E. Grädel, P. Kolaitis, L. Libkin, M. Marx,
M. Vardi, S. Weinstein

Finite Model Theory and its Applications

This document contains Leonid Libkin's chapter *Embedded Finite Models and Constraint Databases*

April 17, 2006

Springer

Berlin Heidelberg New York
Hong Kong London
Milan Paris Tokyo

Contents

1

Embedded Finite Models and Constraint Databases

1.1 Introduction

The goal of this chapter is to answer two questions:

1. How does one store an *infinite* set in a database?
2. And what does it have to do with *finite* model theory?

Clearly, one cannot store an infinite set, but instead one can store a *finite representation* of an infinite set and write queries as if the entire infinite set were stored. This is the key idea behind *constraint databases*, which emerged relatively recently as a very active area of database research. The primary motivation comes from geographical and temporal databases: how does one store a region in a database? More importantly, how does one design a query language that makes the user view a region as if it were an infinite collection of points stored in the database?

Finite representations used in constraint databases are first-order formulae; in geographical applications, one often uses Boolean combinations of linear or polynomial inequalities. One of the most challenging questions in the development of the theory of constraint databases was that of the expressive power: what are the limitations of query languages for constraint databases? These questions were easily reduced to those on the expressiveness of query languages over ordinary *finite* relational databases, with the additional condition that databases may store numbers and arithmetic operations may be used in queries. This is exactly the setting of *embedded finite model theory.*

It turned out that the classical techniques for analyzing the expressive power of relational query languages no longer work in this new setting. In the past several years, however, most questions on the expressive power have been settled, by using new techniques that mix the finite and the infinite, and bring together results from a number of fields such as model theory, algebraic geometry and symbolic computation.

In this chapter we present a variety of results on embedded finite models and constraint databases. The core part of this chapter deals with new techniques for analyzing expressive power in the mixed setting. These techniques, that come in the form of *collapse results*, reduce many questions over constraint databases or embedded finite models to the classical finite model theory setting.

Organization

In Section 1.2, we describe the setting of embedded finite models, and explain connections with relational database theory. Section 1.3 contains a brief introduction into constraint databases.

Section 1.4 gives an overview of collapse results; it also defines different semantics of logical formulae, and introduces the notion of genericity. Sections 1.5 and 1.6 describe collapse results for different semantics and different notions of genericity. In Section 1.7 we look into connections between collapse results and various model-theoretic notions, and in Section 1.8 we describe a close relationship between collapse results and the notion of VC dimension, which is of interest in model theory and machine learning. Section 1.9 presents results on the expressive power of query languages over constraint databases that use two different techniques: reduction to the case of embedded finite models, and the analysis of topological structure of constraint databases.

Sections 1.10 and 1.11 deal with topics motivated by database considerations. Section 1.10 studies query safety, which means guaranteeing finite output for relational databases, and some geometric properties for constraint databases. Section 1.11 briefly analyzes the problems of aggregate operators and higher-order features in constraint databases.

1.2 Relational Databases and Embedded Finite Models

In classical finite model theory, we work with finite structures and deal with sentences like

$$\exists x \exists y \forall z (\neg E(z,x) \vee \neg E(z,y))$$

saying that the diameter of an (undirected) graph with edge-set E is at least 3. In embedded finite model theory, we still work with finite structures but deal with sentences like

$$\exists x \exists y \ (E(x,y) \wedge (y = x \cdot x + 1))$$

saying that there is an edge (x,y) in a graph with $y = x^2 + 1$. It is assumed here that the nodes of a graph come from some domain that is equipped with arithmetic operations such as addition and multiplication; for example, the nodes could be natural, rational, or real numbers.

To illustrate the difference, consider as an example a relational signature of directed graphs, consisting of a single edge-predicate E. Suppose we want to find the composition of E with itself; that is, find pairs (a, b) in a directed graph that are connected by a path of length at most 2. This is done by writing a formula

$$\varphi(x, y) \;\equiv\; \exists z\ (E(x, z) \wedge E(z, y)).$$

This formula gives us a *conjunctive query*; it can be written in a variety of relational database languages: as

$$q(x, y) \text{ :- } E(x, z), E(z, y)$$

in datalog, or

$$\pi_{\#1,\#4}\ (\sigma_{\#2=\#3}\ (R \times R)\ )$$

in relational algebra, or

```
SELECT R1.Source, R2.Destination
FROM R R1, R R2
WHERE R1.Destination=R2.Source
```

in SQL.

Now suppose that the nodes of the graph are natural numbers, and we are only willing to consider paths $E(x, z), E(z, y)$ in which x, y, z are related by some condition: for example, $x + y = z$. It is straightforward to rewrite the above query in first-order logic as

$$\varphi'(x, y) \;\equiv\; \exists z\ (E(x, z) \wedge E(z, y) \wedge (x + y = z)),$$

or in SQL:

```
SELECT R1.Source, R2.Destination
FROM R R1, R R2
WHERE R1.Destination=R2.Source
      AND R1.Source + R2.Destination = R2.Source
```

But what about relational algebra? The most natural way seems to be:

$$\pi_{\#1,\#4}\ (\sigma_{(\#2=\#3)\wedge(\#1+\#4=\#2)}\ (R \times R)\ );$$

however, relational algebra does not allow arithmetic operations in its selection predicates.

At the first glance, this is easy to remedy: just add arithmetic predicates to the selection conditions. While this seems to be easy, there appear to be two serious problems.

Expressive Power We know that first-order logic, and thus relational algebra, cannot express most recursive and counting queries, such as the transitive closure of a relation, or the parity of a set. However, this was proved under the assumption that only equality and order comparisons are allowed on nodes of graphs. How does one prove the analogous result (if it is true) if nodes are numbers, and arithmetic operations are used in formulae?

It appears that the standard techniques for proving expressivity bounds are not directly applicable in this case. Tools based on locality cannot tell us anything meaningful due to the presence of order; 0-1 laws are inapplicable altogether, and games become unmanageable as the duplicator must maintain partial isomorphism not only for the graph edges, but also for all the arithmetic predicates as well. It thus seems that entirely different techniques are needed to solve the problem of the expressive power in this setting.

Query Evaluation It is clear that the query φ' above can be evaluated by the usual bottom-up technique: we first construct $R \times R$, then select all the tuples (a, b, c, d) with $b = c$ and $a + d = b$, and then project out the first and the last components. However, what if the condition is not $x + y = z$ but z being a perfect square? The query will then be rewritten as

$$\varphi''(x, y) \;\equiv\; \exists z \; (E(x, z) \wedge E(z, y) \wedge (\exists u \; (z = u \cdot u))),$$

and the selection condition will have to evaluate $\exists u \; (z = u \cdot u)$ with u ranging over the infinite set of natural numbers! In this particular case, it appears that the evaluation is possible: one does not have to check all $u \in \mathbb{N}$, but only $u \leq z$. However, one can have more complex conditions, for example: $\exists x_1 \ldots \exists x_k \; p(x_1, \ldots, x_k) = 0$, where p is some polynomial with integer coefficients. The truth value of this sentence cannot be determined algorithmically, as this would imply solving Hilbert's 10th problem. Thus, it is not always possible to evaluate queries with arithmetic conditions. In general, one would encounter this problem dealing with any undecidable theory.

To give another example of potential problems with query evaluation, consider the following query $\psi(x)$ saying that x^2 belongs to S:

$$\exists y \; S(y) \wedge (x \cdot x = y).$$

This query is clearly evaluable, but its output depends on whether one works with real numbers, or integers, for example: over the reals, the output is $\{-\sqrt{a}, \sqrt{a} \mid a \in S\}$, but over the integers one has to select integers from this set. Thus, the output is different depending on the range of quantifier $\exists y$: whether it is $\mathbb{R}$ or $\mathbb{Z}$. Also, it is is not immediately clear how a query processor can look at the query above, and transform the declarative specification involving a quantifier over an infinite set into a finite evaluable query like $\{-\sqrt{a}, \sqrt{a} \mid a \in S\}$.

To deal with these problems, we now have to give a formal definition of the setting. Intuitively, we deal with finite relational structures whose elements come from some interpreted domain with some interpreted operations. Formally, the object of our study is the following:

Definition 1.1. *Let $\mathfrak{M} = \langle U, \Omega \rangle$ be an infinite structure on a set U, where the signature Ω contains some function, predicate, and constant symbols. Let SC be a relational signature $\{R_1, \ldots, R_l\}$ where each relation symbol R_i has arity $p_i > 0$. Then an* embedded finite model *(that is, an SC-structure embedded into $\mathfrak{M}$) is a structure*

$$D \quad = \quad \langle A, R_1^D, \ldots, R_l^D \rangle,$$

where each R_i^D is a finite *subset of U^{p_i}, and A is the union of all elements that occur in the relations $R_1^D, \ldots, R_l^D$. The set A is called the* active domain *of D, and is denoted by $adom(D)$.* □

Examples of structures $\mathfrak{M}$ that will be used most often are real and natural numbers with various arithmetic operations, for example, $\langle \mathbb{N}, +, \cdot \rangle$, the real ordered field $\langle \mathbb{R}, +, \cdot, 0, 1, <, \rangle$ or the real ordered group $\langle \mathbb{R}, +, -, 0, 1, < \rangle$.

The notation SC comes from the database name *schema* for the relational vocabulary of a finite structure.

In the setting where we mix finite and infinite structures, first-order logic (FO) must be defined carefully. Note that we have two different universes that can be quantified over: the universe U of the infinite structure $\mathfrak{M}$, and the active domain A of the finite structure D.

Definition 1.2. *Given a structure $\mathfrak{M} = \langle U, \Omega \rangle$ and a relational signature SC,* first-order logic (FO) over $\mathfrak{M}$ and SC*, denoted by* FO($SC, \mathfrak{M}$)*, is defined as follows:*

- *Any atomic FO formula in the language of $\mathfrak{M}$ is an atomic* FO($SC, \mathfrak{M}$) *formula. For any p-ary symbol R from SC and terms $t_1, \ldots, t_p$ in the language of $\mathfrak{M}$, $R(t_1, \ldots, t_p)$ is an atomic* FO($SC, \mathfrak{M}$) *formula.*
- *Formulae of* FO($SC, \mathfrak{M}$) *are closed under the Boolean connectives ($\vee$, $\wedge$, and $\neg$).*
- *If φ is a* FO($SC, \mathfrak{M}$) *formula, then the following:*

$$\exists x\ \varphi \qquad \forall x\ \varphi \qquad \exists x \in adom\ \varphi \qquad \forall x \in adom\ \varphi$$

are FO($SC, \mathfrak{M}$) *formulae.*

The class of first-order formulae in the language of $\mathfrak{M}$ will be denoted by FO($\mathfrak{M}$) (that is, the formulae built up from atomic $\mathfrak{M}$-formulae by Boolean connectives and quantification $\exists, \forall$). The class of formulae not using the symbols from Ω will be denoted by FO(SC) (in this case all four quantifiers are allowed).

The notions of free and bound variables are standard. For the semantics, given a $\mathrm{FO}(SC, \mathfrak{M})$ formula $\varphi(x_1, \ldots, x_n)$, and $\vec{a} = (a_1, \ldots, a_n) \in U^n$, we define the relation $(\mathfrak{M}, D) \models \varphi(\vec{a})$. When $\mathfrak{M}$ is understood, we usually write just $D \models \varphi(\vec{a})$. The notion of satisfaction is standard, with only the case of quantification requiring explanation. Let $\varphi(x, \vec{y})$ be a formula, and $\vec{b}$ be a tuple of elements of U, of the same length as $\vec{y}$. Then:

$$
\begin{array}{rcl}
(\mathfrak{M}, D) \models \exists x\ \varphi(x, \vec{b}) & \Leftrightarrow & (\mathfrak{M}, D) \models \varphi(a, \vec{b}) \text{ for some } a \in U \\
(\mathfrak{M}, D) \models \forall x\ \varphi(x, \vec{b}) & \Leftrightarrow & (\mathfrak{M}, D) \models \varphi(a, \vec{b}) \text{ for all } a \in U \\
(\mathfrak{M}, D) \models \exists x \in adom\ \varphi(x, \vec{b}) & \Leftrightarrow & (\mathfrak{M}, D) \models \varphi(a, \vec{b}) \text{ for some } a \in adom(D) \\
(\mathfrak{M}, D) \models \forall x \in adom\ \varphi(x, \vec{b}) & \Leftrightarrow & (\mathfrak{M}, D) \models \varphi(a, \vec{b}) \text{ for all } a \in adom(D).
\end{array}
$$

The quantifiers $\exists x \in adom\ \varphi$ and $\forall x \in adom\ \varphi$ are called *active-domain* quantifiers. Note that they are definable with the unrestricted quantifiers $\exists$ and $\forall$, as $adom(D)$ is definable by a FO formula. However, we find it more convenient to have them explicitly in the syntax so that we can use both restricted and unrestricted quantifiers in the same formula.

Definition 1.3. *By* $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ *we denote the fragment of* $\mathrm{FO}(SC, \mathfrak{M})$ *that only uses quantifiers* $\exists x \in adom$ *and* $\forall x \in adom$*. Formulae in this fragment are called the* active-domain semantics formulae*.*

Sometimes we shall also refer to the standard interpretation of the unrestricted quantifiers $\exists$ and $\forall$ as the *natural* semantics of first-order formulae, and to the class $\mathrm{FO}(SC, \mathfrak{M})$ as the class of *natural semantics formulae.*

Our goal is to study $\mathrm{FO}(SC, \mathfrak{M})$. In particular, we show that the solutions to the crucial problems of expressive power and query evaluation depend heavily on the model-theoretic properties of $\mathfrak{M}$. In fact, we shall see the full range of expressivity – from all computable properties to just $\mathrm{FO}_{\mathrm{act}}(SC)$-definable – for different structures $\mathfrak{M}$. Of course it is highly undesirable to have a query language that expresses all computable queries, since in the database setting we want to keep the complexity low, and we want queries to be optimizable. The latter situation is much more attractive, since essentially one deals with the familiar relational calculus on finite databases.

1.3 Constraint Databases

The field of constraint databases (CDB) was initiated in 1990, and since then has become a well-established topic in the database field. It grew out of the research on Datalog and Constraint Logic Programming (CLP). The original motivation was to combine work in these two areas, with the goal of obtaining a database-style, optimizable version of constraint logic programming. The key idea was that the notion of a tuple in a relational database could be replaced by a conjunction of constraints from an appropriate language (for example, linear arithmetic constraints), and that many of the features of the relational model could be extended in an appropriate way. In particular, standard query languages such as those based on first-order logic and Datalog could be extended, at least in principle, to such a model.

The primary motivation for constraint databases comes from the field of spatial and spatio-temporal databases, and geographical information systems (GIS). One wants to store an infinite set – say, a region on the plane – in a database and query it *as if* all the points (infinitely many) were stored. This is clearly impossible. However, it is possible to store a *finite representation* of an infinite set, and make this completely transparent to the user, who still can access the data as though infinitely many points were stored.

To illustrate how infinite geometric objects can be represented with different classes of constraints, we use the following examples:

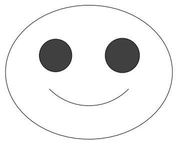

Fig. 1.1. An example of two-variable polynomial constraints

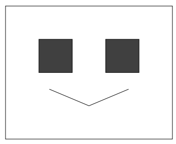

Fig. 1.2. An example of two-variable linear arithmetic constraints

Consider Figure 1.1. This figure can be described, using polynomial inequalities with integer coefficients:

$$\begin{aligned}&(x^2/25 + y^2/16 = 1) \vee (x^2 + 4x + y^2 - 2y \leq 4)\\ &\quad \vee (x^2 - 4x + y^2 - 2y \leq -4) \vee (x^2 + y^2 - 2y = 8 \wedge y < -1)\ .\end{aligned}$$

The first equality describes the outer ellipse of the figure, the second and third disjuncts describe the "eyes", and the last disjunct describes the "mouth".

If we restrict ourselves to inequalities involving linear functions, the face in Figure 1.1 can no longer be defined. It can, however, be approximated as follows:

$$\begin{aligned}&(-5 \leq x \leq 5 \wedge y = -4) \vee (-5 \leq x \leq 5 \wedge y = 4)\\ &\quad \vee (x = 5 \wedge -4 \leq y \leq 4) \vee (x = -5 \wedge -4 \leq y \leq 4)\\ &\quad \vee (-3 \leq x \leq -1 \wedge 0 \leq y \leq 2) \vee (1 \leq x \leq 3 \wedge 0 \leq y \leq 2)\\ &\quad \vee (3y = -x - 6 \wedge -2 \leq y \leq -1) \vee (3y = x - 6 \wedge -2 \leq y \leq -1)\ .\end{aligned}$$

The first four disjuncts describe the outer rectangle. The next two disjuncts describe the "eyes", with the last two describing the "mouth".

What makes the sets depicted in Figures 1.1 and 1.2 special is that they are *definable* by FO formulae over some structures, in this case, the real field and the real ordered group.

Definition 1.4. *Given a structure* $\mathfrak{M} = \langle U, \Omega \rangle$*, a set* $X \subseteq U^n$ *is called* $\mathfrak{M}$-definable *(or definable over* $\mathfrak{M}$*, or just* definable *if* $\mathfrak{M}$ *is understood) if there exists a* FO *formula* $\varphi(x_1, \ldots, x_n)$ *in the language of* $\mathfrak{M}$ *such that*

$$X \quad = \quad \{(a_1, \ldots, a_n) \in U^n \ \mid \ \mathfrak{M} \models \varphi(a_1, \ldots, a_n)\}.$$

We now consider two classes of definable sets especially relevant in the context of constraint databases.

Definition 1.5. *We use abbreviations* $\mathbf{R}$ *for the real field (that is,* $\langle \mathbb{R}, +, \cdot, 0, 1, < \rangle$*) and* $\mathbf{R}_{\text{lin}}$ *for the real ordered group (*$\langle \mathbb{R}, +, -, 0, 1, < \rangle$*). Sets definable over* $\mathbf{R}$ *are called* semi-algebraic *and sets definable over* $\mathbf{R}_{\text{lin}}$ *are called* semi-linear*.*

A remarkable property of both $\mathbf{R}_{\text{lin}}$ and $\mathbf{R}$ is that they admit *quantifier-elimination*; that is, every formula is equivalent to a quantifier-free one. For $\mathbf{R}_{\text{lin}}$ this is a simple consequence of Fourier-Motzkin elimination; for $\mathbf{R}$, this is the celebrated result of Tarski's.

Thus, every semi-algebraic set in $\mathbb{R}^n$ is a Boolean combination of sets given by polynomial equalities and inequalities of the form

$$p(x_1, \ldots, x_n) \ \ \{=, >, <\} \ \ 0,$$

where p is a polynomial (with rational or integer coefficients). Similarly, a semi-linear set in $\mathbb{R}^n$ is a Boolean combination of sets given by linear equalities and inequalities of the form

$$a_1 \cdot x_1 + \ldots + a_n \cdot x_n \ \ \{=, >, <\} \ \ b,$$

where the a_is and b are rational or integer coefficients. That is, a semi-linear set is a Boolean combination of half-spaces and hyperplanes in $\mathbb{R}^n$.

The set shown in Figure 1.1 is semi-algebraic, and the set shown in Figure 1.2 is semi-linear. In general, the majority of geographical applications represent regions by linear constraints; that is, regions are semi-linear sets. If linear constraints are not sufficient, one can use polynomial constraints instead.

We are now ready to present a mathematical model of constraint databases.

Definition 1.6. *Let* $\mathfrak{M} = \langle U, \Omega \rangle$ *be an infinite structure on a set* U*, and let* SC *be a relational signature* $\{R_1, \ldots, R_l\}$ *where each relation* R_i *has arity* $p_i > 0$*. Then a* constraint database *of schema* SC *is a tuple*

$$\mathbf{D} \quad = \quad \langle R_1^{\mathbf{D}}, \ldots, R_l^{\mathbf{D}} \rangle,$$

where each $R_i^{\mathbf{D}}$ *is a* definable *subset of* U^{p_i}. *The superscript* $\mathbf{D}$ *is omitted if it is clear from the context.* □

Thus, the only difference between the definition of a constraint database and an embedded finite model is that in the former we interpret the SC-predicates by definable sets, and in the latter – by finite sets.

The definition of $\mathrm{FO}(SC, \mathfrak{M})$ is the same for constraint databases as it is for embedded finite models, except that we do not use the restricted quantification $\exists x \in adom$ and $\forall x \in adom$. The quantifiers are thus interpreted as ranging over the entire infinite set U. As linear and polynomial constraints play a special role in the theory of constraint databases, we introduce a special notation for them.

Definition 1.7. *If* $\mathfrak{M}$ *is the real field, we write* $\mathrm{FO} + \textsc{Poly}(SC)$ *for* $\mathrm{FO}(SC, \mathbf{R})$, *or just* $\mathrm{FO} + \textsc{Poly}$ *if* SC *is clear from the context. If* $\mathfrak{M}$ *is the real ordered group, we write* $\mathrm{FO} + \textsc{Lin}(SC)$ *(or just* $\mathrm{FO} + \textsc{Lin}$*) for* $\mathrm{FO}(SC, \mathbf{R}_{\mathrm{lin}})$.

The notation $\mathrm{FO} + \textsc{Poly}$ stands for FO with polynomial constraints, and $\mathrm{FO} + \textsc{Lin}$ for FO with linear constraints. An example of definability in $\mathrm{FO} + \textsc{Poly}$ is the property that all points in a relation S lie on a common circle: $\exists a \exists b \exists r\ (\forall x \forall y\ S(x, y) \rightarrow (x - a)^2 + (y - b)^2 = r^2)$. In general, $\mathrm{FO} + \textsc{Poly}$ can define many useful topological concepts such as closure, interior and boundary. These are definable in $\mathrm{FO} + \textsc{Lin}$ as well. For example, the $\mathrm{FO} + \textsc{Lin}$ query $\alpha(x, y)$:

$$\forall \epsilon > 0 \exists x' \exists y' \big(S(x', y') \wedge (x - \epsilon < x' < x + \epsilon) \wedge (y - \epsilon < y' < y + \epsilon)\big)$$

tests if the pair (x, y) is in the closure of a set $S \subseteq \mathbb{R}^2$.

In $\mathrm{FO} + \textsc{Poly}$ one can also define the convex hull of a set. To see how this is done in the two-dimensional case, assume that a semi-algebraic set $S \in \mathbb{R}^2$ is given. Then $\varphi(x, y)$ given by

$$\exists x_1, y_1, x_2, y_2, x_3, y_3 \quad \exists \lambda_1, \lambda_2, \lambda_3 \quad \left(\begin{array}{l} \quad S(x_1, y_1) \wedge S(x_2, y_2) \wedge S(x_3, y_3) \\ \wedge\ \lambda_1 \geq 0 \wedge \lambda_2 \geq 0 \wedge \lambda_3 \geq 0 \\ \wedge\ \lambda_1 + \lambda_2 + \lambda_3 = 1 \\ \wedge\ (x = \lambda_1 \cdot x_1 + \lambda_2 \cdot x_2 + \lambda_3 \cdot x_3) \\ \wedge\ (y = \lambda_1 \cdot y_1 + \lambda_2 \cdot y_2 + \lambda_3 \cdot y_3) \end{array} \right)$$

is true on (x, y) iff $(x, y) \in \mathrm{conv}(S)$. In general, to definite the convex hull of a set S in $\mathbb{R}^n$, one uses Carathéodory's theorem stating that $\vec{x}$ is in the convex hull of $S \subseteq \mathbb{R}^n$ iff $\vec{x}$ is in the convex hull of some $n + 1$ points in S, and codes this by an FO formula just as we did above for the case of $\mathbb{R}^2$.

We note again that these examples demonstrate the crucial property of constraint databases: query languages based on FO view the database as if it were infinitely many tuples stored in memory. We refer to the database relations in exactly the same way we do for the usual relational databases.

Now that we defined constraint databases and saw some examples of querying, we consider the same issues we addressed in the context of embedded finite models: expressive power and query evaluation.

Expressive power We saw that FO + POLY is a rather expressive language to talk about properties of semi-algebraic sets, and that many topological properties of semi-linear sets can already be expressed in the weaker language FO + LIN. We next turn to a very basic topological property: *connectivity.* Suppose we are given a semi-algebraic or semi-linear set S, and we want to test if it is topologically connected. Can we do this in FO + POLY or FO + LIN?

At first, it seems that the answer is "no." Indeed, it appears that topological connectivity is rather close to graph connectivity: take an undirected graph G and embed it in $\mathbb{R}^3$ without self-intersections. Then the embedding is topologically connected iff G is a connected graph. However, we only know that FO cannot express graph connectivity; there is nothing yet that tells us that similar bounds exist for FO + LIN and FO + POLY.

Query Evaluation Suppose we are given an FO($SC, \mathfrak{M}$) query $\varphi(\vec{x})$ and a constraint database $\mathbf{D}$ over $\mathfrak{M}$. How does one evaluate φ on $\mathbf{D}$? The answer to this is very simple – one just puts the definition of relations in $\mathbf{D}$ into φ. For example, if $\varphi(x) \equiv \exists y\ (S(x,y) \wedge (p_1(x,y) > 0))$ and S is given by $p_2(x,y) < 0$, where p_1, p_2 are polynomials, then by putting the definition of S into φ we obtain a new formula $\varphi^{\mathbf{D}}(x) \equiv \exists y\ ((p_2(x,y) < 0) \wedge (p_1(x,y) > 0))$. As this is an FO formula, it gives us a constraint database.

This may look a little bit like cheating, and of course it is. For example, how does one check that $\mathbf{D} \models \varphi(1)$? To do so, one must be able to check if $\varphi^{\mathbf{D}}(1)$ is true in $\mathbf{R}$; in general, one must be able to check if $\varphi^{\mathbf{D}}(\vec{a})$ is true in a given structure $\mathfrak{M}$, where $\varphi^{\mathbf{D}}$ is the result of substituting definitions of relations in SC in the query φ.

This can only be done if the FO theory of the underlying structure $\mathfrak{M}$ is *decidable.* This property certainly holds for $\mathbf{R}_{\text{lin}}$ and $\mathbf{R}$ (in fact, they satisfy a much stronger property of having quantifier-elimination); however, for many structures this property does not hold (for example, $\langle \mathbb{N}, +, \cdot \rangle$).

We shall see in the remainder of this chapter that the correspondence between the problems of topological connectivity of constraint databases and graph connectivity in the embedded setting is not an accident: in fact, the majority of expressivity bounds for constraint databases are obtained by rather simple reductions to embedded finite models.

1.4 Collapse and Genericity: An Overview

The next five sections will deal primarily with the setting of embedded finite models. In this short section, we give an overview of the main results.

Many results on expressive power use the notion of *genericity* which comes from the classical relational database setting. Informally, this notion is sometimes stated as *data independence principle*: when one evaluates queries on relational databases, exact values of elements stored in a database are not important. For example, the answer to the query: "Does a graph have diameter 2?" is the same for the graph $\{(1,2),(1,3),(1,4)\}$ and for the graph $\{(a,b),(a,c),(a,d)\}$, which is obtained by the mapping $1 \mapsto a, 2 \mapsto b, 3 \mapsto c, 4 \mapsto d$.

In general, generic queries commute with permutations of the domain. Queries expressible in $\mathrm{FO}(SC, \mathfrak{M})$ need not be generic: for example, the query given by $\exists x\ S(x) \wedge x > 1$ is true on $S = \{2\}$ but false on $S = \{0\}$. However, as all queries definable in standard relational languages – relational calculus, Datalog, etc. – are generic, to reduce questions about $\mathrm{FO}(SC, \mathfrak{M})$ to those in ordinary finite-model theory, it suffices to restrict one's attention to generic queries.

We now define genericity of *Boolean queries* (which are just classes of SC-structures) and *non-Boolean queries* (which map a finite SC-structure to a finite subset of U^m, $m > 0$). We also define genericity in the ordered as well as unordered setting. The reason for considering the ordered setting separately is twofold: first, most structures of interest in applications are ordered, and second, in several proofs we need to introduce the order relation to obtain the desired results.

Given a function $\pi : U \to U$, we extend it to finite SC-structures D by replacing each occurrence of $a \in adom(D)$ with $\pi(a)$.

Definition 1.8.

- *A Boolean query Q is* totally generic *(*order-generic*) if for every partial injective function (partial monotone injective function, resp.) π defined on $adom(D)$, $Q(D) = Q(\pi(D))$.*
- *A non-Boolean query Q is* totally generic *(*order-generic*) if for every partial injective function (partial monotone injective function, resp.) π defined on $adom(D) \cup adom(Q(D))$, $\pi(Q(D)) = Q(\pi(D))$.*

Order-genericity of course assumes that U is linearly ordered. Clearly, total genericity is stronger than order-genericity. Examples of totally generic queries are all queries definable in relational algebra, Datalog, the *While* language, and in fact in almost every language studied in relational database theory. As a concrete example, consider the parity query. Since for any injective $\pi : U \to U$ it is the case that $card(X) = card(\pi(X))$, parity is totally generic.

Examples of order-generic queries include queries definable in relational calculus and datalog with order (that is, order comparisons are allowed in selection predicates and Datalog rules).

Approaches to Proving Expressivity Bounds

How can one prove bounds on $\mathrm{FO}(SC, \mathfrak{M})$? Probably, by reducing the problem to something we know about. And we know a lot about FO over finite structures, ordered or unordered. In our terms, this is either $\mathrm{FO}_{\mathrm{act}}(SC, \langle U, \emptyset \rangle)$, which we denote by $\mathrm{FO}_{\mathrm{act}}(SC)$ (that is, there are no operations on U, and everything is restricted to the active domain), or $\mathrm{FO}_{\mathrm{act}}(SC, \langle U, < \rangle)$, which will be denoted by $\mathrm{FO}_{\mathrm{act}}(SC, <)$ (that is, the only predicate on U is the order $<$).

To reduce the expressivity of $\mathrm{FO}(SC, \mathfrak{M})$ to $\mathrm{FO}_{\mathrm{act}}(SC, <)$ or $\mathrm{FO}_{\mathrm{act}}(SC)$, we have to deal with two problems: unrestricted quantification over U, and the presence of $\mathfrak{M}$-definable constraints in formulae. Figure 1.3 illustrates possible approaches to the problem.

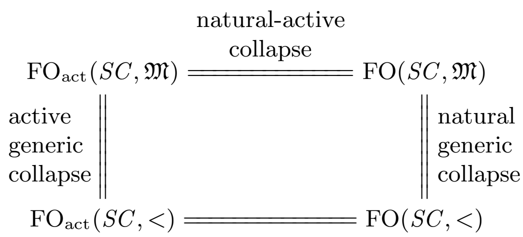

Fig. 1.3. Approaches to proving bounds for $\mathrm{FO}(SC, \mathfrak{M})$

We need to go from the upper right corner to the lower left corner. One possibility is to move left first and then down. To move left, we must prove that for a given $\mathfrak{M}$, $\mathrm{FO}(SC, \mathfrak{M})$ and $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ have the same power. That is, all unrestricted quantification can be eliminated. This will be called *natural-active collapse*. To move down, we would have liked to prove $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M}) = \mathrm{FO}_{\mathrm{act}}(SC, <)$, but this is impossible due to the following.

Lemma 1.9. $\mathrm{FO}_{\mathrm{act}}(SC, <)$ *only defines order-generic queries.* □

On the other hand, queries definable in $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ need not be generic. Thus, we attempt to prove the next best thing: that all generic queries in $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ and $\mathrm{FO}_{\mathrm{act}}(SC, <)$ are the same. This is called *active generic collapse*.

Another possibility is to go from the right upper corner down first. For the same reasons as before, we have to restrict ourselves to generic queries, and attempt to prove that any generic query in $\mathrm{FO}(SC, \mathfrak{M})$ is definable in $\mathrm{FO}(SC, <)$. This is called *natural generic collapse*. Then, to go left, we have to prove the natural-active collapse over a very simple structure $\langle U, < \rangle$.

Let us now summarize the definitions of collapse results we will be proving here.

Definition 1.10. *We say that a structure* $\mathfrak{M}$ *admits:*

- natural-active collapse *if* $\mathrm{FO}(SC, \mathfrak{M}) = \mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ *for any* SC*;*
- active generic collapse *if, for any* SC*, the classes of order-generic queries in* $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ *and* $\mathrm{FO}_{\mathrm{act}}(SC, <)$ *are the same (assuming* $\mathfrak{M}$ *is ordered);*
- natural generic collapse *if, for any* SC*, the classes of order-generic queries in* $\mathrm{FO}(SC, \mathfrak{M})$ *and* $\mathrm{FO}(SC, <)$ *are the same (assuming* $\mathfrak{M}$ *is ordered).*

We shall also consider collapse results for totally generic queries, but they will of lesser importance. The next three sections deal with collapse results: Section 1.5 discusses the active generic collapse, Section 1.6 the natural-active collapse, and Section 1.6.7 the natural generic collapse.

1.5 Active-Generic Collapse

Our goal is to prove the active generic collapse over *any* ordered structure. We do it by proving a Ramsey property, defined below, and then showing that it implies the collapse.

