrak{B}) = \mathsf{core}_\Sigma(\mathfrak{A})$;
2. $\mathsf{tp}^1_{\mathfrak{B},\Sigma}(a) = \mathsf{tp}^1_{\mathfrak{A},\Sigma}(a)$ *for all* $a \in \mathsf{core}_\Sigma(\mathfrak{A})$;
3. $\mathsf{tp}^1_{\mathfrak{B},\Sigma}(B \setminus \mathsf{core}_\Sigma(\mathfrak{A})) = \mathsf{tp}^1_{\mathfrak{A},\Sigma}(A \setminus \mathsf{core}_\Sigma(\mathfrak{A}))$;
4. $|B| \leq 2^{|\Sigma|}(1 + \mathsf{tower}(4^{|\Sigma|+4}, |\phi|))$.

It follows that if a model as in (II) exists, then there exists one of size at most $2^{|\Sigma|}(1 + \mathsf{tower}(4^{|\Sigma|+4}, |\phi|))$ and thus we can iterate over all candidate structures $\mathfrak{B}$ up to this size, check whether $\mathfrak{B}$ is a model of ϕ that satisfies Conditions (d) to (f), and then iterate over all models $\mathfrak{B}'$ of ϕ with $B' = B$ to check that $\mathfrak{B}$ is CP-minimal.

Condition (I) requires more work. We use a mosaic approach, that is, we attempt to assemble the structure $\mathfrak{A}$ required by Condition (I) by combining small pieces called *mosaics*. Fix a pair $(\mathfrak{A}_0, T_{\overline{\mathsf{core}}})$. A mosaic for $(\mathfrak{A}_0, T_{\overline{\mathsf{core}}})$ is a decorated finite structure whose universe contains $\mathfrak{A}_0$ and possibly elements from a fixed set U^+ of $2 \cdot \mathsf{ar}$ elements where ar is the maximum arity of predicates in ϕ.

The purpose of the decoration is to trace partial homomorphisms from CQs in q through the mosaics, as follows. A *match triple* for a structure $\mathfrak{B}$ takes the form $(p, \widehat{p}, h)$ such that p is a CQ in q, $\widehat{p} \subseteq p$, and h is a partial map from $\mathsf{var}(\widehat{p})$ to B that is a homomorphism from $\widehat{p}|_{\mathsf{dom}(h)}$ to $\mathfrak{B}$ where $\widehat{p}|_{\mathsf{dom}(h)}$ denotes the restriction of $\widehat{p}$ to the variables in the domain of h. Intuitively, $\mathfrak{B}$ is a mosaic and the triple $(p, \widehat{p}, h)$ expresses that a homomorphism from $\widehat{p}$ to $\mathfrak{A}$ exists, with the variables in $\mathsf{dom}(h)$ being mapped to the current piece $\mathfrak{B}$ and the variables in $\mathsf{var}(\widehat{p}) \setminus \mathsf{dom}(h)$ mapped to other pieces of $\mathfrak{A}$. A match triple is *complete* if $\widehat{p} = p$ and *incomplete* otherwise. To make $\mathfrak{A}$ a countermodel, we must avoid complete match triples. A *specification* for a structure $\mathfrak{B}$ is a set S of match triples for $\mathfrak{B}$ and we call S *saturated* if the following conditions are satisfied:

- if p is a CQ in q, $\widehat{p} \subseteq p$, and h is a homomorphism from $\widehat{p}$ to $\mathfrak{B}$, then $(p, \widehat{p}, h) \in S$;
- if $(p, \widehat{p}, h), (p, \widehat{p}', h') \in S$ and $h(x) = h'(x)$ is defined for all $x \in \mathsf{var}(\widehat{p}) \cap \mathsf{var}(\widehat{p}')$, then $(p, \widehat{p} \cup \widehat{p}', h \cup h') \in S$.

In the following, we use the symbols introduced in $(*)$.

