SIAM J. COMPUT.
Vol. 35, No. 6, pp. 1386–1439


# LINEARIZATION AND COMPLETENESS RESULTS FOR TERMINATING TRANSITIVE CLOSURE QUERIES ON SPATIAL DATABASES*

FLORIS GEERTS†, BART KUIJPERS†, AND JAN VAN DEN BUSSCHE†

**Abstract.** We study queries to spatial databases, where spatial data are modeled as semialgebraic sets, using the relational calculus with polynomial inequalities as a basic query language. We work with the extension of the relational calculus with terminating transitive closures. The main result is that this language can express the linearization of semialgebraic databases. We also show that the sublanguage with linear inequalities only can express all computable queries on semilinear databases. As a consequence of these results, we obtain a completeness result for topological queries on semialgebraic databases.

**Key words.** constraint databases, real algebraic geometry, transitive closure logics, query languages

**AMS subject classifications.** 68P15, 14P10, 57R05, 03B70

**DOI.** 10.1137/S0097539702410065

**1. Introduction.** Spatial database systems [1, 8, 12, 24, 25, 42] are concerned with the representation and manipulation of data that have a geometric or topological interpretation. Conceptually, spatial databases store geometric figures, which are possibly infinite sets of points in a real space $\mathbf{R}^n$. The framework of constraint databases [34], introduced by Kanellakis, Kuper, and Revesz [27], provides an elegant and powerful model for spatial databases. In the setting of the constraint model, a geometric figure is finitely represented as a Boolean combination of polynomial equalities and inequalities over the real numbers. Such figures are known as semialgebraic sets. Special cases of figures definable by linear polynomials are known as semilinear sets [6].

The relational calculus or first-order logic, expanded with polynomial equalities and inequalities and evaluated over the semialgebraic sets (viewed as relations over the reals) stored in the database, serves as a basic spatial query language and is denoted by FO+POLY. The special case of queries expressed using linear equalities and inequalities is denoted by FO+LIN. Several authors have argued that the restriction to linear polynomial constraints provides a sufficiently general framework for spatial database applications [21, 46, 47]. Indeed, in geographic information systems (GIS), which form one of the main application areas of spatial databases, linear representations are used to model spatial objects [34, Chapter 9]. Existing implementations of the constraint model, for instance, the work on the system DEDALE [19, 20, 21], are also restricted to linear polynomial constraints. Indeed, for these constraints, the evaluation of queries expressed in FO+LIN is conceptually easier and can be computed by numerous efficient algorithms for geometric operations on linear figures [38]. The computational complexity of evaluating an FO+LIN query on linear constraint

*Received by the editors June 22, 2002; accepted for publication (in revised form) November 13, 2005; published electronically April 7, 2006.
http://www.siam.org/journals/sicomp/35-6/41006.html
†Theoretical Computer Science Group, Hasselt University and Transnational University of Limburg, Agoralaan, Gebouw D, B-3590 Diepenbeek, Belgium (floris.geerts@uhasselt.be, bart.kuijpers@uhasselt.be, jan.vandenbussche@uhasselt.be). The first author is a postdoctoral researcher of the FWO-Vlaanderen.

databases ($NC^1$) is also slightly lower than that of evaluating an FO+Poly query on polynomial constraint databases (NC) [2, 22, 41].

Since the expressive power of the basic query languages FO+Poly and FO+Lin is rather limited [34, Chapters 5 and 6], it makes sense to consider more powerful extensions.

Various extensions with recursion have already been introduced and studied. Grumbach and Kuper [18] defined syntactic variants of DATALOG with linear constraints which capture exactly the queries on linear constraint databases in the plane, which have PTIME and PSPACE data complexity. Kreutzer [30] defines several recursive languages capturing PTIME and PSPACE on a restricted class of linear constraint databases. Termination properties of DATALOG with polynomial constraints are investigated by Kuijpers et al. [31] and Kuijpers and Smits [33].

In this paper, we study the expressive power of FO+Poly (and FO+Lin) extended with the transitive closure operator (TC). Transitive closure is a simple form of recursion and we apply it only in a simple way; specifically, we do not apply TC to formulas with extra free variables (parameters), as is allowed in the standard definition of transitive closure logic [11].

In the first part of the paper, we show that when we extend the TC operator with explicit stop conditions, which we denote by TCS, the language FO+Lin+TCS is computationally complete on the class of databases definable by linear polynomials with integer coefficients (**Z**-linear databases). This means that for every partial computable query $Q$, there is a formula $\varphi$ such that for every **Z**-linear database $D$, the evaluation of $\varphi$ on $D$ terminates if and only if $Q(D)$ is defined and results in $Q(D)$. It remains an open problem whether FO+Lin+TC (without explicit stop conditions) is also computationally complete in this sense. We point out that recently, Kreutzer [29] defined an extension of FO+Lin with a different transitive closure operator and proved completeness on linear constraint databases as well (see the end of section 3 for more details).

In the second part of the paper, we investigate the expressive power of FO+Poly+TCS on general polynomial constraint databases. In contrast to the linear case, we have not been able to establish the computational completeness. Yet, we will show that the language is complete as far as all Boolean topological queries are concerned.

In order to prove this result, we show that there is a formula of FO+Poly+TC (no stop conditions are needed) that expresses *linearization*: when evaluated on an arbitrary semialgebraic set $A$, it results in a semilinear set $\widehat{A}$ topologically equivalent (i.e., homeomorphic) to $A$. Moreover, $\widehat{A}$ can be assumed to be a **Z**-linear set.

Our linearization formula always terminates, in the sense that on any input $A$, every application of the TC operator in the formula converges after a finite number of stages. In the case when $A$ is bounded, the linearization formula can be sharpened to produce a set $\widehat{A}$ that is arbitrarily close to the input set $A$.

The components of the linearization formula require a number of new geometric constructions in FO+Poly. More specifically, we introduce the uniform cone radius decomposition of semialgebraic sets. Using the result of Geerts [14], we show that this decomposition can be defined in FO+Poly. Also, we define the regular decomposition of semialgebraic sets and use the results of Rannou [39] to show that this decomposition is expressible in FO+Poly.

The linearization algorithm also implies that semialgebraic sets in $\mathbf{R}^n$ can be linearized, a fact which has been known for a long time [7]. The standard constructive linearization (or triangulation) algorithm for semialgebraic sets, which is attributed to Hardt [26], can be found in the standard textbook on real algebraic geometry [6, section 9.2] and in the more recent book on algorithms in real algebraic geometry [3, Chapter 5].

The difference of the existing linearization algorithm for semialgebraic sets is that the polynomials appearing in the description of the semialgebraic sets are used explicitly. This is not possible in our setting because we can only interact with the semialgebraic set using queries. Because of this, our algorithm is not likely to be as efficient as the existing algorithm (we did not compute the exact complexity though). Moreover, our linearization is based on the local conical behavior of semialgebraic sets, and the inductive construction based on these cones might be of interest in real algebraic geometry.

Finally, we use the linearization formula in the following two ways to show the expressibility in FO+Poly+TC of two common queries which are known to be not expressible in FO+Poly: (1) We show that the connectivity query on polynomial constraint databases is expressible by an always terminating formula in FO+Poly+TC; (2) we show that there is a formula in FO+Poly+TC that always has a terminating evaluation and that evaluates on a given bounded semialgebraic set $A$ to a number that is arbitrarily close to the volume of $A$.

We remark that some of the above results were already described (in considerably less detail) for two dimensions [16] and arbitrary dimensions [13].

This paper is organized as follows. Section 2 gives the definition of polynomial constraint databases and defines the standard first-order query languages. Section 3 extends these languages with a transitive closure operator. Section 4 studies the computational completeness of these extensions and gives some inexpressibility results of the first-order query languages. Section 5 provides geometric tools necessary for the linearization construction. Section 6 presents the construction itself and discusses applications of linearization (testing connectivity and approximating the volume).

**2. Preliminaries.** We denote the set of real numbers by $\mathbf{R}$, the set of algebraic numbers by $\mathbf{A}$, the set of integers by $\mathbf{Z}$, and the set of natural numbers by $\mathbf{N}$.

A *semialgebraic set in* $\mathbf{R}^n$ is a finite union of sets definable by conditions of the form

$$f_1(\vec{x}) = f_2(\vec{x}) = \cdots = f_k(\vec{x}) = 0, \quad g_1(\vec{x}) > 0, \quad g_2(\vec{x}) > 0, \dots, g_\ell(\vec{x}) > 0,$$

where $\vec{x} = (x_1, \dots, x_n) \in \mathbf{R}^n$, and where $f_1(\vec{x}), \dots, f_k(\vec{x}), g_1(\vec{x}), \dots, g_\ell(\vec{x})$ are multivariate polynomials in the variables $x_1, \dots, x_n$ with integer coefficients. A $\mathbf{Z}$-*linear* ($\mathbf{A}$-*linear*) *set in* $\mathbf{R}^n$ is a semialgebraic set which can be defined in terms of linear polynomials with *integer* (*algebraic*) coefficients.

A *database schema* $\mathcal{S}$ is a finite set of relation names, each with a given arity. A *polynomial constraint database* $D$ over $\mathcal{S}$ assigns to each $S \in \mathcal{S}$ a semialgebraic set $S^D$ in $\mathbf{R}^k$, where $k$ is the arity of $S$. A $\mathbf{Z}$-*linear* ($\mathbf{A}$-*linear*) *constraint database* assigns to each $S \in \mathcal{S}$ a $\mathbf{Z}$-linear ($\mathbf{A}$-linear) set $S^D$ in $\mathbf{R}^k$, where $k$ is the arity of $S$. A $k$-ary *query* over $\mathcal{S}$ is a partial function $Q$ that maps each database $D$ over $\mathcal{S}$ to a $k$-ary relation $Q(D) \subseteq \mathbf{R}^k$.

First-order logic over the vocabulary $(+, \times, 0, 1, <)$ expanded with the database schema $\mathcal{S}$ provides a basic query language which we denote by FO+Poly. The sublanguage of FO+Poly consisting of the formulas that do not use multiplication is denoted by FO+Lin.

Every formula $\varphi(x_1,\dots,x_k)$ in FO+Poly expresses a $k$-ary query as follows: Let $D$ be a database over $\mathcal{S}$; then

$$\varphi(D) = \{(a_1,\dots,a_k) \in \mathbf{R}^k \mid \langle \mathbf{R}, D\rangle \models \varphi(a_1,\dots,a_k)\}.$$

Here, by $\langle \mathbf{R}, D\rangle$ we mean the standard structure of the reals $\langle \mathbf{R}; +, \times, 0, 1, <\rangle$ expanded with the relations (semialgebraic sets) in $D$.