We start with a simple example that illustrates the main idea of the proof. Suppose we have a sentence Φ of $\mathrm{FO} + \textsc{Poly}$:

$$\forall x \in adom\ \forall y \in adom\ S(x, y) \rightarrow (\neg(x = y^2) \wedge \neg(y = x^2)).$$

In general, given a sentence, one cannot decide whether it defines a generic query. So assume for a moment that a given sentence happens to express a generic query. How does one show then that this query is definable in FO without polynomial constraints (for example, how does one prove that this query is not parity)? Clearly, one needs a systematic way of finding counterexamples for each non-FO query. This is provided by the following observation. Let $X = \{3^{3^i} \mid i > 0\} \subset \mathbb{N}$. Then, for any $x, y \in X$, we have $x \neq y^2$, because $3^j = 2 \cdot 3^i$ does not hold for any $i, j > 0$. Thus, if $adom(S) \subset X$, then $S \models \Phi$. Now, assume that Φ expresses a generic query Q. Given *any* finite relation S, we can find a monotone embedding π of its active domain into X. Thus, $Q(S) = Q(\pi(S))$ by genericity, and we know that $Q(\pi(S))$ is true. Hence, $Q(S)$ is true for all S, and thus Φ cannot express a non first-order generic query.

This is the basic idea behind the proof of the active generic collapse: we first show that for each formula, its behavior on some infinite set is described by a first-order formula. This is called the Ramsey property. We then show how genericity and the Ramsey property imply the collapse.

1.5.1 Ramsey properties

Definition 1.11. *Let* $\mathfrak{M} = \langle U, \Omega \rangle$ *be an ordered structure. We say that a* $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ *formula* $\varphi(\vec{x})$ *has the* Ramsey property *if the following is true:*

> *Let* X *be an infinite subset of* U*. Then there exists an infinite set* $Y \subseteq X$ *and a* $\mathrm{FO}_{\mathrm{act}}(SC, <)$ *formula* $\psi(\vec{x})$ *such that for any instance* D *of* SC *with* $adom(D) \subset Y$*, and for any* $\vec{a}$ *over* Y*, it is the case that* $D \models \varphi(\vec{a}) \leftrightarrow \psi(\vec{a})$*.*

We speak of the total *Ramsey property if* ψ *is an* $\mathrm{FO}_{\mathrm{act}}$ *formula in the language of* SC *(note the absence of order).*

In the rest of this section, we prove the Ramsey property. Fix an ordered structure $\mathfrak{M} = \langle U, \Omega \rangle$ and a schema SC. The following simple lemma will often be used as a first step in proofs of collapse results. Before stating it, note that for any $\mathrm{FO}(SC, \mathfrak{M})$, subformulae $(x = y)$ can be viewed as both atomic $\mathrm{FO}(SC)$ and atomic $\mathrm{FO}(\mathfrak{M})$ formulae. For the rest of the chapter, we choose to view them as atomic $\mathrm{FO}(\mathfrak{M})$ formulae; that is, atomic $\mathrm{FO}(SC)$ are only those of the form $R(\cdots)$ for $R \in SC$.

Lemma 1.12. *Let* $\varphi(\vec{x})$ *be an* $\mathrm{FO}(SC, \mathfrak{M})$ *formula. Then there exists an equivalent formula* $\psi(\vec{x})$ *such that every atomic subformula of* ψ *is either an* $\mathrm{FO}(SC)$ *formula, or an* $\mathrm{FO}(\mathfrak{M})$ *formula. Furthermore, it can be assumed that none of the variables* $\vec{x}$ *occurs in an* $\mathrm{FO}(SC)$*-atomic subformula of* $\psi(\vec{x})$*. If* φ *is an* $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ *formula, then* ψ *is also an* $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ *formula.*

Proof. Introduce m fresh variables $z_1, \ldots, z_m$, where m is the maximal arity of a relation in SC, and replace any atomic formula of the form $R(t_1(\vec{y}), \ldots, t_l(\vec{y}))$, where $l \leq m$ and the t_is are $\mathfrak{M}$-terms, by $\exists z_1 \in adom \ldots \exists z_l \in adom \ \bigwedge_i (z_i = t_i(\vec{y})) \wedge R(z_1, \ldots, z_l)$. Similarly use existential quantifiers to eliminate $\vec{x}$-variables from $\mathrm{FO}(SC)$-atomic formulae. □

The key in the inductive proof of the Ramsey property is the case of $\mathrm{FO}(\mathfrak{M})$-subformulae. For this, we first recall the infinite version of Ramsey's theorem, in the form most convenient for our purposes.

Theorem 1.13 (Ramsey). *Given an infinite ordered set* X*, and any partition of the set of all ordered* m*-tuples* $x_1 < \ldots < x_m$ *of elements of* X *into* l *classes* $A_1, \ldots, A_l$*, there exists an infinite subset* $Y \subseteq X$ *such that all ordered* m*-tuples of elements of* Y *belong to the same class* A_i*.* □

Lemma 1.14. *Let* $\varphi(\vec{x})$ *be an* $\mathrm{FO}(\mathfrak{M})$*-formula. Then* φ *has the Ramsey property.*

Proof. Consider a (finite) enumeration of all the ways in which the variables $\vec{x}$ may appear in the order of $\mathcal{U}$. For example, if $\vec{x} = (x_1, \ldots, x_4)$, one possibility is $x_1 = x_3, x_2 = x_4$ and $x_1 < x_2$. Let P be such an arrangement, and $\zeta(P)$ a first-order formula that defines it ($x_1 = x_3 \wedge x_2 = x_4 \wedge x_1 < x_3$ in the above example). Note that there are finitely many such arrangements P; let $\mathcal{P}$ be the set of all of those. Each P induces an equivalence relation on $\vec{x}$, for example, $\{(x_1, x_3), (x_2, x_4)\}$ for P above. Let $\vec{x}^P$ be a subtuple of $\vec{x}$ containing a representative for each class (e.g., (x_1, x_4)) and let $\varphi^P(\vec{x}^P)$ be obtained from φ by replacing all variables from an equivalence class by the chosen representative. Then $\varphi(x)$ is equivalent to

$$\bigvee_{P \in \mathcal{P}} \zeta(P) \wedge \varphi^P(\vec{x}^P) \; .$$

We now show the following. Let $\mathcal{P}' \subseteq \mathcal{P}$ and $P_0 \in \mathcal{P}'$. Let $X \subseteq U$ be an infinite set. Assume that $\psi(\vec{x})$ is given by

$$\bigvee_{P \in \mathcal{P}'} \zeta(P) \wedge \varphi^P(\vec{x}^P).$$

Then there exists an infinite set $Y \subseteq X$ and a quantifier-free FO($<$) formula $\gamma_{P_0}(\vec{x})$ such that ψ is equivalent to

$$\gamma_{P_0}(\vec{x}) \;\vee \bigvee_{P \in \mathcal{P}' - \{P_0\}} \zeta(P) \wedge \varphi^P(\vec{x}^P)$$

for tuples $\vec{x}$ of elements of Y.

To see this, suppose that P_0 has m equivalence classes. Consider a partition of tuples of X^m ordered according to P_0 into two classes: A_1 of those tuples for which $\varphi^{P_0}(\vec{x}^{P_0})$ is true, and A_2 of those for which $\varphi^{P_0}(\vec{x}^{P_0})$ is false. By Ramsey's theorem, for some infinite set $Y \subseteq X$ either all ordered tuples over Y^m are in A_1, or all are in A_2. In the first case, ψ is equivalent to $\zeta(P_0) \vee \bigvee_{P \in \mathcal{P}' - \{P_0\}} \zeta(P) \wedge \varphi^P(\vec{x}^P)$, and in the second case ψ is equivalent to $\neg\zeta(P_0) \vee \bigvee_{P \in \mathcal{P}' - \{P_0\}} \zeta(P) \wedge \varphi^P(\vec{x}^P)$, proving the claim.

The lemma now follows by applying this claim inductively to every partition $P \in \mathcal{P}$, passing to smaller infinite sets, while getting rid of all the formulae containing symbols other $=$ and $<$. At the end we have an infinite set over which φ is equivalent to a quantifier-free FO($<$) formula. □

Now a simple inductive argument proves:

Proposition 1.15. *Let $\mathfrak{M}$ be any ordered structure. Then every $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ formula has the Ramsey property.*

Proof. By Lemma 1.12, we assume that every atomic subformula is an $\mathrm{FO}_{\mathrm{act}}(SC)$ formula or an $\mathrm{FO}(\mathfrak{M})$ formula. The base cases for the induction are those of $\mathrm{FO}_{\mathrm{act}}(SC)$ formulae, where there is no need to change the formula or find a subset, and of $\mathrm{FO}(\mathfrak{M})$ atomic formulae, which is given by Lemma 1.14.

Let $\varphi(\vec{x}) = \varphi_1(\vec{x}) \wedge \varphi_2(\vec{x})$, and $X \subseteq \mathcal{U}$ infinite. First, find ψ_1, $Y_1 \subseteq X$ such that for any D and $\vec{a}$ over Y_1, $D \models \varphi_1(\vec{a}) \leftrightarrow \psi_1(\vec{a})$. Next, by using the hypothesis for φ_2 and Y_1, find an infinite $Y_2 \subseteq Y_1$ such that for any D and $\vec{a}$ over Y_2, $D \models \varphi_2(\vec{a}) \leftrightarrow \psi_2(\vec{a})$. Then take $\psi = \psi_1 \wedge \psi_2$ and $Y = Y_2$.

The case of $\varphi = \neg\varphi'$ is trivial.

For the existential case, let $\varphi(\vec{x}) = \exists y \in adom\ \varphi_1(y, \vec{x})$. By the hypothesis, find $Y \subseteq X$ and $\psi_1(y, \vec{x})$ such that for any D and $\vec{a}$ over Y and any $b \in Y$ we have $D \models \varphi_1(b, \vec{a}) \leftrightarrow \psi_1(b, \vec{a})$. Let $\psi(\vec{x}) = \exists y \in adom.\psi_1(y, \vec{x})$. Then, for any D and $\vec{a}$ over Y, $D \models \psi(\vec{a})$ iff $D \models \psi_1(b, \vec{a})$ for some $b \in adom(D)$ iff $D \models \varphi_1(b, \vec{a})$ for some $b \in adom(D)$ iff $D \models \varphi_1(\vec{a})$, thus finishing the proof.
□

It is clear from the proof of Proposition 1.15 that only the case of atomic $\mathrm{FO}(\mathfrak{M})$ formulae requires the introduction of the order relation. Thus, if atomic $\mathrm{FO}(\mathfrak{M})$ formulae had the total Ramsey property over $\mathfrak{M}$, so would all $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ formulae. In general, this cannot be guaranteed for arbitrary $\mathfrak{M}$ (consider, for example, $\langle U, < \rangle$). However, there is an important class of structures on the reals for which this statement can be shown.

We say that $\mathfrak{M} = \langle \mathbb{R}, \Omega \rangle$ is *analytic* if Ω consists of real-analytic functions. For example, $\langle \mathbb{R}, +, \cdot \rangle$ is analytic.

Lemma 1.16. *Let $\mathcal{F} = \{f_i(\vec{x})\}_{i \in I}$ be a countable family of real-analytic functions, where $\vec{x} = (x_1, \ldots, x_l)$. Assume that none of the functions in $\mathcal{F}$ is identically zero. Let $X \subseteq \mathbb{R}$ be a set of cardinality of the continuum. Then there is a set $Y \subseteq X$ of cardinality of the continuum such that for any tuple $\vec{c}$ of l distinct elements of Y, none of $f_i(\vec{c}), i \in I$, equals zero.*

The proof of this result, which we omit here, is a Zorn's lemma argument based on the fact that a non-zero real analytic function can have at most countably many zeros. □

Proposition 1.17. *Let $\mathfrak{M} = \langle \mathbb{R}, \Omega \rangle$ be analytic. Then every* $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ *formula has the total Ramsey property.*

Proof sketch. We only need to modify the proof of Lemma 1.14, to show the total Ramsey property of atomic $\mathrm{FO}(\mathfrak{M})$ formulae. This can be done by using Lemma 1.16 in place of Ramsey's theorem. □

1.5.2 Collapse results

We now show how the Ramsey property implies the active generic collapse. Recall (see Section 1.4) that an m-ary query, $m > 0$, is a mapping from finite SC-structures on U to finite subsets of U^m. We start with the following observation.

Lemma 1.18. *If Q is an order-generic query on SC-structures over an infinite set U, then $adom(Q(D)) \subseteq adom(D)$ for every SC-structure D.*

Proof. First note that for any finite subsets $Y \subset X$ of an infinite ordered set U, any $x \in X - Y$, and any number $n > 0$, we can find monotone injective maps $\pi_1, \ldots, \pi_n$ defined on X such that for all i, j, $\pi_i(Y) = \pi_j(Y)$, but all $\pi_1(x), \ldots, \pi_n(x)$ are distinct. This is true because U has either an infinitely descending or an infinitely ascending chain; in each case it is easy to construct the π_is.

Now suppose that $Z = adom(Q(D)) - adom(D)$ is nonempty for an order-generic query Q. Let $X = adom(Q(D)) \cup adom(D)$, $Y = adom(D)$ and $n = card(Z) + 1$. Construct $\pi_1, \ldots, \pi_n$ as above. Now for any i, j: $\pi_i(Q(D)) = Q(\pi_i(D)) = Q(\pi_j(D)) = \pi_j(Q(D))$; hence $\pi_1(Z) = \ldots = \pi_n(Z)$. In particular, for every $x \in Z$, $\pi_i(x) \in \pi_1(Z)$, whence $card(\pi_1(Z)) = card(Z) \geq n$. This contradiction proves the lemma. □

Lemma 1.19. *Assume that every* $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ *formula has the Ramsey property. Then* $\mathfrak{M}$ *admits the active generic collapse.*

Proof. Let Q be an order-generic query definable in $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$. By the Ramsey property, we find an infinite $X \subseteq U$ and an $\mathrm{FO}_{\mathrm{act}}(SC, <)$-definable Q' that coincides with Q on X. We claim they coincide everywhere. Let D be a SC-structure. Since X is infinite, there exists a partial monotone injective map π from $adom(D)$ into X. Since Q' is $\mathrm{FO}_{\mathrm{act}}(SC, <)$-definable, it is order-generic, and thus Q and Q' do not extend active domains. Hence, $\pi(Q(D)) = Q(\pi(D)) = Q'(\pi(D)) = \pi(Q'(D))$ from which $Q(D) = Q'(D)$ follows. □

We now put Proposition 1.15 and Lemma 1.19 together:

Theorem 1.20. *Every ordered structure admits the active generic collapse.* □

Thus, no matter what functions and predicates in $\mathfrak{M}$, first-order logic cannot express more generic active-semantics queries over it than just $\mathrm{FO}_{\mathrm{act}}(SC, <)$. In particular, we have the following.

Corollary 1.21. *Let* $\mathfrak{M}$ *be an arbitrary structure. Then queries such as parity, majority, connectivity, transitive closure, and acyclicity are not definable in* $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$.

Proof. Assume otherwise, and extend $\mathfrak{M}$ to $\mathfrak{M}^<$ by adding the symbol $<$ to be interpreted as a linear order. Then $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M}^<)$ defines one of those queries, for appropriate SC. Since all the queries listed above are order-generic, we obtain from Theorem 1.20 that $\mathrm{FO}_{\mathrm{act}}(SC, <)$ defines them, which is not the case. □

We conclude by showing a stronger collapse result over analytic structures.

Corollary 1.22. *If* $\mathfrak{M} = \langle \mathbb{R}, \Omega \rangle$ *is analytic, then any totally generic query definable in* $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ *is definable in* $\mathrm{FO}_{\mathrm{act}}(SC)$. □

Indeed, this is a stronger version of collapse, as there exist totally generic queries in $\mathrm{FO}_{\mathrm{act}}(SC, <) - \mathrm{FO}_{\mathrm{act}}(SC)$ (even for very simple vocabularies SC).

1.6 Natural-Active Collapse

So far we have dealt with formulae that only use the restricted quantification $\forall x \in adom$ and $\exists x \in adom$. We next move to unrestricted quantification, where quantifiers are allowed to range over the infinite universe of a structure $\mathfrak{M}$. Our ultimate goal is to prove the natural-active collapse: $\text{FO}(SC, \mathfrak{M}) = \text{FO}_{\text{act}}(SC, \mathfrak{M})$. We start by showing that there is a reason to believe that this may hold for some structures $\mathfrak{M}$, although not for all of them. We then review some notions from model theory that help us distinguish good structures (for which the collapse holds) from bad ones (for which it does not). After that, we give a gentle introduction to the main ideas of the proof of the natural-active collapse, considering a simple case of linear constraints (that is, $\text{FO} + \text{Lin}$) and one unrestricted existential quantifier to eliminate. After that, we present a general proof and an algorithm, and revisit the collapse for generic queries.

1.6.1 Collapse: failure and success

We have seen that the active generic collapse holds for *every* ordered structure Does this extend to the natural-active collapse? To give a negative answer, consider the structure $\mathfrak{N} = \langle \mathbb{N}, +, \cdot \rangle$. (We may include an order relation $<$ as well, but it is definable: $x < y$ iff $\neg(x = y) \wedge \exists z\ (y = x + z)$.) Let SC consist of a single unary predicate S. From the active generic collapse, we know that parity is not definable in $\text{FO}_{\text{act}}(SC, \mathfrak{N})$. However,

Proposition 1.23. *Parity is definable in* $\text{FO}(SC, \mathfrak{N})$. *Consequently,* $\mathfrak{N}$ *does not admit the natural-active collapse.*

Proof. Let $p_1, p_2, \ldots$ enumerate the prime numbers. Consider three predicates on $\mathbb{N}$: $P_0(x)$ holds iff x is prime, $P_1(x, y)$ holds iff y equals p_x, and $P_2(x)$ holds iff x is the product of an even number of distinct primes. Note that P_0, P_1 and P_2 are recursive, and thus definable over $\mathfrak{N}$. The way of expressing parity is then the following: given a set $S = \{x_1, \ldots, x_n\}$ with $x_1 < \ldots < x_n$, we code it as $c_S = p_{x_1} \cdot \ldots \cdot p_{x_n}$. Suppose we have a formula $\varphi(c)$ which holds iff $c = c_S$. Then parity is expressed as

$$\neg\exists x S(x) \ \vee \ \exists c\ (\varphi(c) \wedge P_2(c)).$$

Thus, it remains to show how to express φ. It can be defined by the following formula:

$$\forall p\ P_0(p) \to \begin{pmatrix} (\exists y (c = p \cdot y)) \to \neg\exists y (c = p \cdot p \cdot y) \\ \wedge\ (\exists y (c = p \cdot y)) \leftrightarrow \exists x\ (S(x) \wedge P_1(x, p)) \end{pmatrix}$$

It says that for every prime p that divides c, c is not divisible by p^2, and p is of the form p_x for some $x \in S$, which forces c to be c_S. This completes the proof. $\square$

One may observe that there is nothing specific for parity in the proof above. In particular, the coding scheme can be easily extended to finite SC-structures for any SC, and the fact that every recursive predicate on $\mathbb{N}$ is definable in $\mathfrak{N}$ allows us to state:

Proposition 1.24. *For any SC, every computable property of finite SC-structures is definable in* $\mathrm{FO}(SC, \mathfrak{N})$. □

In fact, $\mathrm{FO}(SC, \mathfrak{N})$ can even express properties that are *not* computable.

Thus, we have witnessed a rather dramatic failure of the natural-active collapse. Is there then something that gives us hope of recovering it for some structures? Let us first look at the simplest possible $\mathfrak{M}$: $\langle U, \emptyset\rangle$. It turns out that in this case the collapse can be proven rather easily.

Theorem 1.25. *For every schema,* $\mathrm{FO}(SC) = \mathrm{FO}_{\mathrm{act}}(SC)$.

Proof. We consider the case of nonempty finite structures. If an $\mathrm{FO}_{\mathrm{act}}(SC)$ formula $\psi(\vec{x})$ equivalent to an $\mathrm{FO}(SC)$ formula $\varphi(\vec{x})$ is found in this case, then for arbitrary finite SC-structures, a formula equivalent to φ is given by $(\exists x \in \mathit{adom}(x = x) \wedge \psi(\vec{x})) \vee (\neg\exists x \in \mathit{adom}(x = x) \wedge \varphi_{\emptyset}(\vec{x}))$, where $\varphi_{\emptyset}(\vec{x})$ is a quantifier-free formula equivalent to the formula obtained from φ by replacing each occurrence of a predicate from SC by *false*.

Now the proof is by induction on the structure of the formula. The cases of atomic formulae and Boolean connectives are obvious. For the existential case, we define a transformation $[\gamma]^x$ that eliminates all free occurrences of variable x from quantifier-free formulae:

- If γ is $(x = x)$, then $[\gamma]^x = \mathit{true}$;
- If γ is $(x = y)$ or $R(\ldots, x, \ldots)$, then $[\gamma]^x = \mathit{false}$;
- If γ is any other atomic formula, then $[\gamma]^x = \gamma$;
- If $\gamma = \gamma_1 \vee \gamma_2$, then $[\gamma]^x = [\gamma_1]^x \vee [\gamma_2]^x$;
- If $\gamma = \neg\gamma'$, then $[\gamma]^x = \neg[\gamma']^x$;

Let $\varphi(\vec{z}) = \exists x \alpha(x, \vec{z})$ where $z = (z_1, \ldots, z_n)$. By the hypothesis, α is equivalent to an $\mathrm{FO}_{\mathrm{act}}(SC)$ formula $\alpha'(x, \vec{z})$. Assume without loss of generality that α' is of the form $\mathbf{Q}y_1 \in \mathit{adom} \ldots \mathbf{Q}y_m \in \mathit{adom}\ \beta(x, \vec{y}, \vec{z})$, where β is quantifier-free.

Define $\varphi_0(\vec{z}) \equiv \exists x \in \mathit{adom}\ \alpha'(x, \vec{z})$, $\varphi_i(\vec{z}) \equiv \alpha'(z_i, \vec{z})$ and $\varphi_\infty(\vec{z}) \equiv \mathbf{Q}y_1 \in \mathit{adom} \ldots \mathbf{Q}y_m \in \mathit{adom}\ [\beta(x, \vec{y}, \vec{z})]^x$. Let

$$\varphi'(\vec{z}) \;\equiv\; \varphi_0 \;\vee\; \Big(\bigvee_{i=1}^{n} \varphi_i\Big) \;\vee\; \varphi_\infty.$$

We now show that $D \models \varphi(\vec{a}) \leftrightarrow \varphi'(\vec{a})$ for every nonempty D and every $\vec{a} \in \mathcal{U}^n$. First note that for every $\vec{b} \in \mathit{adom}(D)^m$, the following three statements are equivalent: (i) $D \models [\beta(x, \vec{b}, \vec{a})]^x$; (ii) for some $c \notin \mathit{adom}(D)$ and not in $\vec{a}$, $D \models \beta(c, \vec{b}, \vec{a})$, (iii) for all $c \notin \mathit{adom}(D)$ and not in $\vec{a}$, $D \models \beta(c, \vec{b}, \vec{a})$. Indeed, these equivalences hold for atomic formulae, and they are preserved under Boolean connectives.

Since all quantified variables y_i range over the active domain, we then obtain that $D \models \varphi_\infty(\vec{a})$ iff for some $c \notin \mathit{adom}(D)$ and not in $\vec{a}$, $D \models \alpha'(c, \vec{a})$. This implies the required equivalence $D \models \varphi(\vec{a}) \leftrightarrow \varphi'(\vec{a})$. □

Thus, the natural-active collapse is a meaningful concept: there are structures that admit it. On the other hand, we know that there are restrictions on structures that admit the collapse. We next discuss such restrictions.

1.6.2 Good structures vs. bad structures: o-minimality

We start with a minimal requirement a structure $\mathfrak{M}$ must satisfy to admit the natural-active collapse. Suppose we have an FO($\mathfrak{M}$) formula, that is, a formula that does not use symbols from SC. What does it mean for it to be equivalent to an $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ formula? In the absence of a finite structure, this means being equivalent to a quantifier-free FO($\mathfrak{M}$) formula. Thus, to admit the collapse, a structure $\mathfrak{M}$ must admit *quantifier-elimination*: that is, for every formula $\varphi(\vec{x})$ of FO($\mathfrak{M}$), there is a quantifier-free FO($\mathfrak{M}$) formula $\psi(\vec{x})$ such that $\mathfrak{M} \models \forall \vec{x}\ \psi(\vec{x}) \leftrightarrow \varphi(\vec{x})$.

Classical model theory provides us with many examples of such structures; some of them were mentioned already in the introduction, a few are listed below.

- $\langle U, < \rangle$ where $<$ is a dense order without endpoints on U.
- $\langle \mathbb{R}, +, -, 0, 1, < \rangle$ – this is a consequence of Fourier elimination.
- $\langle \mathbb{R}, +, \cdot, 0, 1, < \rangle$ – this is, of course, Tarski's classical result on quantifier-elimination for real closed fields.
- $\langle \mathbb{N}, +, <, 0, 1, (\equiv_k)_{k>0} \rangle$ where $x \equiv_k y$ iff $x = y(\mathrm{mod} k)$ – this is Presburger arithmetic.

However, quantifier-elimination alone is not sufficient to guarantee the collapse. Indeed, any structure $\mathfrak{M}$ admits a definitional expansion to some $\mathfrak{M}'$ that has quantifier-elimination (simply by adding new symbols for all definable predicates). Thus, if we take such an expansion $\mathfrak{N}'$ of $\mathfrak{N} = \langle \mathbb{N}, +, \cdot \rangle$, we still have that all computable properties of finite SC-structures are definable in FO($SC, \mathfrak{N}'$), but $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{N}')$ cannot define parity.

To impose additional restrictions, we consider the model-theoretic notion of o-minimality. An ordered structure $\mathfrak{M} = \langle U, \Omega \rangle$ is *o-minimal* if every definable set is a finite union of points and open intervals. Here, definable sets are those of the form $\{x \in U \mid \mathfrak{M} \models \varphi(x)\}$ where φ is a first-order formula in the language of Ω and constants for elements of U.

An interval is given by its *endpoints*, a and b, and it is either an open interval $(a, b) = \{c \mid a < c < b\}$, or closed $[a, b] = \{c \mid a \leq c \leq b\}$, or one of the half-open half-closed versions $[a, b)$ or $(a, b]$; by considering $+\infty$ and $-\infty$ as endpoints, we also have unbounded versions of the above: $\{c \mid c < b\}$, $\{c \mid c \leq b\}$, $\{c \mid c > a\}$, $\{c \mid c \geq a\}$. Also, an equivalent definition of o-minimality is that every definable set is a finite union of intervals.

Let us list some important examples of o-minimal structures.

- $\langle \mathbb{Q}, <, (q)_{q \in \mathbb{Q}} \rangle$ is o-minimal. Indeed, every first-order formula $\varphi(x)$ is equivalent to a quantifier-free one, which is then a Boolean combination of finitely many formulae of the form $x = q$ or $x < q$. Let $q_1 < \ldots < q_k$ be the finite set of all constants that occur in such formulae. Consider then intervals $(-\infty, q_1), \{q_1\}, (q_1, q_2), \{q_2\}, \ldots, \{q_k\}, (q_k, \infty)$. It is clear that the set defined by φ is a union of some of those.
- A more complex example is that of the real field: $\langle \mathbb{R}, +, \cdot, 0, 1, < \rangle$. Consider a formula $\varphi(x)$. Since the real field has quantifier-elimination, $\varphi(x)$ is equivalent to a Boolean combination of formulae of the form $p(x) > 0$, where p is a polynomial with real coefficients. Consider all such polynomials which are not identically zero, and let $q_1 < \ldots < q_k$ be the finite set of all the roots of these polynomials (each can have only finitely many). We thus again obtain that the set defined by $\varphi(x)$ is a union of some intervals among $(-\infty, q_1), \{q_1\}, (q_1, q_2), \{q_2\}, \ldots, \{q_k\}, (q_k, \infty)$, as no polynomial used in the representation of $\varphi(x)$ can change sign on such an interval.
- The same quantifier-elimination argument shows that the real ordered group $\langle \mathbb{R}, +, -, 0, 1, < \rangle$ is o-minimal.
- There are other interesting examples of o-minimal structures, where proving o-minimality is very hard. The most notable one is that of the exponential field: $\langle \mathbb{R}, +, \cdot, e^x \rangle$. Others include the expansion of the real field with the Gamma-function, or restricted analytic functions.

We shall present more properties of o-minimal structures before proving the natural-active collapse in Section 1.6.5.

1.6.3 Collapse theorem and corollaries

Our goal now is to show the following.

Theorem 1.26 (Natural-Active Collapse). *Let $\mathfrak{M} = \langle U, \Omega \rangle$ be an o-minimal structure that admits quantifier elimination. Then it admits the natural-active collapse.*

Furthermore, if the theory of $\mathfrak{M}$ is decidable and the quantifier elimination procedure is effective, then there is an algorithm that for every $\mathrm{FO}(SC, \mathfrak{M})$ *formula constructs an equivalent* $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ *formula.* □

The proof of this theorem will be presented in Section 1.6.5, after we present the main ideas in the simpler case of linear constraints, that is, $\mathfrak{M}$ being $\langle \mathbb{R}, +, -, 0, 1, < \rangle$.

We first state some corollaries of this result. Since the real field and the real-ordered group are o-minimal and admit quantifier elimination, we conclude that they also admit the natural-active collapse.

Corollary 1.27. *Every natural-semantics* FO + LIN *(*FO + POLY*) formula is equivalent to an active-domain semantics* FO + LIN *(*FO + POLY*, resp.) formula.* □

Combining this with the active generic collapse, we obtain:

Corollary 1.28. *Let Q be an order-generic query expressible in* FO + POLY *or* FO + LIN*. Then Q is expressible in* $\text{FO}_{\text{act}}(SC, <)$*. In particular, queries such as parity, majority, connectivity, transitive closure, and acyclicity are not definable in* FO + POLY*.* □

Thus, the expressive power of FO + POLY and FO + LIN is remarkably constrained – they cannot express more generic queries than FO queries over ordered finite structures, despite the fact that they possess great expressive power for nongeneric queries, as we saw in Section 1.3.

Before we present the proof, we give a simple example of a transformation from $\text{FO}(SC, \mathfrak{M})$ to $\text{FO}_{\text{act}}(SC, \mathfrak{M})$. Let SC contain one binary predicate S, and $\mathfrak{M}$ be the real field (that is, we deal with FO + POLY). Consider the sentence

$$\Phi \equiv \exists a \exists b \forall x \forall y \ (S(x, y) \to a \cdot x + b = y)$$

saying that S lies on a line. Note that this can be reformulated as follows: S lies on a line iff every triple of elements of S is collinear. Given three points $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ in $\mathbb{R}^2$, there is a *quantifier-free* FO + POLY formula $\chi(x_1, x_2, x_3, y_1, y_2, y_3)$ testing if these points are collinear. Indeed, such points are collinear iff either $x_1 = x_2 = x_3$, or $y_1 = y_2 = y_3$, or two points coincide, or, in the case when all three points are different, they can be ordered either as $x_{i_1} < x_{i_2} < x_{i_3}$, $y_{i_1} < y_{i_2} < y_{i_3}$, or $x_{i_1} < x_{i_2} < x_{i_3}$, $y_{i_1} > y_{i_2} > y_{i_3}$, and $(x_{i_2} - x_{i_1})(y_{i_3} - y_{i_2}) = (x_{i_3} - x_{i_2})(y_{i_2} - y_{i_1})$. We now express Φ by an equivalent active-domain formula

$$\forall x_1, x_2, x_3, y_1, y_2, y_3 \in adom \ \begin{pmatrix} S(x_1, y_1) \wedge S(x_2, y_2) \wedge S(x_3, y_3) \to \\ \chi(x_1, x_2, x_3, y_1, y_2, y_3) \end{pmatrix}.$$

Of course, this transformation is very ad-hoc, and takes into account the semantics of the original formula Φ. In what follows, we present a more general transformation.

1.6.4 Collapse algorithm: the linear case

The general proof of the natural-active collapse is by induction on the formulae. The cases of atomic formulae and Boolean connectives are simple: for atomic formulae, there is no need to change anything, and then one just propagates the connectives. The only hard case is that of the unrestricted quantification $\exists x\varphi$. We now consider an FO + LIN sentence $\Phi \equiv \exists z\varphi(z)$, where

$$\varphi(z) \equiv \ \mathbf{Q}y_1 \in adom \ldots \mathbf{Q}y_m \in adom \ \alpha(z, \vec{y}),$$

where each $\mathbf{Q}$ is either $\exists$ or $\forall$. (Of course we could have considered an open formula $\Phi(\vec{x})$ with free variables, as we shall do in the next section. However, our goal here is to present the ideas of the proof, so we make the assumption that there are no free variables. It will turn out that they do not add to the complexity of the proof, but they make notation heavier.)

Using Lemma 1.12, we can further assume that α is a Boolean combination of formulae of the form:

1. atomic SC-formulae $R_j(\vec{u})$ where $R_j \in SC$ and $\vec{u}$ only has variables from $\vec{y}$;
2. linear constraints involving z: $z \ \theta \ \sum_{i=1}^{m} a_i \cdot y_i + b$, where θ is $=$ or $<$;
3. linear constraints not involving z: $\sum_{i=1}^{m} a_i \cdot y_i + b \ \theta \ 0$.

Let $f_1(\vec{y}), \ldots, f_p(\vec{y})$ enumerate the (finitely many) functions that occur as right hand sides $\sum_{i=1}^{m} a_i \cdot y_i + b$ of linear constraints in 2) above (that is, those involving z). We also assume that one of the functions f_i is the function $f(\vec{y}) = y_1$.

Fix an SC-structure D, and let $A = adom(D)$. Let

$$B_0 = \{f_i(\vec{a}) \mid i = 1, \ldots, p, \vec{a} \in A^m\}$$

Note that $A \subseteq B_0$. Assume that $B_0 = \{b_1, \ldots, b_k\}$ with $b_1 < \ldots < b_k$.