Definition 1. *A* mosaic *for $(\mathfrak{A}_0, T_{\overline{\mathsf{core}}})$ is a pair $M = (\mathfrak{B}, S)$ where*

- *$\mathfrak{B}$ is a finite structure such that*
  1. $B \subseteq A_0 \cup U^+$;
  2. $\mathfrak{B}|_{A_0} = \mathfrak{A}_0$;
  3. $\mathsf{tp}^1_{\mathfrak{B},\Sigma}(B \setminus \Delta) \subseteq T_{\overline{\mathsf{core}}}$;
  4. *$\mathfrak{B}$ satisfies $\forall \bar{x}\,(\alpha_i \to \varphi_i)$, for $1 \leq i \leq n_\forall$;*
- *S is a saturated specification for $\mathfrak{B}$ that contains only incomplete match triples.*

We use $\mathfrak{B}_M$ to refer to $\mathfrak{B}$ and S_M to refer to S.

Let $\mathcal{M}$ be a set of mosaics for $(\mathfrak{A}_0, T_{\overline{\mathsf{core}}})$. We say that $M \in \mathcal{M}$ is *good in $\mathcal{M}$* if for $1 \leq i \leq n_\exists$, the following condition is satisfied: if $\beta_i = R(\bar{z})$ and $\bar{a} \in R^{\mathfrak{B}}$, then we find a mosaic $M' \in \mathcal{M}$ such that

1. $\mathsf{tp}_{\mathfrak{B}_M,\Sigma}(\bar{a}) = \mathsf{tp}_{\mathfrak{B}_{M'},\Sigma}(\bar{a})$;
2. $\mathfrak{B}_{M'} \models \exists \bar{y}\,(\gamma_i \wedge \psi_i)[\bar{a}]$;
3. if $(p, \widehat{p}, h') \in S_{M'}$, then $(p, \widehat{p}, h) \in S_M$ where h is the restriction of h' to range $A_0 \cup \bar{a}$.

To verify Condition (I), we start with the set of all mosaics for $(\mathfrak{A}_0, T_{\overline{\mathsf{core}}})$ and repeatedly and exhaustively eliminate mosaics that are not good.

Lemma 5. *$\mathfrak{A}_0$ can be extended to a model $\mathfrak{A}$ of ϕ that satisfies Conditions (a) to (c) iff at least one mosaic survives the elimination process.*

At this point, we have established Theorem 6. It should be clear that the presented algorithm establishes membership in TOWER in combined complexity. For data complexity, note that the size M of the stuctures $\mathfrak{A}_0$ in pairs $(\mathfrak{A}_0, T_{\overline{\mathsf{core}}})$ is now k-exponential for a constant k: it is essentially an exponentiation tower of twos followed by $|\phi|$ whose height is independent of D (while $|\phi|$ depends linearly on $|D|$). The same is true for the bound established by Lemma 4 and the size of mosaics.

5 Lower Bounds for the Guarded Fragment

We prove lower bounds that match the upper bounds given in Section 4. Our proofs are formulated in terms of the data complexity of AQ-querying, but we also derive from them tight complexity results for circumscribed consequence.

We start with an EXP lower bound on the data complexity of AQ-querying for the restricted yet natural case where only a single predicate is minimized and no predicate is fixed. It is then of course pointless to use a preference relation in the circumscription pattern. The bound applies even for ontologies $\mathcal{O}$ that are sets of existential rules.

Theorem 7. *AQ-querying in GF is* EXP*-hard in data complexity even for ontologies that are sets of existential rules, with a single minimized predicate and no fixed predicates, and with a fixed signature.*

For UCQ-querying, the same even holds for a fixed signature in which all predicates have arity at most two.

We invite the reader to verify the proof of Theorem 7, provided in the appendix, as a warmup for the proof of the main result of this section, which is up next.