*Example* 2.1. Suppose that $\mathcal{S}$ contains the binary relation name $S$. Then the FO+Poly formula

$$\varphi(x,y) \equiv \exists\varepsilon\forall x'\forall y'\big(\varepsilon > 0 \wedge ((x-x')^2 + (y-y')^2 < \varepsilon \rightarrow S(x',y'))\big)$$

expresses the query that maps any database $D$ over $\mathcal{S}$ to the interior of $S^D$. $\diamond$

FO+Poly queries can be effectively evaluated as follows. Let $\varphi(x_1,\dots,x_k)$ be an FO+Poly formula over schema $\mathcal{S}$, and let $D$ be a database over $\mathcal{S}$. For every $S \in \mathcal{S}$, we represent the set $S^D$ by some quantifier-free polynomial constraint formula $\psi_S(y_1,\dots,y_k)$, where $k$ is the arity of $S$, that defines $S^D$ in the sense that $S^D = \{(a_1,\dots,a_k) \in \mathbf{R}^k \mid \mathbf{R} \models \psi_S(a_1,\dots,a_k)\}$. Now replace in $\varphi$ every subformula of the form $S(z_1,\dots,z_k)$ with $\psi_S(z_1,\dots,z_k)$. Making these replacements for every $S \in \mathcal{S}$, we obtain a polynomial constraint formula which we denote by $\varphi^D$ and which defines $\varphi(D)$ in the sense that $\varphi(D) = \{(a_1,\dots,a_k) \in \mathbf{R}^k \mid \mathbf{R} \models \varphi^D(a_1,\dots,a_k)\}$.

Because first-order logic over the reals admits quantifier elimination [43], we can rewrite $\varphi^D$ in a quantifier-free form from which we can conclude that $\varphi(D)$ is always a semialgebraic set. This is called the closure principle. The reals without multiplication also admit quantifier elimination, so in the same way, if $D$ is semilinear and $\varphi$ is in FO+Lin, then $\varphi(D)$ is also semilinear. Thus, there is also a closure principle for FO+Lin provided we work with semilinear databases. For more information on FO+Poly and FO+Lin queries, we refer the reader to the literature [34].

**3. Transitive closure logics.** Many interesting spatial database queries are not expressible in the first-order query languages FO+Poly and FO+Lin, e.g., the query that asks whether a given set is topologically connected or not. Therefore, it makes sense to consider extensions of FO+Poly (or FO+Lin) with recursion to obtain more powerful query languages. We study one of the most simple recursion constructs in this context, i.e., the transitive closure operator TC.

An immediate observation is that TC cannot be added "just like that" with its standard mathematical semantics without losing the important closure principle.

*Example* 3.1. The transitive closure of the semialgebraic set $\{(x,y) \in \mathbf{R}^2 \mid y = 2x\}$ equals $\{(x,y) \in \mathbf{R}^2 \mid \exists i \in \mathbf{N} : y = 2^i x\}$, which is not a semialgebraic set. $\diamond$

Therefore, we look at the TC operator quite naturally as a programming construct with a purely operational semantics. For example, we will look at the transitive closure example just mentioned simply as a nonterminating computation. Almost all programming languages allow for the expression of nonterminating computations, and it is part of the programmer's job to avoid writing such programs.

A formula in FO+Poly+TC is a formula built in the same way as an FO+Poly formula, but with the following extra formation rule: If $\psi(\vec{x},\vec{y})$ is a formula with $\vec{x}$, $\vec{y}$ $k$-tuples of variables, and $\vec{s}$, $\vec{t}$ are $k$-tuples of terms, then

$$[\mathrm{TC}_{\vec{x};\vec{y}}\,\psi](\vec{s},\vec{t}) \tag{3.1}$$

is also a formula which has as free variables those in $\vec{s}$ and $\vec{t}$. Since the only free variables in $\psi(\vec{x},\vec{y})$ are those in $\vec{x}$ and $\vec{y}$, we do not allow parameters in applications of the TC operator, as are allowed in general transitive closure logic studied in finite model theory [11]. With parameters, it is not so clear how to preserve the simple and elegant operational semantics we define next.

The semantics of a subformula of the above form (3.1) evaluated on a database $D$ is defined in the following operational manner:

1. Evaluate, recursively, $\psi(D)$.
2. Start computing the following iterative sequence of $2k$-ary relations:

$$\begin{aligned} X_0 &:= \psi(D), \\ X_{i+1} &:= X_i \cup \{(\vec{x}, \vec{y}) \in \mathbf{R}^{2k} \mid \exists \vec{z}\,(X_i(\vec{x}, \vec{z}) \wedge X_0(\vec{z}, \vec{y}))\}. \end{aligned}$$

   Stop as soon as an $i$ has been found such that $X_i = X_{i+1}$.
3. The semantics of $[\mathrm{TC}_{\vec{x};\vec{y}}\,\psi](\vec{s}, \vec{t})$ is now defined as the $2k$-ary relation $X_i$.

Since every step in the above algorithm, including the test for $X_i = X_{i+1}$, is expressible in FO+Poly, every step is effective and the only reason why the evaluation may not be effective is that the computation does not terminate. In that case the semantics of the formula (3.1) (and any other formula in which it occurs as subformula) is undefined.

The language FO+Lin+TC consists of all FO+Poly+TC formulas that do not use multiplication.

*Example* 3.2. Let $S$ be a relation name of arity $n$. Consider the following FO+Poly+TC formula:

$$\texttt{connected} \equiv \forall \vec{s} \forall \vec{t}\,\big(\big(S(\vec{s}) \wedge S(\vec{t})\big) \rightarrow [\mathrm{TC}_{\vec{x};\vec{y}}\,\texttt{lineconn}](\vec{s}, \vec{t})\big),$$

where $\texttt{lineconn}(\vec{x}, \vec{y})$ is the formula

$$\forall \lambda\big(0 \leq \lambda \leq 1 \wedge \forall \vec{t}(\vec{t} = \lambda \vec{x} + (1-\lambda)\vec{y} \rightarrow S(\vec{t}))\big).$$

In section 6.5, we will prove that the TC-subformula in connected terminates on all linear constraint databases over $\mathcal{S}$. Note that a pair of points $(\vec{p}, \vec{q})$ belongs to the TC of $\texttt{lineconn}(D)$ (with $D$ semilinear) if and only if $\vec{p}$ and $\vec{q}$ belong to the same connected component of $S^D$. Hence, connected effectively expresses connectivity of semilinear sets. $\diamond$

We will sometimes want to be able to specify an explicit termination condition on transitive closure computations. To this end, we introduce the language FO+Poly+TCS.

Formulas in FO+Poly+TCS are again built in the same way as in FO+Poly but with the following extra formation rule: If $\psi(\vec{x}, \vec{y})$ is a formula with $\vec{x}$, $\vec{y}$ $k$-tuples of variables; $\sigma$ is an FO+Poly sentence (formula without free variables) over the schema $\mathcal{S}$ expanded with a special $2k$-ary relation name $X$; and $\vec{s}$, $\vec{t}$ are $k$-tuples of terms, then

$$[\mathrm{TC}_{\vec{x};\vec{y}}\,\psi \mid \sigma](\vec{s}, \vec{t}) \tag{3.2}$$

is also a formula which has as free variables those in $\vec{s}$ and $\vec{t}$. We call $\sigma$ the *stop condition* of this formula.

The semantics of a subformula of the above form (3.2) evaluated on databases $D$ is defined in the same manner as in the case without stop condition, but now we stop not only in case an $i$ is found such that $X_i = X_{i+1}$, but also in case an $i$ is found such that $(D, X_i) \models \sigma$, whichever case occurs first.

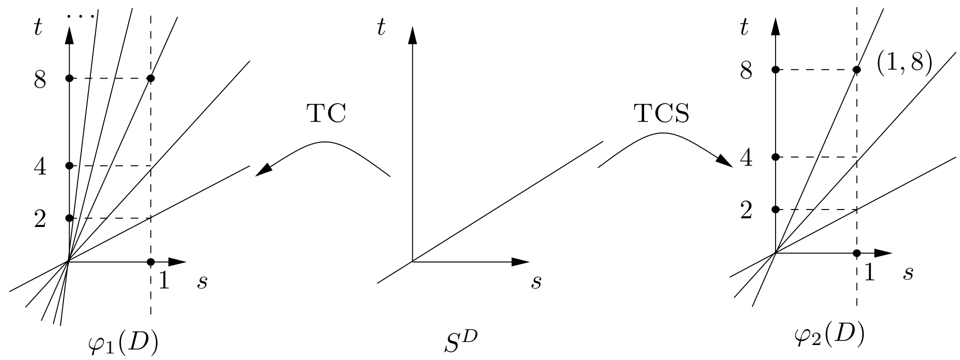


FIG. 3.1. *Illustration of the difference between transitive closure without stop condition (left) and with stop condition (right).*

*Example* 3.3. Let $S$ be a relation name of arity $n$ in $\mathcal{S}$, and consider the FO+ POLY+TCS formula

$$\varphi_1(s,t) \equiv [\mathrm{TC}_{x;y}\, S](s,t) \tag{3.3}$$

and the formula

$$\varphi_2(s,t) \equiv [\mathrm{TC}_{x;y}\, S \mid X(1,8)](s,t). \tag{3.4}$$

On the database $D$ over $S$, where $S^D = \{(x,y) \in \mathbf{R}^2 \mid y = 2x\}$, the evaluation of formula (3.3) does not terminate, but formula (3.4) evaluates in three iterations to $\{(s,t) \in \mathbf{R}^2 \mid t = 2s \vee t = 4s \vee t = 6s \vee t = 8s\}$. An illustration is given in Figure 3.1. $\diamond$

The language FO+LIN+TCS consists of all FO+POLY+TCS formulas that do not use multiplication.

An alternative way of controlling the computation of the transitive closure is provided by Kreutzer [29]. He allows a parametrized transitive closure operator in which the computation of the transitive closure can be restricted to certain paths (after specifying certain starting points).

It can be easily seen that any formula in FO+LIN+TC or FO+POLY+TC can be expressed by an equivalent formula in the corresponding logics of Kreutzer (see Geerts and Kuijpers [17]). Moreover, the transitive closure logic FO+LIN+KTC (the "K" stands for "Kreutzer") is computationally complete on $\mathbf{Z}$-linear constraint databases [29]. As we will see in the next section, the same completeness result holds for FO+LIN+TCS. Hence, FO+LIN+KTC and FO+LIN+TCS are equally expressive on $\mathbf{Z}$-linear constraint databases. Despite this similarity, the way in which queries are expressed in each language is quite different. Indeed, FO+LIN+KTC has an "a priori" character because *starting* points have to be properly selected in order to obtain terminating formula. In FO+LIN+TCS, termination is forced by the *stop* conditions, which are of an "a posteriori" character.