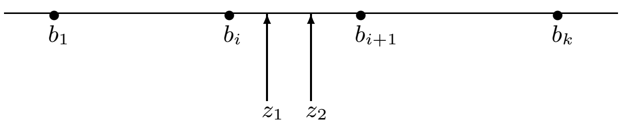

Fig. 1.4. Illustration to the natural-active collapse for the linear case

If $z_1 \in (b_i, b_{i+1})$ satisfies φ, then any other z_2 from this interval satisfies φ. Indeed, the variable z is only used in atomic subformulae of the form 2), that is, $z \ \theta \ f_j(\vec{y})$. Thus, for any instantiation $\vec{a}$ for $\vec{y}$ from the active domain A, we have $D \models \alpha(z_1, \vec{a}) \leftrightarrow \alpha(z_2, \vec{a})$, since the sign of z_1 and z_2 with respect to all $f_j(\vec{a})$ is the same. Since all variables $\vec{y}$ range over A, this implies $D \models \varphi(z_1) \leftrightarrow \varphi(z_2)$. Similarly, we note that for any $z_1, z_2 < b_1$, or for any $z_1, z_2 > b_k$, it is also the case that $D \models \varphi(z_1) \leftrightarrow \varphi(z_2)$.

Thus, if φ is witnessed by an element in an interval (b_i, b_{i+1}), or $(-\infty, b_1)$, or (b_k, ∞), it is witnessed by *every* element of the interval. Hence, if we define

$$B_1 \;=\; \{\frac{b+b'}{2} \mid b, b' \in B_0\} \;\cup\; \{b-1 \mid b \in B_0\} \;\cup\; \{b+1 \mid b \in B_0\},$$

we conclude that $D \models \exists z \varphi(z)$ iff $D \models \varphi(b)$ for some $b \in B_1$.

A nice property of B_1 is that it is definable in FO + Lin under the active-domain semantics. In fact, using the definition of B_1, we just rewrite $\exists z \varphi(z)$ to an equivalent active-domain semantics sentence:

$$\exists \vec{u} \in adom\ \exists \vec{v} \in adom \left(\begin{array}{l} \quad \left(\bigvee_{i=1}^{p} \bigvee_{j=1}^{p} (\varphi([\frac{f_i(\vec{u})+f_j(\vec{v})}{2} \;/\; z])) \right) \\ \vee \left(\bigvee_{i=1}^{p} \varphi([(f_i(\vec{u})-1) \;/\; z]) \right) \\ \vee \left(\bigvee_{i=1}^{p} \varphi([(f_i(\vec{u})+1) \;/\; z]) \right) \end{array} \right)$$

where $f_1, \ldots, f_p$ are all the linear functions used in constraints of the form $z = f_i(\vec{y})$ or $z < f_i(\vec{y})$ in the formula φ, as well as the function $f(\vec{y}) = y_1$.

Note that the proof of the existence of a sentence equivalent to Φ is constructive. Furthermore, the simple proof sketched in this section has the main ingredients of the general proof. To eliminate an unrestricted quantifier from $\varphi(\vec{x}) \equiv \exists z \alpha(z, \vec{x})$, we define some partition of U into a finite union of intervals $\bigcup_i I_i(\vec{x})$, such that:

- if $\varphi(\vec{a})$ is witnessed by $c \in I_i(\vec{a})$, then it is witnessed by any $c' \in I_i(\vec{a})$;
- each interval $I_i(\vec{x})$ is definable by an $\mathrm{FO}(SC, \mathfrak{M})$ formula, parametrically in $\vec{x}$, and so is a representative of each such interval, and
- the maximum number of intervals $I_i(\vec{x})$ is uniformly bounded for all $\vec{x}$.

1.6.5 Collapse algorithm: the general case

We start by listing some important properties of o-minimal structures. The key is the uniform bound on the number of intervals in definable sets.

Theorem 1.29 (Uniform Bounds). *If $\mathfrak{M}$ is o-minimal, and $\gamma(\vec{y}, x)$ is a first-order formula in the language of $\mathfrak{M}$, then there is an integer K_γ such that, for each tuple $\vec{a}$ from U, the set $\{x \mid \mathfrak{M} \models \gamma(\vec{a}, x)\}$ is composed of fewer than K_γ intervals.* □

This is a very strong and deep result. O-minimality simply tells us that for every $\gamma(\vec{y}, x)$ and every $\vec{a}$, the set $\gamma(\mathfrak{M}, \vec{a}) = \{x \mid \mathfrak{M} \models \gamma(\vec{a}, x)\}$ is a finite union of intervals. It is conceivable that the number of intervals in $\gamma(\mathfrak{M}, \vec{a})$ depends on $\vec{a}$ in such a way that there is no bound on this number when $\vec{a}$ ranges over U. The Uniform Bounds theorem tells us that such a situation is impossible: there is an an upper bound on the number of intervals that depends only on γ, and not on $\vec{a}$. As a side remark, the uniform bounds theorem also implies that a structure elementary equivalent to an o-minimal one is o-minimal itself.

We note, however, that for many familiar o-minimal structures, such as the real field or the real ordered group, the Uniform Bounds theorem is trivial. Indeed, for the real field, the proof of o-minimality based on quantifier-elimination (given in Section 1.6.2) immediately yields uniform bounds, as the number of intervals is determined by the number of polynomials used in the formula, and their degrees (recall that the number of intervals is determined by the total number of roots of all nonzero polynomials used in the formula).

For every $\gamma(\vec{y}, x)$ in the language of $\mathfrak{M}$ and constants, and every $\vec{a}$ over $\mathfrak{M}$, by the ith interval of $\gamma(\vec{a}, \cdot)$ we shall mean the ith interval of $\gamma(\mathfrak{M}, \vec{a})$, in the usual ordering on U. We shall use the following simple facts.

- For every formula $\gamma(\vec{y}, x)$, and every i, there exists a first-order formula denoted by $\hat{\gamma}_i(\vec{y}, x)$ such that $\mathfrak{M} \models \hat{\gamma}_i(\vec{a}, c)$ iff c is in the ith interval of $\gamma(\vec{a}, \cdot)$. In what follows, we always assume that the distinguished variable x is the last one.
- If the quantifier-elimination procedure is effective, and atomic sentences of $\mathfrak{M}$ are decidable, then K_γ is computable for each γ. Indeed, for each i, write a sentence $\Gamma_i \equiv \exists x \exists \vec{y}\ \hat{\gamma}_i(\vec{y}, x)$ and check if it is true in $\mathfrak{M}$, using quantifier-elimination and recursiveness of $\mathfrak{M}$. Eventually, we find i such that Γ_i is false; this follows from Theorem 1.29. Thus, K_γ can be taken to be this i.
- Since intervals are first-order definable, we can use them in formulae. For example, given a formula $\gamma(\vec{y}, x)$, a number i, and another formula $\beta(\vec{z}, x)$, we can write a first-order formula $\alpha(\vec{y}, \vec{z}, x)$ saying that every x from the ith interval of $\gamma(\vec{y}, \cdot)$ satisfies $\beta(\vec{z}, x)$. This of course is just $\forall x\ (\hat{\gamma}_i(\vec{y}, x) \to \beta(\vec{z}, x))$, but we shall occasionally use the interval notation in formulae, to simplify the presentation.

Natural-active collapse: eliminating one existential quantifier

This is the key case in proving the collapse, as the proof is by induction on the formulae, and this is the only case where there is a need to do something. We consider an $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ formula

$$\alpha(\vec{x}, z) \quad \equiv \quad \mathbf{Q} y_1 \in adom \ldots \mathbf{Q} y_m \in adom\ \beta(\vec{x}, \vec{y}, z)\ ,$$

where $\beta(\vec{x}, \vec{y}, \vec{z})$ is quantifier-free, and has the following properties:

- every atomic subformula of β is either an FO(SC) formula, or an FO($\mathfrak{M}$) formula (where equalities are considered to be FO($\mathfrak{M}$) formulae);
- there exists at least one FO($\mathfrak{M}$) atomic subformula of β, and at least one $\vec{y}$-variable (that is $m > 0$), and
- z does not occur in atomic FO(SC) subformulae.

Let $\mathcal{F}$ be the collection of all FO($\mathfrak{M}$) atomic subformulae of β, and their negations.

For formulae $\sigma(\vec{x}, \vec{y}, z)$, $\rho(\vec{x}, \vec{y}, z)$ and $\tau(\vec{x}, \vec{y}, z)$ from $\mathcal{F}$, $i \leq K_\rho$, and $j \leq K_\tau$, we let $\sigma_{ij}^{\rho\tau}(\vec{x}, \vec{y}, \vec{s}, \vec{t})$, where $card(\vec{s}) = card(\vec{t}) = card(\vec{y})$, be the formula defined as follows:

$$\sigma_{ij}^{\rho\tau}(\vec{x}, \vec{y}, \vec{s}, \vec{t}) \quad \equiv \quad \forall u \ \big((\hat{\rho}_i(\vec{x}, \vec{s}, u) \wedge \hat{\tau}_j(\vec{x}, \vec{t}, u)) \rightarrow \sigma(\vec{x}, \vec{y}, u)\big) \ .$$

Let $\varphi(\vec{x})$ be $\exists z \ \alpha(\vec{x}, z)$.

Lemma 1.30. *Let D be a nonempty finite SC-structure over $\mathfrak{M}$. Let $\varphi, \alpha, \beta, \mathcal{F}$ be as above. Let $\vec{a}$ be a tuple over U. Then $D \models \varphi(\vec{a})$ if and only if there exist $\vec{b}, \vec{c} \in adom(D)^m$, two formulae $\rho(\vec{x}, \vec{y}, z)$ and $\tau(\vec{x}, \vec{y}, z)$ in $\mathcal{F}$ and $i \leq K_\rho, j \leq K_\tau$ such that for the ith interval of $\rho(\vec{a}, \vec{b}, \cdot)$ and the jth interval of $\tau(\vec{a}, \vec{c}, \cdot)$, denoted by I_0 and I_1 respectively, the following three conditions hold:*

1. *$I_0 \cap I_1 \neq \emptyset$.*
2. *For all $\vec{e} \in adom(D)^m$, and all $c, c' \in I_0 \cap I_1$, we have $\mathfrak{M} \models \sigma(\vec{a}, \vec{e}, c) \leftrightarrow \sigma(\vec{a}, \vec{e}, c')$ for all $\sigma \in \mathcal{F}$.*
3. *$D \models \alpha'(\vec{b}, \vec{c}, \vec{a})$, where $\alpha'(\vec{s}, \vec{t}, \vec{x})$ is obtained from $\alpha(\vec{x}, z)$ by replacing each subformula $\sigma(\vec{x}, \vec{y}, z)$ from $\mathcal{F}$ by $\sigma_{ij}^{\rho\tau}(\vec{x}, \vec{y}, \vec{s}, \vec{t})$.*

Proof. For the *only if* part, assume that $D \models \varphi(\vec{a})$. That is, $D \models \exists z \alpha(\vec{a}, z)$. Let d witness this; that is, $D \models \alpha(\vec{a}, d)$. For every $\vec{e}$ over $adom(D)$, of the same length as $\vec{y}$, and every atomic FO($\mathfrak{M}$) subformula $\rho(\vec{x}, \vec{y}, z)$ of β, we define $I_d(\vec{e}, \rho)$ to be the maximal interval of $\rho(\mathfrak{M}, \vec{a}, \vec{e}) = \{c \mid \mathfrak{M} \models \rho(\vec{a}, \vec{e}, c)\}$ containing d, in the case when $\mathfrak{M} \models \rho(\vec{a}, \vec{e}, d)$, or the the maximal interval of $\neg\rho(\mathfrak{M}, \vec{a}, \vec{e})$ containing d, in the case when $\mathfrak{M} \models \neg\rho(\vec{a}, \vec{e}, d)$. Let $\mathcal{I}_d$ be the collection $\{I_d(\vec{e}, \rho) \mid \vec{e} \in adom(D)^{|\vec{y}|}, \rho \in \mathcal{F}\}$. Since for each $\vec{e}$ and ρ we have $d \in I_d(\vec{e}, \rho)$, we obtain that $\bigcap \mathcal{I}_d \neq \emptyset$.

Now note that for any finite collection of intervals $I_1, \ldots, I_p$, there are two indices i and j such that $\bigcap_{l=1}^{p} I_l = I_i \cap I_j$. Then there are two intervals, I_0 and I_1 in $\mathcal{I}_d$ such that $I_0 \cap I_1 = \bigcap \mathcal{I}_d$. Let $\vec{b}$ be such that I_0 is the ith interval of $\rho(\vec{a}, \vec{b}, \mathfrak{M})$, and $\vec{c}$ be such that I_1 is the jth interval of $\tau(\vec{a}, \vec{c}, \mathfrak{M})$, where $\rho, \tau \in \mathcal{F}$ (that is, ρ, τ are either atomic FO($\mathfrak{M}$) subformulae of φ, or negations of such atomic subformulae).

Let $\vec{e} \in adom(D)^{|\vec{y}|}$. Pick any $\sigma \in \mathcal{F}$ and any $c, c' \in I_0 \cap I_1$. Since $I_0 \cap I_1 = \bigcap \mathcal{I}_d$, we obtain that $c, c' \in I_0 \cap I_1 \subseteq I_d(\vec{e}, \sigma)$, which implies $\mathfrak{M} \models \sigma(\vec{a}, \vec{e}, c) \leftrightarrow \sigma(\vec{a}, \vec{e}, c')$. This proves conditions 1 and 2 in the Lemma.

To prove condition 3, notice that for every FO($\mathfrak{M}$) atomic subformula $\sigma(\vec{x}, \vec{y}, z)$ of φ, and for every $\vec{e} \in adom(D)^{|\vec{y}|}$, we have

$$\sigma(\vec{a}, \vec{e}, d) \quad \leftrightarrow \quad \forall u \in I_0 \cap I_1 \; \sigma(\vec{a}, \vec{e}, u) \; ,$$

since $I_0 \cap I_1 = \bigcap \mathcal{I}_d$.

Now, for any subformula $\gamma(\vec{x}, \vec{y}, z)$ of $\alpha(\vec{x}, z)$, let $\gamma'(\vec{s}, \vec{t}, \vec{x}, \vec{y})$ be the result of replacing each $\sigma(\vec{x}, \vec{y}, z)$ from $\mathcal{F}$ by $\sigma_{ij}^{\rho\tau}(\vec{x}, \vec{y}, \vec{s}, \vec{t})$.

We can now restate the above equivalence as:

$$(*) \qquad D \models \sigma(\vec{a}, \vec{e}, d) \leftrightarrow \sigma'(\vec{a}, \vec{e}, \vec{b}, \vec{c})$$

for every $\vec{e} \in adom(D)^{|\vec{y}|}$ (where $\vec{b}$ and $\vec{c}$ are the tuples necessary to define $I_0 \cap I_1$ above), where $\sigma(\vec{x}, \vec{y}, z)$ is atomic or negated atomic (i.e. $\sigma \in \mathcal{F}$).

The above equivalence is preserved under Boolean combinations and active quantification over variables from $\vec{y}$ in σ. Hence we obtain $(*)$ for every σ that is a subformula of α. Finally, this gives us

$$D \models \alpha(\vec{a}, d) \leftrightarrow \alpha'(\vec{a}, \vec{b}, \vec{c}) \; .$$

Since $D \models \alpha(\vec{a}, d)$, we conclude $D \models \alpha'(\vec{a}, \vec{b}, \vec{c})$, proving 3).

To prove the *if* part, assume that there exist $\vec{b}, \vec{c} \in adom(D)^m$, $\rho, \tau \in \mathcal{F}$, and $i \leq K_\rho, j \leq K_\tau$ such that for I_0, I_1 defined as in the statement of the Lemma, conditions 1, 2, and 3 hold. Let d be an arbitrary element of $I_0 \cap I_1$. We claim that $D \models \alpha(\vec{a}, d)$, thus proving $D \models \varphi(\vec{a})$.

Indeed, for every FO($\mathfrak{M}$) atomic subformula $\sigma(\vec{x}, \vec{y}, z)$ of α, we have

$$\sigma(\vec{a}, \vec{e}, d) \quad \leftrightarrow \quad \forall u \in I_0 \cap I_1 \; \sigma(\vec{a}, \vec{e}, u) \; ,$$

for every $\vec{e}$ over $adom(D)$ – this follows from 2. That is, $\sigma(\vec{a}, \vec{e}, d) \leftrightarrow \sigma_{ij}^{\rho\tau}(\vec{a}, \vec{e}, \vec{b}, \vec{c})$. As before, since this equivalence is preserved under Boolean combinations with FO(SC) atomic formula, and under active-domain quantification over variables from $\vec{y}$, we obtain

$$D \models \alpha(\vec{a}, d) \leftrightarrow \alpha'(\vec{b}, \vec{c}, a) \; ,$$

thus proving $D \models \alpha(\vec{a}, d)$. The lemma is proved. □

The transformation algorithm

The algorithm that converts natural-semantics formulae into active-semantics formulae works by induction on the structure of the formulae. In the case of atomic formulae, there is no need to change anything. For Boolean connectives, suppose $\varphi \equiv \chi \vee \psi$. Let χ_{act} and ψ_{act} be $\text{FO}_{\text{act}}(SC, \mathfrak{M})$ formulae equivalent to χ and γ. Then $\chi_{\text{act}} \vee \psi_{\text{act}}$ is an $\text{FO}_{\text{act}}(SC, \mathfrak{M})$ formula equivalent to φ. We deal with negation and conjunction similarly.

The only nontrivial case is that of an existential quantifier $\exists z \alpha(\vec{x}, z)$. To handle it, we use Lemma 1.30. For now, assume that we deal with nonempty SC-structures. By the induction hypothesis, we assume that α is an $\text{FO}_{\text{act}}(SC, \mathfrak{M})$ formula. We first put α in the form required by Lemma 1.30 by taking conjunction with a true sentence $\exists y \in adom(y = y)$ (since *adom* is nonempty) to ensure that there are quantifiers and atomic $\text{FO}(\mathfrak{M})$ formulae, and then using Lemma 1.12 to separate $\text{FO}(\mathfrak{M})$ and $\text{FO}(SC)$ formulae, and finally putting α in prenex form. Once α is in the right form, we apply Lemma 1.30, noticing that it translates into a first-order description. The step-by-step process of doing so is described in the algorithm NATURAL-ACTIVE shown on the next page. Note that every occurrence of an unrestricted quantifier $\forall$ or $\exists$ is of the form $\forall y \gamma$ or $\exists x \gamma$ where γ is an $\text{FO}(\mathfrak{M})$ formula. Since $\mathfrak{M}$ has quantifier-elimination, this means that every occurrence of unrestricted quantification can be eliminated.

Summing up, we have the following.

Proposition 1.31. *Let $\mathfrak{M}$ be o-minimal and admit quantifier-elimination. Let $\varphi(\vec{x})$ be any $\text{FO}(SC, \mathfrak{M})$ first-order formula, and let φ_{act} be the output of* NATURAL-ACTIVE *on φ. Then, for every nonempty finite SC-structure D, $D \models \forall \vec{x}\ \varphi(\vec{x}) \leftrightarrow \varphi_{\text{act}}(\vec{x})$. Furthermore, if $\mathfrak{M}$ is recursive and the quantifier-elimination procedure is effective, then there is an effective procedure yielding such an φ_{act} on input φ.* □

To conclude the proof of Theorem 1.26, we have to deal with the case of $adom(D)$ being empty. Let $\varphi(\vec{x})$ be an $\text{FO}(SC, \mathfrak{M})$ formula. Let $\varphi'_{\emptyset}(\vec{x})$ be obtained from φ by replacing each occurrence of $R(\cdots)$, where $R \in SC$, by *false*. Note that $\varphi'_{\emptyset}$ is an $\text{FO}(\mathfrak{M})$ formula. Let $\varphi_{\emptyset}$ be a quantifier-free formula equivalent to $\varphi'_{\emptyset}$. A simple induction on formulae shows that for the empty SC-instance, $\emptyset_{SC}$, it is the case that $\emptyset_{SC} \models \varphi(\vec{a})$ iff $\mathfrak{M} \models \varphi_{\emptyset}(\vec{a})$, for every $\vec{a}$. Thus, an $\text{FO}_{\text{act}}(SC, \mathfrak{M})$ formula

$$\begin{aligned}\varphi'(\vec{x}) \quad \equiv \quad & [(\exists x \in adom\ (x = x)) \ \wedge \ \varphi_{\text{act}}(\vec{x})] \\ & \vee [(\neg \exists x \in adom\ (x = x)) \ \wedge \ \varphi_{\emptyset}(\vec{x})]\ ,\end{aligned}$$

has the property that $D \models \forall \vec{x}\ \varphi(\vec{x}) \leftrightarrow \varphi'(\vec{x})$, for arbitrary D. This concludes the proof of Theorem 1.26. □

1.6.6 Collapse without o-minimality

We have seen that quantifier-elimination is necessary for the natural-active collapse. What about o-minimality? It turns out that there are non-o-minimal structures that admit the collapse. Consider the structure $\mathfrak{Z} = \langle \mathbb{Z}, +, < \rangle$. It is not o-minimal: for example, the formula $\varphi(x)$ given by $\exists y\ (y + y = x)$

Algorithm NATURAL–ACTIVE

INPUT: FO($SC, \mathfrak{M}$) formula $\varphi(\vec{x})$
OUTPUT: $\mathrm{FO_{act}}(SC, \mathfrak{M})$ formula $\varphi_{\mathrm{act}}(\vec{x})$

1. If φ is an atomic formula, then $\varphi_{\mathrm{act}} = \varphi$.
2. If $\varphi = \psi * \chi$, then $\varphi_{\mathrm{act}} = \psi_{\mathrm{act}} * \chi_{\mathrm{act}}$ where $* \in \{\vee, \wedge\}$; if $\varphi = \neg\psi$, then $\varphi_{\mathrm{act}} = \neg\psi_{\mathrm{act}}$.
3. If $\varphi = \exists x \in adom\ \psi$, then $\varphi_{\mathrm{act}} = \exists x \in adom\ \psi_{\mathrm{act}}$.
4. Let $\varphi(\vec{x}) = \exists z\ \alpha^0(\vec{x}, z)$.

4.1 Let $\alpha(\vec{x}, z)$ be a formula equivalent to α^0_{act} which is of the form

$$\mathbf{Q}y_1 \in adom \ldots \mathbf{Q}y_m \in adom\ \beta(\vec{x}, \vec{y}, z),$$

where $\beta(\vec{x}, \vec{y}, \vec{z})$ is quantifier-free, and has the following properties: every atomic subformula of β is either a FO(SC) formula, or a FO($\mathfrak{M}$) formula; there exists at least one FO($\mathfrak{M}$) atomic subformula of β, $m > 0$, and z does not occur in FO(SC) subformulae.

4.2 Let $\mathcal{F}$ be the collection of all atomic FO($\mathfrak{M}$) subformulae of α, and their negations.

4.3 Let $K = \max_{\gamma \in \mathcal{F}} K_\gamma$.

4.4 For every pair of formulae $\rho, \sigma \in \mathcal{F}$, and every $i, j < K$, define $\chi^{\rho\sigma}_{ij}(\vec{x}, \vec{s}, \vec{t})$ to be the quantifier-free FO($\mathfrak{M}$) formula equivalent to $\exists u\ \big(\hat{\rho}_i(\vec{x}, \vec{s}, u) \wedge \hat{\sigma}_j(\vec{x}, \vec{t}, u)\big)$. Note that $|\ \vec{s}\ | = |\ \vec{t}\ | = m$.

4.5 For each $\rho, \sigma \in \mathcal{F}$, each $i, j < K$, and each $\tau \in \mathcal{F}$, define $\tau^{\rho\sigma}_{ij}(\vec{x}, \vec{y}, \vec{s}, \vec{t})$ as a quantifier-free formula equivalent to

$$\forall u\ \Big(\hat{\rho}_i(\vec{x}, \vec{s}, u) \wedge \hat{\sigma}_j(\vec{x}, \vec{t}, u) \rightarrow \tau(\vec{x}, \vec{y}, u)\Big)$$

4.6 For each $\rho, \sigma \in \mathcal{F}$, each $i, j < K$, define $\alpha^{\rho\sigma}_{ij}(\vec{x}, \vec{s}, \vec{t})$ as α in which every FO($\mathfrak{M}$) atomic subformula $\tau(\vec{x}, \vec{y}, z) \in \mathcal{F}$ is replaced by $\tau^{\rho\sigma}_{ij}(\vec{x}, \vec{y}, \vec{s}, \vec{t})$.

4.7 Let $\mathrm{same}_\beta(\vec{x}, \vec{r}, u, v)$ be $\bigwedge(\rho(\vec{x}, \vec{r}, u) \leftrightarrow \rho(\vec{x}, \vec{r}, v))$, where the conjunction is taken over all the FO($\mathfrak{M}$) atomic subformulae ρ of β.

4.8 For each $\rho, \sigma \in \mathcal{F}$, each $i, j < K$, define $\eta^{\rho\sigma}_{ij}(\vec{x}, \vec{s}, \vec{t}, \vec{r})$ as a quantifier-free formula equivalent to

$$\forall u, v\ \Big((\hat{\rho}_i(\vec{x}, \vec{s}, u) \wedge \hat{\sigma}_j(\vec{x}, \vec{t}, u) \wedge \hat{\rho}_i(\vec{x}, \vec{s}, v) \wedge \hat{\sigma}_j(\vec{x}, \vec{t}, v)) \rightarrow \mathrm{same}_\beta(\vec{x}, \vec{r}, u, v)\Big)$$

4.9 For each $\rho, \sigma \in \mathcal{F}$, each $i, j < K$, define $\pi^{\rho\sigma}_{ij}(\vec{x}, \vec{s}, \vec{t})$ as $\forall \vec{r} \in adom\ \eta^{\rho\sigma}_{ij}(\vec{x}, \vec{s}, \vec{t}, \vec{r})$.

4.10 Output, as $\varphi_{\mathrm{act}}(\vec{x})$, the formula

$$\exists \vec{s} \in adom\ \exists \vec{t} \in adom \bigvee_{\rho, \sigma \in \mathcal{F}} \bigvee_{i,j<K} (\chi^{\rho\sigma}_{ij}(\vec{x}, \vec{s}, \vec{t}) \wedge \pi^{\rho\sigma}_{ij}(\vec{x}, \vec{s}, \vec{t}) \wedge \alpha^{\rho\sigma}_{ij}(\vec{x}, \vec{s}, \vec{t})).$$

defines the set of even numbers. The same example though shows that the natural-active collapse fails over $\mathfrak{Z}$: the Boolean query $\exists x\ (S(x) \wedge \varphi(x))$ is not expressible in $\mathrm{FO}_{\mathrm{act}}(\{S\}, \mathfrak{Z})$, since φ cannot be expressed by a quantifier-free formula.

However, it is well-known that $\mathfrak{Z}$ admits quantifier-elimination in an extended signature. Let $x \sim_k y$ iff $x = y(\mathrm{mod} k)$. These relations are definable over $\mathfrak{Z}$, and the structure $\mathfrak{Z}_0 = \langle \mathbb{Z}, +, <, 0, 1, (\sim_k)_{k>0} \rangle$ does admit quantifier-elimination. We thus have an example of a structure that has quantifier-elimination, is not o-minimal, and

Proposition 1.32. $\mathfrak{Z}_0$ *admits the natural-active collapse.*

Proof sketch. The proof is again by induction, and we consider the only non-trivial case of existential quantification. To simplify the notation, assume that we have a sentence $\Phi \equiv \exists z \varphi(z)$ where

$$\varphi(z) \equiv \mathbf{Q} y_1 \in adom \ldots \mathbf{Q} y_m \in adom\ \alpha(z, \vec{y}),$$

where each $\mathbf{Q}$ is either $\exists$ or $\forall$.

Using Lemma 1.12, we can assume that α is a Boolean combination of:

1. atomic SC-formulae with free variables among $\vec{y}$;
2. linear constraints $f(z, \vec{y})\ \theta\ 0$, where f is a linear function and θ is a $=$, or $<$, or $\leq$ comparison;
3. constraints of the form $f(z, \vec{y}) \sim_c p$ for $c \in \mathbb{N}$ and $0 \leq p < c$, where again f is a linear function.

Let c be the maximum number for which one of $\sim_c$ relations occurs in α. Let $\chi_i(x)$ enumerate all satisfiable formulae of the form

$$\bigwedge_{1<b\leq c} x \sim_b p_b,$$

where $p_b < b$, and similarly let $\chi_i^m(\vec{y})$ enumerate all satisfiable conjunctions $\chi_{i_1}(y_1) \wedge \ldots \wedge \chi_{i_m}(y_m)$. Then $\varphi(z)$ is equivalent to:

$$\exists z \Big(\bigvee_i \chi_i(z) \wedge \mathbf{Q} y_1 \in adom \ldots \mathbf{Q} y_m \in adom\ \big(\bigvee_j \chi_j^m(\vec{y}) \wedge \alpha(z, \vec{y}) \big) \Big).$$

Note that if we know all the residues for z and $\vec{y}$ modulo all the positive integers not exceeding c, then we can infer the truth value of each constraint of the form $f(z, \vec{y}) \sim_b p$ for every $b \leq c$ and $p_b < b$. Thus, we can assume without loss of generality that constraints of the form $f(z, \vec{y}) \sim_b p$ do not appear in α, unless f is identically z or one of y_is.

To eliminate $\exists z$ from the formula above, we proceed just as in the case of $\mathrm{FO} + \mathrm{LIN}$. Let $g_1(\vec{y}), \ldots, g_l(\vec{y})$ enumerate all the linear functions that occur in constraints of the form $z\theta g_i(\vec{y})$, and the function $g(\vec{y}) = y_1$. Fix a finite set A, and define a set B_0 as $\{g_i(\vec{a}) \mid i \leq l, \vec{a} \in A^m\}$. Note that $A \subseteq B_0$. Let $b_1 < \ldots < b_k$ list the elements of B_0.

Suppose we have a SC-structure D with $adom(D) = A$, and suppose that $\varphi(z_0)$ holds. Assume that $b_i < z_0 < b_{i+1}$. Then the same argument as in the proof of the collapse for FO + LIN shows that any other $z_0' \in (b_i, b_{i+1})$ that agrees with z_0 on all χ_js, also satisfies φ. This shows the following: if there is a z_0 satisfying φ, then there is one such that $|z_0 - b_i| \leq c$ for some b_i. In particular, if $D \models \Phi$, then there exists $z_0 \in B_1$ such that $D \models \varphi(z_0)$, where $B_1 = \{b + p, b - p \mid b \in B_0, 0 \leq p \leq c\}$. Just as in the case of FO + LIN, this set B_1 is definable in FO($SC, \mathfrak{Z}_0$). Thus, under the assumption that α only uses $\sim_k$ relations to compare a variable with a constant, we can rewrite Φ to

$$\exists \vec{u} \in adom \bigvee_{-c \leq b \leq c} \bigvee_{i=1}^{l} \varphi((g_i(\vec{u}) + b) \ / \ z),$$

thus eliminating an unrestricted quantifier $\exists z$. Notice that unlike in the case of FO + LIN, we need m additional active-domain quantifiers (instead of $2m$), as the proof does not require witnesses which are middles of some intervals (b_i, b_{i+1}). □

1.6.7 Natural-generic collapse

The natural generic collapse says that order-generic queries in FO($SC, \mathfrak{M}$) can be expressed in FO($SC, <$). We now derive this collapse result as a corollary to the two collapses shown so far.

Corollary 1.33 (Natural Generic Collapse). *Let $\mathfrak{M} = \langle U, \Omega \rangle$ be an o-minimal structure. Then it admits the natural generic collapse.*

Proof. Let Q be an order-generic query definable in FO($SC, \mathfrak{M}$). Consider a definitional expansion $\mathfrak{M}'$ of $\mathfrak{M}$ by extending Ω with new symbols for all $\mathfrak{M}$-definable predicates. Such $\mathfrak{M}'$ admits quantifier-elimination, and then by the natural-active collapse we obtain that Q is definable in $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M}')$. From the active generic collapse, we conclude that Q is definable in $\mathrm{FO}_{\mathrm{act}}(SC, <)$ (and thus in FO($SC, <$)). □

While the active generic collapse holds for all ordered structures, and for the natural-active collapse the bounds of Theorem 1.26 are the best currently known, Corollary 1.33 was extended to a larger class of structures. The proof of the result is rather involved, but we shall present the statement below.

The new condition on the structures uses the Vapnik-Chervonenkis dimension, a central concept in computational learning theory. Suppose S is an infinite set, and $\mathcal{C} \subseteq 2^S$ is a family of subsets of S. Let $F \subset S$ be finite; we say that $\mathcal{C}$ *shatters* F if the collection $\{F \cap C \mid C \in \mathcal{C}\}$ is 2^F. The *Vapnik-Chervonenkis (VC) dimension* of $\mathcal{C}$, VCdim($\mathcal{C}$), is the maximal cardinality of a finite set shattered by $\mathcal{C}$. If arbitrarily large finite sets are shattered by $\mathcal{C}$, we let $\text{VCdim}(\mathcal{C}) = \infty$.