We show that, when using multiple minimized predicates as well as a preference order, then the data complexity is no longer in k-EXP for any $k \geq 1$. In other words, while for every fixed ontology $\mathcal{O}$, query q, and circumscription pattern CP querying is in k-EXP in data complexity for some k (c.f. Theorem 6), k cannot be uniformly bounded by a constant from above for all $\mathcal{O}$, q, and CP. In combined complexity, AQ-querying is even TOWER-hard.

Theorem 8. *AQ-querying in GF is*

1. TOWER*-hard in combined complexity (under logspace reductions) and*
2. k-EXP*-hard for every* $k \geq 1$ *in data complexity.*

This holds already for circumscribed sets of guarded existential rules and without fixed predicates.

We prove Point 2 as follows. It is known that, for every $\kappa \geq 1$, there is a fixed $(\kappa - 1)$-exponentially space-bounded alternating Turing machine (ATM) whose word problem is κ-EXP-hard (Chandra, Kozen, and Stockmeyer 1981). We provide a reduction from the word problem of each of these ATMs to AQ-querying in GF.[1] Our reductions are uniform across all κ and, as discussed in (Schmitz 2016), this also yields TOWER-hardness in combined complexity.

Let $\kappa \geq 1$ and let $\mathcal{M}$ be a $(\kappa - 1)$-exponentially space-bounded alternating Turing machine (ATM) whose word problem is κ-EXP-hard. We exhibit a set of existential rules $\mathcal{O}$ and a circumscription pattern CP such that given an input $e = e_1 \cdots e_n \in \Sigma^*$ to $\mathcal{M}$, we can construct in polynomial time a database D such that $\mathcal{M}$ accepts e iff $\mathcal{O}, D \models_{\mathsf{CP}} \mathsf{goal}(a)$, where goal is a unary predicate and a a dedicated constant symbol.

One main challenge is to generate a tape of the required length and we first focus on achieving that. To this end, we produce κ linear orders, with the k^{th} order being of length $\mathsf{tower}(k-1, p(n))$. In other words, the first order has length $p(n)$, the second has length $2^{p(n)}$, the third $2^{2^{p(n)}}$, and so on, until the κ^{th} order which has length $\mathsf{tower}(\kappa - 1, p(n))$ and will be used as the tape for the ATM computation. The positions in the $(k+1)^{\text{st}}$ order will be encoded in binary using elements of the k^{th} order as bit positions. For each k, the element of the k^{th} order are marked with the unary predicate ord_k. To guarantee that the encoding of a position in the $(k+1)^{\text{st}}$ order indeed only uses bit positions from the k^{th} tape, the predicates $\mathsf{ord}_1, \ldots, \mathsf{ord}_\kappa$ are minimized.

We also use other minimized predicates, arranged in a preference order as follows:

$$\mathsf{root} \prec \mathsf{err}_1 \prec \mathsf{ord}_1 \prec \cdots \prec \mathsf{err}_\kappa \prec \mathsf{ord}_\kappa \prec \mathsf{err}_{\kappa+1}.$$

The predicate err_k is used to 'report' errors in the k-th order by being made true on the constant a. This shall then make the query predicate goal true on a and in this way rule out erroneous models. The preferred minimization of err_k over ord_k acts as an incentive to avoid such errors. We use an additional predicate $\mathsf{err}_{\kappa+1}$ to detect errors in the ATM computation. To enforce that err_k is reported precisely on a, we use root and include in D

$$\mathsf{root}(a).$$

Any other predicate used is varying, which concludes the definition of CP. We now clarify how error reporting works. Since the minimization of root is preferred over that of all other predicates, in every CP-minimal model $\mathfrak{A}$ we have $\mathsf{root}^{\mathfrak{A}} = \{a^{\mathfrak{A}}\}$. When an error on the k^{th} tape is detected at some element x, we generate an instance y of err_k. We then require err_k to be subsumed by root, so that, in every CP-minimal model, y is actually a. We also require err_k to be subsumed by goal so that goal holds at $a^{\mathfrak{A}}$ whenever an error is detected in the representation of the k^{th} order. Formally, we include in $\mathcal{O}$, for every $k \in \{1, \ldots, \kappa+1\}$, the rules