We point out that termination properties of these logics on general polynomial constraint databases have already been studied [17]. However, a complete comparison of these logics on polynomial constraint databases is left open.

**4. Expressivity results.** In this section, we show a general result on the expressive power of FO+LIN+TCS. More specifically, we prove that FO+LIN+TCS is computationally complete on $\mathbf{Z}$-linear constraint databases (Theorem 4.4). The proof consists of three steps. In the first step, we show that any computable function on the natural numbers can be simulated in FO+LIN+TCS (Lemma 4.1). In the second step, we show that there exists an encoding of **Z**-linear constraint databases by finite sets of rational numbers, and show that both the encoding and the corresponding decoding are expressible in FO+LIN+TCS (Lemmas 4.2 and 4.3). This implies that FO+LIN+TCS is computationally complete on **Z**-linear constraint databases.

For polynomial constraint databases, we show that FO+POLY+TCS is computationally complete for Boolean topological queries. This follows from the completeness on **Z**-linear constraint databases and the existence of an FO+POLY+TC query that, given any polynomial constraint database as input, returns a **Z**-linear constraint database which is topologically equivalent to the input. In this section we show that this "linearization query" is not expressible in FO+POLY. The FO+POLY+TC construction of the linearization query will be presented in section 6 (following preparations in section 5.1).

**4.1. Recursive functions on the natural numbers.** We first show that FO+LIN+TCS is computationally complete on the set of natural numbers **N**.

LEMMA 4.1. *For every partial computable function $f : \mathbf{N}^k \to \mathbf{N}$, there exists a formula $\varphi_f(y)$ in* FO+LIN+TCS *over the schema $\mathcal{S} = \{S\}$, with $S$ a $k$-ary relation, such that for any database $D$ over $\mathcal{S}$ with $S^D = \{(n_1, \dots, n_k)\}$, we have that $\varphi_f(D)$ is defined if and only if $f(n_1, \dots, n_k)$ is defined, and in this case $\varphi_f(D) = \{f(n_1, \dots, n_k)\}$.*

*Proof.* We show this by simulating the run of a nondeterministic $p$-counter machine $M_f$ which computes $f$. Here $M_f = (Q, \delta, q_0, q_f)$, where $Q$ is a finite set of internal states, $q_0 \in Q$ is the initial state, and $q_f \in Q$ is the final (halting) state. The set $\delta$ contains quadruples of the form $[q, i, s, q'] \in Q \times \{1, \dots, p\} \times \{Z, P\} \times Q$ or $[q, i, d, q'] \in Q \times \{1, \dots, p\} \times \{-, +\} \times Q$. The quadruple $[q, i, s, q']$ means that if $M_f$ is in state $q$ and the $i$th counter is equal to zero (when $s = Z$) or positive (when $s = P$), then change the state into $q'$. The quadruple $[q, i, d, q']$ means that if $M_f$ is in state $q$, then increase the $i$th counter by one (when $d = +$), or decrease the $i$th counter by one (when $d = -$), and change the state into $q'$. We assume that $Q = \{0, 1, \dots, m-1, m\}$, $q_0 = 0$, and $q_f = m$. Moreover, we assume that $p \geqslant k$ and that the initial configuration of $M_f$ when computing $f(n_1, \dots, n_k)$ has $n_1, \dots, n_k$ as the values of the first $k$ counters. When a halting state is reached, we assume that the first counter contains $f(n_1, \dots, n_k)$.

We define the first-order formula $\Psi_{\text{step}}(q, n_1, \dots, n_p, q', n'_1, \dots, n'_p)$, which describes a single step in a run of $M_f$. The formula $\Psi_{\text{step}}$ is the disjunction of the following formulas for $[q, i, s, q']$ and $[q, i, d, q']$ in $\delta$:

$$\Psi_{[q,i,Z,q']} \equiv Q(q) \wedge Q(q') \wedge n'_i = n_i = 0 \wedge \bigwedge_{j \in \{1, \dots, i-1, i+1, \dots, p\}} n_j = n'_j,$$

$$\Psi_{[q,i,P,q']} \equiv Q(q) \wedge Q(q') \wedge n'_i = n_i > 0 \wedge \bigwedge_{j \in \{1, \dots, i-1, i+1, \dots, p\}} n_j = n'_j,$$

$$\Psi_{[q,i,+,q']} \equiv Q(q) \wedge Q(q') \wedge n'_i = n_i + 1 \wedge \bigwedge_{j \in \{1, \dots, i-1, i+1, \dots, p\}} n_j = n'_j,$$

$$\Psi_{[q,i,-,q']} \equiv Q(q) \wedge Q(q') \wedge n'_i = n_i - 1 \wedge \bigwedge_{j \in \{1, \dots, i-1, i+1, \dots, p\}} n_j = n'_j.$$

We use the stop condition $\sigma$, which checks whether the final state has been reached starting from the initial state:

$$\sigma \equiv \exists y_1 \cdots \exists y_p \exists n_1 \cdots \exists n_k \big(S(n_1, \ldots, n_k) \wedge X(0, n_1, \ldots, n_k, \vec{0}_{p-k}, m, y_1, \ldots, y_p)\big).$$

Here, $\vec{0}_\ell$ denotes the $\ell$-tuple $(0, \ldots, 0)$.

The desired formula $\varphi_f(y)$ extracts $f(n_1, \ldots, n_k)$ from the first counter (represented by the variable $y$) when the stop condition is satisfied:

$$\begin{aligned}&\exists y_2 \cdots \exists y_p \exists n_1 \cdots \exists n_k (S(n_1, \ldots, n_k)\\ &\quad \wedge [\mathrm{TC}_{q,\vec{n};q',\vec{n}'}\, \Psi_{\text{step}} \mid \sigma](0, n_1, \ldots, n_k, \vec{0}_{p-k}, m, y, y_2, \ldots, y_p)). \qquad \square\end{aligned}$$

**4.2. Finite representation of Z-linear constraint databases.**

LEMMA 4.2. *There exists an encoding of* **Z***-linear constraint databases into finite relational databases over the rationals, and a corresponding decoding, which are both expressible in* FO+LIN+TCS.

*Proof.* It was shown by Vandeurzen [46] and Vandeurzen, Gyssens, and Van Gucht [48] that any **Z**-linear set in $\mathbf{R}^n$ has a finite geometric representation by means of a finite set over **Q** consisting of $(n+1)^2$-ary tuples. Basically, this geometric representation contains the projective coordinates[1] of a complete triangulation of the **Z**-linear set. Moreover, this representation can be expressed in FO+POLY. Vandeurzen [46] and Vandeurzen, Gyssens, and Van Gucht [48] actually show that this representation can be expressed in an extension of FO+LIN with some limited amount of multiplicative power. Also, the corresponding decoding, which computes the **Z**-linear constraint database given its finite geometric representation, can be expressed in this logic.

Hence, the lemma follows if we can show that FO+LIN+TCS can perform this limited amount of multiplication.

More specifically, we have to be able to express the multiplication of rationals $q_i$ from a finite set $S = \{q_1, \ldots, q_m\}$ with a real number $x$, i.e., $q_i x$ for $i = 1, \ldots, m$. First, we express how integers $n_i$ and $d_i$ can be computed in FO+LIN+TCS such that $q_i = \frac{n_i}{d_i}$ for $i = 1, \ldots, m$.

We assume that all rational numbers in the set $S$ are positive. The case of all negative rational numbers is completely analogous. If both positive and negative rational numbers occur in the set, we separate the positive from the negative and treat both sets separately.

Consider the following enumeration *enum* of pairs of natural numbers: *enum* is a mapping from $\mathbf{N} \times \mathbf{N}$ to $\mathbf{N} \times \mathbf{N}$ defined by

$$\mathit{enum} : (i, j) \mapsto \begin{cases} (i+1, j-1) & \text{if } j > 0, \\ (0, i+1) & \text{if } j = 0. \end{cases}$$

For every pair $(p, q) \in \mathbf{N} \times \mathbf{N}$, there clearly exists $k \in \mathbf{N}$ such that $\mathit{enum}^k(0,0) = (p, q)$. We shall interpret $(p, q)$ as the rational number $\frac{p}{q}$ in case $q \neq 0$ and as 0 otherwise.

Given a rational number $q$ and two natural numbers $n$ and $d$, we can test in FO+LIN+TCS whether $q = \frac{n}{d}$. This test can be performed as follows. Let $\mathit{frac} : \mathbf{R}^3 \to \mathbf{R}^3$ be the mapping defined as

$$\mathit{frac} : (q, j, v) \mapsto (q, j-1, v+q).$$

[1] Projective coordinates are used to deal with unbounded databases and the unbounded simplices in their triangulation.

Then for given $q \in \mathbf{Q}$ and $n, d \in \mathbf{N}$, we have that $q = \frac{n}{d}$ if and only if $\mathit{frac}^d(q, d, 0) = (q, 0, n)$.

To find the numerator and denominator of a rational number $q$, we will enumerate all pairs of natural numbers $(n, d) = \mathit{enum}^k(0, 0)$, $k = 0, 1, \ldots$ and test for each pair whether $\mathit{frac}^d(q, d, 0) = (q, 0, n)$. For this, we combine *enum* and *frac* into a partial mapping $\mathit{tryall} : \mathbf{R}^5 \to \mathbf{R}^5$ defined as

$$(q, i, j, u, v) \mapsto \begin{cases} (q, i, j, u', v') & \text{with } (q, u', v') = \mathit{frac}(q, u, v) \quad \text{if } u \geqslant 1, \\ (q, i', j', j', 0) & \text{with } (i', j') = \mathit{enum}(i, j) \quad \text{if } u = 0. \end{cases}$$

We claim that $q = \frac{n}{d}$ for $n, d \in \mathbf{N}$ if and only if $\mathit{tryall}^k(q, 0, 0, 0, 0) = (q, n, d, 0, n)$. Indeed, starting from $(q, 0, 0, 0, 0)$ the iterates of *tryall* behave as follows. Suppose we are at the $k$th iterate. If the third coordinate of $\mathit{tryall}^k(q, 0, 0, 0, 0)$ is zero, a new pair of natural numbers is generated (using the *enum* mapping). Assume that $\mathit{tryall}^{k+1}(q, 0, 0, 0, 0) = (q, i, j, j, 0)$ and suppose that $j > 0$ (otherwise we jump to a new pair of natural numbers immediately). Then, using the $frac$ mapping, we end up after $j$ more iterations at $\mathit{tryall}^{k+j+1}(q, 0, 0, 0, 0) = \mathit{tryall}^j(q, i, j, j, 0) = (q, i, j, 0, jq)$ ($frac$ reduces the fourth coordinate by one in each iteration). Note that if $i = jq$, then we have found a numerator $i$ and denominator $j$ of $q$. In any case, we move on to $\mathit{tryall}^{k+j+2}(q, 0, 0, 0, 0) = (q, i', j', j', 0)$, where $(i', j')$ is the next pair of natural numbers, and the above process starts again. In this way, the iterates of *tryall* visit every pair of natural numbers starting from $(q, 0, 0, 0, 0)$; between two consecutive pairs, it is checked whether the first pair is a numerator/denominator pair for $q$. The mapping *tryall* can clearly be expressed by an FO+Lin formula,

$$\psi_{\mathit{tryall}}(q, i, j, u, v, q', i', j', u', v'),$$

expressing that $\mathit{tryall}(q, i, j, u, v) = (q', i', j', u', v')$.