This applies to first-order structures as follows. Let $\mathfrak{M} = \langle U, \Omega \rangle$, and let $\varphi(\vec{x}, \vec{y})$ be a formula in the language of $\mathfrak{M}$ with $|\vec{x}| = n, |\vec{y}| = m$. For each $\vec{a} \in U^n$, define $\varphi(\vec{a}, \mathfrak{M}) = \{\vec{b} \in U^m \mid \mathfrak{M} \models \varphi(\vec{a}, \vec{b})\}$, and let $F_\varphi(\mathfrak{M})$ be $\{\varphi(\vec{a}, \mathfrak{M}) \mid \vec{a} \in \mathcal{U}^n\}$. Families of sets arising in such a way are called *definable families.*

Definition 1.34. $\mathfrak{M}$ *is said to have finite VC dimension if every definable family in* $\mathfrak{M}$ *has finite VC dimension.* □

Examples of structures that have finite VC dimension include:

- Every o-minimal structure;
- $\langle \mathbb{N}, +, < \rangle$ and $\langle \mathbb{Z}, +, < \rangle$;
- every linear order;
- ordered Abelian groups (that is, Abelian groups in which addition is monotone with respect to the order).

In particular, the class is a proper extension of the class of all o-minimal structures. The following is a deep result that we present here without a proof:

Theorem 1.35. *Let* $\mathfrak{M}$ *be an ordered structure that has finite VC dimension. Then* $\mathfrak{M}$ *admits the natural generic collapse.* □

We shall discuss the relationship between VC dimension and various forms of collapse in Section 1.8.

The diagram in Figure 1.3 summarizes what has been achieved towards proving the collapse results.

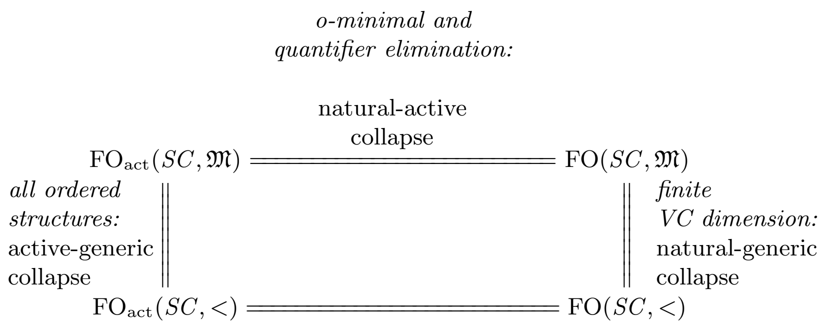

Fig. 1.5. Summary of the collapse results

1.7 Model theory and collapse results

While most collapse results proved so far apply to o-minimal structures, we have seen a couple of examples outside of the o-minimal world. So it is natural to ask what really causes the natural-active or other forms of collapse: are there some properties of the underlying structure that cause it to happen?

The goal of this section is to give a partial answer to this question. We start by presenting a technical condition, called *pseudo-finite homogeneity*, that ensures a form of collapse that is closely related to the natural-active collapse. We then describe a couple of model-theoretic conditions that are often easy to verify, and that imply pseudo-finite homogeneity, and thus the collapse. We will see a number of examples of collapse outside of the o-minimal context that are implied by those conditions.

We start with the following definition.

Definition 1.36. *We say that a structure* $\mathfrak{M}$ *admits the* restricted quantifier collapse *if for every* SC*, every* $\mathrm{FO}(SC, \mathfrak{M})$ *formula is equivalent to an* $\mathrm{FO}(SC, \mathfrak{M})$ *formula in which* SC*-relations do not appear in the scope of unrestricted quantifiers.*

For example, in the formula $\exists x \in adom \forall y \in adom\ (S(x, y) \rightarrow \forall z \exists u\ x^2 + y = z^2 + u)$, the SC-relation S only appears in the scope of two active-domain quantifies $\exists x \in adom$ and $\forall y \in adom$. However, for the formula $\exists u \exists v\ (\forall x \in adom \forall y \in adom\ S(x, y) \rightarrow y = u \cdot x + v)$ this is not the case, as S appears in the scope of the quantifiers $\exists u$ and $\exists v$.

Note that if $\mathfrak{M}$ admits the restricted quantifier collapse, and if $\mathfrak{M}'$ is the expansion of $\mathfrak{M}$ with all definable predicates, then every $\mathrm{FO}(SC, \mathfrak{M})$ formula is equivalent to an $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M}')$ formula. In particular, if $\mathfrak{M}$ admits quantifier-elimination, then the restricted quantifier collapse implies the natural-active collapse. Furthermore, the restricted quantifier collapse always implies the natural generic collapse. Thus:

restricted quantifier collapse	+	QE	=	natural-active collapse	⇒	restricted quantifier collapse	⇒	natural generic collapse

Remark. Although we provide all the necessary model-theoretic definitions here, the reader needs some infinite model theory background to understand the proofs in this section. In particular, many proofs using techniques from classical, infinite model theory, are only sketched. We still encourage the reader without such a background to read this section (perhaps skipping the proofs) to see many new examples of collapse results.

We shall also assume that we deal with structures in a finite or countable language; this assumption can easily be avoided at the expense of some additional arguments involving infinite cardinals, which we prefer not to deal with here.

1.7.1 Pseudo-finite homogeneity

We start with a few definitions from model theory. For a structure $\mathfrak{M}$, its theory is denoted by $\mathrm{Th}(\mathfrak{M})$. Two structures $\mathfrak{M}_1$ and $\mathfrak{M}_2$ of the same language are elementarily equivalent (written $\mathfrak{M}_1 \equiv \mathfrak{M}_2$) if their theories are the same; that is, if they satisfy the same FO-sentences. For a subset A of $\mathfrak{M}$, and an n-tuple $\vec{a}$, the *n-type of $\vec{a}$ over A* (or just type, if n is understood), $\mathrm{tp}_{\mathfrak{M}}(\vec{a}/A)$, is the set of all formulae in n free variables, in the language of $\mathfrak{M}$ plus constants for the elements of A, that are satisfied by $\vec{a}$.

A model $\mathfrak{M}$ is called *ω-saturated* if every consistent 1-type over a finite subset of $\mathfrak{M}$ is realized in $\mathfrak{M}$. It is known that for every $\mathfrak{M}$, there exists an ω-saturated elementary extension $\mathfrak{M}'$.

Let $L(SC, \mathfrak{M})$ be the language that is the expansion of $L(\mathfrak{M})$, the language of $\mathfrak{M}$, with all the relation symbols in SC. A structure in this language is a pair $(\mathfrak{M}', D)$ where $\mathfrak{M}'$ is a structure in the language of $\mathfrak{M}$ and D is an interpretation of SC symbols over $\mathfrak{M}'$ (not necessarily finite). Let $\mathcal{F}(SC, \mathfrak{M})$ be the theory of all $L(SC, \mathfrak{M})$ structures $(\mathfrak{M}', D)$ where $\mathfrak{M}' \models \mathrm{Th}(\mathfrak{M})$ and D is finite. We now call a SC-structure D on $\mathfrak{M}$ *pseudo-finite* if $(\mathfrak{M}, D) \models \mathcal{F}(SC, \mathfrak{M})$.

Definition 1.37. *We say that $\mathfrak{M}$ has* ω-pseudo-finite homogeneity property*, or* ω-PFH *for short, if for any model $\mathfrak{M}'$ of* $\mathrm{Th}(\mathfrak{M})$*, any two pseudo-finite SC-structures D_1, D_2 on $\mathfrak{M}'$, and any bijective and $L(\mathfrak{M})$-elementary map $h : D_1 \to D_2$ such that $(\mathfrak{M}', D_1, D_2, h)$ is ω-saturated, it is the case that for every $a \in \mathfrak{M}'$, there exists $b \in \mathfrak{M}'$ such that $h \cup \{(a, b)\}$ is elementary.*

Theorem 1.38. *If $\mathfrak{M}$ has ω-PFH, then it admits the restricted quantifier collapse.*

Proof sketch. Let φ be an $\mathrm{FO}(SC, \mathfrak{M})$ sentence. Assume that φ is not equivalent to any restricted quantifier sentence. Let α_i enumerate all restricted quantifier $\mathrm{FO}(SC, \mathfrak{M})$ sentences; then for every α_i we can find a model $(\mathfrak{M}_i, D_i^1, D_i^2)$ such that $\mathfrak{M}_i \equiv \mathfrak{M}$, $D_i^1 \models \varphi$, $D_i^2 \models \neg\varphi$, and D_i^1, D_i^2 agree on α_i. By compactness, we have a model $(\mathfrak{M}', D_1, D_2)$ such that D_1, D_2 agree on all $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ sentences, $D_1 \models \varphi$ and $D_2 \models \neg\varphi$.

A standard model-theoretic argument shows that we can further assume that there is a partial $L(SC, \mathfrak{M})$-isomorphism $h : D_1 \to D_2$ that is also an elementary map in the language of $\mathfrak{M}$, and that furthermore, $(\mathfrak{M}, D_1, D_2, h)$ is ω-saturated. By ω-PFH, for any $k > 0$, h can be extended k times back and forth to an $L(\mathfrak{M})$-elementary map, which is a partial $L(SC, \mathfrak{M})$-isomorphism since its domain includes $adom(D_1)$ and range includes $adom(D_2)$. Thus, h is an $L(SC, \mathfrak{M})$-elementary map, which contradicts $D_1 \models \varphi$ and $D_2 \models \neg\varphi$. □

The notion of pseudo-finite homogeneity may not be a very easy one to check for a given structure; however, other model-theoretic properties imply it, and thus they imply the restricted quantifier collapse. We shall see two examples below.

1.7.2 Finite cover property and collapse

Similarly to the definition of ω-saturation, we can define ω_1-saturated structures by requiring that types over countable sets (rather than just finite sets) be realized. By requiring that the structure $(\mathfrak{M}', D_1, D_2, h)$ in the definition of PFH be ω_1-saturated, we obtain a stronger notion of ω_1-PFH.

We now say that $\mathfrak{M}$ has the *pseudo-finite saturation property*, or PFS for short, if for any model $\mathfrak{M}'$ of the theory of $\mathfrak{M}$ and any pseudo-finite set A in $\mathfrak{M}'$ such that $(\mathfrak{M}', A)$ is ω_1-saturated, every consistent 1-type over A is realized in $(\mathfrak{M}', (a)_{a\in A})$. This property is easier to connect to other model-theoretic properties, and furthermore:

Proposition 1.39. *Pseudo-finite saturation implies ω_1-PFH, and thus it implies the restricted quantifier collapse.*

Proof. Let $(\mathfrak{M}', D_1, D_2, h)$ be ω_1-saturated, where D_1, D_2 are pseudo-finite. Let $a \in \mathfrak{M}'$, $A = adom(D_1)$, and let $p = \mathrm{tp}_{\mathfrak{M}'}(a/A)$. Let $h(p) = \{\varphi(x, h(\vec{a})) \mid \varphi(x, \vec{a}) \in p\}$. Then $h(p)$ is a type over $adom(D_2)$; by pseudo-finite saturation it is realized by some $b \in \mathfrak{M}'$, and thus $h \cup \{(a, b)\}$ is elementary. □

One known result about pseudo-finite saturation is that it holds for structures that do not have the finite cover property. Recall that a structure $\mathfrak{M}$ has the *finite cover property* if there is a formula $\varphi(x, \vec{y})$ such that for every $n > 0$, one can find tuples $\vec{a}_1, \ldots, \vec{a}_n$ such that $\exists x \bigwedge_{j\neq i} \varphi(x, \vec{a}_j)$ holds for each $i \leq n$, but $\exists x \bigwedge_{j\leq n} \varphi(x, \vec{a}_j)$ does not hold. Since every $\mathfrak{M}$ that does not have the cover property has pseudo-finite saturation, it also admits the restricted quantifier collapse.

In model theory, a number of examples of structures without the finite cover property have been collected; for example, every structure whose theory is categorical in every uncountable power is such. Some of the best known examples are:

- The complex numbers field $\langle \mathbb{C}, +, \cdot \rangle$ (in fact, any algebraically closed field of characteristic p, where p is zero or prime).
- $\langle \mathbb{N}, \pi \rangle$, where $\pi : \mathbb{N} \to \mathbb{N}$ is a permutation without finite cycles.
- $\langle \mathbb{N}, succ \rangle$.

Corollary 1.40. *The three structures above admit the restricted quantifier collapse.* □

As another example, we consider the first-order theory of finitely many successor relations. It is a decidable theory (in fact, even the monadic second-order theory is decidable, by a classical result by Büchi) with many applications in computer science. Let Σ be a finite alphabet, and let Σ^* be the set of all finite strings over Σ, with ϵ being the empty string. For each $a \in \Sigma$, let f_a be the unary function that appends a at the end of its argument: $f_a(x) = x \cdot a$. We now have the following:

Proposition 1.41. *For any finite Σ, the structure $\langle \Sigma^*, \epsilon, (f_a)_{a\in\Sigma}\rangle$ admits the restricted quantifier collapse.*

Proof sketch. We show that $\mathfrak{M} = \langle \Sigma^*, \epsilon, (f_a)_{a\in\Sigma}\rangle$ does not have the finite cover property. We need a bit of preparation. Our proof will use the following known result: $\mathfrak{M}$ does not have the finite cover property if (a) no formula $\alpha(\vec{x}, \vec{y})$ defines an infinite linear order on $\mathfrak{M}$, and (b) for every $\alpha(\vec{x}, \vec{y})$, there is a formula $\beta(\vec{x})$ such that $\beta(\vec{x})$ holds iff the number of $\vec{y}$ for which $\alpha(\vec{x}, \vec{y})$ holds, is infinite[1].

Let $g_a(x)$ be the following definable function: if the last symbol of x is a, then g removes it, otherwise $g_a(x) = x$. Then it is easy to see that $\langle \Sigma^*, \epsilon, (f_a)_{a\in\Sigma}\rangle$ has quantifier-elimination in the language $\langle (f_a, g_a)_{a\in\Sigma}, \epsilon\rangle$. Using this, one easily concludes (a).

To show (b), define the distance between two strings x and y, $d(x, y)$, as the minimal length of a term t built from f_a, g_a, ϵ such that $t(x) = y$. If one thinks of Σ^* as an infinite $|\Sigma|$-ary tree, then $d(u, v)$ is simply the distance in this tree. We define $d(\vec{x}, \vec{y})$ as the minimal distance between a component of $\vec{x}$ and a component of $\vec{y}$. Note that for each fixed i, $d(\vec{x}, \vec{y}) < i$ is definable.

Given a formula $\alpha(\vec{x}, \vec{y})$, assume without loss of generality that it is a Boolean combination of formulae $v = t(u)$, where v, u are variables among $\vec{x}, \vec{y}$ and ts are terms. Let k be the maximum length of a term in α. Define $\gamma(\vec{x})$ as

$$\forall \vec{y}\ \alpha(\vec{x}, \vec{y}) \rightarrow \bigwedge_i (d(y_i, \vec{x}) < m(k+2)) \wedge (d(y_i, \epsilon) < m(k+2)),$$

where m is the length of $\vec{y}$. We claim that $\gamma(\vec{x})$ holds iff the number of $\vec{y}$ such that $\alpha(\vec{x}, \vec{y})$ holds is finite. Then we take $\beta \equiv \neg\gamma$.

One direction is trivial. Assume $\gamma(\vec{x})$ does not hold; then one can find $\vec{y}$ for which $\alpha(\vec{x}, \vec{y})$ holds, and divide $\vec{y}$ into two parts, $\vec{y}_1$ and $\vec{y}_2$, such that $d(\vec{y}_2, (\vec{y}_1, \vec{x}, \epsilon)) > k+1$. Let now s be a sufficiently long string; define $s\cdot\vec{y}_2$ as the result of adding s as a prefix to all strings in $\vec{y}_2$. It is clear that $\alpha(\vec{x}, (\vec{y}_1, s\cdot\vec{y}_2))$ still holds, which completes the proof since s is arbitrary. □

The results of this section have some limitations; in particular, all structures with the pseudo-finite saturation property are stable, which means that one cannot define infinite linear orders in them. To deal with ordered structures (which are the ones most typically used in applications), we present a different model-theoretic notion that implies ω-PFS.

---

[1] The reader familiar with this subject will notice that our condition (b) is not sufficient to conclude that $\mathfrak{M}$ does not have the finite cover property: instead, one would need to show a stronger property (b'): namely, for any formula $\alpha(\vec{x}, \vec{y}, \vec{z})$ such that $\alpha(\mathfrak{M}, \vec{z})$ is an equivalence relation $E_{\vec{z}}$ for every $\vec{z}$, there is a formula $\beta(\vec{z})$ such that $\beta(\vec{z})$ holds iff $E_{\vec{z}}$ has finitely many equivalence classes. However, using quantifier-elimination for $\mathfrak{M}$, one can show that $\mathfrak{M}$ eliminates imaginaries and thus each equivalence relation is of the form $\{(\vec{u}, \vec{w}) \mid f(\vec{u}) = f(\vec{v})\}$ for some definable function f. Therefore, (b) implies (b').

1.7.3 Isolation and collapse

Let $\mathfrak{M}$ be a structure, A its subset, and p a 1-type over A. Let p' be a subset of p. We say that p' *isolates* p if p is the only type over A that contains p'.

Definition 1.42. *We say that* $\mathfrak{M}$ *has the* isolation property *if for every model* $\mathfrak{M}'$ *of the theory of* $\mathfrak{M}$*, any pseudo-finite set* A *in* $\mathfrak{M}'$*, and any element* a*, there is a finite set* $A' \subseteq A$ *such that* $\mathrm{tp}_{\mathfrak{M}'}(a/A')$ *isolates* $\mathrm{tp}_{\mathfrak{M}'}(a/A)$*.*

This gives us a number of new examples of structures that admit the restricted quantifier collapse, thanks to the following:

Proposition 1.43. *If* $\mathfrak{M}$ *has the isolation property, then it has* ω*-PFH (and thus admits the restricted quantifier collapse).*

Proof. Assume we have an ω-saturated $(\mathfrak{M}', D_1, D_2, h)$, where D_1, D_2 are two pseudo-finite SC-structures, and h is elementary. Let $a \in \mathfrak{M}'$. Let $A_i = adom(D_i)$; then A_1, A_2 are pseudo-finite sets. Let $p = \mathrm{tp}_{\mathfrak{M}'}(a/A_1)$; by isolation, there is a finite set $A_1' \subseteq A_1$ such that $p' = \mathrm{tp}_{\mathfrak{M}'}(a/A_1')$ isolates p.

For each $\varphi \in p$, let φ_h be φ in which every $c \in A_1$ is replaced by $h(c)$. Since h is elementary, any finite conjunction of formulae φ_h, $\varphi \in p$, is satisfiable in $\mathfrak{M}'$, and thus by compactness $h(p) = \{\varphi_h \mid \varphi \in p\}$ is consistent. Furthermore, a straightforward compactness argument shows that $h(p')$ isolates $h(p)$. Since A_1' is finite, $h(p')$ is countable, and thus by saturation it is realized by an element $b \in \mathfrak{M}'$. Since $h(p')$ isolates $h(p)$, b is of type $h(p)$, which shows that $h \cup \{(a, b)\}$ is elementary. □

As the simplest example of the isolation property, consider the theory of linear order, whose models are ordered sets $\langle U, < \rangle$. Let $\mathfrak{M}$ be such a structure, and let A be a pseudo-finite set. For any $a \in U$, and any finite set A_0, either there are two consecutive elements of A_0, say $b < c$, such that $(b, c) \cap A_0 = \emptyset$ and $b \leq a \leq c$, or $a > m$, where m is the maximal element of A_0, or $a < m'$, where m' is the minimal element of A_0. As this condition is FO-definable, it must be true for the pseudo-finite set A. We claim that $\mathrm{tp}_{\mathfrak{M}}(a/A)$ is isolated by $\mathrm{tp}_{\mathfrak{M}}(a/A')$, where $A' = \{b, c\}$, or $A' = \{m\}$, or $A' = \{m'\}$, depending on which of the three cases is true. We prove this for the case of $b \leq a \leq c$; the other cases are similar.

To show that $\mathrm{tp}_{\mathfrak{M}}(a/A')$ isolates $\mathrm{tp}_{\mathfrak{M}}(a/A)$, we must prove that for any a', $(\mathfrak{M}, a, b, c) \equiv (\mathfrak{M}, a', b, c)$ implies $(\mathfrak{M}', a, (d)_{d \in A}) \equiv (\mathfrak{M}', a', (d)_{d \in A})$. This is easy to see by an Ehrenfeucht-Fraïssé game argument. By the assumption, the duplicator has a winning strategy on $([b, c], a)$ and $([b, c], a')$. For the winning strategy on $(\mathfrak{M}', a, (d)_{d \in A})$ and $(\mathfrak{M}', a', (d)_{d \in A})$, the duplicator uses the above strategy for moves in the interval $[b, c]$, and copies spoiler's moves elsewhere.

What about more complex examples? First, it is easy to extend the example above to the case of ordered sets with some additional unary relations. That is:

Corollary 1.44. *Let $\mathfrak{M}$ be a structure with one binary relation, interpreted as a linear ordering, and finitely many unary relations. Then $\mathfrak{M}$ admits the restricted quantifier collapse.* □

As our next example, we revisit the theory of k successor relations, that is, $\langle \Sigma^*, \epsilon, (f_a)_{a\in\Sigma}\rangle$. This structure is the infinite k-ary tree in which we only have successor relations available. Most often it is considered in the context of monadic second-order logic, which can define the *prefix* relation in addition to the successor relations. So we now consider an extension, $\langle \Sigma^*, \epsilon, (f_a)_{a\in\Sigma}, \prec\rangle$, where $x \prec y$ means that x is a prefix of y. The question is: does this structure admit the collapse?

The technique of Section 1.7.2 does not work here, since $\langle \Sigma^*, \epsilon, (f_a)_{a\in\Sigma}, \prec\rangle$ does have the finite cover property: structures that do not have it, cannot define an infinite linear order; on the other hand, it is easy to define the lexicographic ordering in the presence of $\prec$. This turns out to be one of the examples where isolation does the job.

Proposition 1.45. $\mathfrak{M} = \langle \Sigma^*, \epsilon, (f_a)_{a\in\Sigma}, \prec\rangle$ *admits the restricted quantifier collapse.*

Proof. Let $x \preceq y$ mean $x \prec y$ or $x = y$. Let $x \sqcap y$ be the longest common prefix of strings x and y, and let $x - y$ be defined as follows: if $x = y \cdot z$, then $x - y = z$; if $y \not\preceq x$, then $x - y = \epsilon$. Let $L \subseteq \Sigma^*$ be a star-free language. Define P_L to be the set of pairs of strings (x, y) such that $y \preceq x$ and $x - y \in L$. It is not hard to show that P_L is definable in $\langle \Sigma^*, \epsilon, (f_a)_{a\in\Sigma}, \prec\rangle$ (using the fact that star-free languages are exactly those definable over strings considered as finite models).

Before we prove the collapse, we collect a few more properties of $\mathfrak{M}$. The following is true for any finite (and hence pseudo-finite) set A in any structure $\mathfrak{M}'$ elementarily equivalent to $\mathfrak{M}$. The meet of all elements of A equals to the meet of some two elements of A. Moreover, for any $c \in \mathfrak{M}'$, $c \sqcap A$, the longest prefix of c that is also a prefix of some element of A, equals $c \sqcap a$ for some $a \in A$. Furthermore, there exist four not necessarily distinct elements $a_1, a_2, a_3, a_4 \in A$ such that $a_1 \sqcap a_2 \preceq c \sqcap A \preceq a_3 \sqcap a_4$, and there are no a', a'' such that $a_1 \sqcap a_2 \prec a' \sqcap a'' \prec a_3 \sqcap a_4$.

We shall use the following known result on definability in $\mathfrak{M}$. Every formula $\varphi(\vec{x})$ is equivalent to a disjunction of the formulae $\alpha_i(\vec{x}) \wedge \beta_i(\vec{x})$ such that the following is true. Each $\alpha_i(\vec{x})$ is a quantifier-free formula that specifies, for each x_i, x_j, x_k, x_l, whether $x_i \sqcap x_j = \epsilon$, and whether $x_i \sqcap x_j \prec x_k \sqcap x_l$. Each $\beta_i(\vec{x})$ is a conjunction of the formulae $P_L(x_i \sqcap x_j, x_k \sqcap x_l)$ where $\alpha_i(\vec{x})$ implies that there are no elements of the form $x_p \sqcap x_q$ such that $x_i \sqcap x_j \prec x_p \sqcap x_q \prec x_k \sqcap x_l$.

We now show that $\mathfrak{M}$ has the isolation property, and thus admits the restricted quantifier collapse. Let $\mathfrak{M}'$ be elementarily equivalent to $\mathfrak{M}$, A a pseudo-finite set, and $c \in \mathfrak{M}'$. Find (at most) four elements $a_1, a_2, a_3, a_4 \in A$ such that $a_1 \sqcap a_2 \preceq c \sqcap A \preceq a_3 \sqcap a_4$, and there are no a', a'' such that $a_1 \sqcap a_2 \prec a' \sqcap a'' \prec a_3 \sqcap a_4$. Then the above result characterizing definability in $\mathfrak{M}$ easily implies that $\text{tp}_{\mathfrak{M}'}(c/\{a_1, a_2, a_3, a_4\})$ isolates $\text{tp}_{\mathfrak{M}}(c/A)$. □

The notion of isolation could as well be called ω-isolation: a type over a set is isolated by a type of a subset of cardinality $< \omega$. We could then introduce a notion of λ-isolation for any cardinal λ. The cardinal λ of interest to us here is ω_1; the notion of ω_1-isolation says that $\mathrm{tp}_{\mathfrak{M}}(a/A)$, for A pseudo-finite, is isolated by $\mathrm{tp}_{\mathfrak{M}}(a/A')$, where $A' \subseteq A$ is finite or countable. Just as ω-isolation implies ω-PFH and the restricted quantifier collapse, ω_1-isolation implies ω_1-PFH, and thus the collapse. We shall now use ω_1-isolation to give an alternative proof of the restricted quantifier collapse for $\mathfrak{Z} = \langle \mathbb{Z}, +, < \rangle$. We already know this result: Proposition 1.32 showed the natural-active collapse for $\mathfrak{Z}_0$, which is an expansion of $\mathfrak{Z}$ that has quantifier-elimination. But we provide the proof below to illustrate the power of model-theoretic techniques.

Proposition 1.46. $\mathfrak{Z}$ *admits the restricted quantifier collapse.*

Proof. Let $\mathfrak{M}$ be a model of $\mathrm{Th}(\mathfrak{Z})$, and A a pseudo-finite set in $\mathfrak{M}$. Since A is pseudo-finite, for any a, either there exist $a_1 < a_2 \in A$ such that $a_1 \leq a \leq a_2$ and $(a_1, a_2) \cap A = \emptyset$, or $a > m$, where m is the maximal element of A, or $a < m'$, where m' is the minimal element of A. We assume, without loss of generality, that we deal with the first case.

Let $f(\vec{y})$ be a linear function with integer coefficients. For any finite set A, and an element a, we have a uniquely defined tuple $\vec{b}_-^{A,f}$ of elements of A such that $f(\vec{b}_-^{A,f}) \leq a$, and for any other tuple $\vec{c}$ of elements of A, either $f(\vec{c}) > a$, or $f(\vec{c}) < f(\vec{b}_-^{A,f})$, or $f(\vec{c}) = f(\vec{b}_-^{A,f})$ and $\vec{c}$ is above $\vec{b}_-^{A,f}$ in the lexicographic ordering. In other words, $\vec{b}_-^{A,f}$ is the lexicographically smallest tuple of elements of A on which f reaches its maximum value which does not exceed a. Since the above can be stated in FO, such a tuple $\vec{b}_-^{A,f}$ is uniquely determined for a pseudo-finite set A.

Similarly, define $\vec{b}_+^{A,f}$ to be the lexicographically smallest tuple of elements of A on which f reaches its minimum value which lies above a. Again, this is well-defined for a pseudo-finite set A.

We now let A' be the set that has a_1, a_2 and all the components of all $\vec{b}_-^{A,f}$ and $\vec{b}_+^{A,f}$ as f ranges over all linear functions with integer coefficients. Since such tuples are unique for each f, the set A' is countable. We claim that $\mathrm{tp}_{\mathfrak{Z}}(a/A')$ isolates $\mathrm{tp}_{\mathfrak{Z}}(a/A)$.

For this, it is convenient to use $\mathfrak{Z}_0$, the expansion of $\mathfrak{Z}$ with $\sim_k, k > 1$, that admits quantifier-elimination. Suppose $\mathrm{tp}_{\mathfrak{Z}}(a/A') = \mathrm{tp}_{\mathfrak{Z}}(a'/A')$; it then suffices to show that $\mathfrak{Z}_0$-atomic types of a and a' over A are the same. As $\mathrm{tp}_{\mathfrak{Z}}(a/A')$ specifies all $a - a_1 \sim_k n_k$ and $a_2 - a \sim_k n'_k$ relations for all $k > 1$, and all constants $a_1, a_2 \in A'$, a and a' agree on all the formulae $f(x, \vec{y}) - g(x, \vec{y}) \sim_k n_k$, where f, g are linear functions, $\vec{y}$ takes values in A, and $0 \leq n_k < k$. By quantifier elimination for $\mathfrak{Z}_0$, we may assume that other atomic formulae are of the form $x \theta f(\vec{y})$, where f is a linear function with integer coefficients, and θ is one of $<, >, =$. Suppose that $a > f(\vec{b})$ holds for some $\vec{b}$ over A. Then either $f(\vec{b}) < f(\vec{b}_-^{A,f})$, or $f(\vec{b}) = f(\vec{b}_-^{A,f})$ and $\vec{b}_-^{A,f}$ is lexicographically smaller than $\vec{b}$. Since all the components of $\vec{b}_-^{A,f}$ are in A' and the types of a and a'

over A' are the same, we conclude that $a' > f(\vec{b}_-^{A,f})$ and thus $a' > f(\vec{b})$. The cases of θ being $>$ and $=$ are similar. Hence, $\mathrm{tp}_{\mathfrak{Z}_0}(a/A) = \mathrm{tp}_{\mathfrak{Z}_0}(a'/A)$, which proves ω_1-isolation. □

In conclusion, we remark that the techniques of the two previous subsections – using the finite cover property or isolation to prove the collapse – are completely disjoint. While every structure that does not have the finite cover property is stable, every structure with the isolation property is unstable; in particular, one can define an infinite linear order on such a structure.

1.8 VC dimension and collapse results

In this section we consider the relationship between the VC (Vapnik-Chervonenkis) dimension, a concept from statistics and learning theory, and collapse results. We have seen one powerful result (Theorem 1.35): any structure whose definable families have finite VC dimension, admits the natural generic collapse. It turns out that VC dimension is even closer related to collapse results: namely,

natural-active collapse	⇒	restricted quantifier collapse	⇒	finite VC dimension	⇒	natural-generic collapse,

as the result below demonstrates.

Theorem 1.47. *Let $\mathfrak{M}$ admit the restricted quantifier collapse. Then $\mathfrak{M}$-definable families have finite VC dimension.*

Proof. In the proof we shall use a complexity class AC^0/poly defined as follows. (We use a slightly nonstandard definition, in terms of FO-formulae rather than circuits, as it is more convenient for our purposes.) Consider a class of finite SC-structures $\mathcal{C}$, and assume that $adom(D)$ of size n is always of the form $\{0,\dots,n-1\}$. Such a class belongs to AC^0/poly if there exists a vocabulary SC' disjoint from SC, a function h from $\mathbb{N}$ to SC'-structures, and a sentence $\Phi_{\mathcal{C}}$ of $\mathrm{FO}(SC \cup SC')$ such that (a) $adom(h(n)) \subseteq \{0,\dots,n-1\}$, and (b) for each SC-structures of size n, $D \in \mathcal{C}$ iff $(D, h(n)) \models \Phi_{\mathcal{C}}$. In other words, we use $\Phi_{\mathcal{C}}$ to decide if $D \in \mathcal{C}$, and $\Phi_{\mathcal{C}}$ uses D as well as some polynomial-size 'advice' $h(n)$. Some strong lower bounds have been proved for AC^0/poly; they imply, for example, that parity, and importantly for us, 3-colorability, are not in AC^0/poly.

Now assume that $\mathfrak{M}$ admits the restricted quantifier collapse, and has infinite VC dimension. We obtain a contradiction by showing that 3-colorability is in AC^0/poly.

To proceed, we need the following known (and nontrivial) result: if $\mathfrak{M}$ has infinite VC dimension, then there is a formula $\varphi(\vec{x}, y)$ (where y is a single variable) that defines a family of infinite VC dimension. Take this formula φ; then for each n, there is a set $Y_n \subset \mathfrak{M}$ of size n that is shattered by $\{\varphi(\vec{a}, \mathfrak{M}) \mid \vec{a}\}$.

Now expand the language of $\mathfrak{M}$ with a binary relation E (to be interpreted as a finite graph), and consider the sentence Ψ:

$$\exists \vec{x}_1 \exists \vec{x}_2 \exists \vec{x}_3 \left[\forall y \in adom \begin{pmatrix} (\varphi(\vec{x}_1, y) \wedge \neg\varphi(\vec{x}_2, y) \wedge \neg\varphi(\vec{x}_3, y)) \\ \vee (\neg\varphi(\vec{x}_1, y) \wedge \varphi(\vec{x}_2, y) \wedge \neg\varphi(\vec{x}_3, y)) \\ \vee (\neg\varphi(\vec{x}_1, y) \wedge \varphi(\vec{x}_2, y) \wedge \varphi(\vec{x}_3, y)) \end{pmatrix} \wedge \right.$$

$$\left. \forall y_1 \in adom \forall y_2 \in adom \; E(y_1, y_2) \rightarrow \neg \begin{pmatrix} (\varphi(\vec{x}_1, y_1) \wedge \varphi(\vec{x}_1, y_2)) \\ \vee (\varphi(\vec{x}_2, y_1) \wedge \varphi(\vec{x}_2, y_2)) \\ \vee (\varphi(\vec{x}_3, y_1) \wedge \varphi(\vec{x}_3, y_2)) \end{pmatrix} \right]$$

The fact that φ defines a family that shatters each Y_n lets us model second-order quantifiers over Y_n; in particular, for any graph G with $adom(G) \subseteq Y_n$, $G \models \Psi$ iff G is 3-colorable.