$$\mathsf{err}_k(x) \to \mathsf{root}(x), \mathsf{goal}(x).$$

We do not want distinct orders to share elements and report an error if they do. We also require a not to be used as an order element. For $1 \leq i < j \leq \kappa$, add

$$\mathsf{ord}_i(x), \mathsf{ord}_j(x) \to \exists y\, \mathsf{err}_j(y)$$
$$\mathsf{ord}_k(x), \mathsf{root}(x) \to \exists y\, \mathsf{err}_k(y).$$

Elements of the first order are represented in the database D as constants c_i^1, $1 \leq i \leq p(n)$:

$$\mathsf{ord}_1(c_i^1) \text{ for } 0 \leq i < p(n).$$

The k^{th} order is represented by the binary predicate succ_k, for $1 \leq k \leq \kappa$. We use a unary predicate $\overline{\mathsf{end}}$ to mark the elements of orders that are not the final element. For all $k \in \{1, \ldots, \kappa\}$, we add the rules

$$\mathsf{succ}_k(x, x') \to \mathsf{ord}_k(x), \mathsf{ord}_k(x')$$
$$\mathsf{ord}_k(x), \overline{\mathsf{end}}(x) \to \exists x'\, \mathsf{succ}_k(x, x').$$

For the first order, we ensure the intended interpretation of succ_1 via a binary predicate $\overline{\mathsf{succ}}_1$, the following facts in D:

$$\overline{\mathsf{succ}}_1(c_i^1, c_j^1) \text{ for } 0 \leq i, j < p(n) \text{ with } j \neq i+1,$$
$$\overline{\mathsf{start}}(c_i^1) \text{ for } 0 < i < p(n)$$
$$\overline{\mathsf{end}}(c_i^1) \text{ for } 0 \leq i < p(n) - 1,$$

[1] We use ATMs for uniformity with the proof of Theorem 7. We could also work with deterministic Turing machines which, however, would only simplify the proof in a minor way.

$$\mathsf{ones}_{\triangleleft}(x,y), \overline{\mathsf{start}}(y), \mathsf{ord}_{k-1}(y) \rightarrow \exists y'\ \mathsf{ones}_{\triangleleft}(x,y'), \mathsf{succ}_{k-1}(y',y), \mathsf{bit}_{k,1}(x,y') \quad (1)$$

$$\mathsf{zeros}_{\triangleleft}(x,y), \overline{\mathsf{start}}(y), \mathsf{ord}_{k-1}(y) \rightarrow \exists y'\ \mathsf{zeros}_{\triangleleft}(x,y'), \mathsf{succ}_{k-1}(y',y), \mathsf{bit}_{k,0}(x,y') \quad (2)$$

$$\mathsf{copy}_{\triangleright}(x,x',y), \overline{\mathsf{end}}(y), \mathsf{ord}_{k-1}(y) \rightarrow \exists y'\ \mathsf{copy}_{\triangleright}(x,x',y'), \mathsf{succ}_{k-1}(y',y), \mathsf{copy}(x,x',y') \quad (3)$$

Figure 1: Additional rules used in the proof of Theorem 8 for every $k \in \{2, \dots, \kappa\}$.

and the rule

$$\mathsf{succ}_1(x,y), \overline{\mathsf{succ}}_1(x,y) \rightarrow \exists z\ \mathsf{err}_1(z).$$

Note that we also introduced a $\overline{\mathsf{start}}$ predicate, for later use.