Let $\Psi(q, i, j, u, v, q', i', j', u', v')$ be the formula

$$q \geqslant 0 \land i \geqslant 0 \land j \geqslant 0 \land i' \geqslant 0 \land j' \geqslant 0 \land u \geqslant 0 \land q = q' \\ \land \psi_{\mathit{tryall}}(q, i, j, u, v, q', i', j', u', v').$$

Given a finite set of rational numbers $S = \{q_1, \ldots, q_m\}$, we obtain a denominator and numerator for all these numbers by taking the transitive closure

$$[\mathrm{TC}_{q,i,j,u,v;q',i',j',u',v'} \, \Psi \mid \sigma](\vec{s}, \vec{t}), \tag{4.1}$$

where $\vec{s}$ and $\vec{t}$ are 5-tuples of variables, and where

$$\sigma \equiv \forall q(S(q) \to \exists n \exists d X(q, 0, 0, 0, 0, q, n, d, 0, n)).$$

This condition stops the computation of the transitive closure of $\Psi$ when, for each rational number $q$ in $S$, there exists a $k$ such that $\mathit{tryall}^k(q, 0, 0, 0, 0) = (q, n, d, 0, n)$, or in other words, when a pair of natural numbers $(n, d)$ has been encountered such that $q = \frac{n}{d}$. If multiple pairs $(n, d)$ represent the same rational number in $S$, we select the pair with the smallest value of $n$. Thus, we obtain for each $q \in S$ a unique denominator and numerator.

We are now ready to show how to express the multiplication of rational numbers from a finite set $S$ with a real number. By what we just showed, we may assume that the rational numbers are represented as numerator/denominator pairs; i.e., we may assume that $S = \{(n_1, d_1), \ldots, (n_m, d_m)\}$.

Let max be the largest natural number occurring in $S$. We first compute any multiplication of the form $rn$ with $r \in \mathbf{R}$ and $n \in \{0, 1, \ldots, \max\}$.

For this, we define the following formula $\mathtt{natmult}(x, y, z, x', y', z')$:

$$\begin{aligned} & x = x' \wedge y' = y - 1 \wedge z' \\ & \quad = z + x \wedge \exists \max(\exists n(S(\max, n) \vee S(n, \max)) \\ & \qquad\qquad\qquad \wedge \forall n \forall d(S(n, d) \rightarrow n \leqslant \max \wedge d \leqslant \max) \wedge 0 \leqslant y \wedge y \leqslant \max). \end{aligned}$$

Then the formula

$$\mathtt{mult}(a, b, c) \equiv [\mathrm{TC}_{x,y,z;x',y',z'}\, \mathtt{natmult}](a, b, 0, a, 0, c)$$

holds if and only if $ab = c$ for $a \in \mathbf{R}$, $b \in \mathbf{N}$ and $b \leqslant \max$. In this way, we can retrieve any multiple up to max of any real number.

Finally, we define $\mathtt{ratmult}(z, y, n, d) \equiv \exists u(\mathtt{mult}(z, d, u) \wedge \mathtt{mult}(y, n, u))$. This formula holds for $(z, y, n, d)$ if and only if $z = yq$ with $z, y \in \mathbf{R}$, and $q = \frac{n}{d}$ with $(n, d) \in S$. $\square$

**4.3. Natural number representation.**

LEMMA 4.3. *There exists an encoding of finite relations over the rational numbers into single natural numbers, and a corresponding decoding, which are both expressible in* FO+LIN+TCS.

*Proof.* We assume that the relation to be encoded involves positive rational numbers only. The general case can be dealt with by splitting the relation into "sign-homogeneous" pieces, dealing with each piece separately and encoding the tuple of natural numbers obtained for each piece again into a single natural number.

In the proof of Lemma 4.2, we have seen that in FO+LIN+TCS we can go from rational numbers (out of a finite set) to denominator/numerator pairs and back. Hence, we can actually assume that the relation to be encoded involves positive natural numbers only.

We will encode this in two steps. In the first step, we encode a finite relation over $\mathbf{N}$ into a finite subset of $\mathbf{N}$. In the second step, we encode a finite subset of $\mathbf{N}$ into a single natural number. Since queries can be composed, we can treat these two encoding steps (and their corresponding decoding steps) separately.

*Encoding, first step.* A finite $k$-ary relation $s$ over $\mathbf{N}$ can be encoded into a finite subset $\mathrm{Enc}_1(s)$ of $\mathbf{N}$:

$$\mathrm{Enc}_1(s) := \left\{ \prod_{i=1}^{k} p_i^{n_i} \mid (n_1, \ldots, n_k) \in s \right\}.$$

Here, $p_i$ denotes the $i$th prime number.

Now let $S$ be a $k$-ary relation name. We will construct an FO+LIN+TC formula $\epsilon_1$ over $\{S\}$ such that for any database $D$ where $S^D$ is finite and involves natural numbers only, $\epsilon_1(D) = \mathrm{Enc}_1(S^D)$. For notational simplicity, we give the construction only for the case $k = 2$; the general case is analogous.

Consider the following formula $\psi(x_1, x_2, y, x'_1, x'_2, y')$:

$$\begin{aligned} & \exists u_1 \exists u_2 (S(u_1, u_2) \wedge x_1 \leq u_1 \wedge x_2 \leq u_2) \\ & \qquad\qquad \wedge ((x_1 > 0 \wedge x'_1 = x_1 - 1 \wedge x'_2 = x_2 \wedge y' = 2y) \\ & \qquad\qquad \vee (x_1 = 0 \wedge x_2 > 0 \wedge x'_1 = x_1 \wedge x'_2 = x_2 - 1 \wedge y' = 3y)). \end{aligned}$$

Here, $y' = 2y$ is an abbreviation for $y' = y + y$, and similarly for $y' = 3y$; note that 2 and 3 are the first two prime numbers.

We now define the mapping $p(x_1, x_2, y) = (x'_1, x'_2, y)$ if and only if $\psi(x_1, x_2, y, x'_1, x'_2, y')$. As long as $k \leqslant x_1$, we have that $p^k(x_1, x_2, y) = (x_1 - k, x_2, y2^k)$. As soon as $k > x_1$, $p^k(x_1, x_2, y)$ is undefined. If $k = x_1$, we can compute further iterates and have that $p^{k+\ell}(x_1, x_2, y) = p^\ell(0, x_2 - \ell, y2^{x_1}3^\ell)$ as long as $\ell \leqslant x_2$. Iterates again become undefined in case $\ell > x_2$. Finally, if $\ell = x_2$ then $p^{k+\ell}(x_1, x_2, y) = (0, 0, y2^{x_1}3^{x_2})$, and we obtain the encoding for $(x_1, x_2)$ for $y = 1$. No further iterates are defined starting from $(0, 0, y')$.

We will compute the iterates of $p$ using transitive closure and check for each $(n_1, n_2)$ whether there exists a $k$ such that $p^k(n_1, n_2, 1) = (0, 0, y)$. More specifically, the desired formula $\epsilon_1(y)$ is equal to

$$\exists n_1 \exists n_2 \big(S(n_1, n_2) \wedge [\mathrm{TC}_{x_1,x_2,y;x'_1,x'_2,y'}\, \psi](n_1, n_2, 1, 0, 0, y)\big).$$

The discussion above shows that this formula gives the correct answer. The condition $S(u_1, u_2) \wedge x_1 \leq u_1 \wedge x_2 \leq u_2$ in $\psi$ bounds the values of $x_1$ and $x_2$, and hence ensures that the transitive closure computation always terminates.

*Decoding, first step.* Let $S$ be a unary relation name. We will construct an FO+LIN+TC formula $\delta_1$ over $\{S\}$ such that for any database $D$ where $S^D$ equals $\mathrm{Enc}_1(r)$ for some $r$, we have $\delta_1(D) = r$. As above, we give the construction only for the case $k = 2$.

Consider now the following formula $\psi(x_1, x_2, y, x'_1, x'_2, y')$:

$$\begin{aligned} x_1 \geq 0 \wedge x_2 \geq 0 \wedge y \geq 1 \wedge ((x'_1 = x_1 + 1 \wedge x'_2 = x_2 \wedge y' = 2y) \\ \vee (x'_1 = x_1 \wedge x'_2 = x_2 + 1 \wedge y' = 3y)) \wedge \exists u(S(u) \wedge y' \leq u). \end{aligned}$$

An analysis similar to that for $\mathrm{Enc}_1$ shows that when we define $q(x_1, x_2, y) = (x'_1, x'_2, y')$ if and only if $\psi(x_1, x_2, y, x'_1, x'_2, y')$, the iterates of $q$ satisfy $q^k(0, 0, 1) = (n_1, n_2, u)$ if and only if $u = 2^{n_1}3^{n_2}$.

Then the desired formula $\delta_1(n_1, n_2)$ is

$$\exists u(S(u) \wedge [\mathrm{TC}_{x_1,x_2,y;x'_1,x'_2,y'}\, \psi](0, 0, 1, n_1, n_2, u)).$$

The condition $\exists u(S(u) \wedge y' \leq u)$ in $\psi$ bounds the value of $y'$, and hence ensures the termination of the computation of the transitive closure.

*Encoding, second step.* A finite ordered subset $s = \{n_1, \dots, n_\ell\}$ of $\mathbf{N}$ can be encoded into a single natural number $\mathrm{Enc}_2(s) := \prod_{i=1}^{\ell} p_i^{n_i}$.

Let $S$ be a unary relation name. We will construct an FO+LIN+TCS formula $\epsilon_2$ over $\{S\}$ such that for any database $D$ where $S^D$ is a finite subset of $\mathbf{N}$, we have $\epsilon_2(D) = \{\mathrm{Enc}_2(S^D)\}$.

We will use the following auxiliary FO+LIN+TCS formulas; we will explain later how to get them (except for `min` and `max`, which are easy to get).