Since $\mathfrak{M}$ admits the restricted quantifier collapse, we may assume that Ψ is equivalent to a sentence Ψ' of the form

$$\mathbf{Q} z_1 \in adom \ldots \mathbf{Q} z_m \in adom \;\; \alpha(\vec{z})$$

where α is a Boolean combination of formulae $E(z_i, z_j)$ and formulae $\beta_l(\vec{z})$, $l \leq k$, over $\mathfrak{M}$.

For each β_l having p free variables, introduce a new p-ary relation symbol R_l. Let $SC' = \{R_l \mid l \leq k\}$. Next, for each n, fix a bijection $\pi_n : \{0, \ldots, n-1\} \rightarrow Y_n$. Let $h(n)$ be a SC'-structure on $\{0, \ldots, n-1\}$ in which a tuple $(a_1, \ldots, a_p)$ belongs to R_l iff $\beta_l(\pi_n(a_1), \ldots, \pi_n(a_p))$ holds in $\mathfrak{M}$. Finally, let Ψ'' be Ψ' in which every subformula $\beta_l(\vec{u})$ is replaced by $R_l(\vec{u})$. We then conclude that for any graph G on nodes $\{0, \ldots, n-1\}$, $(G, h(n)) \models \Psi''$ iff G is 3-colorable, which contradicts the fact that 3-colorability is not in AC^0/poly. This proves the theorem. $\square$

A natural question, then, is the following: what kind of bounds on $\mathrm{FO}(SC, \mathfrak{M})$ can one show for structures $\mathfrak{M}$ of *infinite* VC dimension? Clearly we cannot hope to prove the natural-active or the restricted quantifier collapse; but is it possible to prove some meaningful bounds, and if so, how?

While our understanding of the limits of collapse results is by no means complete, in the remainder of this section we give three examples of very different behavior of FO over finite models embedded into structures with infinite VC dimension.

- In some cases, there is no collapse at all. We have seen that any computable query over finite SC-structures can be expressed in $\mathrm{FO}(SC, \mathfrak{N})$, and $\mathfrak{N}$ has infinite VC dimension. (To see this directly, assume as we did before that a set $X = \{x_1, \ldots, x_k\}$ with $x_1 < \ldots < x_k$ is coded by $2^{x_1} \cdot 3^{x_2} \cdot \ldots \cdot p_k^{x_k}$ where p_k is the kth prime. Let $\varphi(x, y)$ say that y in the set coded by x. Then the family $\{\varphi(n, \mathfrak{N}) \mid n \in \mathbb{N}\}$ has infinite VC dimension.)

- In another example, we get a collapse to a logic which is more powerful than FO. Namely, we shall show in Section 1.8.1, that over the random graph $\mathcal{RG}$, $\mathrm{FO}(SC, \mathcal{RG})$ collapses to active MSO, $\mathrm{MSO}_{\mathrm{act}}(SC, \mathcal{RG})$. Recall that MSO, monadic second-order logic, extends FO with quantification over sets. In the active version $\mathrm{MSO}_{\mathrm{act}}$, this set quantification is over subsets of $adom(D)$.
- In the last example, we do not know whether the natural-generic collapse could be proved. Nevertheless, we succeed in showing that generic queries can be evaluated in AC^0. As AC^0 is one of very few complexity classes for which lower bounds have been proved, this suffices to conclude that queries such as parity are not expressible. The structure for which this result is proved (in Section 1.8.2), extends $\langle \Sigma^*, f_a, \prec\rangle$ from the previous section by adding string length comparisons.

1.8.1 Random graph and collapse to MSO

In this section, we give an example of a nicely-behaved structure, with decidable theory and quantifier elimination, that does not admit the natural-active collapse. This structure, however, admits a collapse to monadic second-order logic, MSO.

The structure is the random graph $\mathcal{RG} = \langle U, E\rangle$ on a countably infinite set U: that is, any model that satisfies every sentence that is true in almost all finite undirected graphs. Here 'almost all' is with respect to the uniform probability distribution: $E(a, b)$ holds with probability $1/2$, independently for each pair (a, b). It is known that the set of all such sentences forms a complete theory with infinite models, and that this theory is decidable and ω-categorical. The latter means that up to isomorphism, there is only one countable model.

Other, non-probabilistic descriptions of $\mathcal{RG}$ exist. For example, let $U = \{u_0, u_1, \ldots\}$, and define E as follows: $(u_i, u_j) \in E$ iff either the $i$the bit on the binary representation of j, or the jth bit in the binary representation of i, is 1.

The random graph satisfies the following *extension axioms*, for each $n > 0$:

$$\forall x_1, \ldots, x_n \bigwedge_{i \neq j} x_i \neq x_j \rightarrow \Big(\bigwedge_{M \subseteq \{1,\ldots,n\}} \exists z \notin \vec{x} \; (\bigwedge_{i \in M} E(z, x_i) \wedge \bigwedge_{j \notin M} \neg E(z, x_j)) \Big)$$

In other words, let T be a finite subset of U and $S \subseteq T$. Then the extension axioms say that there exists $z \notin T$ such that for all $x \in S$, $(z, x) \in E$, and for all $x \in T - S$, $(z, x) \notin E$. It is immediately clear from the extension axioms that $\mathcal{RG}$ has infinite VC dimension; in fact, the family definable by the formula $E(x, y)$ shatters arbitrarily large finite sets.

Recall that MSO, monadic second-order logic, is a restriction of second-order logic in which second-order variables range over sets. In the active-domain fragment of MSO, they range over subsets of $adom(D)$.

Theorem 1.48. $\mathrm{FO}(SC, \mathcal{RG}) = \mathrm{MSO}_{\mathrm{act}}(SC, \mathcal{RG})$.

Proof. The idea is to use the extension axioms to model MSO queries. Consider an $\mathrm{MSO}_{\mathrm{act}}$ formula

$$\varphi(\vec{x}) \;\equiv\; \mathbf{Q}X_1 \subseteq adom \ldots \mathbf{Q}X_m \subseteq adom\; \mathbf{Q}y_1 \in adom \ldots \mathbf{Q}y_n \in adom\; \alpha(\vec{X}, \vec{x}, \vec{y}),$$

where the X_is are second-order variables, the y_js are first-order variables, and α is a Boolean combination of SC- and $\mathcal{RG}$-formulae in variables $\vec{x}, \vec{y}$, and formulae $X_i(x_j)$ and $X_i(y_j)$. Construct a new $\mathrm{FO}(SC, \mathcal{RG})$ formula $\varphi'(\vec{x})$ by replacing each $\mathbf{Q}X_i \subseteq adom$ with $\mathbf{Q}z_i \notin adom \cup \vec{x}$ (which is FO-definable), and changing every atomic subformula $X_i(u)$ to $E(z_i, u)$. It is then easy to see, from the extension axioms, that φ' is equivalent to φ.

For the other direction, proceed by induction on the formulae. The only nontrivial case is that of unrestricted existential quantification. Suppose we have an $\mathrm{MSO}_{\mathrm{act}}(SC, \mathcal{RG})$ formula $\varphi(\vec{x}, z)$, with $\vec{x} = (x_1, \ldots, x_n)$, of the form

$$\mathbf{Q}\vec{X} \subseteq adom\;\; \mathbf{Q}\vec{y} \in adom\;\; \alpha(\vec{X}, \vec{x}, \vec{y}, z),$$

where α again is a Boolean combination of atomic SC- and $\mathcal{RG}$-formulae, as well as formulae $X_i(u)$, where u is one of the first-order variables $z, \vec{x}, \vec{y}$. We want to find an $\mathrm{MSO}_{\mathrm{act}}$ formula equivalent to $\exists z\; \varphi$.

Such a formula is a disjunction of $\exists z \in adom\; \varphi \;\vee\; \bigvee_i \varphi(\vec{x}, x_i) \;\vee\; \exists z \notin adom\; \varphi$. The former is an $\mathrm{MSO}_{\mathrm{act}}(SC, \mathcal{RG})$ formula. To eliminate z from the latter, all we have to know about z is its connections to $\vec{x}$ and to the active domain in the random graph; the former is taken care of by a disjunction listing all subsets of $\{1, \ldots, n\}$, and the latter by a second-order quantifier over the active domain. For $I \subseteq \{1, \ldots, n\}$, let $\chi_I(\vec{x})$ be a quantifier-free formula saying that no x_i, x_j with $i \in I, j \notin I$, could be equal. Introduce a new second-order variable Z and define an $\mathrm{MSO}_{\mathrm{act}}$ formula $\psi(\vec{x})$ as

$$\exists Z \subseteq adom \bigvee_{I \subseteq \{1,\ldots,n\}} \Big(\chi_I(\vec{x}) \wedge \mathbf{Q}\vec{X} \subseteq adom\;\; \mathbf{Q}\vec{y} \in adom\;\; \alpha_I^Z(\vec{X}, Z, \vec{x}, \vec{y})\Big),$$

where $\alpha_I^Z(\vec{X}, Z, \vec{x}, \vec{y})$ is obtained from α by:

1. replacing each $E(z, x_i)$ by *true* for $i \in I$ and *false* for $i \notin I$,
2. replacing each $E(z, y_j)$ by $Z(y_j)$, and
3. replacing each $X_i(z)$ by *false*.

The extension axioms then ensure that ψ is equivalent to $\exists z \notin adom\; \varphi$. $\square$

Thus, $\mathcal{RG}$ provides an example of a structure with quantifier-elimination and decidable first-order theory that does not admit the natural-active collapse. At the same time, one can establish meaningful bounds on the expressiveness of queries over $\mathcal{RG}$: for example, each generic query in $\mathrm{FO}(SC, \mathcal{RG})$

is in $\mathrm{MSO}_{\mathrm{act}}(SC)$. (This does not immediately follow from the active-generic collapse, as we do not include any order relation. One can show that the order is not needed by modifying the proof of Lemma 1.14 using some special properties of $\mathcal{RG}$.) Thus, every generic query in $\mathrm{FO}(SC, \mathcal{RG})$ can be evaluated in PSPACE (in fact, even in the polynomial hierarchy).

1.8.2 Complexity bounds for generic queries

We now revisit the structure $\langle \Sigma^*, (f_a)_{a \in \Sigma}, \prec \rangle$ from Section 1.7.3. Recall that Σ here is a finite alphabet, Σ^* is the set of all finite strings over Σ, f_a is a function that adds a at the end of its argument, and $\prec$ is the prefix relation. We now extend it to a structure $\mathcal{S} = \langle \Sigma^*, (f_a)_{a \in \Sigma}, \prec, \mathrm{el} \rangle$, which adds a binary predicate el interpreted as follows: $\mathrm{el}(x, y)$ iff $|x| = |y|$, where $|\ |$ stands for the length of a finite string.

Despite looking rather arbitrary, this structure naturally arises in the study of logical properties of formal languages, and has a number of nice properties. For example, subsets of Σ^* definable in $\mathcal{S}$ are precisely the regular languages. Moreover, in a certain sense, $\mathcal{S}$ is the most general structure whose definable relations are precisely tuples of strings accepted by finite automata. That is, any other structure on Σ^* whose definable relations are tuples accepted by finite automata, can be interpreted in $\mathcal{S}$. The characterization of definable relations via automata also implies the decidability of the theory of $\mathcal{S}$.

Using the isolation property, we proved the restricted quantifier collapse for $\langle \Sigma^*, (f_a)_{a \in \Sigma}, \prec \rangle$. However, it is impossible to prove the collapse for $\mathcal{S}$ as its definable families may have infinite VC dimension. To see this, let $\Sigma = \{a, b\}$, and consider a formula $\varphi(x, y)$ saying: there is a prefix of x that has the same length as y and ends with an a:

$$\exists z \exists v \ \big(z \preceq x \ \wedge \ \mathrm{el}(z, y) \ \wedge \ f_a(v) = z\big)$$

For each n, let $A_n = \{b^i \mid i \leq n\}$, and let A be an arbitrary subset of A_n. Let s_A be a string of length n whose ith position is a iff $b^i \in A$. Then for each $i \leq n$, $\varphi(s_A, b^i)$ holds iff $b^i \in A$. This shows that arbitrarily large finite sets can be shattered by families definable in $\mathcal{S}$.

This still leaves open the possibility of proving the natural-generic collapse for $\mathcal{S}$; however, we do not know if it holds in $\mathcal{S}$. Still, we can prove reasonably good bounds for $\mathrm{FO}(SC, \mathcal{S})$. For this, we need the complexity class $\mathrm{AC}^0/\mathrm{poly}$ used in Theorem 1.47. As this class is a very modest extension of $\mathrm{FO}_{\mathrm{act}}(SC, <$), some good bounds can be derived.

Proposition 1.49. *Every generic query in* $\mathrm{FO}(SC, \mathcal{S})$ *can be evaluated in* $\mathrm{AC}^0/\mathrm{poly}$. *In particular, queries such as parity and connectivity are not expressible in* $\mathrm{FO}(SC, \mathcal{S})$.

Proof sketch. First, we explain the complexity model used here, which is applicable to evaluation of *generic queries*. Given a SC-structure D with

$|adom(D)| = n$, we code elements of the active domain by the numbers $0, \ldots, n-1$ represented in binary, and then code tuples and relations in a standard fashion, using special delimiter characters. Using this coding, one shows that every generic sentence Φ can be evaluated in AC^0/poly. This is done in three steps:

1. First, we show that it suffices to restrict quantification to stings of length at most m_D, where $m_D = \max\{|x| \mid x \in adom(D)\}$. This is proved by an Ehrenfeucht-Fraïssé game argument. More precisely, one shows the following. Let $\Sigma^{\leq m} = \{x \in \Sigma^* \mid |x| \leq m\}$. Then, for each SC, there is a fixed constant l_{SC}, such that if the duplicator can win in $k + l_{SC}$ moves on the restrictions of $(\mathcal{S}, D_1)$ and $(\mathcal{S}, D_2)$ to to $(\Sigma^{\leq m_{D_1}}, D_1)$ and $(\Sigma^{\leq m_{D_1}}, D_2)$, then the duplicator can win in k moves on $(\mathcal{S}, D_1)$ and $(\mathcal{S}, D_2)$.
2. Second, define an ordering $<$ on Σ^*: $x < y$ if either $|x| < |y|$, or $|x| = |y|$ and x is lexicographically less than y. Viewing Σ^* as an infinite tree, this amounts to traversing it, level by level, from left to right. Now, by genericity, we may assume that $adom(D)$ is an initial segment of this ordering $<$.
3. Finally, we define an advice function f that for each n codes all the relations of $\mathcal{S}$ on the first n' elements of Σ^* in the order $<$. Here n' is the number of all strings of length at most m, where m is the length of the nth string in the $<$-order. For a given SC-structure D with $|adom(D)| = n$, f codes all the relations of $\mathcal{S}$ on $\Sigma^{\leq m_D}$. Assuming that $adom(D)$ is an initial segment of $<$, we conclude that the size of $f(n)$ is polynomial in n. By 1), we know that quantification over $\Sigma^{\leq m_D}$ suffices. As $f(n)$ provides all the information about $\mathcal{S}$ on $\Sigma^{\leq m_D}$, we conclude that with f, a generic query can be expressed in FO, and thus it belongs to AC^0/poly. □

1.9 Expressiveness of Constraint Query Languages

In this section we return to constraint databases and study the expressive power of standard query languages such as FO + Lin and FO + Poly. We shall mostly deal with the fundamental topological property of connectivity, which is also important in many applications of constraint databases as spatial databases. That is, we deal with the following problem:

Problem:	CONNECTIVITY
Input:	an $\mathfrak{M}$-definable set $S \subseteq \mathbb{R}^k$;
Output:	True, if S is topologically connected, and false, otherwise.

The question is whether CONNECTIVITY is definable in $\mathrm{FO}(SC, \mathfrak{M})$ where SC consists of just S. We shall mostly deal with the cases where $\mathfrak{M}$ is the real field or the real ordered group (and thus S is semi-algebraic or semi-linear);

then by definability we mean definability in FO + POLY and FO + LIN. We remarked in Section 1.3 that the problem looks akin to the problem of finite graph connectivity, simply because any finite graph can be embedded into $\mathbb{R}^3$ without self-intersections, and the result of the embedding is topologically connected iff the original graph is connected. At that point, we did not know if FO + POLY and FO + LIN define graph connectivity. Now we know that they do not. However, we choose a different and less ad-hoc way to proceed, as the results we present here give us more than nondefinability of connectivity, and can be used for dimensions 1 and 2 as well.

In the next section, we shall see a reduction from topological connectivity to some definability problem for embedded finite models. In Section 1.9.2 we present a different technique, based on the topological structure of definable sets. In Section 1.9.3 we study queries that separate FO + POLY from FO + LIN.

1.9.1 Reductions to the finite case

Recall that MAJORITY is the following problem: "given two finite sets A and B, is $card(A) > card(B)$?" We now prove the following.

Proposition 1.50. *Assume that* FO + POLY *can define* CONNECTIVITY *when the input is restricted to semi-linear sets. Then* FO + POLY *can define* MAJORITY.

Proof. Suppose we are given two finite sets A and B. Assume without loss of generality that $a, b > 0$ for all $a \in A$ and $b \in B$ (if not, add $\max_{a \in A} |a| + 1$ to all elements of A, and likewise for B; this can be defined in FO + LIN). Let $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$, where $a_1 < \ldots < a_n$ and $b_1 < \ldots < b_m$. This is shown in Figure 1.6 for $n = 6$ and $m = 4$.

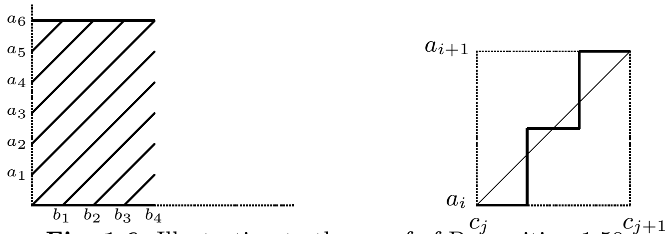

Fig. 1.6. Illustration to the proof of Proposition 1.50

Let $C = B \cup \{0\}$. Assume that $C = \{c_1, \ldots, c_{m+1}\}$ where $c_1 = 0$, and $c_i = b_{i-1}$ for $1 < i \leq m + 1$. For each $1 \leq i < n$ and $1 \leq j < m + 1$, define a semi-linear set X_{ij} in $\mathbb{R}^2$ as the union of the following 5 sets:

$$\begin{array}{l} X_{ij}^1 = \{(x,y) \mid y = a_i, c_j \le x \le \frac{2c_j}{3} + \frac{c_{j+1}}{3}\} \\ X_{ij}^2 = \{(x,y) \mid x = \frac{2c_j}{3} + \frac{c_{j+1}}{3}, a_i \le y \le \frac{a_i + a_{i+1}}{2}\} \\ X_{ij}^3 = \{(x,y) \mid y = \frac{a_i + a_{i+1}}{2}, \frac{2c_j}{3} + \frac{c_{j+1}}{3} \le x \le \frac{c_j}{3} + \frac{2c_{j+1}}{3}\} \\ X_{ij}^4 = \{(x,y) \mid x = \frac{c_j}{3} + \frac{2c_{j+1}}{3}, \frac{a_i + a_{i+1}}{2} \le y \le a_{i+1}\} \\ X_{ij}^5 = \{(x,y) \mid y = a_{i+1}, \frac{c_j}{3} + \frac{2c_{j+1}}{3} \le x \le c_{j+1}\}. \end{array}$$

This is shown in the right picture in Figure 1.6: the five sets corresponds to the five segments of the thick line. We then define a set X as

$$\{(x,0) \mid a_1 \le x \le a_n\} \ \cup \ \{(x,a_n) \mid a_1 \le x \le a_n\} \ \cup \ \bigcup_{i=1}^{n-1} \bigcup_{j=1}^{m} X_{ij}.$$

This set is shown in the left picture in Figure 1.6 (in fact, we show lines as straight, but it should be kept in mind that in every rectangle $[c_j, c_{j+1}] \times [a_i, a_{i+1}]$ it is given by X_{ij}).

We next observe that X is definable in $\mathrm{FO} + \mathrm{Lin}$ from A and B. Indeed, C is definable, and then every X_{ij} is definable, as follows from its definition. (The main reason for going from (c_j, a_i) to (c_{j+1}, a_{i+1}) by "steps" rather than a straight line was to achieve definability in $\mathrm{FO} + \mathrm{Lin}$.) Secondly, $\mathit{card}(B) \ge \mathit{card}(A)$ iff the set X is connected – this is because the "line" from $(0,0)$ reaches the ceiling iff $\mathit{card}(B) \ge \mathit{card}(A)$. Thus, X is connected iff MAJORITY is false on A and B, which completes the proof. □

We immediately derive from this and the fact that $\mathrm{FO} + \mathrm{Lin}$ suffices to construct X from A and B:

Corollary 1.51. *Neither* $\mathrm{FO} + \mathrm{Lin}$ *nor* $\mathrm{FO} + \mathrm{Poly}$ *can define* CONNECTIVITY. *Furthermore,* CONNECTIVITY *is not definable in* $\mathrm{FO}(\{S\}, \mathfrak{M})$ *if* $\mathfrak{M}$ *is an o-minimal expansion of the real field* $\mathbf{R}$. □

The reduction technique is not limited to the CONNECTIVITY problem. We invite the reader to draw simple pictures that give similar reductions for problems like homeomorphism of two 2-dimensional sets, existence of exactly one (or at most one, or at least one) hole, or being simply connected.

1.9.2 Topological properties

In this section we give a different proof that topological connectivity is not definable in $\mathrm{FO} + \mathrm{Poly}$. The proof relies on topological properties of semi-algebraic sets, and on a criterion for indistinguishability of two sets in $\mathbb{R}^2$ by certain $\mathrm{FO} + \mathrm{Poly}$ queries.

Note that connectivity is a query about topological properties of its input. Formally, a Boolean query Q on sets in $\mathbb{R}^k$ is called *topological* if it is invariant under homeomorphisms: for any homeomorphism $h : \mathbb{R}^k \to \mathbb{R}^k$ and any $S \subseteq \mathbb{R}^k$, $Q(S)$ is true iff $Q(h(S))$ is true. Examples of topological queries are connectivity, having exactly one hole, having exactly k connected components. Examples of nontopological queries are properties such as "being a line", "containing the origin" etc.

It turns out that the expressive power of FO + POLY with respect to topological queries on $\mathbb{R}^2$ can be nicely characterized. The characterization is based on the fact that every semi-algebraic set S is locally *conic* around any point. This is illustrated in Figure 1.7: there is a small neighborhood of a point $\vec{x}$ such that the intersection of this neighborhood with the set S is isotopic to the cone with center in $\vec{x}$ and the base that is the intersection of S with the boundary of the neighborhood.

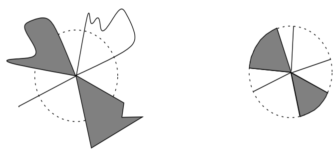

Fig. 1.7. Cones

More precisely, let $B_\epsilon(\vec{x})$ be $\{\vec{y} \in \mathbb{R}^2 \mid \parallel \vec{y} - \vec{x} \parallel \leq \epsilon\}$ and $B^\circ_\epsilon(\vec{x}) = \{\vec{y} \in \mathbb{R}^2 \mid \parallel \vec{y} - \vec{x} \parallel = \epsilon\}$. Then for each semi-algebraic set S and $\vec{x} \in \mathbb{R}^2$, there is $\epsilon > 0$ such that $S \cap B_\epsilon(\vec{x})$ is isotopic to the cone with the center in $\vec{x}$ and the base $B^\circ_\epsilon(\vec{x}) \cap S$. Furthermore, for any $\epsilon' < \epsilon$, $B_{\epsilon'}(\vec{x}) \cap S$ is isotopic to the same cone, so we can talk about the topological type of a cone of S around $\vec{x}$. We shall use $\mathsf{tp}_S(\vec{x})$ to denote the topological type of such a cone.

There are four cone types that are of special interest: the full cone, the half-cone, the line, and the empty cone, shown in Figure 1.8. The first is the cone type of a point in the interior of a set S. The second is the cone type of a point on the boundary of a two-dimensional region. The third is the type of a point in a one-dimensional segment of S. And the last one is the type of a point outside of S, or of an isolated point of S. It turns out that for any closed semi-algebraic set $S \subseteq \mathbb{R}^2$, these are the only cone types that can be realized by infinitely many points – all other cone types have only finitely many realizers.

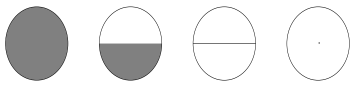

Fig. 1.8. Four cone types

We write $S \sim_{\mathsf{tp}} S'$ if for every topological type T of a cone,

$$card(\{x \in S \mid \mathsf{tp}_S(x) = T\}) \quad = \quad card(\{x \in S' \mid \mathsf{tp}_{S'}(x) = T\}).$$

Note that this condition is somewhat reminiscent of that for Hanf-locality, which says that each local neighborhood must have equally many realizers in two structures.

Cone types characterize the expressive power of FO + Poly with respect to topological queries as follows.

Theorem 1.52. *Let* Q *be a topological* FO + Poly *query over the schema with one binary relation, and let* $S \sim_{\mathsf{tp}} S'$*, where* S, S' *are closed semi-algebraic sets in* $\mathbb{R}^2$*. Then* $Q(S)$ *is true iff* $Q(S')$ *is true.* □

The proof of this result is rather involved. The main idea is as follows. It is possible to define a set of elementary transformations on closed semi-algebraic subsets of $\mathbb{R}^2$ such that these transformations preserve elementary equivalence with respect to topological FO + Poly sentences, and such that every two sets satisfying $S \sim_{\mathsf{tp}} S'$ can be transformed to the same subset of $\mathbb{R}^2$.

Another proof that connectivity is not in FO + Poly

Suppose that connectivity is tested by a (topological) FO + Poly query Q. Consider S_1 and S_2 shown in Figure 1.9: S_1 is a disk, and S_2 is a disjoint union of two disks. Both S_1 and S_2 realize the same cone types (the full, the half, and the empty cones), and both have infinitely many realizers for each of these types. Thus, $S_1 \sim_{\mathsf{tp}} S_2$, and by Theorem 1.52 we must have $Q(S_1)$ iff $Q(S_2)$. Thus, Q cannot define connectivity, as S_1 is connected, and S_2 is not.

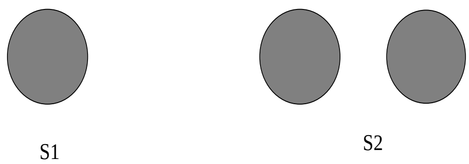

Fig. 1.9. Proving that connectivity is not in FO + Poly

It is natural to ask whether Theorem 1.52 can be extended to schemas with two or more relation symbols, in particular, to topological queries over multiple regions on the plane. It turns out that the answer is negative.

Suppose that we have two relation symbols, S and T, and assume that S is interpreted as the area shown in light grey, and T as the area shown in dark grey. Figure 1.10 gives two instances of (S, T): in instance $\mathcal{I}_1$ on the left, T lies inside S, and in instance $\mathcal{I}_2$ on the right, S lies inside T.

We can see that $\mathcal{I}_1 \sim_{\mathsf{tp}} \mathcal{I}_2$, as both instances realize the same cone types. At the same time, $\mathcal{I}_1$ and $\mathcal{I}_2$ can be separated by a topological FO + Poly query. The latter statement is by no means trivial. An obvious way to separate $\mathcal{I}_1$ from $\mathcal{I}_2$ would be by saying: "traversing any line from $-\infty$ to $+\infty$, we

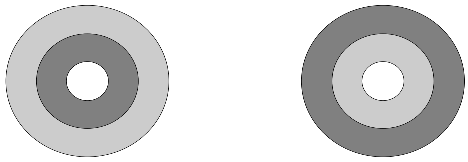

Fig. 1.10. Topological equivalence for multiple regions

first enter S and then T". However, it is easy to show that this property, while expressible in FO + Poly, is not topological. Nevertheless, a rather complicated construction yields a topological FO + Poly query that separates $\mathcal{I}_1$ from $\mathcal{I}_2$.

1.9.3 Linear vs. polynomial constraints

All expressivity bounds proved so far, in the finite and infinite contexts, apply to both FO + Lin and FO + Poly. In this section we show a few queries that separate the two. As $\mathbf{R}$ and $\mathbf{R}_{\text{lin}}$ share many model-theoretic properties, in particular, most properties that were crucial for proving collapse results, new techniques are needed to separate them.

Most separation results are based on the simple observation that multiplication is not definable in $\mathbf{R}_{\text{lin}}$ (indeed, by quantifier-elimination, every $\mathbf{R}_{\text{lin}}$-definable function is piece-wise linear). To show that an FO + Poly query Q is not expressible in FO + Lin we then prove that adding Q to FO + Lin would enable us to define multiplication.

We start with two examples, that can be stated for either finite or semi-linear sets. For both queries, the input is a set $S \subseteq \mathbb{R}^2$. The queries are:

- conv(S), which returns the convex hull of S, and
- collinear(S), which returns the set of triples $s_1, s_2, s_3 \in S$ (that is, a subset of $\mathbb{R}^6$) which are collinear.

We have already seen that conv($\cdot$) is an FO + Poly query. collinear($\cdot$) is expressible in FO + Poly as well, as FO + Poly can test if any three given points (x_1, y_1), (x_2, y_2), (x_3, y_3), are collinear.

Proposition 1.53. *Neither* conv *nor* collinear *is expressible in* FO + Lin*, even if their arguments are finite sets.*

Proof sketch. The main idea is illustrated in Figure 1.11. Assume that collinear is definable in FO + Lin. Suppose we are given four distinct points u, v, w, s in $\mathbb{R}^2$. Then, in FO + Lin, we can test if the lines $l(u, v)$ and $l(w, s)$ passing through u, v and w, s respectively, are parallel. Indeed, such lines are not parallel iff there is a point p such that both collinear(u, v, p) and collinear(w, s, p) hold (Figure 1.11, (b)).

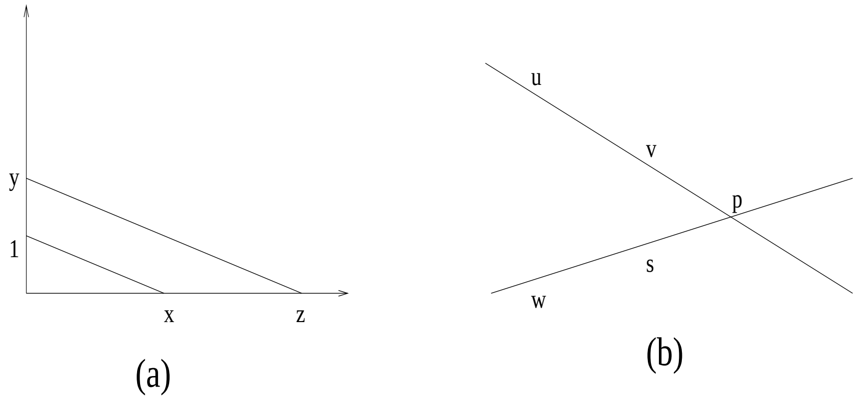

Fig. 1.11. Inexpressibility of conv and collinear in FO + LIN

However, testing if two lines are parallel is sufficient to define multiplication, as shown in Figure 1.11, (a). If the lines passing through $(0, 1)$ and $(x, 0)$, and $(0, y)$ and $(z, 0)$ are parallel, then $z = x \cdot y$. Thus, collinear is not a FO + LIN query.

Finally, conv is not expressible, since three distinct points are collinear iff one of them in the convex hull of two others. $\square$

Note that the query convex(S), testing if an n-dimensional semi-linear set $S \subseteq \mathbb{R}^n$ is convex, can be defined in FO + LIN, as S is convex iff for every two points $(x_1, \ldots, x_n), (y_1, \ldots, y_n) \in S$, the point $(\frac{1}{2}(x_1 + y_1), \ldots, \frac{1}{2}(x_n + y_n))$ is in S. Another positive expressibility result is testing whether a semi-linear set $S \subseteq \mathbb{R}^2$ is a line, since S is a line if either it is a vertical line, or it is the graph of a function, and for any $\vec{x}, \vec{y}, \vec{z} \in S$, $\vec{x} + (\vec{y} - \vec{z}) \in S$. All these conditions are FO + LIN-expressible.