For the k^{th} order, with $k \in \{2, \dots, \kappa\}$, the positions of elements are represented by the two binary predicates $\mathsf{bit}_{k,0}$ and $\mathsf{bit}_{k,1}$ pointing to the elements of the $(k-1)^{\text{st}}$ order, which serve as bit positions. Intuitively, $\mathsf{bit}_{k,b}(x,y)$ says that the y^{th} bit in the binary encoding of the position of element x in the k^{th} order is b. We add the following rules, for every $k \in \{2, \dots, \kappa\}$ and $b \in \{0,1\}$:

$$\begin{aligned}
\mathsf{bit}_{k,b}(x,y) &\rightarrow \mathsf{ord}_k(x), \mathsf{ord}_{k-1}(y)\\
\mathsf{bit}_{k,0}(x,y) &\rightarrow \overline{\mathsf{end}}(x)\\
\mathsf{bit}_{k,1}(x,y) &\rightarrow \overline{\mathsf{start}}(x)\\
\mathsf{bit}_{k,0}(x,y), \mathsf{bit}_{k,1}(x,y) &\rightarrow \exists z\ \mathsf{err}_k(z).
\end{aligned}$$

We need to guarantee that the encoding of positions is incremented when moving along the predicate succ_k, generally assuming that the least significant bit position is the first element in the order. We use a binary predicate fz_k (for First Zero) and the following rules, for all $k \in \{2, \dots, \kappa\}$:

$$\begin{aligned}
\mathsf{ord}_k(x), \overline{\mathsf{end}}(x) &\rightarrow \exists y\ \mathsf{fz}_k(x,y)\\
\mathsf{fz}_k(x,y) &\rightarrow \mathsf{bit}_{k,0}(x,y), \mathsf{ones}_{\triangleleft}(x,y).
\end{aligned}$$

The second rule makes sure that the position represented by y has value 0 and that all positions to the left of y have value 1. The latter is enforced by the binary predicate $\mathsf{ones}_{\triangleleft}$ which propagates to every position strictly to the left of y, enforcing a bit value of 1; see Rule 1 in Figure 1.

The following rules introduce a ternary predicate nextfz_k extending each instance of $\mathsf{succ}_k(x,x')$ to further include the position of the first zero in the encoding of x. We use nextfz_k to properly set up the bit values in the encoding of the position of x'. Add, for every $k \in \{2, \dots, \kappa\}$,

$$\begin{aligned}
\mathsf{succ}_k(x,x') &\rightarrow \exists y\ \mathsf{nextfz}_k(x,x',y), \mathsf{fz}_k(x,y)\\
\mathsf{nextfz}_k(x,x',y) &\rightarrow \mathsf{bit}_{k,1}(x',y), \mathsf{zeros}_{\triangleleft}(x',y), \mathsf{copy}_{\triangleright}(x,x',y).
\end{aligned}$$

Predicate $\mathsf{zeros}_{\triangleleft}(x',y)$ enforces that all 1 bits to the left of the first zero in the encoding of the position of x, which is at position y, are flipped to 0s in the encoding of the position of x'. The 0 in position y for x is flipped to a 1 for x'. All other positions keep their bit values thanks to predicate $\mathsf{copy}_{\triangleright}$ which instantiates a copy that, in turn, complies with the following rule for $b = 0, 1$:

$$\mathsf{copy}(x,x',y), \mathsf{bit}_{k,b}(x,y) \rightarrow \mathsf{bit}_{k,b}(x',y).$$

Details can be found as Rules 2 and 3 in Figure 1.

As explained above, the κ^{th} order has the desired length and we use its elements as positions of tape cells in the ATM computation. We show in the appendix how to encode that computation. The challenging part is to ensure that the tape symbols that are not under the head are preserved when the ATM makes a transition. This is enforced by a mechanism similar to the propagation of the predicate $\mathsf{ones}_{\triangleleft}$ above.

The (straightforward) polynomial time reduction from circumscribed AQ-querying to circumscribed consequence given in the proof of Theorem 3 also applies to GF. Thus, Theorem 8 also yields the following.