- Formulas `card`, `min`, and `max` over $\{S\}$, with the property that for any $D$ where $S^D$ is finite of cardinality $\ell$, $\texttt{card}(D) = \{\ell\}$; $\texttt{min}(D) = \{\min S^D\}$; and $\texttt{max}(D) = \{\max S^D\}$.
- Formulas `prime`, `mult`, and `nat`, over $\{M\}$, with $M$ a unary relation name, with the property that for any $D$ where $M^D = \{m\}$ is a natural number singleton,
  - $\texttt{prime}(D) = \{p_m\}$;
  - $\texttt{mult}(D) = \{(x, y, z) \in \mathbf{R}^3 \mid xy = z \text{ and } y \in \mathbf{N} \text{ and } y \leq m\}$; and
  - $\texttt{nat}(D) = \{0, 1, 2, \dots, m\}$.

- Formula $\mathtt{pow}$ over $\{M, M_2\}$, with $M$, $M_2$ unary relation names, with the property that for any $D$ where $M^D = \{m\}$ and $M_2^D = \{m_2\}$ are natural number singletons, $\mathtt{pow}(D) = \{(x, y, z) \in \mathbf{R}^3 \mid x^y = z \text{ and } x \in \mathbf{N} \text{ and } x \leq m \text{ and } y \in \mathbf{N} \text{ and } y \leq m_2\}$.

Using composition, we also obtain that

- $\mathtt{maxprime} \equiv \mathtt{prime}(\mathtt{card})$, defining $p_\ell$ where $\ell$ is the cardinality of $S$;
- $\mathtt{nat}' \equiv \mathtt{nat}(\mathtt{maxprime})$, defining $\{0, 1, 2, \dots, p_\ell\}$; and
- $\mathtt{pow}' \equiv \mathtt{pow}(\mathtt{maxprime}, \mathtt{max})$, defining exponentiation of natural numbers $\leq p_\ell$ by natural numbers $\leq \max S$.

We furthermore construct the following formulas:

- $\mathtt{mult}'$, obtained from $\mathtt{mult}$ by replacing each occurrence of a subformula $M(u)$ with
$$\exists p_\ell \exists m(\mathtt{maxprime}(p_\ell) \wedge \mathtt{max}(m) \wedge \mathtt{pow}'(p_\ell, m, u)).$$
This formula defines multiplication by natural numbers $\leq p_\ell^{\max S}$.
- $\mathtt{isprime}(p)$, which defines $\{p_1, p_2, \dots, p_\ell\}$:
$$\mathtt{nat}'(p) \wedge p > 1 \wedge \neg\exists u \exists v(\mathtt{nat}'(u) \wedge \mathtt{nat}'(v) \wedge u > 1 \wedge v > 1 \wedge \mathtt{mult}'(u, v, p)).$$
- $\mathtt{succ}(x, x')$, which specifies the next element after $x$ in $S$ (or $\max(S)+1$) and is given by the formula
$$(\neg\mathtt{max}(x) \wedge S(x') \wedge x < x' \\ \wedge \neg\exists x''(S(x'') \wedge x < x'' < x')) \vee (\mathtt{max}(x) \wedge x' = x + 1).$$
- $\mathtt{next}(p, p')$, which specifies the next prime number greater than $p$ and smaller than or equal to $p_\ell$ (or $p_\ell + 1$) and is given by the formula
$$(\neg\mathtt{maxprime}(p) \wedge \mathtt{isprime}(p') \wedge p < p' \\ \wedge \neg\exists p''(\mathtt{isprime}(p'') \wedge p < p'' < p')) \vee (\mathtt{maxprime}(p) \wedge p' = p + 1).$$

We need to compute the product $\prod_{i=1}^{\ell} p_i^{n_i}$. Consider now the following formula $\psi(x, p, y, x', p', y')$:
$$S(x) \wedge \mathtt{succ}(x, x') \wedge \mathtt{next}(p, p') \wedge \exists y''(\mathtt{pow}'(p, x, y'') \wedge \mathtt{mult}'(y, y'', y')).$$

Note that the variables $y$ and $y'$ are related by $y' = p^x y$. In order to find the desired product, we have to compute the transitive closure of $\psi$ and check which $y'$-value is in the transitive closure with $(n_1, 2, 1)$ and $(m+1, p_\ell+1, y')$. More explicitly, the desired formula $\epsilon_2(n)$ is
$$\exists n_1 \exists m \exists p_\ell(\mathtt{min}(n_1) \wedge \mathtt{max}(m) \wedge \mathtt{maxprime}(p_\ell) \\ \wedge [\mathrm{TC}_{x,p,y;x',p',y'}\, \psi](n_1, 2, 1, m+1, p_\ell+1, n)).$$

It remains to show how the auxiliary formulas can be constructed. Formula $\mathtt{card}(\ell)$ can be written as
$$\exists n_1 \exists m(\mathtt{min}(n_1) \wedge \mathtt{max}(m) \\ \wedge [\mathrm{TC}_{x,c;x',c'}\, S(x) \wedge \mathtt{succ}(x, x') \wedge c' = c + 1](n_1, 0, m+1, \ell)),$$

where $\mathtt{succ}(x, x')$ is as above.

From the computationally completeness of FO+LIN+TCS (Lemma 4.1), we derive directly the formula $\mathtt{prime}$.

For formula $\mathtt{mult}$, consider the following formula $\psi(x, y, u, x', y', u')$:

$$x' = x \wedge y' = y - 1 \wedge u' = u + x \wedge 0 < y \wedge \exists m(M(m) \wedge y \leq m).$$

Then $\mathtt{mult}(x, y, z)$ is $[\mathrm{TC}_{x,y,u;x',y',u'}\ \psi](x, y, 0, x, 0, z)$.

Formula $\mathtt{nat}(n)$ can be written as

$$n = 0 \vee [\mathrm{TC}_{x;x'}\ (0 \leq x \wedge \exists m(M(m) \wedge x < m) \wedge x' = x + 1)](0, n).$$

Finally, for formula $\mathtt{pow}$, consider the following formula $\psi(x, u, v; x', u', v')$:

$$\begin{aligned}&\mathtt{nat}(x) \wedge \exists m(M(m) \wedge x < m) \wedge 0 \leq u \wedge \exists m_2(M_2(m_2) \wedge u < m_2) \wedge u' \\ &\qquad = u + 1 \wedge \mathtt{mult}(v, x, v').\end{aligned}$$

Then $\mathtt{pow}(x, y, z)$ is $(y = 0 \wedge z = 1) \vee [\mathrm{TC}_{x,u,v;x',u',v'}\ \psi](x, 0, 1, x, y, z)$.

*Decoding, second step.* Let $E$ be a unary relation name. We will construct an FO+LIN+TCS formula $\delta_2$ over $\{E\}$ such that for any database $D$ where $E^D$ is a singleton $\{e\}$ such that $e$ equals $Enc_2(s)$ for some $s$, we have $\delta_2(D) = s$.

By Lemma 4.1, we have formulas $\mathtt{highprime}$ and $\mathtt{highexp}$ over $\{E\}$ such that for any $D$ as above, we have $\mathtt{highprime}(D) = \{p_\ell\}$ and $\mathtt{highexp}(D) = \{m\}$, where $p_\ell$ is the highest prime factor of $e$, and $m$ is the highest exponent of a prime number in the prime factorization of $n$. Composing the formula $\mathtt{pow}$ of above with these two formulas, we obtain a formula defining exponentiation of natural numbers $\leq p_\ell$ by natural numbers $\leq m$, which we again denote by $\mathtt{pow}'$. Also, analogously to the way we constructed the formula $\mathtt{isprime}$ above, we obtain a formula defining $\{p_1, p_2, \ldots, p_\ell\}$, which we again denote by $\mathtt{isprime}$.

We need a formula $\mathtt{divisor}$ that finds all divisors of a natural number. First, consider the following formula $\psi(u, v, u', v')$:

$$0 \leq u \wedge \exists e(E(e) \wedge u \leq e) \wedge v \geq 1 \wedge v' = v \wedge u' = u - v$$

and let $\mathtt{divisor}(d)$ be the formula

$$\exists e(E(e) \wedge [\mathrm{TC}_{u,v;u',v'}\ \psi](e, d, 0, d)).$$

Then, the desired formula $\delta_2(n)$ is

$$\begin{aligned}&\exists p(\mathtt{isprime}(p) \wedge \exists d(\mathtt{pow}'(p, n, d) \wedge \mathtt{divisor}(d)) \\ &\qquad \wedge \neg\exists n' \exists d'(\mathtt{pow}'(p, n', d') \wedge \mathtt{divisor}(d') \wedge n' > n)). \qquad \square\end{aligned}$$

**4.4. Completeness result for Z-linear constraint databases.**

THEOREM 4.4. *For every partially computable query $Q$ on* **Z***-linear constraint databases, there exists an* FO+LIN+TCS *formula $\varphi$ such that for each database $D$, $\varphi(D)$ is defined if and only if $Q(D)$ is, and in this case $\varphi(D)$ and $Q(D)$ are equal.*

*Proof.* The proof follows directly from the lemmas above, as is illustrated in the following diagram. Let $D$ be a **Z**-linear constraint database over a schema

$\mathcal{S} = \{S_1, \ldots, S_k\}$, and let $Q$ be an arbitrary partially computable query:

$$\begin{tikzcd}
D \arrow[rr, "Q"] \arrow[d, "(\text{Lemma 4.2})"'] & & Q(D) \\
\{S_{1,\text{fin}}, \ldots, S_{k,\text{fin}}\} \arrow[d, "(\text{Lemma 4.3})"'] & & S_{\text{fin}} \arrow[u, "(\text{Lemma 4.2})"'] \\
(n_1, \ldots, n_k) \in \mathbf{N}^k \arrow[rr, "f_Q", "(\text{Lemma 4.1})"'] & & n_{Q(D)} \in \mathbf{N} \arrow[u, "(\text{Lemma 4.3})"']
\end{tikzcd}$$

First, each $S_i^D$, $i = 1, \ldots, k$, is encoded in a finite relations $S_{i,\text{fin}}$, which in its turn is encoded in a natural number $n_i$. In this way, a $k$-tuple $(n_1, \ldots, n_k)$ is obtained. Since $Q$ is computable, there exists a partial computable function $f_Q$ which implements $Q$ on these encodings. Let $n_Q(D)$ be the result of $f_Q$ on input $(n_1, \ldots, n_k)$. This integer is decoded into a finite relation $S_{\text{fin}}$ which in its turn is decoded into a $\mathbf{Z}$-linear constraint database $D'$. This database is then the result of the query $Q$ on the input database $D$, i.e., $D' = Q(D)$. ∎

**4.5. Implications for polynomial constraint databases.** For polynomial constraint databases, we cannot prove completeness and have to settle for less. Although finite representations of polynomial constraint databases exist, it is not known whether a finite encoding can be expressed in FO+POLY+TCS.