We consider one more example: the query ExistsLine(S) is true iff the set $S \subseteq \mathbb{R}^2$ contains the graph of a line, $\{(x, y) \mid ax + b = y\}$ for some $a, b \in \mathbb{R}$. Along the same lines as the proof of Proposition 1.53, we can show that ExistsLine is not definable in FO + LIN. Indeed, let $u, w \geq 0$ and $v > 1$, and consider the set $S_{u,v,w} \subseteq \mathbb{R}^2$ defined as follows:

$$S_{u,v,w} = \left\{ (x, y) \;\middle|\; \begin{array}{l} \quad x \leq 0, y \leq 0 \\ \text{or } 0 \leq x \leq 1, 0 \leq y \leq v \\ \text{or } 1 \leq x \leq u, v \leq y \\ \text{or } u \leq x, w \leq y. \end{array} \right\}.$$

This set is shown in Figure 1.12. It is easy to see that ExistsLine$(S_{u,v,w})$ is true iff $w \leq u \cdot v$; thus, in FO + LIN + ExistsLine one can define, for example, the set $\{(x, y) \mid y = x^2, x > 1\}$, which is clearly not FO + LIN-definable. Hence, ExistsLine is not an FO + LIN query.

However, not all results separating FO + LIN and FO + POLY are so simple. Consider the following FO + POLY query $\psi(x_1, x_2, y_1, y_2)$:

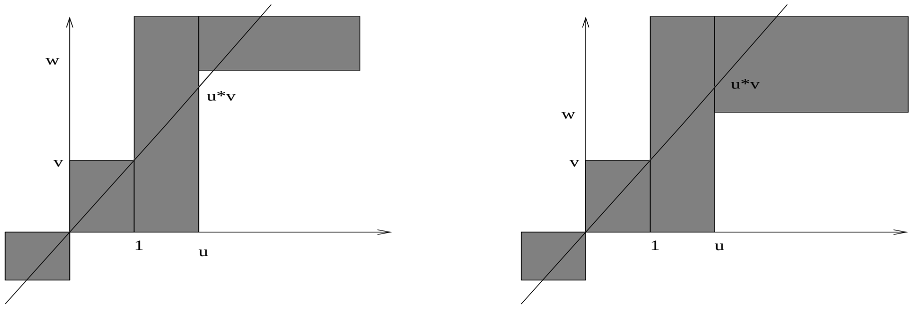

Fig. 1.12. Proving that "contains a line" is not in FO + Lin

$$\forall\lambda\ \Big((0 \le \lambda \le 1)\ \rightarrow\ S(\lambda\cdot x_1 + (1-\lambda)\cdot x_2,\ \ \lambda\cdot y_1 + (1-\lambda)\cdot y_2)\Big),$$

saying that the segment between (x_1, y_1) and (x_2, y_2) is contained in $S \subseteq \mathbb{R}^2$. By the same method we used for ExistsLine, one can show that this is not an FO + Lin query. But now consider a slight modification of this query: suppose we want to know if the segment connecting two points on the *boundary* of a set S lies entirely in S. It turns out that this query is inexpressible in FO + Lin; the proof of this fact, however, is far from obvious.

1.10 Query Safety

In the previous sections, we worked with different kinds of objects: arbitrary FO($SC, \mathfrak{M}$) formulae (for which we proved results like the natural-active collapse) and *queries* definable in FO($SC, \mathfrak{M}$) (for which we proved results like the active generic collapse). Queries, unlike arbitrary formulae, are required to have certain *closure* properties: they return finite outputs on embedded finite models.

This notion of closure is well known in the classical relational database theory under the name of *safety*: one is often interested in looking at only those formulae in $\mathrm{FO_{act}}(SC)$ that return finite results. For example, assuming an infinite domain U and one relation S, the formula $\neg S(x)$ produces the infinite set $U - adom(D)$. It is known that for $\mathrm{FO_{act}}(SC)$, one can identify a recursive subset of safe formulae; that is, the set of formulae that always return finite results on finite SC-structures, and such that every formula with this property is equivalent to one from this set.

In this section we consider the problem of safety in the context of embedded finite models and constraint databases. For the former, we encounter a familiar situation that the behavior of formulae depends greatly on the properties of the underlying structures. For some structures, most notably $\mathbf{R}_{\mathrm{lin}}$ and $\mathbf{R}$ (linear and polynomial constraints) we give nice syntactic characterization. The safety problem also arises in the context of constraint databases. Although the flavor is quite different, we show that it reduces to the finite safety problem.

1.10.1 Finite and infinite query safety

Recall that the output of an $\mathrm{FO}(SC, \mathfrak{M})$ formula $\varphi(x_1, \ldots, x_n)$ on a finite SC-structure D is $\varphi(D) \stackrel{\text{def}}{=} \{\vec{a} \in U^n \mid D \models \varphi(\vec{a})\}$.

Definition 1.54. *An* $\mathrm{FO}(SC, \mathfrak{M})$ *formula* $\varphi(\vec{x})$ *is* safe *on a finite SC-structure D if $\varphi(D)$ is finite. A formula is safe it it is safe on every finite structure.* □

We now define the following problems:

Problem:	SAFETY
Input:	an $\mathrm{FO}(SC, \mathfrak{M})$ formula $\varphi(\vec{x})$;
Output:	True, if φ is safe, and false, otherwise.
Problem:	STATE-SAFETY
Input:	an $\mathrm{FO}(SC, \mathfrak{M})$ formula $\varphi(\vec{x})$ and a finite SC-structure D;
Output:	True, if φ is safe on D, and false, otherwise.

It is known that in general the SAFETY problem is undecidable even for $\mathfrak{M} = \langle U, \emptyset \rangle$ and φ an $\mathrm{FO}_{\mathrm{act}}(SC)$ formula. On the other hand, one can easily show:

Proposition 1.55. *Let $\mathfrak{M} = \langle U, \emptyset \rangle$. Then the* STATE-SAFETY *problem is decidable.*

Proof sketch. By Theorem 1.25, we can assume that φ is an $\mathrm{FO}_{\mathrm{act}}(SC)$ formula. Then it is safe on D iff every tuple in the output only contains elements of $adom(D)$ (by genericity, if at least one tuple contains some element $c \notin adom(D)$, then any other $c' \notin adom(D)$ can be substituted for c). This condition can be easily tested by considering a set $c_1, \ldots, c_m$ of distinct elements not in $adom(D)$, where m is the number of free variables in φ, and checking all tuples in $adom(D) \cup \{c_1, \ldots, c_m\}$. □

We now turn to the safety problem for constraint databases. Consider a situation when we have a linear constraint database $\mathbf{D}$, but we want to write queries against $\mathbf{D}$ in $\mathrm{FO} + \textsc{Poly}$. The main reasons for considering this situation are the following. Linear constraints are used to represent spatial data in many applications, they have several advantages over polynomial constraints: the quantifier-elimination procedure is less costly, and numerous algorithms have been developed to deal with figures represented by linear constraints. As $\mathrm{FO} + \textsc{Lin}$ is more limited than $\mathrm{FO} + \textsc{Poly}$ (for example, it cannot define the convex hull of a set), one may want to use $\mathrm{FO} + \textsc{Poly}$ to get extra expressive power.

However, as soon as the class of constraints used in queries is more general than the class used to define constraint database instances, we encounter the safety problem again: the output of an $\mathrm{FO} + \textsc{Poly}$ query may fail to be semi-linear. More generally, if constraint databases are required to have certain geometric properties, then the safety problem is whether those geometric properties are preserved by a given query language. Section 1.10.4 deals with this problem.

1.10.2 Safe translations

The main goal of this section is to show that safety of formulae is greatly affected by the properties of the underlying structure $\mathfrak{M}$. To state these results formally, we use the following concept.

Definition 1.56. *We say that there is a* safe translation *of* $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ *formulae, if there is a function* $\varphi \to \varphi_{\mathrm{safe}}$ *on* $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ *formulae such that for every* φ*,*

1. φ_{safe} *is safe, and*
2. *if* φ *is safe for* D*, then* $\varphi(D) = \varphi_{\mathrm{safe}}(D)$*.*

A translation is canonical if $\varphi_{\mathrm{safe}}(D) = \emptyset$ *whenever* φ *is not safe on* D*. A translation is recursive if the function* $\varphi \to \varphi_{\mathrm{safe}}$ *is recursive.* □

It turns out that recursive safe translations need not exist even for structures with decidable theories.

Proposition 1.57. *There exists a structure* $\mathfrak{M}$ *that is recursive, has a decidable first-order theory, and for which there is no recursive safe translation of* $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ *formulae.*

Proof sketch. Consider the structure $\mathfrak{M}$ whose domain U is the disjoint union of

- the set of Turing machines, appropriately coded as strings;
- the set of input strings to a Turing machine;
- the set of *traces*, i.e., full descriptions of a partial run of a Turing machine on an input word.

The signature of $\mathfrak{M}$ consists of one ternary relation P, which holds of a triple (M, w, t) iff t is a trace for Turing machine M on input word w. The key point is that there is no structure or ordering on the traces themselves: hence one cannot determine in first-order logic whether or not a trace is maximal. In fact, using a quantifier-elimination argument, one can show that the first-order theory of $\mathfrak{M}$ is decidable.

Let SC contain a single unary relation S. For any Turing machine M, let $\varphi^M(t)$ be the query $\exists! w \in \mathit{adom}(w = w) \wedge \exists w \in \mathit{adom} P(M, w, t)$. That is, if $S = \{w\}$, φ^M checks if t is a trace of M on w.

Assume we have a recursive safe translation, and consider $\varphi^M_{\text{safe}}(t)$. If we could check the equivalence of φ^M_{safe} and φ^M, we would be able to enumerate all machines that halt on every input, which is clearly impossible.

Next, to verify the equivalence of φ^M_{safe} and φ^M, we simply turn them into $\text{FO}(\mathfrak{M})$ formulae $\psi^M(w,t)$ and $\psi^M_{\text{safe}}(w,t)$ by replacing each subformula of the form $S(z)$ by $w = z$. The resulting $\text{FO}(\mathfrak{M})$ formulae then are true for (w,t) iff t is in the output of the corresponding query on input $\{w\}$. Thus, $\forall w \forall t\ (\psi^M(w,t) \leftrightarrow \psi^M_{\text{safe}}(w,t))$ holds iff φ^M_{safe} and φ^M are equivalent. The result now follows from the decidability of the theory of $\mathfrak{M}$. □

If one drops the condition that the theory of $\mathfrak{M}$ be decidable, but insists on computable functions and predicates in Ω, the situation is even worse: there need not be any safe translations at all (recursive or not).

Proposition 1.58. *There is a structure $\mathfrak{M} = \langle \mathbb{N}, P\rangle$, where P is a computable predicate, such that there is no safe translation of $\text{FO}_{\text{act}}(SC, \mathfrak{M})$ formulae.*

Proof. Let P be a ternary predicate defined as: $P(i,j,k)$ iff the ith Turing machine on the input j makes at least k moves (assuming some standard encoding of machines and inputs). Consider the schema that consists of a single binary relation S. Assume to the contrary that there is a safe translation over $\mathfrak{M}$. Let $\varphi(k) \equiv \exists i, j \in adom\ S(i,j) \wedge P(i,j,k)$, and let $\psi(k)$ be φ_{safe}. Note that ψ is an active-domain formula in the language of S and P. We now show how to use ψ to decide the halting problem.

Suppose we are given the ith machine M_i and the input j. We assume without loss of generality that M_i makes at least one move on j. Define a database D in which S consists of a single tuple (i,j). Since we know that ψ is safe, we then compute the minimum number l such that $D \not\models \psi(l)$. It is computable since a) it exists, and b) for each k, it is decidable whether $D \models \psi(k)$.

Assume that $D \models \varphi(l)$. Then M_i does not halt on j. Indeed, if M_i halts, then $\varphi(D)$ is finite, and hence $\varphi(D) = \psi(D)$, but we have $l \in \varphi(D) - \psi(D)$. Assume that $D \not\models \varphi(l)$. Then M_i makes $k < l$ moves on j, and thus halts. Hence, $D \models \varphi(l)$ iff M_i halts on j. Since it is decidable whether $D \models \varphi(l)$, we get a contradiction. □

On the other hand, for some structures $\mathfrak{M}$, recursive safe translations can be obtained.

Proposition 1.59. *Let $\mathfrak{M}$ be o-minimal, based on a dense order, admit effective quantifier-elimination, and have a decidable theory (for example, $\mathfrak{M}$ can be $\mathbf{R}_{\text{lin}}$ or $\mathbf{R}$). Then there exists a recursive canonical safe translation of $\text{FO}_{\text{act}}(SC, \mathfrak{M})$ formulae.*

Proof sketch. Given an $\text{FO}_{\text{act}}(SC, \mathfrak{M})$ formula φ, let $\alpha(x)$ be a formula defining the active domain of the output of φ. Let Ψ be an $\text{FO}_{\text{act}}(SC, \mathfrak{M})$ sentence equivalent to

$$\neg\exists x_1, x_2\ ((x_1 < x_2) \wedge (\forall x\ x_1 < x < x_2 \rightarrow \alpha(x)))$$

(it exists by the natural-active collapse). Define φ_{safe} as $\varphi \wedge \Psi$. The proposition then follows from the following easy claim: $D \models \Psi$ iff $\varphi(D)$ is finite. □

Corollary 1.60. *Let $\mathfrak{M}$ be as in Proposition 1.59. Then the state-safety problem over $\mathfrak{M}$ is decidable.* □

Thus, to obtain nice syntactic characterization of safe queries, we must deal with structures having good properties (just as in the case of collapse results).

1.10.3 Finite query safety: characterization

To give an idea of the characterization of safety we are about to provide, let us modify slightly an example we used in Section 1.2: $\varphi(x) \equiv (x > 1) \wedge \exists y\ S(y) \wedge (x \cdot x = y)$. Assuming that the underlying structure is the real field $\mathbf{R}$, the output of this formula is contained in the output of $\exists y\ S(y) \wedge (x \cdot x = y)$, which is $\{-\sqrt{a}, \sqrt{a} \mid a \in S\}$. Thus, there is an upper bound on the output of φ, which is given by applying certain functions to the active domain. This is the central idea of the *range-restriction* we are about to define. But first we introduce the notion of effective syntax for safe queries.

Definition 1.61. *We say that a class $\mathcal{Q}$ of queries* captures *the class of safe queries in* $\mathrm{FO}(SC, \mathfrak{M})$ *if every query in $\mathcal{Q}$ is safe and definable in* $\mathrm{FO}(SC, \mathfrak{M})$*, and every safe* $\mathrm{FO}(SC, \mathfrak{M})$ *query is equivalent to a query in $\mathcal{Q}$.*

If there exists a recursively enumerable class $\mathcal{Q}$ of queries that captures safe queries in $\mathrm{FO}(SC, \mathfrak{M})$*, we say that the class of safe* $\mathrm{FO}(SC, \mathfrak{M})$ *queries has* effective syntax. □

Proposition 1.57 (more precisely, the construction presented in the proof of Proposition 1.57) implies that there are structures $\mathfrak{M}$ with decidable first-order theory but without effective syntax for safe $\mathrm{FO}(SC, \mathfrak{M})$. Proposition 1.59, on the other hand, shows that there is an effective syntax for FO + Lin and FO + Poly queries, as one can express, in the language, if the output of a query is finite. This way of guaranteeing effective syntax is quite inelegant, and tells us nothing about the structure of safe queries. Below we present a much better description, based on the notion of definable functions.

Definition 1.62. *Given $\mathfrak{M} = \langle U, \Omega \rangle$, a function $f : U^k \rightarrow U$ is* $\mathfrak{M}$-definable *(or just definable if $\mathfrak{M}$ is understood) if its graph $\{(a_1, \ldots, a_k, a) \in U^{k+1} \mid a = f(a_1, \ldots, a_k)\}$ is an $\mathfrak{M}$-definable set.* □

From now on, we assume $adom(D) \neq \emptyset$. The case of empty SC-structures can be dealt with easily, as in this case an $\mathrm{FO}(SC, \mathfrak{M})$ formula reduces to a fixed $\mathrm{FO}(\mathfrak{M})$ formula, whose finiteness can be tested in the o-minimal case.

Definition 1.63. *Given* $\mathfrak{M} = \langle U, \Omega \rangle$, *a* query in range-restricted form *is a pair* $Q = (F, \varphi(x_1, \ldots, x_n))$, *where* $\varphi(\vec{x})$ *is an* $\mathrm{FO}(SC, \mathfrak{M})$ *formula, and* F *is a finite collection of definable functions.*

The semantics is defined as follows. First, for a set X, *let*

$$F(X) \quad = \quad \{f(\vec{a}) \mid f \in F, \quad \vec{a} \in X^{arity(f)} \}.$$

Then, for any finite SC*-structure* D, *define*

$$Q(D) \quad = \quad \varphi(D) \cap (F(adom(D)))^n.$$

That is, the finite set $F(adom(D))$ provides an upper bound on the output of Q (every constant in $Q(D)$ must be contained in $F(adom(D))$) and then φ is evaluated within this set. Since F is finite, and every function in F is definable, we obtain the following.

Lemma 1.64. *Every query in range-restricted form over* $\mathfrak{M}$ *is safe and definable in* $\mathrm{FO}(SC, \mathfrak{M})$. □

We now can state the main result of the section.

Theorem 1.65. *Let* $\mathfrak{M}$ *be any o-minimal structure based on a dense linear order. Assume that there is at least one definable constant in* $\mathfrak{M}$. *Then there is a function* Make_Safe *that takes as input an* $\mathrm{FO}(SC, \mathfrak{M})$ *formula* $\varphi(\vec{x})$, *and outputs a finite set* F *of definable functions such that the query* $Q = (F, \varphi)$ *is equivalent to* φ *on any finite* SC*-structure* D *on which* φ *is safe. Furthermore, if* $\mathfrak{M}$ *is decidable and has effective quantifier-elimination, then* Make_Safe *is recursive.* □

The proof of this theorem will be given in the rest of this section. But first we state some corollaries.

Corollary 1.66 (Range-restricted = Safe). *Let* $\mathfrak{M}$ *be as in Theorem 1.65. Then the class of range-restricted queries captures the class of safe* $\mathrm{FO}(\mathfrak{M}, SC)$ *queries.* □

We now consider specifically the cases of polynomial and linear constraints.

Definition 1.67. *a) A query in the* linear range-restricted form *is a pair* $Q = (F, \varphi)$ *where* φ *is a* $\mathrm{FO} + \mathrm{LIN}$ *formula, and* F *is a finite collection of linear functions (that is, functions of the form* $\langle \vec{a}, \vec{x} \rangle + b$*). The semantics is defined in the same way as for range-restricted queries above.*

b) A query in the polynomial range-restricted form *is a pair* $Q = (P, \varphi(x_1, \ldots, x_n))$ *where* φ *is a* $\mathrm{FO} + \mathrm{POLY}$ *formula, and* P *is a finite collection of multivariate polynomials with a distinguished variable* z. *The semantics is defined as follows. For a set* X, *and* $p(z, \vec{y})$, *let* $p(X)$ *be the set of all roots of polynomials of the form* $p(z, \vec{a})$, *where* $\vec{a}$ *is a tuple over* X, *provided such a univariate polynomial is not identically zero. Let* $P(X) = \bigcup_{p \in P} p(X)$. *Then* $Q(D)$ *is defined as* $\varphi(D) \cap (P(adom(D)))^n$. □

Corollary 1.68. *a) The class of queries in the linear range-restricted form captures the class of safe* FO + LIN *queries.*

b) The class of queries in the polynomial range-restricted form captures the class of safe FO + POLY *queries.*

Proof. a) A function definable over $\mathbf{R}_{\text{lin}}$ is piece-wise linear. Thus it suffices to apply Theorem 1.65, and take all the linear functions of which functions in F are composed.

b) Similarly, we apply Theorem 1.65 and obtain a set F of semi-algebraic functions. Each semi-algebraic function $f(\vec{y})$ is known to be algebraic. That is, there exists a polynomial $p(z, \vec{y})$ such that $p(z, \vec{y}) = 0$ iff $z = f(\vec{y})$. The result follows from this. □

Algebraic formulae and the proof of Theorem 1.65

We first give an analog of range-restriction using certain FO($\mathfrak{M}$) formulae, and then show how to derive a set F of definable functions from such a characterization. The FO($\mathfrak{M}$) formulae we shall use are *algebraic* formulae. They have distinguished parameters, which we shall always denote by $\vec{y}$ and separate from the single other variable by a semicolon. Assume that $\vec{y}$ is of length m. An FO($\mathfrak{M}$) formula $\gamma(x; \vec{y})$ is called algebraic if for each $\vec{b}$ in U^m there are only finitely many $a \in U$ that satisfy $\gamma(a, \vec{b})$. For example, the formula $\gamma(x; y) \equiv (x^2 = y)$ is algebraic over $\mathbf{R}$.

From the Uniform Bounds theorem (Theorem 1.29), we obtain the following useful fact about algebraic formulae.

Lemma 1.69. *Let $\mathfrak{M}$ be o-minimal, and $\gamma(x; \vec{y})$ algebraic. Then there exists a number K such that for any $\vec{b} \in U^m$, the set $\{a \in U \mid \mathfrak{M} \models \gamma(\vec{a}; \vec{b})\}$ has fewer than K elements.* □

We now need a syntactic characterization of algebraic formulae over o-minimal structures. Let $\Xi = \{\xi_1(x; \vec{y}), \ldots, \xi_k(x; \vec{y})\}$ be a collection of formulae. Let

$$\text{same}_\Xi(x, x'; \vec{y}) \quad \equiv \quad \bigwedge_{i=1}^{k} (\xi_i(x; \vec{y}) \leftrightarrow \xi_i(x'; \vec{y})).$$

Now define

$$\beta_\Xi(x; \vec{y}) \;\equiv\; \forall x', x'' \; \big(x' < x < x'' \rightarrow (\exists z \; x' \leq z \leq x'' \wedge \neg\text{same}_\Xi(x, z; \vec{y}))\big).$$

Proposition 1.70. *Let $\mathfrak{M}$ be an o-minimal structure based on a dense order. Then a formula $\gamma(x; \vec{y})$ is algebraic iff there exists a collection of* FO($\mathfrak{M}$) *formulae Ξ such that γ is equivalent to β_Ξ.*

Proof. Let Ξ be a collection of formulae, and assume that β_Ξ is not algebraic. That is, for some $\vec{b}$ over U, $\beta_\Xi(\mathfrak{M}; \vec{b}) = \{a \mid \mathfrak{M} \models \beta_\Xi(a; \vec{b})\}$ is infinite. Since $\mathfrak{M}$ is o-minimal, $\beta_\Xi(\mathfrak{M}; \vec{b})$ is a finite union of points and intervals. Since $<$ is dense, this means that there exist $a_0 < b_0 \in U$ such that $[a_0, b_0] \subseteq \beta_\Xi(\mathfrak{M}; \vec{b})$. We now consider the formulae $\xi'_i(x) = \xi_i(x; \vec{b})$ for all $\xi_i \in \Xi$. Since both $\xi'_i(\mathfrak{M}) = \xi_i(\mathfrak{M}; \vec{b})$ and $\neg\xi'_i(\mathfrak{M}) = \neg\xi_i(\mathfrak{M}; \vec{b})$ are finite unions of intervals and $<$ is dense, for every non-degenerate interval J, it is the case that either $J \cap \xi'_i(\mathfrak{M})$ or $J \cap \neg\xi'_i(\mathfrak{M})$ contains an infinite (closed) interval. Using this, we construct a sequence of intervals as follows: $I_0 = [a_0, b_0]$, $I_1 \subseteq I_0$ is an interval that is contained either in $I_0 \cap \xi'_1(\mathfrak{M})$ or in $I_0 \cap \neg\xi'_1(\mathfrak{M})$. At the jth step, $I_j \subseteq I_{j-1}$ is an interval that is contained either in $I_{j-1} \cap \xi'_j(\mathfrak{M})$ or in $I_{j-1} \cap \neg\xi'_j(\mathfrak{M})$. Let $I = I_k$. Then, for any $c, d \in I$, $\mathfrak{M} \models \xi_i(c, \vec{b}) \leftrightarrow \xi_i(d; \vec{b})$.

Since $I = [a', b'] \subseteq [a_0, b_0]$ and $\mathfrak{M} \models \beta_\Xi(c; \vec{b})$ for all $c \in I$, we obtain that, for every $c \in (a', b')$, there exists $d \in [a', b']$ such that $\mathfrak{M} \models \neg\mathrm{same}_\Xi(c, d; \vec{b})$. That is, for some $\xi_i \in \Xi$, $\mathfrak{M} \models \neg(\xi_i(c; \vec{b}) \leftrightarrow \xi_i(d; \vec{b}))$, which is impossible by construction of I. This proves that β_Ξ is algebraic.

For the converse, we let, for any $\gamma(x; \vec{y})$, Ξ consist of just γ. That is, $\beta_\Xi(x; \vec{y})$ is

$$\forall x', x'' \ \big(x' < x < x'' \to (\exists z \ x' \leq z \leq x'' \wedge \neg(\gamma(x; \vec{y}) \leftrightarrow \gamma(z; \vec{y})))\big).$$

We claim that γ and β_Ξ are equivalent, if γ is algebraic. Fix any $\vec{b}$ of the same length as $\vec{y}$, and assume that $\gamma(a; \vec{b})$ holds. If $\beta_\Xi(a; \vec{b})$ does not hold, then there exist $a' < a < a''$ such that for every $c \in [a', a'']$, $\gamma(c; \vec{b}) \leftrightarrow \gamma(a; \vec{b})$ holds; thus, $\gamma(c; \vec{b})$ holds for infinitely many c, contradicting algebraicity of γ. Hence, $\beta_\Xi(a; \vec{b})$ holds. Conversely, assume that $\beta_\Xi(a; \vec{b})$ holds. If $\gamma(a; \vec{b})$ does not hold, then there is an interval containing a on which $\gamma(\cdot; \vec{b})$ does not hold. Indeed, $\neg\gamma(\mathfrak{M}; \vec{b})$ is a finite union of intervals, whose complement is a finite set of points, so the above observation follows from the density of the ordering. We now pick $a' < a''$ such that $\gamma(\cdot; \vec{b})$ does not hold on $[a', a'']$. Since $\beta_\Xi(a; \vec{b})$ holds, we find $c \in [a', a'']$ such that $\neg(\gamma(a; \vec{b}) \leftrightarrow \gamma(c; \vec{b}))$ holds; that is, $\gamma(c; \vec{b})$ holds for $c \in [a', a'']$, which is impossible. Thus, we conclude that $\gamma(a; \vec{b})$ holds, proving that for any $\vec{b}$, $\forall x \ (\gamma(x; \vec{b}) \leftrightarrow \beta_\Xi(x; \vec{b}))$. This finishes the proof. $\square$

Given an algebraic formula $\gamma(x; \vec{y})$ and a set $X \subseteq U$, let $\gamma(X)$ be the set of all a that make true $\gamma(a; \vec{b})$, as $\vec{b}$ ranges over tuples of elements of X. Note that if X is finite, then so is $\gamma(X)$.

We now define a query in the *algebraic range-restricted form* as a pair $Q = (\gamma(x; \vec{y}), \varphi(x_1, \ldots, x_n))$, where φ is an $\mathrm{FO}(SC, \mathfrak{M})$ formula, and γ is an algebraic $\mathrm{FO}(\mathfrak{M})$ formula. The semantics is defined as $Q(D) = \varphi(D) \cap (\gamma(adom(D)))^n$. Clearly, Q is safe.

Proposition 1.71. *Let $\mathfrak{M}$ be any o-minimal structure based on a dense linear order. Then there is a function* Make_Safe$'$ *that takes as input an* $\mathrm{FO}(SC, \mathfrak{M})$ *formula $\varphi(x_1, \ldots, x_n)$, and outputs an algebraic formula $\gamma(x; \vec{y})$ such that the query $Q = (\gamma, \varphi)$ is equivalent to φ on all structures D for which φ is safe.*

Furthermore, if $\mathfrak{M}$ *has effective quantifier-elimination, then* Make_Safe$'$ *is recursive.*

Proof. Let $\psi(z)$ be a one-variable $\mathrm{FO}(SC, \mathfrak{M})$ formula that defines the active domain of the output of φ. That is, it is the disjunction of all formulae $\exists \vec{x}^{(i)} \varphi(z, \vec{x}^{(i)})$ where $\vec{x}^{(i)}$ is $\vec{x}$ except the ith component, and $(z, \vec{x}^{(i)})$ is the tuple in which z is inserted in the ith position. Note that φ is safe on D iff ψ is.

Let $\mathfrak{M}'$ be a definable expansion of $\mathfrak{M}$ that has quantifier-elimination, and hence admits the natural active-collapse. We can thus assume that ψ is an $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M}')$ formula. Let

$$\psi(z) \;\equiv\; \mathbf{Q} w_1 \in adom \ldots \mathbf{Q} w_l \in adom \;\; \alpha(z, \vec{w})$$

where $\alpha(z, \vec{w})$ is quantifier-free, and all atomic subformulae $R(\cdots)$ contain only variables, excluding z. Let $\Xi = \{\xi_i(z, \vec{w}) \mid i = 1, \ldots, k\}$ be the collection of all $\mathrm{FO}(\mathfrak{M}')$-atomic subformulae of α. We may assume without loss of generality that the length of $\vec{w}$ is nonzero, and that Ξ is nonempty (just as we did in the proof of the natural-active collapse).

Define $\mathrm{same}_{\Xi}(a, b, \vec{w})$, as before, to be $\bigwedge_{i=1}^{k} (\xi_i(a, \vec{w}) \leftrightarrow \xi_i(b, \vec{w}))$, and define $\gamma(x; \vec{w})$ to be $\beta_{\Xi}(x; \vec{w})$. We let Make_Safe(ψ) output γ. Note that γ is actually an $\mathrm{FO}(\mathfrak{M})$ formula, since $\mathfrak{M}'$ is a definable expansion.

Since γ is algebraic by Proposition 1.70, we must show that $\{a \mid D \models \psi(a)\} = \{a \in \gamma(D) \mid D \models \psi(a)\}$ for every nonempty database for which ψ is safe.

Assume otherwise; that is, for some nonempty D for which ψ is safe, we have $D \models \psi(a)$ but $a \notin \gamma(D)$. Let $\vec{c}_1, \ldots, \vec{c}_M$ be an enumeration of all tuples of the length of $\vec{w}$ of elements of $adom(D)$. Note that $M > 0$. Since $a \notin \gamma(D)$, we have that for each $i = 1, \ldots, M$, there exist a'_i, a''_i such that $a'_i < a < a''_i$ and $\mathfrak{M} \models \mathrm{same}_{\Xi}(a, c, \vec{c}_i)$ for all $c \in [a'_i, a''_i]$.

Let $b' = \max\{a'_i\}, b'' = \min\{a''_i\}$. We have $b' < a < b''$, and for each $\vec{c}$ (of length of $\vec{w}$) over the active domain, we have $\xi_i(a; \vec{c}) \leftrightarrow \xi_i(c, \vec{c})$ for every $c \in [b', b'']$. From this, by a simple induction on the structure of the formula (using the fact that z does not appear in any atomic formula $R(\cdots)$), we obtain that $D \models \alpha(a, \vec{c}) \leftrightarrow \alpha(c, \vec{c})$ for every $\vec{c}$ over $adom(D)$ and every $c \in [b', b'']$, and thus $D \models \psi(a) \leftrightarrow \psi(c)$, which implies that ψ is not safe for D. This contradiction proves correctness of Make_Safe$'$, and the proposition. □

To conclude the proof of Theorem 1.65, we have to show how to obtain definable functions from algebraic formulae.

Proposition 1.72. *Let* $\mathfrak{M}$ *be o-minimal, such that there is at least one definable constant. Let* $\gamma(x; \vec{y})$ *be algebraic. Then there is a finite collection* F *of definable functions* $f(\vec{y})$ *such that* $\gamma(X) \subseteq F(X)$ *for any set* $X \subseteq U$*. Moreover, if* $\mathfrak{M}$ *is decidable, then the set* F *can be found effectively.*

Proof. Let c be a definable constant over $\mathfrak{M}$. Given γ, let K be an integer such that the set $\{a \in U \mid \mathfrak{M} \models \gamma(a, \vec{b})\}$ has fewer than K element for every $\vec{b}$ (see Lemma 1.69). For each $i < K$, define $f_i(\vec{y})$ to be the ith element (in the order $<$) that makes true $\gamma(\cdot, \vec{y})$, if it exists, and c, if there is no such element. Let $F = \{f_i \mid i < K\}$. Clearly, each f_i is a definable function and $\gamma(X) \subseteq F(X)$. If $\mathfrak{M}$ is decidable, then K can be found, and thus F can be constructed effectively. □

We finally complete the proof of Theorem 1.65. Given an $\mathrm{FO}(SC, \mathfrak{M})$ formula $\varphi(\vec{x})$, we first apply Proposition 1.71 to get an algebraic formula γ giving a bound on the output (if it is finite), and then apply Proposition 1.72 to get a set functions F that puts a bound on the output of φ. If $\mathfrak{M}$ is decidable and quantifier-elimination is effective, then γ can be effectively found (as the natural-active collapse is effective), and there is an algorithm for constructing F from γ. □

1.10.4 Infinite query safety: reduction

The question of query safety over constraint database reduces to preserving certain *geometric properties* of regions in $\mathbb{R}^k$. If $\mathfrak{M} = \langle U, \Omega\rangle$ is an infinite structure, let $\mathsf{DS}(\mathfrak{M})$ be the class of definable sets over $\mathfrak{M}$, that is, $\mathsf{DS}(\mathfrak{M}) = \bigcup_{n<\omega} \mathsf{DS}_n(\mathfrak{M})$ and $\mathsf{DS}_n(\mathfrak{M})$ is the collection of definable subsets of U^n. We use SAlg_n for semi-algebraic sets in $\mathbb{R}^n$.