Corollary 1. *Circumscribed consequence in GF is* TOWER-*hard.*

6 FO^2 with Counting: C^2

We observe that in C^2, circumscribed consequence and circumscribed AQ-querying are decidable. This is achieved by combining a result from (Wies, Piskac, and Kuncak 2009) with ideas from (Bonatti et al. 2015a).

Recall that Presburger arithmetic is the first-order theory of the natural numbers with addition and equality. BAPA is a multisorted theory that combines Presburger arithmetic with the theory of (uninterpreted) sets and their cardinalities. We refer to (Kuncak, Nguyen, and Rinard 2006) for full details and only remark that numerical variables are denoted with x, y, z, set variables with B, and set cardinality with $|B|$.

For a structure $\mathfrak{A}$ and a 1-type t, we write $t^{\mathfrak{A}}$ to denote the set of elements $\{a \in A \mid \mathsf{tp}^1_{\mathfrak{A}}(a) = t\}$. The following was proved in (Wies, Piskac, and Kuncak 2009), making intense use of the results of (Pratt-Hartmann 2005).

Theorem 9. *Let ϕ be a C^2 sentence and let $t_1, \dots, t_n$ be the 1-types for ϕ. One can compute a formula $\chi_\phi(x_1, \dots, x_n)$ of Presburger arithmetic such that*

1. *for every model $\mathfrak{A}$ of ϕ, $\chi_\phi[|t_1^{\mathfrak{A}}|, \dots, |t_n^{\mathfrak{A}}|]$ is true;*
2. *if $\chi_\phi[k_1, \dots, k_n]$ is true, then there is a model $\mathfrak{A}$ of ϕ with $|t_i^{\mathfrak{A}}| = k_i$ for $1 \leq i \leq n$.*

The above provides a reduction from consequence in C^2 to unsatisfiability in BAPA: the C^2 consequence $\phi \models \psi$ holds iff the BAPA sentence $\exists B_1 \cdots \exists B_n\, \chi_{\phi \wedge \neg\psi}[|B_1|/x_1, \dots, |B_n|/x_n]$ is unsatisfiable. We extend this to circumscribed consequence.

Assume that we want to decide $\phi \models_{\mathsf{CP}} \psi$, with ϕ, ψ two C^2 sentences and $\mathsf{CP} = (\prec, \mathsf{M}, \mathsf{F}, \mathsf{V})$. We may assume w.l.o.g. that ϕ and ψ contain the same predicates and thus have the same 1-types. With ϑ, we denote the BAPA formula

$$\chi_\phi[|B_1|/x_1, \dots, |B_n|/x_n] \wedge \bigwedge_{A \in \mathsf{M} \cup \mathsf{F}} \big(B_A = \bigcup_{t_i \mid A(x_1) \in t_i} B_i\big)$$

and we define ϑ' to be like ϑ, but using set variables B'_i in place of B_i and B'_A in place of B_A. Let $\overline{B}$ be the tuple of set variables in ϑ and let $\overline{B}'$ be the corresponding tuple for ϑ'. We write $\overline{B}' <_{\mathsf{CP}} \overline{B}$ to denote the conjunction of

- $B_A = B'_A$ for all $A \in \mathsf{F}$;
- for all $A \in \mathsf{M}$: $B'_A \not\subseteq B_A \rightarrow \bigvee_{A' \in \mathsf{M} \mid A' \prec A} B'_{A'} \subsetneq B_{A'}$;
- $\bigvee_{A \in \mathsf{M}} \big(B'_A \subsetneq B_A \wedge \bigwedge_{A' \in \mathsf{M} \mid A' \prec A} B'_{A'} = B_{A'} \big)$.