Let $A$ be a semialgebraic set in $\mathbf{R}^n$. An *algebraic linearization* of $A$ is an $\mathbf{A}$-linear set $\widehat{A}$ in $\mathbf{R}^n$ such that $A$ and $\widehat{A}$ are topologically equivalent. A *rational linearization* of $A$ is a $\mathbf{Z}$-linear set $\widehat{A}_{\text{rat}}$ in $\mathbf{R}^n$ such that $A$ and $\widehat{A}_{\text{rat}}$ are topologically equivalent.

For $\vec{x} \in \mathbf{R}^n$, we define $\|\vec{x}\| = \sqrt{x_1^2 + \cdots + x_n^2}$. A linearization approximates the set $A$ also from a metric point of view if the following condition is satisfied: for every point $\vec{p}$ in $A$, $\|\vec{p} - h(\vec{p})\| < \varepsilon$ for a fixed $\varepsilon > 0$, where $h$ is a homeomorphism of $\mathbf{R}^n$ such that $h(A) = \widehat{A}$. If this condition is satisfied for a (rational) linearization, we call this linearization a *(rational) $\varepsilon$-approximation* of the set $A$. We will denote rational and algebraic $\varepsilon$-approximations, respectively, by $\widehat{A}_{\text{rat},\varepsilon}$ and $\widehat{A}_\varepsilon$.

*Example* 4.1. Consider the planar semialgebraic set $A = \{(x, y) \in \mathbf{R}^2 \mid x^2 + y^2 = 2\}$. Let $\varepsilon = \frac{1}{2}$. In Figure 4.1, we have drawn an algebraic $\varepsilon$-approximation $\widehat{A}_\varepsilon = \{(x, y) \in \mathbf{R}^2 \mid \max\{|x|, |y|\} = \sqrt{2}\}$, a rational $\varepsilon$-approximation $\widehat{A}_{\text{rat},\varepsilon} = \{(x, y) \in \mathbf{R}^2 \mid \max\{|x|, |y|\} = 1\}$, and a linearization $\widehat{A}$ which is not an $\varepsilon$-approximation. ◇

Algebraic and rational linearizations exist for any semialgebraic set. This is no longer true for $\varepsilon$-approximations, where the existence is guaranteed only for bounded semialgebraic sets. Consider, e.g., the semialgebraic set $\{(x, y) \in \mathbf{R}^2 \mid y = x^2\}$. It is easy to see that this parabola cannot be approximated by a finite number of line segments, and hence has no $\varepsilon$-approximation for any $\varepsilon > 0$.

Let $\mathcal{S} = \{S\}$, with $S$ an $n$-ary relation name. We define for any polynomial constraint database $D$ over $\mathcal{S}$ an *algebraic (rational) linearization query* $Q_{\text{lin}}$ ($Q_{\text{rat}-\text{lin}}$) as a query such that $Q_{\text{rat}}(D)$ ($Q_{\text{rat-lin}(D)}$) is an algebraic (rational) linearization of $S^D$.

Similarly, for any $\varepsilon > 0$ and any polynomial constraint database $D$ over $\mathcal{S}$ such that $S^D$ is a bounded semialgebraic set, we define an *algebraic (rational) $\varepsilon$-approximation query* $Q_\varepsilon$ ($Q_{\text{rat},\varepsilon}$) as a query such that $Q_\varepsilon(D)$ ($Q_{\text{rat},\varepsilon}(D)$) is an algebraic (rational) $\varepsilon$-approximation of $S^D$.

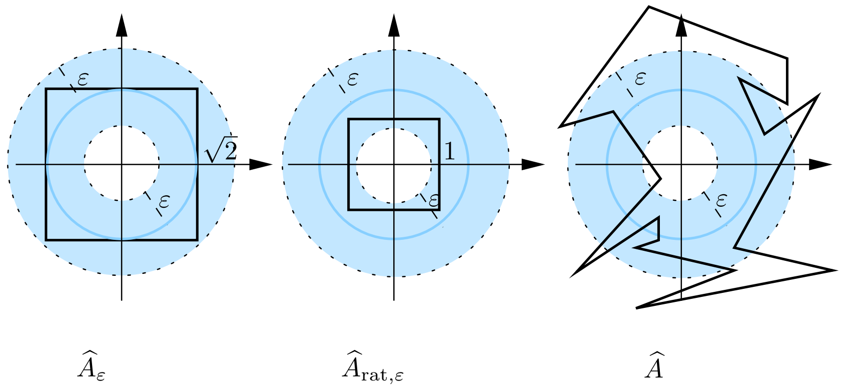


FIG. 4.1. *Let $A$ be the circle (grey). Left: an algebraic $\varepsilon$-approximation. Middle: a rational $\varepsilon$-approximation. Right: an algebraic linearization.*

It is an open question whether some algebraic or rational linearization query can be expressed in FO+POLY. With respect to the $\varepsilon$-approximation query, neither the algebraic nor the rational version can be expressed in FO+POLY.

PROPOSITION 4.5. *Let $\varepsilon > 0$ be a real number. No $\varepsilon$-approximation query is expressible in* FO+POLY.

*Proof.* Let $\mathcal{S} = \{S\}$, with $S$ a binary relation name. Let $D$ be a polynomial constraint database over $\mathcal{S}$. Consider the following FO+POLY formulas over $\mathcal{S}$:

- A formula `circle` such that for any database $D$ over $\mathcal{S}$, `circle(D)` is either the circle through the points of $S^D$, if $S^D$ consists of three noncollinear points, or $\mathtt{circle}(D) = \emptyset$. This formula is easily seen to be in FO+POLY.
- A formula `cornerpoints` such that for any database $D$ over $\mathcal{S}$, $\mathtt{cornerpoints}(D)$ is either the set of points in which $S^D$ is not locally a straight line, in the case when $S^D$ is semilinear, or $\mathtt{cornerpoints}(D) = \emptyset$, otherwise. By a result of Dumortier et al. [10], it is expressible in FO+POLY whether a semialgebraic set is semilinear. Hence, `cornerpoints` is expressible in FO+POLY.

Assume that the query $Q_\varepsilon$ (and similarly, $Q_{\mathrm{rat},\varepsilon}$) is expressible in FO+POLY. Let $\varepsilon$`-approx` be the formula which expresses $Q_\varepsilon$. Then the formula

$$\varphi \equiv \mathtt{cornerpoints}(\varepsilon\mathtt{-approx}(\mathtt{circle}))$$

is also in FO+POLY. However, the number of points in $\varphi(D)$, $|\varphi(D)|$ can be made arbitrarily large by choosing $D$ such that $S^D$ consist of three points far enough apart. This contradicts the dichotomy theorem of Benedikt and Libkin [4], which guarantees the existence of a polynomial $p_\varphi$ such that $|\varphi(D)| < p_\varphi(|S^D|) = p_\varphi(3)$ in the case when $|\varphi(D)|$ is finite. ∎

In contrast to the negative expressiveness result in Proposition 4.5, we will prove that all kinds of linearizations are expressible in FO+POLY+TC. Indeed, in section 6 we show that there exists

- an FO+POLY+TC expressible algebraic linearization query (Theorem 6.6);
- an FO+POLY+TC expressible rational linearization query (Theorem 6.9);
- an FO+POLY+TC expressible algebraic $\varepsilon$-approximation query (Theorem 6.7);
- an FO+POLY+TC expressible rational $\varepsilon$-approximation query (Theorem 6.10).

We shall denote the FO+POLY+TC formula, which expresses the rational linearization by `ratlin`. Let $Q$ be a partially computable Boolean topological query. Since $Q$ is partially computable, it is in particular partially computable on **Z**-linear constraint databases, and therefore by Theorem 4.4 is expressible on these databases by a formula $\varphi_Q$ in FO+LIN+TCS.

Because $Q$ is topological, $Q(D)$ is true if and only if $\varphi_Q(\texttt{ratlin}(D))$ is true. Hence, we have proven the following theorem.

THEOREM 4.6. *For every partially computable Boolean topological query $Q$ on polynomial constraint databases, there exists an* FO+POLY+TCS *formula $\varphi$ such that for each database $D$, $\varphi(D)$ is defined if and only if $Q(D)$ is defined, and in this case $\varphi(D)$ and $Q(D)$ are equal.*

**5. Geometrical properties of semialgebraic sets.** In this section, we discuss a number of topological properties of spatial databases that can be expressed in first-order logic. They are used in the construction of the linearization of polynomial constraint databases in the next section.

We will use the following notation. Let $A \subseteq \mathbf{R}^n$; the closure of $A$ is denoted by $\mathrm{cl}(A)$, and $\mathrm{int}(A)$ indicates the interior of $A$. We denote $\mathrm{cl}(A) - \mathrm{int}(A)$ (the boundary of $A$) by $\partial A$.

**5.1. The cone radius.** Let $A$ be a semialgebraic set in $\mathbf{R}^n$, and let $\vec{p}$ be a point in $\mathbf{R}^n$. We define the *cone with base $A$ and top $\vec{p}$* as the union of all closed line segments between $\vec{p}$ and points in $A$. Formally, this is the set $\{t\vec{b} + (1-t)\vec{p} \mid \vec{b} \in A, 0 \leq t \leq 1\}$ and we denote this set by $\mathrm{Cone}(A, \vec{p})$.

For a point $\vec{p} \in \mathbf{R}^n$, and $\varepsilon > 0$, we denote the closed ball centered at $\vec{p}$ with radius $\varepsilon$ by $B^n(\vec{p}, \varepsilon)$, and we denote the sphere centered at $\vec{p}$ with radius $\varepsilon$ by $S^{n-1}(\vec{p}, \varepsilon)$.

The local conic structure of semialgebraic sets characterizes the local topology of semialgebraic sets.

THEOREM 5.1 (local conic structure; see Theorem 9.3.6 of [6]). *Let $A$ be a semialgebraic set in $\mathbf{R}^n$ and $\vec{p}$ be a point of $\mathrm{cl}(A)$. Then there is a real number $\varepsilon > 0$ such that intersection $B^n(\vec{p}, \varepsilon) \cap A$ is homeomorphic to the set $\mathrm{Cone}(S^{n-1}(\vec{p}, \varepsilon) \cap A, \vec{p})$, in the case when $\vec{p} \in A$, and homeomorphic to $\mathrm{Cone}(S^{n-1}(\vec{p}, \varepsilon) \cap A, \vec{p}) - \{\vec{p}\}$ otherwise.*

Before we can state a "box" version of this theorem, we need the following definitions. Consider a $2n$-tuple $B = (a_1, b_1, \ldots, a_n, b_n) \in \mathbf{R}^{2n}$ with $a_i \leqslant b_i$ for each $i$. One can associate with each such tuple an $n$-ary relation $|B|$ in $\mathbf{R}^n$:

$$|B| := \{(x_1, \ldots, x_n) \in \mathbf{R}^n \mid (a_1 \leqslant x_1 \leqslant b_1) \wedge \cdots \wedge (a_n \leqslant x_n \leqslant b_n)\}.$$

We call $B$ a *box* in $\mathbf{R}^n$, and $|B|$ is the *geometric realization of $B$*. The *dimension* of a box is the number of pairs $(a_i, b_i)$ with $a_i \neq b_i$. The *diameter* of a box $B$, $\mathrm{diam}(B)$, equals $(\sum_{i=1}^n (b_i - a_i)^2)^{1/2}$. The *center* of $B$ is the point $((a_1+b_1)/2, \ldots, (a_n+b_n)/2)$.