Let SC consist of an m-ary relation symbol S, and let $\psi(x_1, \ldots, x_n)$ be an $\mathrm{FO}(SC, \mathfrak{M})$ formula. It defines a map from $\mathsf{DS}_m(\mathfrak{M})$ to $\mathsf{DS}_n(\mathfrak{M})$ as follows: for any $X \in \mathsf{DS}_m(\mathfrak{M})$, $\psi(X) = \{\vec{y} \mid (\mathfrak{M}, X) \models \psi(\vec{y})\}$.

Let now $\mathcal{C}$ be a class of objects in $\mathsf{DS}(\mathfrak{M})$. We say that an $\mathrm{FO}(SC, \mathfrak{M})$ formula ψ *preserves* $\mathcal{C}$ if for any $X \in \mathcal{C}$, $\psi(X) \in \mathcal{C}$. The safety question for constraint databases is the following. Is there effective syntax for the class of $\mathcal{C}$-preserving queries?

We now show how this problem can be reduced to finite query safety for embedded finite models.

Definition 1.73. *The class $\mathcal{C}$ has a* canonical representation *in* $\mathsf{DS}(\mathfrak{M})$ *if there is a recursive injective function $g : \mathbb{N} \to \mathbb{N}$ with computable inverse, and for each n, two functions $code_n : 2^{U^n} \to 2^{U^m}$ and $decode_n : 2^{U^m} \to 2^{U^n}$, where $m = g(n)$, such that:*

1. *$decode_n \circ code_n(x) = x$ if $x \in \mathsf{DS}_n(\mathfrak{M})$;*
2. *$\mid code_n(x) \mid < \omega$ if $x \in \mathcal{C}$; $decode_n(x) \in \mathcal{C}$ if x is finite;*
3. *$code_n$ is $\mathrm{FO}(\mathfrak{M})$-definable on $\mathsf{DS}_n(\mathfrak{M})$;*
4. *$decode_n$ is $\mathrm{FO}(\mathfrak{M})$-definable on finite sets.*

Intuitively, the canonical representation is a finite representation of $\mathcal{C}$ within $\mathsf{DS}(\mathfrak{M})$ that can be defined in first-order logic over $\mathfrak{M}$. For example, one approach to obtaining a canonical representation of convex polytopes would be to compute their vertices. This suffices to reconstruct the polytope, and the vertices can be defined by a first-order formula.

Similarly to the finite case, we say that there is effective syntax for $\mathcal{C}$-preserving $\mathrm{FO}(SC, \mathfrak{M})$ formulae if there exists a recursively enumerable set of $\mathcal{C}$-preserving $\mathrm{FO}(SC, \mathfrak{M})$ formulae such that every $\mathcal{C}$-preserving $\mathrm{FO}(SC, \mathfrak{M})$ formula is equivalent to a formula in this set.

Theorem 1.74. *Let $\mathfrak{M} = \langle U, \Omega \rangle$ be o-minimal, based on a dense order, decidable, and have effective quantifier-elimination. Suppose $\mathcal{C}$ is a class that has a canonical representation in* $\mathsf{DS}(\mathfrak{M})$*. Then there is effective syntax for $\mathcal{C}$-preserving* $\mathrm{FO}(SC, \mathfrak{M})$ *formulae.*

Proof. Consider an enumeration of all safe $\mathrm{FO}(SC, \mathfrak{M})$ queries $\langle \varphi_i \rangle$ on finite structures (from Proposition 1.71, we know that it exists). Let φ use an extra relation symbol of arity m, and assume that n is such that $g(n) = m$ (where g comes from the definition of canonical representations). Let φ_i have l parameters, and again let k be such that $g(k) = l$. If n and k are found for a given φ_i, we let ψ be:

$$decode_k \circ \varphi_i \circ code_n.$$

This produces the required enumeration. We have to check that every query of the form$decode_k \circ \varphi_i \circ code_n$ preserves $\mathcal{C}$, and for every $\mathcal{C}$ preserving ψ, we can get φ such that $decode \circ \varphi \circ code$ coincides with ψ. The first one is clear: if we have $X \in \mathcal{C}$, then $code_n(X)$ is finite, hence $\varphi_i(code_n(X))$ is finite too, and thus the output of $decode_k$ is in $\mathcal{C}$.

For the converse, suppose we have a $\mathcal{C}$-preserving query $\psi : \mathsf{DS}_n(\mathfrak{M}) \to \mathsf{DS}_k(\mathfrak{M})$. Define α as follows: $\alpha = code_k \circ \psi \circ decode_n$. That is, α is a query $\mathsf{DS}_m(\mathfrak{M}) \to \mathsf{DS}_l(\mathfrak{M})$. Given this, notice that

$$decode_k \circ \alpha \circ code_n = decode_k \circ code_k \circ \psi \circ decode_n \circ code_n = \psi$$

on $\mathsf{DS}_n(\mathfrak{M})$. Thus, it remains to show that α is safe. Let $X \subset U^m$ be finite. Then $decode_n(X) \in \mathcal{C}$, $decode_n(X) \subset U^n$, and $Y = \psi(decode_n(X)) \in \mathsf{DS}_k(\mathfrak{M})$ is in $\mathcal{C}$, too. Hence, $code_k(Y)$ is finite. □

We now give two applications for semi-algebraic sets and $\mathrm{FO} + \textsc{Poly}$. The first one gives an example of a geometric class for which coding is easy.

Proposition 1.75. *The class of convex polytopes has a canonical representation in* SAlg*. Consequently, the class of* $\mathrm{FO} + \textsc{Poly}$ *queries preserving the property of being a convex polytope has effective syntax.*

Proof. Given a convex polytope X in $\mathbb{R}^n$, its vertices can be found as $V(X) = \{\vec{x} \in \mathbb{R}^n \mid \vec{x} \in X, \vec{x} \notin \mathrm{conv}(X - \vec{x})\}$, where $\mathrm{conv}(\cdot)$ denotes the convex hull. Thus, $V(X)$ is definable in $\mathrm{FO} + \textsc{Poly}$. We now define $code_n$.

To simplify the notation, we let it produce a pair of n-ary relations, but it can be straightforwardly coded by one relation. If $X = \text{conv}(V(X))$, then $\mathit{code}_n(X) = (V(X), \emptyset)$; otherwise, $\mathit{code}_n(X) = (\mathbb{R}^n, X)$. The function $\mathit{decode}_n : 2^{\mathbb{R}^n} \times 2^{\mathbb{R}^n} \to 2^{\mathbb{R}^n}$ is defined as follows:

$$\mathit{decode}_n(Y, Z) = \begin{cases} \bigcup_{(\vec{y}_1,\ldots,\vec{y}_{n+1})\in Y} \quad \text{conv}(\{\vec{y}_1, \ldots, \vec{y}_{n+1}\}) & \text{if } Y \neq \mathbb{R}^n, \\ Z & \text{otherwise.} \end{cases}$$

Clearly, $\mathit{decode}_n \circ \mathit{code}_n$ is the identity function for any semi-algebraic set; these functions are also first-order definable. If X is a polytope, $V(X)$ is finite, and by Carathéodory's theorem each point of X is contained in the convex hull of at most $n+1$ vertices of X. Hence, $\mathit{card}(\mathit{code}_n(X)) \leq \mathit{card}(V(X))^{n+1}$. If (Y, Z) is finite, then $\mathit{decode}_n(Y)$ is $\text{conv}(Y)$, and thus a convex polytope. This proves the proposition. □

The second example deals with the case of $\mathcal{C}$ being a class of semi-linear sets. We now give two different approaches to showing the following.

Theorem 1.76. *There is an effective syntax for the class of* FO + Poly *queries preserving semi-linearity.* □

One approach to showing this is to prove that the class of semi-linear sets has a canonical representation in the class of semi-algebraic sets. This is true, although the coding scheme is quite complex and not very intuitive. Another way of showing this theorem is based on the proposition below.

Proposition 1.77. *For any $n > 0$, there is an* FO + Poly *sentence over SC containing one n-ary relation symbol, which tests if the input (which is a semi-algebraic set $S \subseteq \mathbb{R}^n$) is semi-linear.* □

Then effective syntax for FO + Poly queries preserving semi-linearity can be obtained simply by inserting tests for the input and output being semi-linear, and returning the empty set if semi-linearity is not preserved. However, the decision procedure is not much simpler than the canonical representation, and we are thus very far from a usable language for FO + Poly-definable queries preserving semi-linearity. But the very fact that such a language exists is an interesting and nontrivial property of FO + Poly.

1.10.5 Deciding safety

Safety of $\text{FO}_{\text{act}}(SC)$ formulae is already undecidable. However, there are some nice syntactic subclasses of $\text{FO}_{\text{act}}(SC)$ for which safety is guaranteed. We now consider one such subclass – *conjunctive queries*. The class of conjunctive queries is defined as a $\{\exists, \wedge\}$-fragment of $\text{FO}_{\text{act}}(SC)$, that is, as the set of formulae built from atomic formulae $S(\cdot)$, where $S \in SC$, using conjunction and existential quantification only. Outputs of such formulae cannot extend the active domain, and hence they are safe. We now consider a natural analog of conjunctive queries over embedded finite models. Although they are no longer guaranteed to produce output containing only elements of the active domain, safety remains decidable for underlying structures such as $\mathbf{R}_{\text{lin}}$ and $\mathbf{R}$.

A *conjunctive query (CQ)* is an $\text{FO}(SC, \mathfrak{M})$ formula of the form

$$\varphi(\vec{x}) \quad \equiv \quad \exists \vec{y} \in adom\ \alpha_1(\vec{x}, \vec{y}) \wedge \ldots \wedge \alpha_k(\vec{x}, \vec{y}) \wedge \gamma(\vec{x}, \vec{y}),$$

where $\alpha_1(\vec{x}, \vec{y}), \ldots, \alpha_k(\vec{x}, \vec{y}), k \geq 0$ are formulae of the form $S(\vec{u})$, $S \in SC$ and $\vec{u}$ a subtuple of $(\vec{x}, \vec{y})$, and γ is an $\text{FO}(\mathfrak{M})$ formula.

Theorem 1.78. *Let $\mathfrak{M}$ be o-minimal, based on a dense order, decidable, and admit effective quantifier-elimination. Then it is decidable if a given conjunctive query in* $\text{FO}(SC, \mathfrak{M})$ *is safe.*

Proof. Given two formulae $\varphi(\vec{x})$ and $\psi(\vec{x})$, by containment $\varphi \subseteq \psi$ we mean $\varphi(D) \subseteq \psi(D)$ for any finite D. From Proposition 1.71 we obtain that for any $\text{FO}(SC, \mathfrak{M})$ formula $\varphi(\vec{x})$, there exists an active-semantics CQ $\psi(\vec{x})$ such that φ is safe iff $\varphi \subseteq \psi$. The theorem now follows from the lemma below.

Lemma 1.79. *Let $\mathfrak{M}$ be as in Theorem 1.78. Then containment is decidable for conjunctive queries.*

Proof. Suppose we are given CQs $\varphi(\vec{x})$ and $\psi(\vec{x})$. We claim that one can effectively find a number k such that $\varphi \subseteq \psi$ iff for every D with at most k tuples, $\varphi(D) \subseteq \psi(D)$. This clearly implies the result, as the latter condition can be expressed as an $\text{FO}(\mathfrak{M})$ sentence.

To prove the claim, assume that $\varphi(\vec{x})$ is $\exists \vec{y} \in adom \bigwedge_{i=1}^{l} \alpha_i(\vec{u}_i) \wedge \gamma(\vec{x}, \vec{y})$. We claim that k can be taken to be l plus the length of $\vec{y}$. Indeed, assume there is $\vec{a} \in \varphi(D) - \psi(D)$. Let $\vec{b}$ witness $D \models \varphi(\vec{a})$; we then see that there is a structure D' that contains at most k tuples from D such that $D' \models \varphi(\vec{a})$ (it has to contain enough tuples to ensure that all elements of $\vec{b}$ are in $adom(D')$, and that $\bigwedge_{i=1}^{l} \alpha_i(\vec{u}_i)$ holds. But then $D' \models \neg\psi(\vec{a})$, for otherwise we would have $D \models \psi(\vec{a})$. Thus, any counterexample to containment is witnessed by a $\leq k$-element structure. This finishes the proof Lemma 1.79 and the Theorem. □

The proof can be extended to show a slightly more general result:

Corollary 1.80. *It is decidable whether any Boolean combination of* $\text{FO} + \text{Lin}$ *or* $\text{FO} + \text{Poly}$ *conjunctive queries is safe.* □

Note, however, that safety of conjunctive queries is not decidable over every structure.

Proposition 1.81. *Let $\mathfrak{N} = \langle \mathbb{N}, +, \cdot \rangle$. Then safety of conjunctive queries in* $\text{FO}(SC, \mathfrak{N})$ *is undecidable, for any SC.*

Proof. Define $\varphi(\vec{x})$ to be $p(\vec{x}) = 0$ for some Diophantine equation. This is a CQ in $\text{FO}(SC, \mathfrak{M})$, and it is safe iff $p(\vec{x}) = 0$ has finitely many solutions. However, this property of Diophantine equations is undecidable. □

Some decidability results can be shown for constraint databases as well. We give only one example here, for the case of queries preserving the property of being a convex polytope.

Lemma 1.82. *Let $\varphi(x_1, \ldots, x_n)$ be a union of* $\mathrm{FO} + \textsc{Poly}$ *conjunctive queries that mention one m-ary relational symbol S. Then one can effectively find two numbers k and l such that φ is preserving the property of being a convex polytope iff for every convex polytope $\mathbf{D}$ in $\mathbb{R}^m$ with at most k vertices, the output $\varphi(\mathbf{D})$ is a convex polytope with at most l vertices in $\mathbb{R}^n$.* □

With this, one can show:

Proposition 1.83. *It is decidable if a union of conjunctive* $\mathrm{FO} + \textsc{Poly}$ *queries preserves the property of being a convex polytope.*

Proof. Note that for each i, there is an $\mathrm{FO} + \textsc{Poly}$ query ψ_i for each i that tests if a set $\mathbf{D}$ is a convex polytope with at most i vertices: it checks that the set of vertices $V(\mathbf{D}) = \{x \in \mathbf{D} \mid x \notin \mathrm{conv}(\mathbf{D} - x)\}$ has at most i elements, and that $\mathbf{D} = \mathrm{conv}(V(\mathbf{D}))$. In order to check if φ in $\mathrm{FO} + \textsc{Poly}$ is preserving convex polytopes, one applies Lemma 1.82 to compute the numbers k and l, and then writes a sentence saying that for every set V in $\mathbb{R}^m$ with at most k elements, applying φ to $\mathrm{conv}(V)$ yields a polytope with at most l vertices. Since conv and ψ_l are definable, this property can be expressed as an $\mathrm{FO}(\mathbf{R})$ sentence. The proposition now follows from the decidability of the theory of $\mathbf{R}$. □

1.10.6 Dichotomy theorem for embedded finite models

We now show a simple but powerful combinatorial structure theorem, saying that over a well-behaved structure, outputs of safe queries cannot grow arbitrarily large in terms of the size of the input. We use the notation $\mathsf{size}(D)$ for the size of a finite structure, measured here as the total number of tuples. It can equivalently be measured as the cardinality of the active domain, or the number of tuples multiplied by their arity, and all the results will hold.

Theorem 1.84 (Dichotomy Theorem). *Let $\mathfrak{M}$ be o-minimal and based on a dense order. Let $\varphi(\vec{x})$ be an $\mathrm{FO}(SC, \mathfrak{M})$ formula. Then there exists a polynomial $p_\varphi : \mathbb{R} \to \mathbb{R}$ such that, for any finite SC-structure D, either $\varphi(D)$ is infinite, or $\mathsf{size}(\varphi(D)) \leq p_\varphi(\mathsf{size}(D))$.*

Proof. Expand Ω by one constant (this does not violate o-minimality) and apply Theorem 1.65. □

The dichotomy theorem can also be stated in terms of a function measuring the growth of the output size. Define $\mathit{growth}_\varphi : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ as

$$\mathit{growth}_\varphi(n) = \max\{\mathsf{size}(\varphi(D)) \mid \mathsf{size}(D) = n\}.$$

Corollary 1.85. *Let* $\varphi(\vec{x})$ *be an* $\mathrm{FO}(SC, \mathfrak{M})$ *formula for* $\mathfrak{M}$ *as in Theorem 1.84. Then there exists a polynomial* p_φ *such that, for every* $n \in \mathbb{N}$*, either* $growth_\varphi(n) = \infty$*, or* $growth_\varphi(n) \leq p_\varphi(n)$. □

As we have often seen in this chapter, the assumptions on the structure are extremely important. Below we show that the Dichotomy Theorem fails over some simple decidable structures on the natural numbers.

Proposition 1.86. *Let* $\mathfrak{M} = \langle \mathbb{N}, +, <, 1\rangle$*. Then there exists an* $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ *formula* $\varphi(x)$ *such that* $growth_\varphi(n) = 2^n$ *for every* $n > 0$.

Proof. Let SC consist of one unary relation S. We show that there exists an $\mathrm{FO}_{\mathrm{act}}(SC, \mathfrak{M})$ sentence Ψ such that $S \models \Psi$ iff S is of the form $S_n = \{2^i \mid 1 \leq i \leq n\}$. This is done by letting Ψ be

$$\begin{array}{l} (\exists x \in adom\ x = 1 + 1 \wedge S(x)) \\ \wedge\ (\forall x \in adom\ x = 1 + 1 \vee x > 1 + 1) \\ \wedge\ (\forall x \in adom\ x = 1 + 1 \vee \exists y \in adom\ y + y = x) \\ \wedge\ (\forall x \in adom\ (\forall y \in adom\ y < x \vee y = x) \vee (\exists y \in adom\ y = x + x)) \end{array}$$

Now define $\varphi(x)$ as $\Psi \wedge \neg(x < 1) \wedge (\exists y \in adom\ x < y \vee x = y)$. Then, for S not of the form S_n, we have $\varphi(S) = \emptyset$, and $\varphi(S_n) = \{1, 2, 3, \ldots, 2^n\}$. Since $card(S_n) = n$, this implies $growth_\varphi(n) = 2^n$ for $n > 0$. □

The dichotomy theorem gives easy expressivity bounds based on the growth of the output size. For example, even if we use exponentiation, we still cannot express any queries with superpolynomial growth, since $\langle \mathbb{R}, +, \cdot, e^x\rangle$ is o-minimal.

To give another application, consider the following problem: given a polyhedron P and $\epsilon > 0$, find a triangulation of P of mesh $< \epsilon$. That is, a triangulation such that the diameter of each simplex (triangle in dimension 2) is less than ϵ. Every polyhedron admits such a triangulation. The output of such a query can be structured in several ways, for example, by storing the information about the face structure of the triangulation. We only impose one requirement that the vertices of the triangulation be computable.

Proposition 1.87. *Let* $\mathfrak{M} = \langle \mathbb{R}, \Omega\rangle$ *be an o-minimal expansion of the real field* $\mathbf{R}$*. Then there is no* $\mathrm{FO}(SC, \mathfrak{M})$ *formula that finds a triangulation of a given polygon with a given mesh. This continues to hold if we restrict to convex polytopes on a plane.*

Proof. Suppose such a formula exists; now consider a new query that does the following. Its input is one binary relation containing a set X of points $\vec{x}_1, \ldots, \vec{x}_n$ on the real plane, and one unary relation containing a single real number $\epsilon > 0$. First, in FO + Poly, construct $\mathrm{conv}(X)$, and then find vertices of a triangulation with mesh $< \epsilon$. This is clearly a safe query, so by the Dichotomy Theorem, there exists a polynomial p such that the number of vertices of the triangulation is at most $m = p(n + 1)$ ($n + 1$ is the size of the input). Let d be the maximal distance between the points $\vec{x}_i, \vec{x}_j$ (and thus the diameter of conv(X)). Since the segment $[\vec{x}_i, \vec{x}_j]$ with $d(\vec{x}_i, \vec{x}_j) = d$ must be covered by the simplexes of the triangulation, it is possible to find a number ϵ such that it cannot be covered by fewer than $m+1$ triangles of diameter ϵ, and hence the number of points in the triangulation is $> m$. This contradiction proves the proposition. □

Analogs of the growth bounds result can be obtained in the constraint database setting as well; we give one example below.

Proposition 1.88. *Let $\varphi(\vec{x})$ be an* FO + POLY *formula that preserves the property of being a convex polytope. Then there exists a polynomial p_φ such that, whenever $\mathbf{D}$ is a convex polytope with n vertices, $\varphi(\mathbf{D})$ has at most $p_\varphi(n)$ vertices.* □

1.11 Database Considerations

In this section, we consider two aspects of embedded finite models and constraint databases motivated primarily by database considerations: adding aggregate functions, and higher-order features.

1.11.1 Aggregate operators

Aggregation operators like COUNT, SUM, and AVG form an indispensable part of database query languages for the relational data model. How can they be used in the settings of embedded finite models and constraint databases?

We shall now briefly consider two aggregate operators. The *average* operator, present in all commercial database systems, returns the average value of a column of a relation. The *volume* operator, used in geographical information system, returns the volume (or area) of a set. Here we investigate the possibility of incorporating these operators into languages like FO + POLY and FO + LIN.

Let $\varphi(\vec{x}, \vec{y})$ be a formula in FO$(SC, \mathfrak{M})$, with $\vec{x}$ and $\vec{y}$ being of length n and m, respectively. We define, for $\vec{a} \in U^n$, $\varphi(\vec{a}, D)$ to be $\{\vec{b} \in U^m \mid D \models \varphi(\vec{x}, \vec{y})\}$.

Let AVG(C) be the average value of a finite set $C \subset \mathbb{R}$; we let AVG$(C) = 0$ if C is empty or infinite. We say that the *average operator* AVG is definable over $\mathfrak{M}$ if for every vocabulary SC and every FO$(SC, \mathfrak{M})$ formula $\varphi(\vec{x}, y)$ there exists an FO$(SC, \mathfrak{M})$ formula $\psi(\vec{x}, z)$ such that for every SC-structure D, $D \models \psi(\vec{a}, c)$ iff $c = $ AVG$(\varphi(\vec{a}, D))$.

An easy application of collapse results shows:

Proposition 1.89. *Let $\mathfrak{M} = \langle \mathbb{R}, \Omega \rangle$ be o-minimal, and such that the expansion $\mathfrak{M} = \langle \mathbb{R}, \Omega, +, \cdot \rangle$ is o-minimal as well (for example, $\mathbf{R}, \mathbf{R}_{\text{lin}}$). Then the average operator* AVG *is not definable over $\mathfrak{M}$.* □

We leave this as an exercise, but we shall soon prove a more general result. Since AVG is not definable, one may consider several ways to overcome this. One possibility is to *approximate* it, rather than define it precisely. What could such an approximation be? Clearly, we cannot hope to define an ϵ-interval around the value of $\text{AVG}(\varphi(\vec{a}, D))$, as then that value would be definable as the center of the interval. Instead, we settle for a bit less: we want to produce a formula defining a nonempty set that lies in that ϵ-interval.

We say that the average operator AVG^ϵ, $\epsilon > 0$, is definable over $\mathfrak{M}$ if for every vocabulary SC and every $\text{FO}(SC, \mathfrak{M})$ formula $\varphi(\vec{x}, y)$ there exists an $\text{FO}(SC, \mathfrak{M})$ formula $\psi(\vec{x}, z)$ such that for every SC-structure D, and every $\vec{a}$, the following two conditions hold:

1. $D \models \exists z\ \psi(\vec{a}, z)$ (that is, $\psi(\vec{a}, D) \neq \emptyset$); and
2. if $D \models \psi(\vec{a}, c)$, then $|c - \text{AVG}(\varphi(\vec{a}, D))| < \epsilon$.

We say that the average operator AVG_I^ϵ, $\epsilon > 0$, is definable over $\mathfrak{M}$ if the above is true whenever $\varphi(\vec{a}, D) \subseteq I = [0, 1]$.

We now show the inexpressibility result for these queries. Recall that all previous inexpressibility results (with the exception of the result on topological queries) were proved by reductions to generic queries. Here we cannot easily find such reductions, as approximating queries are extremely nongeneric: they do not say anything about the behavior on the ϵ-interval, other than that some point of the interval satisfies the formula. The proof below shows a way to circumvent the problem of "extremely nongeneric" queries.

Theorem 1.90. *Let $\mathfrak{M} = \langle \mathbb{R}, \Omega \rangle$ be o-minimal, and such that the expansion $\mathfrak{M} = \langle \mathbb{R}, \Omega, +, \cdot \rangle$ is o-minimal as well. Then the average operators AVG^ϵ (for any $\epsilon > 0$) and AVG_I^ϵ (for $0 < \epsilon < 1/2$) are not definable over $\mathfrak{M}$.*

Proof. Let SC consist of two unary relations, U_1 and U_2. Let $c_1, c_2 > 1$ be two real numbers. We say that Φ is a (c_1, c_2)-*separating sentence* if for any finite instance D of SC, it is the case that $card(U_1) > c_1 \cdot card(U_2)$ implies $D \models \Phi$ and $card(U_2) > c_2 \cdot card(U_1)$ implies $D \models \neg\Phi$. Note that this definition says nothing about the case when $\frac{1}{c_2} \cdot card(U_2) \leq card(U_1) \leq c_1 \cdot card(U_2)$, and thus direct application of bounds on expressiveness of generic queries is impossible.

Lemma 1.91. *Let $\mathfrak{M}$ be as in the theorem, $c_1, c_2 > 1$, and SC as above. Then no (c_1, c_2)-separating sentence is definable in $\text{FO}(SC, \mathfrak{M})$.*

Proof of the lemma. Assume that there is a (c_1, c_2)-separating sentence Φ. From the natural-active collapse, we conclude that there is an $\text{FO}_{\text{act}}(SC, \mathfrak{M}')$ (c_1, c_2)-separating sentence Φ' for some definable expansion $\mathfrak{M}'$ of $\mathfrak{M}$ that has quantifier-elimination. From the Ramsey property of active-semantics formulae (Proposition 1.15) we obtain that there is an infinite set $Y \subseteq U$ and an $\text{FO}_{\text{act}}(SC, <)$-sentence Ψ such that for every SC-structure D with

$adom(D) \subset Y$, we have: $D \models \Phi'$ iff $D \models \Psi$. Thus, it remains to show that $\mathrm{FO}_{\mathrm{act}}(SC,<)$ cannot express a (c_1,c_2)-separating sentence Ψ, on instances over an infinite set.

Assume it can; and let q be the quantifier rank of Ψ. We now consider two instances over Y. In both instances D_1 and D_2 all elements of U_1 precede U_2 in the linear order $<$. In D_1, $card(U_1) = \lceil c_1(2^q+1)\rceil$ and $card(U_2) = 2^q+1$; in D_2, $card(U_1) = 2^q+1$ and $card(U_2) = \lceil c_2(2^q+1)\rceil$. Since Ψ is a (c_1,c_2)-separating sentence, we must have $D_1 \models \Psi$ and $D_2 \models \neg\Psi$. It is then easy to obtain a contradiction by showing that $D_1 \models \Psi$ iff $D_2 \models \Psi$. This is done by proving that the duplicator can win in a q-round Ehrenfeucht-Fraïssé game on D_1 and D_2. This follows from the fact that for every $n, m > 2^q$, the duplicator can win a q-round game on two ordered sets of cardinalities n and m. Thus, for D_1 and D_2, the duplicator picks a separate strategy for U_1 and U_2, and whenever the spoiler plays in U_1, the duplicator forgets about the moves in U_2 and responds in U_1 using the strategy for U_1, and likewise in the case when the spoiler plays in U_2. □

Now assume AVG_I^ϵ is definable. Again, SC consists of two unary predicates, U_1 and U_2. Let $\Delta = (1-2\epsilon)/16$. Given two finite sets U_1 and U_2, we translate them into intervals $[0,\Delta]$ and $[1-\Delta,1]$. By translating a finite set X with $\min X = c, \max X = d > c$ into an interval $[a,b]$ we mean that we map it to the set X' containing exactly the numbers of the form $a + \frac{(x-c)(b-a)}{d-c}$ where $x \in X$; clearly $X' \subset [a,b]$. As the next step, we define $U_1^0 = U_1' \cup \{4\Delta - x \mid x \in U_1'\}$ and $U_2^0 = U_2' \cup \{2-4\Delta - x \mid x \in U_2'\}$. One observes $U_1^0 \subseteq [0,4\Delta]$ and $U_2^0 \subseteq [1-4\Delta,1]$.

The preceding shows that U_1^0 and U_2^0 are $\mathrm{FO}+\mathrm{POLY}$-definable. Thus, the set $C = U_1^0 \cup U_2^0 \subset [0,1]$ is definable in $\mathrm{FO}+\mathrm{POLY}$. Now easy calculations show that

$$\mathrm{AVG}(C) \quad = \quad \frac{1}{8} - \frac{\epsilon}{4} + \frac{m}{n+m} \cdot \frac{3+2\epsilon}{4}$$

where n is the cardinality of U_1 and m is the cardinality of U_2.

We now define a sentence Φ by letting $D \models \Phi$ iff $\mathrm{AVG}^\epsilon(C) = \mathrm{AVG}_I^\epsilon(C) > 1/2$. Let $c_0 = 1 + \frac{16\epsilon}{3-6\epsilon} > 1$. Assume $m > c_0 \cdot n$. Plugging this into the equation for $\mathrm{AVG}(C)$, we derive $\mathrm{AVG}(C) > 1/2 + \epsilon$; thus, in this case $\mathrm{AVG}^\epsilon(C) > 1/2$ no matter which ϵ-approximation of the average is picked, and thus $D \models \Phi$. Similarly, if we assume $n > c_0 \cdot m$, we derive $\mathrm{AVG}(C) < 1/2 - \epsilon$, and thus $\mathrm{AVG}^\epsilon(C) < 1/2$ and $D \models \neg\Phi$. Hence, Φ is a (c_0,c_0)-separating sentence, which is definable in $\mathrm{FO}(SC, \langle \mathbb{R}, \Omega, +, \cdot\rangle)$. This contradiction proves the theorem. □

We now briefly consider the spatial aggregate operator *volume*. First, it is easy to see that it is not definable in the languages $\mathrm{FO}+\mathrm{LIN}$ and $\mathrm{FO}+\mathrm{POLY}$. As was mentioned earlier, those languages have the following fundamental closure property: on a semi-linear constraint database $\mathbf{D}$, an $\mathrm{FO}+\mathrm{LIN}$ query returns a semi-linear set, and likewise, on a semi-algebraic constraint database, an $\mathrm{FO}+\mathrm{POLY}$ query returns a semi-algebraic set.

This closure property can no longer be guaranteed if one allows volume operators, that is, operators Vol that for every formula $\varphi(\vec{x}, \vec{y})$, produce a formula $\psi(\vec{x}, z) \equiv \text{Vol}\vec{y}\ \varphi(\vec{x}, \vec{y})$ such that $\mathbf{D} \models \psi(\vec{a}, v)$ iff $v = \text{Vol}(\varphi(\vec{a}, \mathbf{D}))$. To see this for the semi-linear case, consider a semi-linear set $S \subseteq \mathbb{R}^3$ defined by $(x > 0) \wedge (0 < y < x) \wedge (0 < z < x)$. Let $\varphi(x, y, z)$ be $S(x, y, z)$. Then $\text{Vol}(y, z)\ \varphi(x, y, z)$ is true on a pair (a, v) with $a > 0$ iff $v = a^2$, which shows the failure of closure. In the case of semi-algebraic sets, one can define functions such as $\ln x$ or $\arctan(x)$ with the help of volume. These functions are not semi-algebraic.

Volume is not definable, but can it be approximated? The reason to think that this may be the case is the following result. Suppose $\varphi(\vec{x}, \vec{y})$ is an FO($\mathbf{R}$) formula, defining a semi-algebraic set $S \subseteq [0, 1]^{n+m}$. Then, for every $\epsilon > 0$, there is an FO($\mathbf{R}$) formula $\psi_\epsilon(\vec{x}, z)$ such that, for every $\vec{a} \in [0, 1]^n$, $\mathbf{R} \models \exists z\ \varphi(\vec{a}, z)$, and for any $0 \leq v \leq 1$ such that $\mathbf{R} \models \varphi(\vec{a}, v)$, we have $|v - V| < \epsilon$, for V being the volume of the set $\{\vec{b} \in [0, 1]^m \mid \mathbf{R} \models \varphi(\vec{a}, \vec{b})\}$.

To achieve approximability of volume in FO + Poly, we only have to replace FO($\mathbf{R}$) formulae by FO($SC, \mathbf{R}$) (that is, FO + Poly) formulae. This motivates the following definition. We say that, for $\epsilon > 0$, the operator Vol_I^ϵ is definable in FO + Poly if, for every SC and every FO + Poly formula $\varphi(\vec{x}, \vec{y})$, there exists a formula $\psi(\vec{x}, z)$ such that, for any semi-algebraic constraint database $\mathbf{D}$, and every $\vec{a} \in [0, 1]^n$, the following holds:

1. $\mathbf{D} \models \exists z\ \psi(\vec{a}, z)$, and
2. if $\mathbf{D} \models \psi(\vec{a}, v)$, then $0 \leq v \leq 1$ and $|v - \text{Vol}(\varphi(\vec{a}, D) \cap [0, 1]^m)| < \epsilon$.

However, it turns out that this innocent looking move from FO($\mathbf{R}$) to FO + Poly (that is, FO($SC, \mathbf{R}$)) changes the picture completely.

Theorem 1.92. *The operator* Vol_I^ϵ *is not definable in* FO + Poly*, for any* $\epsilon < 1/2$.