Now let χ denote the BAPA sentence

$$\exists \overline{B} \, \big(\vartheta \wedge \chi_\psi[|B_1|/x_1, \ldots, |B_n|/x_n] \wedge \\ \forall \overline{B}' \, \big(\overline{B}' <_{\mathsf{CP}} \overline{B} \rightarrow \neg \vartheta' \big) \big).$$

It can be verified that $\phi \models_{\mathsf{CP}} \psi$ iff χ is unsatisfiable. Since satisfiability in BAPA is decidable (Feferman and Vaught 1959; Kuncak, Nguyen, and Rinard 2006), we obtain decidability of circumscribed consequence in C^2. This carries over to circumscribed AQ-querying in the same straightforward way as for FO^2.

Theorem 10. *In C^2, circumscribed consequence and circumscribed AQ-querying are decidable.*

Since BAPA is also decidable over finite models, we also obtain the version of Theorem 10 where circumscribed consequence and querying are defined w.r.t. finite models.

7 Conclusion

We have studied the impact on computational complexity of adding circumscription to decidable fragments of first-order logic, which turns out to be remarkably varied: while FO^2 is very tame and does not have higher complexity than $\mathcal{ALC}$ in its circumscribed version, GF suffers from a dramatic complexity explosion. We remark that there is a close connection between circumscription and querying with closed predicates as studied in (Ngo, Ortiz, and Simkus 2016; Lutz, Seylan, and Wolter 2019), see also Example 1. More details are in (Lutz, Manière, and Nolte 2023). As an example, Theorem 7 also applies to AQ-querying of guarded existential rules with a single unary closed predicate. This, in turn, is related to results in (Benedikt et al. 2016).

Several interesting questions remain open. What is the exact complexity of circumscribed consequence in C^2? We speculate that by making careful use of the techniques in (Pratt-Hartmann 2005), one can bring it down to $\text{coNExp}^{\text{NP}}$. What is the complexity of circumscribed consequence in GF with only a single minimized predicate or with multiple such predicates but no preference order? Is circumscribed UCQ-querying in GF finitely controllable? What is the complexity of circumscribed querying with less expressive classes of existential rules such as inclusion dependencies? Is circumscribed consequence decidable in the unary / guarded negation fragments of FO? Note that satisfiability in the latter fragment is known to be reducible to UCQ-querying in GF (Bárány, ten Cate, and Segoufin 2015), but that this reduction relies on arguments based on treeifications of some subformulas of interest, a technique that cannot be applied in presence of circumscription as discussed with Example 2.

Acknowledgments

The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany and by the Sächsische Staatsministerium für Wissenschaft Kultur und Tourismus in the program Center of Excellence for AI-research "Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig", project identification number: ScaDS.AI

This work is partly supported by BMBF (Federal Ministry of Education and Research) in DAAD project 57616814 (SECAI, School of Embedded Composite AI) as part of the program Konrad Zuse Schools of Excellence in Artificial Intelligence.

References

Andréka, H.; Németi, I.; and van Benthem, J. 1998. Modal languages and bounded fragments of predicate logic. *J. Philos. Log.* 27(3):217–274.

Baader, F., and Hollunder, B. 1995. Embedding defaults into terminological knowledge representation formalisms. *J. Autom. Reason.* 14(1):149–180.

Bárány, V.; Gottlob, G.; and Otto, M. 2014. Querying the guarded fragment. *J. of Log. Methods Comput. Sci.* 10(2).

Bárány, V.; ten Cate, B.; and Segoufin, L. 2015. Guarded negation. *J. ACM* 62(3):22:1–22:26.

Benedikt, M.; Bourhis, P.; ten Cate, B.; and Puppis, G. 2016. Querying visible and invisible information. In *Proc. of LICS*, 297–306. ACM.

Bonatti, P. A.; Faella, M.; Lutz, C.; Sauro, L.; and Wolter, F. 2015a. Decidability of circumscribed description logics revisited. In *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation - Essays Dedicated to Gerhard Brewka on the Occasion of His 60th Birthday*, volume 9060 of *LNCS*, 112–124. Springer.

Bonatti, P. A.; Faella, M.; Petrova, I. M.; and Sauro, L. 2015b. A new semantics for overriding in description logics. *Artif. Intell.* 222:1–48.