THEOREM 5.2 (see [14]). *Let $A$ be a semialgebraic set in $\mathbf{R}^n$ and $\vec{p}$ a point of $\mathrm{cl}(A)$. Then there is a real number $\varepsilon > 0$ such that for any $n$-dimensional box $B$ in $\mathbf{R}^n$ such that*

1. $\vec{p} \in \mathrm{int}(|B|)$, *and*
2. $|B| \subseteq (p_1 - \varepsilon, p_1 + \varepsilon) \times \cdots \times (p_n - \varepsilon, p_n + \varepsilon)$,

*we have that the intersection $A \cap |B|$ is homeomorphic to the set Cone$(A \cap \partial|B|, \vec{p})$, in the case when $\vec{p} \in A$, and homeomorphic to the set Cone$(A \cap \partial|B|, \vec{p}) - \{\vec{p}\}$ otherwise.*

Any positive real number $\varepsilon$ as in Theorem 5.2 is called a *cone radius* of $A$ in $\vec{p}$ (see Figure 5.1).

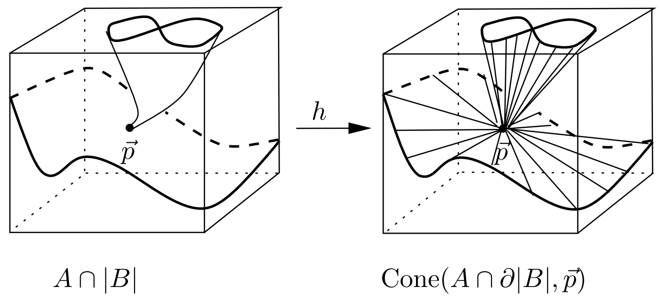


FIG. 5.1. *The local conic structure of semialgebraic sets.*

Let $\mathcal{S} = \{S\}$, with $S$ an $n$-ary relation name. We define the *cone radius query* $Q_{\mathrm{radius}}$ as a query which maps any polynomial constraint database $D$ over $\mathcal{S}$ to a set of pairs $(\vec{p}, r) \in \mathbf{R}^n \times \mathbf{R}$ such that for every $\vec{p} \in \mathrm{cl}(S^D)$ there exists at least one pair $(\vec{p}, r) \in Q_{\mathrm{radius}}(D)$, and for every $(\vec{p}, r) \in Q_{\mathrm{radius}}(D)$, $r$ is a cone radius of $S^D$ in $\vec{p}$.

THEOREM 5.3 (see [14]). *The cone radius query defined above is expressible in* FO+POLY.

The FO+POLY formula over $\mathcal{S}$, constructed in [14] and whose existence is referred to in Theorem 5.3, will be denoted by `radius`. The exact properties of this formula are not important (except for the fact that, for each point $\vec{p}$, it assigns an open interval $(0, r) \subset \mathbf{R}$ such that for each $r' \in (0, r)$, $r'$ is a cone radius) until the proof of Claim 6.1. There we have to go back to the construction of `radius` for the cone radius query as presented in [14].

As observed above, for each point $\vec{p}$,

$$\{r' \mid (\vec{p}, r') \in \mathtt{radius}(D)\} = (0, r).$$

Define $r_{\vec{p}}$ to be the cone radius $r/2$. Moreover, let `uniqueradius` be the FO+POLY formula over $\mathcal{S}$ such that for each point $\vec{p} \in \mathrm{cl}(S^D)$, $(\vec{p}, r_{\vec{p}})$ is in $\mathtt{uniqueradius}(D)$. Basically, `uniqueradius` assigns a unique cone radius to each point.

For a given semialgebraic set $A$ in $\mathbf{R}^n$, we now define the semialgebraic mapping[2] $\gamma_{\mathrm{cone},A}$ from $\mathrm{cl}(A)$ to $\mathbf{R}$ which maps each point $\vec{p} \in \mathrm{cl}(A)$ to the unique cone radius $r_{\vec{p}} \in \mathbf{R}$ given by $\mathtt{uniqueradius}(D)$, where $S^D = A$.

**5.2. The uniform cone radius decomposition.** Although every point of a semialgebraic set has a cone radius which is strictly greater than zero (Theorem 5.2), we are now interested in finding a *uniform cone radius* for a semialgebraic set. We define a uniform cone radius of a semialgebraic set $A \subseteq \mathbf{R}^n$ as a real number $\varepsilon_A > 0$ such that $\varepsilon_A$ is a cone radius of $A$ in all points of $A$. For any $X \subseteq A \subseteq \mathbf{R}^n$, we define a uniform cone radius of $X$ with respect to $A$ as a real number $\varepsilon > 0$ such that $\varepsilon$ is a cone radius of $A$ in all points of $X$.

A first observation is that a uniform cone radius of a semialgebraic set does not always exist.

*Example* 5.1. Consider the set shown in Figure 5.2. We have drawn the maximal cone radius around the points $\vec{p}_1$, $\vec{p}_2$, $\vec{p}_3$, $\vec{p}_4$, and $\vec{p}_5$. It is clear that the closer these points are to the point $\vec{p}$, the smaller their maximal cone radius is. Because we can make the maximal cone radius arbitrarily small by taking points very close to $\vec{p}$, we may conclude that the set shown in this figure has no uniform cone radius. $\diamond$

[2] A mapping is called semialgebraic if its graph is a semialgebraic set.

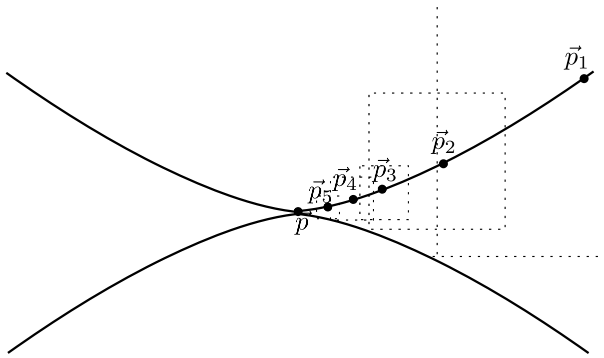


FIG. 5.2. *Example of a semialgebraic set which does not have a uniform cone radius.*

Let $A$ be a semialgebraic set in $\mathbf{R}^n$. We define the *$\varepsilon$-neighborhood* of $A$ as

$$A^\varepsilon := \{\vec{x} \in \mathbf{R}^n \mid \exists \vec{y}\,(\vec{y} \in A \wedge \|\vec{x} - \vec{y}\| < \varepsilon)\}.$$

We will frequently use the following notation. Let $U_0, \dots, U_m$ be pairwise disjoint semialgebraic subsets of $\mathrm{cl}(A)$, which satisfy the condition that for any $m$-tuple $(\varepsilon_0, \dots, \varepsilon_m)$ of positive real numbers, and for $i = 0, \dots, m$, the sets

$$\inf \left\{ \gamma_{\mathrm{cone},A} \left( U_i - \bigcup_{j=i+1}^{m} U_j^{\varepsilon_j} \right) \right\} > 0. \tag{5.1}$$

Note that these sets have a uniform cone radius with respect to $A$. Hence, we say that the sets $U_0, \dots, U_m$ form a *uniform cone radius collection* of $\mathrm{cl}(A)$.

When the sets $U_0, \dots, U_m$ of a uniform cone radius collection of $A$ form a decomposition of $\mathrm{cl}(A)$, i.e.,

$$\mathrm{cl}(A) = U_0 \cup \cdots \cup U_m,$$

then we call $U_0, \dots, U_m$ a *uniform cone radius decomposition of* $\mathrm{cl}(A)$.

We now show how to construct such a uniform cone radius decomposition of $\mathrm{cl}(A)$. For any closed subset $X \subseteq \mathrm{cl}(A)$, we define

$$\Gamma_{\mathrm{nc}}(X) := \{\vec{p} \in X \mid \gamma_{\mathrm{cone},A}\,|_X \text{ is not continuous in } \vec{p}\}. \tag{5.2}$$

Let $\Delta_0 := \mathrm{cl}(A)$, and let $\Delta_{i+1} := \mathrm{cl}(\Gamma_{\mathrm{nc}}(\Delta_i)) \cap \Delta_i$. We define for $k = 0, 1, \dots$ the sets

$$C_k := \Delta_k - \Delta_{k+1}. \tag{5.3}$$

By taking $f = \gamma_{\mathrm{cone},A}$ in the following lemma, we obtain that $\Gamma_{\mathrm{nc}}(X)$ is semialgebraic and $\dim(\Gamma_{\mathrm{nc}}(X)) < \dim X$.

LEMMA 5.4. *For each semialgebraic set $X$ in $\mathbf{R}^n$ and each semialgebraic function $f : X \to \mathbf{R}$, the set $\Gamma(f) = \{\vec{p} \in X \mid f(\vec{p})$ is not continuous in $\vec{p}\}$ is semialgebraic and* $\dim(\Gamma(f)) < \dim X$.

*Proof.* The set

$$\Gamma(f) = \{\vec{p} \in \mathbf{R}^n \mid (\exists \varepsilon > 0)(\forall \delta > 0)\exists \vec{q} \in \mathbf{R}^n(\vec{q} \in X \cap B^n(\vec{p}, \delta) \wedge |f(\vec{p}) - f(\vec{q})| > \varepsilon)\}$$

is clearly semialgebraic. This proves the first assertion.