Proof sketch. The proof is again by reduction to separating sentences; however, the reduction is more involved than that for the Avg operator. In particular, the reduction can only be carried out if the input constraint database is finite and has an initial segment of natural numbers as its active domain. To prove that FO + Poly cannot define a separating sentence on such structures, one can no longer use games, and instead has to rely on circuit lower bounds. □

Note that the bound $1/2$ is tight: for every $\epsilon > 1/2$, Vol_I^ϵ is definable, as the cases of volume being 0 or 1 can be tested in FO + Poly, and in all other cases, $1/2$ is an approximation.

1.11.2 Higher-order features

So far we have only dealt with first-order logic over embedded finite models and constraint databases. As we showed a number of limitations of FO($SC, \mathfrak{M}$)

in both contexts, it is natural to ask how to extend it to overcome those shortcomings. The question arises in both the embedded and the constraint settings. In the first case, the solution is rather easy, and essentially follows the standard techniques of (finite) model theory, such as adding fixpoint operators or second-order quantification. Still, one has to be careful to avoid getting undecidable languages over nice structures, such as the real field. In the constraint setting, the answer to this question is a bit trickier, but we shall see that nice languages can still be obtained that express properties such as topological connectivity.

In the embedded case, we deal here only with adding second-order quantification, but the reader should see that one can similarly add fixpoint or transitive closure operators, for example. In the case of constraint databases, we specifically consider the case of topological connectivity, although other topological queries inexpressible in FO + POLY could be considered as well.

Second-order logic over embedded finite models.

One can define this logic in the general way as $\mathrm{SO}(SC, \mathfrak{M})$ by extending $\mathrm{FO}(SC, \mathfrak{M})$ with second-order quantifiers

$$\exists S\ \varphi \qquad \forall S\ \varphi$$

where S is a relation symbol not in SC. The semantics is that for some $S \subseteq U^k$, φ holds, where k is the arity of S (or for all S, in the case of the universal quantifier). Alternatively, we can define the active-semantics version of the above, where quantifiers are

$$\exists S \in adom\ \varphi \qquad \forall S \in adom\ \varphi,$$

and the semantics changes in the way that S must be a subset of $adom(D)^k$. We shall denote the fragment of $\mathrm{SO}(SC, \mathfrak{M})$ in which all – first-order and second-order – quantifiers range over the active domain, by $\mathrm{SO}_{\mathrm{act}}(SC, \mathfrak{M})$.

We start by noticing the following:

Proposition 1.93. *The active generic collapse holds over every structure* $\mathfrak{M}$ *for second-order logic. That is, every order-generic query definable in* $\mathrm{SO}_{\mathrm{act}}(SC, \mathfrak{M})$ *is definable in* $\mathrm{SO}_{\mathrm{act}}(SC)$.

Proof. Expand $\mathfrak{M}$ to $\mathfrak{M}^<$ by adding a symbol $<$ interpreted as a linear order (if it is not there already). The proof now follows the proof for first-order logic, by establishing the Ramsey property (the proof that the Ramsey property implies the collapse does not change). As the proof of the Ramsey property is by induction on the formulae, the only additional case to consider is that of second-order quantification. It is almost the same as the case of first-order quantification (see the proof of Proposition 1.15). Note that the order relation $<$ can be eliminated from $\mathrm{SO}_{\mathrm{act}}(SC, <)$ formulae, as it is definable in second-order logic. □

Establishing the natural-active collapse is harder, as the most naive approach cannot possibly succeed.

Proposition 1.94. *Every computable property of finite SC-structures is expressible in* $\mathrm{SO}(SC, \mathbf{R})$.

Proof. In second-order logic over $\mathbf{R}$ (in fact, even $\mathbf{R}_{\mathrm{lin}}$) one can define the set of natural numbers by the formula $\varphi(n)$:

$$\exists P\ [P(0) \wedge (\forall x\ (0 < |x| < 1 \rightarrow \neg P(x))) \wedge (\forall x > 0\ (P(x) \leftrightarrow P(x-1)))] \wedge P(n).$$

Then, for any finite SC-structure over $\mathbb{R}$, one can state in second-order logic that there exists an isomorphic structure over $\mathbb{N}$; in first-order logic over $\mathbb{N}$ one can then test any property of this structure. □

At the same time, every generic query in $\mathrm{SO}_{\mathrm{act}}(SC, \mathbf{R})$ is in $\mathrm{SO}_{\mathrm{act}}(SC)$ and thus its complexity is in the polynomial hierarchy; hence $\mathrm{SO}(SC, \mathbf{R}) \neq \mathrm{SO}_{\mathrm{act}}(SC, \mathbf{R})$.

To overcome this problem, we introduce a *hybrid second-order logic* $\mathrm{HSO}(SC, \mathfrak{M})$ as a restriction of $\mathrm{SO}(SC, \mathfrak{M})$ in which all second-order quantifiers range over the active domain (but first-order quantifiers can still range over U). Then $\mathrm{HSO}_{\mathrm{act}}(SC, \mathfrak{M})$ is the restriction of $\mathrm{HSO}(SC, \mathfrak{M})$ in which all first-order quantifiers range over the active domain.

Proposition 1.95. *Let* $\mathfrak{M}$ *be o-minimal and admit quantifier-elimination. Then hybrid second-order logic has the natural-active collapse over* $\mathfrak{M}$*: that is,* $\mathrm{HSO}(SC, \mathfrak{M}) = \mathrm{HSO}_{\mathrm{act}}(SC, \mathfrak{M})$. *Furthermore, if the theory of* $\mathfrak{M}$ *is decidable and quantifier-elimination is effective, then there is an effective transformation of* $\mathrm{HSO}(SC, \mathfrak{M})$ *formulae into equivalent* $\mathrm{HSO}_{\mathrm{act}}(SC, \mathfrak{M})$ *formulae.* □

The proof of this result is very similar to the proof in the first-order case. It is by induction on the formulae, with only the case of $\exists z \alpha$ being nontrivial. In this case, one proves the exact analog of Lemma 1.30, by using essentially the same proof, as the equivalences $(*)$ in that proof are preserved under the addition of active-domain second-order quantifiers.

Thus, every generic query in $\mathrm{HSO}(SC, \mathbf{R})$ is definable in $\mathrm{SO}_{\mathrm{act}}(SC)$; that is, the behavior of hybrid second-order logic is similar to that of first-order logic, as one can apply known bounds from finite-model theory in the embedded context.

Connectivity and constraint databases

While it was shown that topological connectivity is not definable in languages such as FO + Lin and FO + Poly, it is a very useful query in many applications of spatial databases, and one would want to have a language capable of expressing it. The situation is somewhat similar to first-order logic on finite relational structures. As FO cannot express graph connectivity or transitive closure, one enriches the logic by adding fixpoint, or transitive closure operators, or second-order quantification, to give it enough power to express some desirable queries.

A similar approach is unlikely to work for constraint databases. Adding fixpoints straightforwardly to FO + LIN or FO + POLY, one loses the crucial closure property. To see this, note that by iterating a semi-linear relation $x = 2y$, one obtains relations $x = 4y$, $x = 8y$, ..., $x = 2^n y$, ..., and thus one can define the set of all powers of 2. This set is not semi-linear (nor semi-algebraic), which shows that FO + LIN and FO + POLY are not closed under fixpoint operators.

To remedy this, we take the simplest possible approach: if we need topological connectivity, just add it to the language. In this way we obtain languages FO + POLY + C and FO + LIN + C by extending the definition of the language by the following: for every formula $\varphi(\vec{x}, \vec{y})$, there is a new formula

$$\psi(\vec{x}) \quad \equiv \quad \mathsf{C}\vec{y}\ \varphi(\vec{x}, \vec{y}).$$

The semantics is as follows. Given a constraint database $\mathbf{D}$, and a tuple $\vec{a}$ of the same length as $\vec{x}$, let $\varphi(\vec{a}, \mathbf{D}) = \{\vec{b} \mid \mathbf{D} \models \varphi(\vec{a}, \vec{b})\}$. Then

$$\mathbf{D} \models \psi(\vec{a}) \quad \text{iff} \quad \varphi(\vec{a}, \mathbf{D}) \text{ is connected.}$$

The main property of these languages is that they are closed; the proofs, however, are quite different for the semi-algebraic and the semi-linear case.

Proposition 1.96. FO + POLY + C *is closed; that is, on a semi-algebraic constraint database, an* FO + POLY + C *query produces a semi-algebraic set.*

Proof. The proof is by induction on the formulae. The only nontrivial case is that of $\psi(\vec{x}) \equiv \mathsf{C}\vec{y}\ \varphi(\vec{x}, \vec{y})$. Assume that on $\mathbf{D}$, φ defines a set $S \subseteq \mathbb{R}^{n+m}$, where n is the length of $\vec{x}$ and m is the length of $\vec{y}$. Let $S_{\vec{a}}$ denote the set $\{\vec{b} \mid (\vec{a}, \vec{b}) \in S\} \subseteq \mathbb{R}^m$ for $\vec{a} \in \mathbb{R}^n$. A result in algebraic geometry known as the local triviality theorem states that for any semi-algebraic set S as above, there is a partition $\mathbb{R}^n = Y_1 \cup \ldots \cup Y_k$ such that each Y_i is semi-algebraic, and for $\vec{a}_1, \vec{a}_2 \in Y_i$, the sets $S_{\vec{a}_1}$ and $S_{\vec{a}_2}$ are homeomorphic. In particular, either all sets $S_{\vec{a}}, \vec{a} \in Y_i$ are connected, or none of them is. Hence, the result of ψ on $\mathbf{D}$ is a union of some Y_is, and thus semi-algebraic. □

The reason we cannot use the same proof for FO + LIN is that the local triviality theorem fails over $\mathbf{R}_{\text{lin}}$. In the proof above, we only used a part of that theorem, which says that the fibers $S_{\vec{a}}$ have finitely many topological types. But it also asserts that there are semi-algebraic homeomorphisms between sets $S_{\vec{a}_1}$ and $S_{\vec{a}_2}$, $\vec{a}_1, \vec{a}_2 \in Y_i$. An analog of this statement does not hold for semi-linear sets, and hence the local triviality theorem is not applicable in the semi-linear case. (In fact, one can prove local triviality for o-minimal *expansions* of the real field $\mathbf{R}$.)

There are two ways of circumventing the problem. One, quite complex, is to show that the first part of the local triviality theorem still holds for the case of semi-linear sets. But we can also give a simple direct proof of closure of $\mathrm{FO} + \mathrm{LIN} + \mathsf{C}$, which does not require the local triviality theorem.

Proposition 1.97. $\mathrm{FO} + \mathrm{LIN} + \mathsf{C}$ *is closed; that is, on a semi-linear constraint database, an* $\mathrm{FO} + \mathrm{LIN} + \mathsf{C}$ *query produces a semi-linear set.*

Proof. The proof again is by induction on the formulae, and we only consider the case of of $\psi(\vec{x}) \equiv \mathsf{C}\vec{y}\ \varphi(\vec{x}, \vec{y})$. Assume that on $\mathbf{D}$, φ defines a semi-linear set $S \subseteq \mathbb{R}^{n+m}$. Since S is semi-linear, it has a representation of the form

$$\bigvee_{i=1}^{k} \psi_i, \qquad \psi_i \ \equiv \ \bigwedge_{j=1}^{l_i} \langle \vec{a}_{ij}, \vec{x} \rangle \ \theta \ \langle \vec{b}_{ij}, \vec{y} \rangle + c_{ij}$$

where $\langle \cdot, \cdot \rangle$ denotes inner product. Let Z^i be the subset of $\mathbb{R}^{n+m}$ defined by ψ_i. For every $\vec{a} \in \mathbb{R}^n$, the set $Z^i_{\vec{a}}$ is a convex polyhedron, and thus it is connected (unless it is empty).

Let $T_1, \ldots, T_r$ be an arbitrary collection of semi-linear sets in $\mathbb{R}^p$. Define a relation $T_i \approx T_j$ if $\mathrm{cl}(T_i) \cap T_j \neq \emptyset$ or $\mathrm{cl}(T_j) \cap T_i \neq \emptyset$, where $\mathrm{cl}(\cdot)$ denotes the closure of a set. Then $T_1 \cup \ldots \cup T_k$ is connected iff the undirected graph with T_is as vertices and $\approx$ as the edge relation, is connected.

Using this, we conclude the proof as follows. Given an undirected graph G on nodes $1, \ldots, k$, we write $\vec{a} \to_{\mathbf{D}} G$ if

$$\text{there is an edge } (i, j) \text{ in } G \quad \text{iff} \quad Z^i_{\vec{a}} \approx Z^j_{\vec{a}}.$$

We have seen earlier that closure is $\mathrm{FO} + \mathrm{LIN}$-definable. Hence, there is an $\mathrm{FO} + \mathrm{LIN}$ formula $\alpha_G(\vec{x})$ such that $\mathbf{D} \models \alpha_G(\vec{a})$ iff $\vec{a} \to_{\mathbf{D}} G$. This, and the statement of the previous paragraph, imply that

$$\bigvee_{G \text{ connected}} \alpha_G(\vec{x})$$

is equivalent to $\psi(\vec{x})$, where the disjunction is taken over connected undirected graphs on $\{1, \ldots, k\}$. This proves closure, since the above is an $\mathrm{FO} + \mathrm{LIN}$ formula. □

Note that the formula produced in the proof of Proposition 1.97 may be very large, as the number of connected graphs on a k-element set is exponential in k. It turns out that a much more compact formula can always be obtained; the proof of this, however, is much more involved than the simple proof we showed above. See the bibliographic comments for more detail.

1.12 Bibliographic notes

Sections 1.2 and 1.3

For a general introduction to finite model theory, see [32] and previous chapters of this book. A standard reference on database theory is [1], which also covers many topics of finite model theory. Constraint databases were introduced in [50]; for a comprehensive treatment of this topic, see [56]. Mixing the finite and the infinite in the database context is discussed in a number of papers; see, for example, [26, 38]. The semi-algebraic "face" example is taken from [23], the semi-linear one from [56].

Other approaches to combining the finite and the infinite in model theory include metafinite structures [38] (which, in our terminology, can be described as triples consisting of a finite structure D, an infinite structure $\mathfrak{M}$, and a set of functions from $adom(D)$ to tuples over $\mathfrak{M}$), recursive structures [44] (infinite structures in which every relation is computable, and thus has a finite description by means of a Turing machine), and automatic structures [21, 52] in which predicates are given by finite automata, as opposed to arbitrary Turing machines.

Section 1.4

The notion of genericity is standard in relational databases, see [47, 1]. Various forms of collapse results were introduced in [11, 46, 62].

Section 1.5

The active generic collapse was proved independently in [11] and [60]. The Ramsey property is from [11], and the proof given here follows closely the one in [15]. Analytic signatures and total collapse are also discussed in [15]. For a survey on Ramsey theory, see [37]. That there exist properties definable in $\text{FO}_{\text{act}}(SC, <)$ but not $\text{FO}_{\text{act}}(SC)$ is shown in [1] (the result is attributed to Gurevich).

Section 1.6

Proposition 1.23 is a standard exercise on coding in first-order logic over $\langle \mathbb{N}, +, \cdot \rangle$ (cf. [33]); in this form the result was explicitly stated in [42]. The natural-active collapse without interpreted structure (Theorem 1.25) was proved in [46]. An earlier weaker result [3] showed that unrestricted quantification can always be replaced by quantification over some finite superset of the active domain ("4 Russians Theorem").

The concept of o-minimality was introduced by [63], and has been extensively studied in the model-theoretic literature; see [76] for an overview. O-minimality of the exponential field is from [80]; [75] shows that it does not have quantifier-elimination. The uniform bounds theorem (Theorem 1.29) is from [64]. For general model-theoretic properties of structures, see standard texts such as [27].

The natural-active collapse (Theorem 1.26) is from [15]. It was proved earlier by nonconstructive means in [14]. The linear case, sketched in Section 1.6.4, was proved in [62]. (See also [72]). The material of Section 1.6.5 is from [15], except for Proposition 1.32, which is from [35]. A version of the algorithm for the natural-active collapse adapted to FO + Poly was presented in [58].

A different proof of the natural-active collapse for FO + Poly was given in [8]. It only applies to finite structures in which all relations are unary, but achieves much better complexity bounds than the general algorithm presented here.

The natural-generic collapse (Section 1.6.7) was the first collapse result proved for polynomial constraints, see [11]. That proof used the technique of nonstandard universes; here we derived the result as a corollary of the natural-active collapse. Some extensions of this collapse results are known, for example, for quasi-o-minimal structures [9] (which include all o-minimal ones, as well as $\langle \mathbb{N}, +, < \rangle$) and for a larger class of structures with finite VC dimension (Theorem 1.35) [7].

More expressivity bounds were proved in [28] which showed that parity is not definable in FO + Poly even if the input is a set of natural numbers such that the distance between two consecutive elements is 1 or 2. It also extended some expressivity bounds to algebraically closed fields.

Section 1.7

For general model-theoretic background, the reader is referred to [27, 45]. The notion of pseudo-finite homogeneity was introduced in [9, 35]. Theorem 1.38 is from [35], as are the notion of pseudo-finite saturation and Proposition 1.39. The proof of Proposition 1.41 uses the fact that term algebras are stable, and some conditions for showing that a structure does not have the finite cover property; those can be found in [45, 65].

The isolation property, Proposition 1.43 and Corollary 1.44 are from [9]. Proposition 1.45 is from [20]. Proposition 1.46 is a special case of a more general result (that shows the isolation property for quasi-o-minimal structures) in [9]; see also [35].

Section 1.8

For more on VC dimension and its applications in learning theory, see [5, 22]. For applications in logic, and for the basic facts used in the proof of Theorem 1.47, see [57, 70, 76]. In particular, [57] shows that o-minimal structures have finite VC dimension.

The class AC^0/poly used in the Section is a standard complexity class (a.k.a. non-uniform AC^0), see, for example, [49]. Bounds for AC^0/poly implying inexpressibility of queries such as parity and connectivity can be found in [4, 36, 30].

Theorem 1.47 is from [20]. The material of Section 1.8.1 is partly from [60] (which showed one direction of Theorem 1.48; the other direction is from [18]). In [15] it is shown how to use the random ternary relation to express even more queries (for example, parity), thereby refuting a conjecture from [42] that tied such expressivity results to decidability of the theory of the underlying structure. For basic information about the random graph (and more generally, random structures, the reader is referred to [32, 45]).

The material of Section 1.8.2, including Proposition 1.49, is from [19] (which gives a slightly better complexity bound). The structure $\mathcal{S}$ was studied in [24], where the connection with regular languages was shown, in [21], which showed how to interpret automatic structures in it, and in [20], where further model-theoretic properties, including infinite VC dimension, were proved.

Section 1.9

The material on reductions (Section 1.9.1) is from [42], which shows many inexpressibility results for $\mathrm{FO} + \mathrm{POLY}$ by reducing them to parity. Topological properties (Section 1.9.2) of constraint databases were studied in [61, 54, 55, 69]. The conical local structure of semi-algebraic sets is described in texts [23, 10]. Theorem 1.52 is from [54]. The failure of Theorem 1.52 for multiple regions was shown by [40].

Section 1.9.3 is based on [78], which contains many examples of queries expressible and inexpressible in $\mathrm{FO} + \mathrm{LIN}$. More examples of the power of $\mathrm{FO} + \mathrm{LIN}$ can be found in [2], which also conjectured that ExistsLine is not expressible in $\mathrm{FO} + \mathrm{LIN}$. That was first proved in [13], but the proof was very complicated; the simple proof given here is due to [68]. The result on the line segment connecting two boundary points is due to [13].

Section 1.10

Safety is a central notion in relational database theory, see [1]. See [79] for undecidability for first-order logic. Safety with scalar functions was studied in [34]. The state-safety problem was introduced in [3, 6], where decidability was proved for some structures (e.g., $\langle \mathbb{N}, < \rangle$).

The concept of safe translation is from [16]. Proposition 1.57 is from [71] (where a complete description of the structure and the proof of decidability can be found). Propositions 1.58 and 1.59 are from [16]. Extensions to Datalog are discussed in [66, 73].

Section 1.10.3 follows closely [16], except that here we present range-restriction in terms of definable functions, rather than just algebraic formulae. For properties of semi-linear and semi-algebraic functions used in the proof of Corollary 1.68, see [59, 76].

The reduction from infinite safety to finite safety (Theorem 1.74), as well as the canonical representation for convex polytopes, is from [16]. More examples of canonical representations can be found in [16]. The first proof of Theorem 1.76 is based on applying Theorem 1.74 to canonical representations for semi-linear sets, given in [77]. The other proof uses decidability of semi-linearity, proved in [31].

The decidability result for safety of conjunctive queries over o-minimal structures is from [16]; it uses decidability of containment proved in [48]. (See also Chapter 2 of [56] which discusses some subtle points related to the decidability result of [48].) Undecidability of finiteness of the set of solutions of a Diophantine equation (which proves Proposition 1.81) is from [29]. Proposition 1.83 is from [16]. All results in the Section on the dichotomy theorem are from [16].

Section 1.11

Aggregation is a standard feature of database query languages [1, 74]. The results dealing with the average operator are from [17]. How to play a game on ordered sets is described in [43].

That volumes can be approximated for first-order formulae over the real field was shown in [51, 53]. Theorem 1.92 showing that these results do not extend to constraint databases is from [17].

Hybrid logics were introduced in [15], where collapse results were proved. There exist higher-order logics capturing complexity classes over constraint databases defined with order [39] and with linear constraints [41]. The material on connectivity is from [12]. The local triviality theorem used in the proof of Proposition 1.96 can be found in [23, 10, 76]. The proof of Proposition 1.97 in [12] is more involved and relies on special properties of cylindric decompositions [25] of semi-linear sets; the simple proof presented here is due to [81] (the simplicity is achieved at the expense of exponential-size formulae).

References

1. S. Abiteboul, R. Hull and V. Vianu. *Foundations of Databases.* Addison-Wesley, 1995.
2. F. Afrati, T. Andronikos, T. Kavalieros. On the expressiveness of query languages with linear constraints: capturing desirable spatial properties. In *Constraint Databases and Applications*, Springer LNCS 1191, 1997, pages 105–115.
3. A.K. Ailamazyan, M.M. Gilula, A.P. Stolboushkin and G.F. Shvarts. Reduction of a relational model with infinite domains to the finite-domain case. *Doklady Akademii Nauk SSSR*, 286 (1) (1986), 308–311. Translation in *Soviet Physics – Doklady*, 31 (1986), 11–13.
4. M. Ajtai. Σ_1^1 formulae on finite structures. *Annals of Pure and Applied Logic*, 24 (1983), 1–48.
5. M. Anthony and N. Biggs. *Computational Learning Theory.* Cambridge Univ. Press, 1992.
6. A. Avron and J. Hirshfeld. On first order database query languages. In *IEEE Symposium on Logic in Computer Science*, 1991, pages 226–231.
7. J. Baldwin and M. Benedikt. Stability theory, permutations of indiscernibles, and embedded finite models. *Trans. Amer. Math. Soc.* 352 (2000), 4937–4969.
8. S. Basu. New results on quantifier elimination over real closed fields and applications to constraint databases. *Journal of the ACM*, 46 (1999), 537–555.
9. O. Belagradek, A. Stolboushkin, M. Taitslin. Extended order-generic queries. *Annals of Pure and Applied Logic*, 97 (1999), 85–125.
10. R. Benedetti and J.-J. Risler. *Real Algebraic and Semi-algebraic Sets.* Hermann, Paris, 1990.
11. M. Benedikt, G. Dong, L. Libkin and L. Wong. Relational expressive power of constraint query languages. *Journal of the ACM* 45 (1998), 1–34.
12. M. Benedikt, M. Grohe, L. Libkin and L. Segoufin. Reachability and connectivity queries in constraint databases. In *ACM Symposium on Principles of Database Systems*, 2000, pages 104–115.
13. M. Benedikt, H. J. Keisler. Definability over linear constraints. In *Conf. on Commputer Science Logic*, 2000, pages 217–231.
14. M. Benedikt and L. Libkin. On the structure of queries in constraint query languages. In *IEEE Symposium on Logic in Computer Science*, 1996, pages 25–34.

15. M. Benedikt and L. Libkin. Relational queries over interpreted structures. *Journal of the ACM*, 47 (2000), 644–680.
16. M. Benedikt and L. Libkin. Safe constraint queries. *SIAM J. Comput.*, 29 (2000), 1652–1682.
17. M. Benedikt and L. Libkin. Aggregate operators in constraint query languages. *Journal of Computer and System Sciences*, 64 (2002), 628–654.
18. M. Benedikt and L. Libkin. Unpublished notes, 1999.
19. M. Benedikt, L. Libkin, T. Schwentick, L. Segoufin. String operations in query languages. In *ACM Symposium on Principles of Database Systems*, 2001, pages 183–194.
20. M. Benedikt, L. Libkin, T. Schwentick, L. Segoufin. A model-theoretic approach to regular string relations. In *IEEE Symposium on Logic in Computer Science*, 2001, pages 431–440.
21. A. Blumensath, E. Grädel. Automatic structures. In *IEEE Symposium on Logic in Computer Science*, 2000, pages 51–62.
22. A. Blumer, A. Ehrenfeucht, D. Haussler, M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM* 36 (1989), 929–965.
23. J. Bochnak, M. Coste, M.-F. Roy. *Real Algebraic Geometry.* Springer Verlag, 1998.
24. V. Bruyère, G. Hansel, C. Michaux, R. Villemaire. Logic and p-recognizable sets of integers. *Bull. Belg. Math. Soc.* 1 (1994), 191–238.
25. B.F. Caviness and J.R. Johnson, Eds. *Quantifier Elimination and Cylindrical Algebraic Decomposition.* Springer Verlag, 1998.
26. A. Chandra and D. Harel. Computable queries for relational databases. *Journal of Computer and System Sciences* 21(2):156–178, 1980.
27. C.C. Chang and H.J. Keisler. *Model Theory.* North Holland, 1990.
28. O. Chapuis and P. Koiran. Definability of geometric properties in algebraically closed fields. *Mathematical Logic Quarterly*, 45 (1999), 533–550.
29. M. Davis. On the number of solutions of Diophantine equations. *Proc. AMS* 35 (1972), 552–554.
30. L. Denenberg, Y. Gurevich and S. Shelah. Definability by constant-depth polynomial-size circuits. *Information and Control* 70 (1986), 216–240.
31. F. Dumortier, M. Gyssens, L. Vandeurzen, and D. Van Gucht. On the decidability of semi-linearity of semi-algebraic sets, and its implications for spatial databases. *Journal of Computer and System Sciences*, 58 (1999), 535–571. Correction in *JCSS*, 59 (1999), 557–562.
32. H.-D. Ebbinghaus and J. Flum. *Finite Model Theory.* Springer Verlag, 1995.
33. H. B. Enderton. *A Mathematical Introduction to Logic.* Academic Press, New York, 1972.
34. M. Escobar-Molano, R. Hull and D. Jacobs. Safety and translation of calculus queries with scalar functions. In *ACM Symposium on Principles of Database Systems*, 1993, pages 253–264.
35. J. Flum and M. Ziegler. Pseudo-finite homogeneity and saturation. *J. Symbolic Logic* 64 (1999), 1689–1699.
36. M. Furst, J. Saxe, M. Sipser. Parity, circuits, and the polynomial-time hierarchy. *Math. Systems Theory* 17 (1984), 13–27.
37. R.L. Graham, B.L. Rothschild and J.H. Spencer. *Ramsey Theory.* John Wiley & Sons, New York, 1990.
38. E. Grädel and Y. Gurevich. Metafinite model theory. *Information and Computation*, 140 (1998), 26–81.

39. E. Grädel and S. Kreutzer. Descriptive complexity theory for constraint databases. In *Conf. on Computer Science Logic*, 1999, pages 67–81.
40. M. Grohe, L. Segoufin. On first-order topological queries. *ACM Transactions on Computational Logic* 3 (2002), 336–358.
41. S. Grumbach, G. Kuper. Tractable recursion over geometric data. In *Constraint Programming 1997*, pages 450–462.
42. S. Grumbach and J. Su. Queries with arithmetical constraints. *Theoretical Computer Science* 173 (1997), 151–181.
43. Y. Gurevich. Toward logic tailored for computational complexity. In *Computation and Proof Theory*, Springer, 1984, pages 175–216.
44. D. Harel. Towards a theory of recursive structures. In *Conf. Math. Foundations of Comp. Sci.*, 1998, pages 36–53.
45. W. Hodges. *Model Theory*. Cambridge, 1993.
46. R. Hull and J. Su. Domain independence and the relational calculus. *Acta Informatica* 31:513–524, 1994.
47. R. Hull and C. K. Yap. The format model: a theory of database organization. *Journal of the ACM* 31 (1984), 518–537.
48. O.H. Ibarra and J. Su. A technique for proving decidability of containment and equivalence of linear constraint queries. *Journal of Computer and System Sciences*, 59 (1999), 1–28.
49. N. Immerman. *Descriptive Complexity*. Springer Verlag, 1999.
50. P. Kanellakis, G. Kuper, and P. Revesz. Constraint query languages. *Journal of Computer and System Sciences*, 51 (1995), 26–52. Extended abstract in *ACM Symposium on Principles of Database Systems*, 1990, pages 299–313.
51. M. Karpinski and A. Macintyre. Approximating the volume of general Pfaffian bodies. In *Structures in Logic and Computer Science: A selection of Essays in Honor of A. Ehrenfeucht*, Springer LNCS 1261, 1997, pages 162–173.
52. B. Khoussainov and A. Nerode. Automatic presentations of structures. In *Logic and Computational Complexity*, Springer LNCS vol. 960, 1994, pages 367–392.
53. P. Koiran. Approximating the volume of definable sets. In *Symposium on Foundations of Computer Science*, 1995, pages 134-141,
54. B. Kuijpers, J. Paredaens and J. Van den Bussche. Topological elementary equivalence of closed semi-algebraic sets in the real plane. *J. Symbolic Logic* 65 (2000), 1530–1555.
55. B. Kuijpers and J. Van den Bussche. On capturing first-order topological properties of planar spatial databases. In *Int. Conf. on Database Theory*, 1999, pages 187–198.
56. G. Kuper, L. Libkin and J. Paredaens, eds. *Constraint Databases*. Springer Verlag, 2000.
57. M. C. Laskowski. Vapnik-Chervonenkis classes of definable sets. *J. London Math. Soc.*, 45:377–384, 1992.
58. L. Libkin. *Elements of Finite Model Theory*. Springer 2004.
59. D. Marker, M. Messmer and A. Pillay. *Model Theory of Fields*. Springer Verlag, 1996.
60. M. Otto and J. Van den Bussche. First-order queries on databases embedded in an infinite structure. *Information Processing Letters* 60 (1996), 37–41.
61. C. Papadimitriou, D. Suciu and V. Vianu. Topological queries in spatial databases. *Journal of Computer and System Sciences*, 58 (1999), 29–53.
62. J. Paredaens, J. Van den Bussche, and D. Van Gucht. First-order queries on finite structures over the reals. *SIAM J. Comput.* 27 (1998), 1747–1763.

63. A. Pillay, C. Steinhorn. Definable sets in ordered structures. *Bulletin of the AMS* 11 (1984), 159–162.
64. A. Pillay, C. Steinhorn. Definable sets in ordered structures. III. *Trans. AMS* 309 (1988), 469–476.
65. B. Poizat. *A Course in Model Theory*. Springer Verlag, 2000.
66. P. Revesz. Safe query languages for constraint databases. *ACM TODS* 23 (1998), 58–99.
67. R.T. Rockafellar. *Convex Analysis*. Princeton Univ. Press, 1970.
68. L. Segoufin. Private communication, 2001.
69. L. Segoufin and V. Vianu. Querying spatial databases via topological invariants. *Journal of Computer and System Sciences*, 61 (2000), 270–301.
70. S. Shelah. Stability, the f.c.p., and superstability. *Ann. of Math. Logic* 3 (1971), 271–362.
71. A. Stolboushkin and M. Taitslin. Finite queries do not have effective syntax. *Information and Computation*, 153 (1999), 99–116.
72. A. Stolboushkin and M.Taitslin. Linear vs. order constraint queries over rational databases. In *ACM Symposium on Principles of Database Systems*, 1996, pages 17–27.
73. A. Stolboushkin and M. Taitslin. Safe stratified datalog with integer order does not have syntax. *ACM TODS* 23 (1998), 100–109.
74. J. D. Ullman and J. Widom. *A First Course in Database Systems*. Prentice-Hall, 1997.
75. L. van den Dries. Remarks on Tarski's problem concerning (R,+,*,exp). In *Logic Colloquium'82*, North Holland, 1984, pages 97–121.
76. L. van den Dries. *Tame Topology and O-Minimal Structures*. Cambridge, 1998.
77. L. Vandeurzen, M. Gyssens and D. Van Gucht. An expressive language for linear spatial database queries. In *ACM Symposium on Principles of Database Systems*, 1998, pages 109–118.
78. L. Vandeurzen, M. Gyssens and D. Van Gucht. On the expressiveness of linear-constraint query languages for spatial databases. *Theoretical Computer Science* 254 (2001), 423–463.
79. M.Y. Vardi. The decision problem for database dependencies. *Information Processing Letters*, 12 (1981), 251–254.
80. A.J. Wilkie. Model completeness results for expansions of the ordered field of real numbers by restricted Pfaffian functions and the exponential function. *J. Amer. Math. Soc.* 9 (1996), 1051–1094.
81. M. Yannakakis. Private communication, 1999.

Index