Bonatti, P. A.; Lutz, C.; and Wolter, F. 2009. The complexity of circumscription in description logic. *J. Artif. Intell. Res.* 35:717–773.

Calì, A.; Gottlob, G.; and Lukasiewicz, T. 2009. A general datalog-based framework for tractable query answering over ontologies. In *Proc. of PODS*, 77–86. ACM.

Chandra, A. K.; Kozen, D. C.; and Stockmeyer, L. J. 1981. Alternation. *J. ACM* 28(1):114–133.

Donini, F. M.; Nardi, D.; and Rosati, R. 2002. Description logics of minimal knowledge and negation as failure. *ACM Trans. Comput. Log.* 3(2):177–225.

Feferman, S., and Vaught, R. 1959. The first order properties of products of algebraic systems. *Fundamenta Mathematicae* 47(1):57–103.

Giordano, L.; Gliozzi, V.; Olivetti, N.; and Pozzato, G. L. 2013. A non-monotonic description logic for reasoning about typicality. *Artif. Intell.* 195:165–202.

Grädel, E., and Otto, M. 2014. The freedoms of (guarded) bisimulation. In *Johan van Benthem on Logic and Information Dynamics*. Springer. 3–31.

Grädel, E.; Kolaitis, P. G.; and Vardi, M. Y. 1997. On the decision problem for two-variable first-order logic. *Bull. of Symb. Log.* 3(1):53–69.

Grädel, E.; Otto, M.; and Rosen, E. 1997. Two-variable logic with counting is decidable. In *Proc. of LICS*, 306–317. IEEE Computer Society.

Grädel, E. 1999. On the restraining power of guards. *J. Symb. Log.* 64(4):1719–1742.

Kuncak, V.; Nguyen, H. H.; and Rinard, M. C. 2006. Deciding boolean algebra with presburger arithmetic. *J. Autom. Reason.* 36(3):213–239.

Lutz, C., and Manière, Q. 2024. Adding circumscription to decidable fragments of first-order logic: A complexity rollercoaster. arXiv:2407.20822 [cs.AI].

Lutz, C.; Manière, Q.; and Nolte, R. 2023. Querying circumscribed description logic knowledge bases. In *Proc. of KR*, 482–491.

Lutz, C.; Seylan, I.; and Wolter, F. 2019. The data complexity of ontology-mediated queries with closed predicates. *Log. Methods Comput. Sci.* 15(3).

Mortimer, M. 1975. On languages with two variables. *Math. Log. Q.* 21(1):135–140.

Ngo, N.; Ortiz, M.; and Simkus, M. 2016. Closed predicates in description logics: Results on combined complexity. In *Proc. of KR*, 237–246. AAAI Press.

Pacholski, L.; Szwast, W.; and Tendera, L. 1997. Complexity of two-variable logic with counting. In *Proc. of LICS*, 318–327. IEEE Computer Society.

Pratt-Hartmann, I. 2005. Complexity of the two-variable fragment with counting quantifiers. *J. Log. Lang. Inf.* 14(3):369–395.

Rosati, R. 2006. On the decidability and finite controllability of query processing in databases with incomplete information. In *Proc. of PODS*, 356–365. ACM.

Schmitz, S. 2016. Complexity hierarchies beyond elementary. *ACM Trans. Comput. Theory* 8(1).

Scott, D. 1962. A decision method for validity of sentences in two variables. *J. of Symb. Log.* 27.

Stefano, F. D.; Ortiz, M.; and Simkus, M. 2023. Description logics with pointwise circumscription. In *Proc. of IJCAI*, 3167–3175. ijcai.org.

Wies, T.; Piskac, R.; and Kuncak, V. 2009. Combining theories with shared set operations. In Ghilardi, S., and Sebastiani, R., eds., *Proc. of FroCoS*, volume 5749 of *LNCS*, 366–382. Springer.