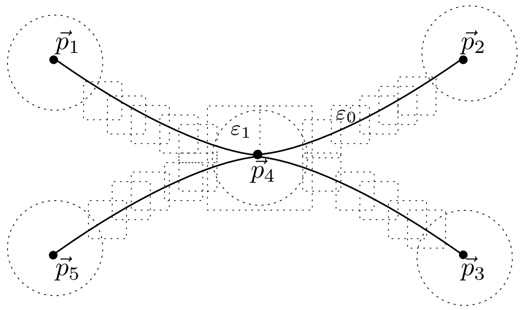


FIG. 5.3. *The points $\vec{p}_1$, $\vec{p}_2$, $\vec{p}_3$, $\vec{p}_4$, and $\vec{p}_5$ form the part $C_1$ which has $\varepsilon_1$ as uniform cone radius. As can be seen, the set $C_0 = A - C_1^{\varepsilon_1}$ has a uniform cone radius $\varepsilon_0$.*

We prove the second assertion by contradiction. Let $d = \dim X$ and suppose that $\dim(\Gamma(f)) = d$. Then there exists a semialgebraic cell $V \subseteq \Gamma(f)$ of dimension $d$. By the cell decomposition theorem of semialgebraic sets [44, Theorem 2.11] there exists a semialgebraic cell decomposition of $V$ into a finite number of semialgebraic cells,

$$V = V_1 \cup \cdots \cup V_k \cup V_{k+1} \cup \cdots \cup V_\ell,$$

with $\dim(V_i) = d$ for $i = 1, \ldots, k$ and $\dim(V_j) < d$ for $j = k+1, \ldots, \ell$, such that

$$f|_{V_i} \text{ is continuous for every } i = 1, \ldots, \ell. \tag{5.4}$$

Since $V_i \subseteq V$ has dimension $d$ for $i = 1, \ldots, k$, $V_i$ is open in $V$, and $V_i$ is also open in $X$ for $i = 1, \ldots, k$. From (5.4) we deduce that each $V_i$ for $i = 1, \ldots, k$ is included in $X - \Gamma(f)$, which is impossible since $V \subseteq \Gamma(f)$. Hence, $\dim(\Gamma(f)) < d$. ∎

An immediate consequence of this lemma is that from $i = n+1$ on, the $C_i$'s are all empty. Let us denote by $m$ the latest index such that $C_m$ is nonempty. Thus, $m \leqslant n$.

We now prove that for any tuple $(\varepsilon_0, \ldots, \varepsilon_m)$ of positive real numbers, the sets

$$C_i - \bigcup_{j=i+1}^{m} C_j^{\varepsilon_j} \quad \text{for } i = 0, 1, \ldots, m$$

have a uniform cone radius. Since $C_m = \Delta_m$ is closed, $\gamma_{\mathrm{cone},A}(C_m)$ is also closed and therefore has a minimum which is strictly positive. Hence, $C_m$ has a uniform cone radius. For $i > 0$ there exists an $\eta < \min\{\varepsilon_0, \ldots, \varepsilon_m\}$ such that

$$C_i - \bigcup_{j=i+1}^{m} C_j^{\varepsilon_j} \subseteq Z := \Delta_i - \Delta_{i+1}^{\eta}. \tag{5.5}$$

The set $Z$ is closed and the restriction $\gamma_{\mathrm{cone},A} \mid Z$ is continuous. Hence, $\gamma_{\mathrm{cone},A}(Z)$ is closed in $\mathbf{R}$ and has a minimum which is strictly positive. We may conclude that $Z$ has a uniform cone radius, and by (5.5), so has $C_i - \bigcup_{j=i+1}^{m} C_j^{\varepsilon_j}$. Thus, $C_0, \ldots, C_m$ is a uniform cone radius decomposition of $\mathrm{cl}(A)$.

*Example* 5.2. In Figure 5.3, we have shown the uniform cone radius decomposition of the set depicted in Figure 5.2. ⋄

Let $\mathcal{S} = \{S\}$, with $S$ an $n$-ary relation name. We define the $n+1$ queries $Q_k^{\text{uniform}}$ such that for any polynomial constraint database $D$ over $\mathcal{S}$,

$$Q_k^{\text{uniform}}(D) := C_k$$

for $k = 0, 1, \ldots, n$, with $C_0, \ldots, C_n$ being the uniform cone radius decomposition of $\text{cl}(S^D)$.

Because $\gamma_{\text{cone},S^D}$ equals $\texttt{uniqueradius}(D)$, and by Theorem 5.3 the formula $\texttt{uniqueradius}$ is in FO+POLY, the following lemma is immediate.

LEMMA 5.5. *The queries $Q_{k\text{-uniform}}$, $k = 0, 1, \ldots, n$, are expressible in* FO+POLY.

**5.3. The regular decomposition.** In this section, we construct a decomposition of semialgebraic sets such that a certain regularity condition is satisfied on each part of the decomposition. In order to define this regularity condition, we need to define the tangent space to a semialgebraic set in a point. The following definitions are taken from Rannou [39].

Let $A$ be a semialgebraic set in $\mathbf{R}^n$. The *secants limit set* of $A$ in a point $\vec{p} \in A$ is defined as the set

$$\text{limsec}_{\vec{p}} A := \bigcap_{\eta > 0} \text{cl}(\{\lambda(\vec{u} - \vec{v}) \in \mathbf{R}^n \mid \lambda \in \mathbf{R} \text{ and } \vec{u}, \vec{v} \in A \cap B^n(\vec{p}, \eta)\}).$$

If $\text{limsec}_{\vec{p}} A$ is a vector space, then we define the *tangent space of $A$ in $\vec{p}$* as $\text{T}_{\vec{p}} A := \vec{p} + \text{limsec}_{\vec{p}} A$. If $\text{limsec}_{\vec{p}} A$ is not a vector space, then the tangent space of $A$ in $\vec{p}$ is undefined.

Let $\mathcal{S} = \{S\}$, with $S$ an $n$-ary relation name. We define the query $Q_{\text{tangent}}$ as the query such that for any polynomial constraint database $D$ over $\mathcal{S}$,

$$Q_{\text{tangent}}(D) := \{(\vec{x}, \vec{v}) \in S^D \times \mathbf{R}^n \mid \text{T}_{\vec{x}} S^D \text{ exists in } \vec{x} \text{ and } \vec{v} \in \text{T}_{\vec{x}} S^D\}.$$

LEMMA 5.6. *The query $Q_{tangent}$ is expressible in* FO+POLY.

*Proof.* It is shown by Rannou [39, Lemma 2] that the definition of the secant limit set of a set in a point can be translated into a first-order formula over the reals. Since it is straightforward to check in FO+POLY whether a secant limit set is a vector space (i.e., we need to check whether, for all $\vec{s}, \vec{t}$ in a secant limit set, the sum $\vec{s} + \vec{t}$ is also an element of this secant limit set), the lemma is immediate. ∎

The set $A$ is *regular in $\vec{p}$* if and only if $\text{T}_{\vec{p}} A$ exists and there exists a neighborhood $U$ of $\vec{p}$ such that the orthogonal projection of $A \cap U$ on $\text{T}_{\vec{p}} A$ is bijective. A set is *regular* if it is regular in all of its points.

A finite number of pairwise disjoint regular sets $R_1, \ldots, R_k$ is called a *regular decomposition of $A$* if $A = R_1 \cup \cdots \cup R_k$.

We now show that every semialgebraic set $A$ has a regular decomposition.

We denote the set of points where $A$ is regular and where the local dimension of $A$ is $k$ by $Reg_k(A)$. Note that $Reg_k(A)$ is either empty or $\dim Reg_k(A) = k$.

Define inductively for $k = n, n-1, \ldots, 0$, the sets

$$R_k := Reg_k\left(A - \bigcup_{j=k+1}^{n} R_j\right). \tag{5.6}$$

These sets are pairwise disjoint and form a decomposition of $A$, i.e.,

$$A = R_n \cup R_{n-1} \cup \cdots \cup R_0. \tag{5.7}$$

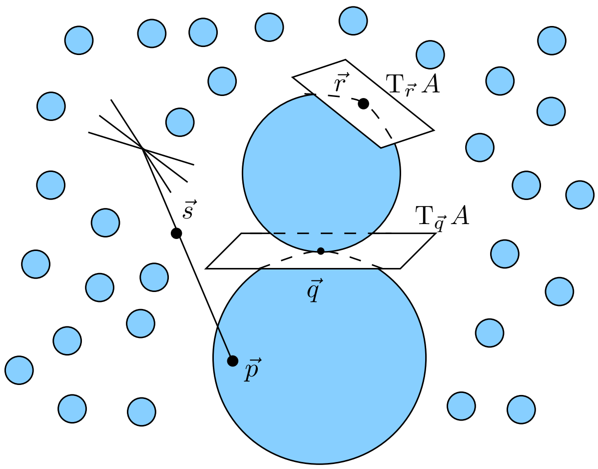


FIG. 5.4. *The snowman $A$ has no tangent space in $\vec{p}$, $A$ has a tangent space in $\vec{q}$ and $\vec{r}$ but is not regular in these points, and $A$ is regular in $\vec{s}$.*

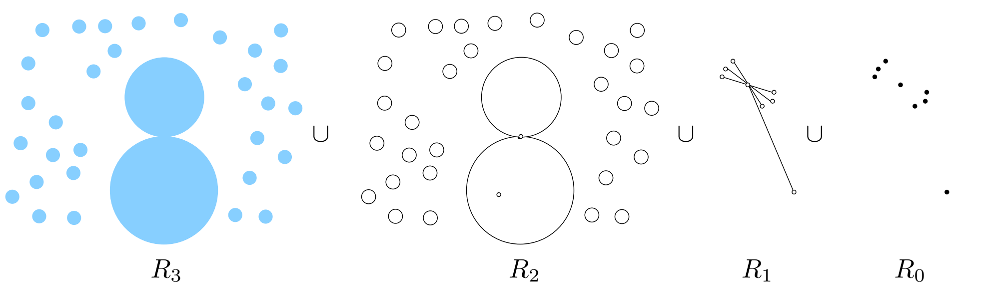


FIG. 5.5. *The three-dimensional set $A$ of Figure 5.4 is decomposed into four parts $R_0$, $R_1$, $R_2$, and $R_3$ according to the construction of the regular decomposition.*

Note that $n+1$ parts are really sufficient because, for any semialgebraic set $X \subseteq \mathbf{R}^n$ of dimension $d$, $X - Reg_d(X)$ has a strictly lower dimension than $X$ [45].

Moreover, by (5.6) each $R_k$ is regular, and hence we define the *regular decomposition of $A$* as the $n+1$ sets $R_0, \ldots, R_n$.

*Example* 5.3. In Figure 5.4, we have illustrated the three possible cases: $T_{\vec{p}}\,A$ does not exist; $T_{\vec{q}}\,A$ and $T_{\vec{r}}\,A$ exist but $A$ is not regular in $\vec{q}$ and $\vec{r}$; and finally, $A$ is regular in $\vec{s}$. In Figure 5.5, we have drawn an example of the regular decomposition of a three-dimensional set in $\mathbf{R}^3$. $\diamond$

Let $\mathcal{S} = \{S\}$, with $S$ an $n$-ary relation name. We define the $n+1$ queries $Q_k^{\text{reg}}$ as the queries such that for every polynomial constraint database $D$,

$$Q_k^{\text{reg}}(D) := R_k$$

for $k = 0, \ldots, n$, with $R_0, \ldots, R_n$ the regular decomposition of $S^D$.

It was proved by Rannou [39, Proposition 2] that checking whether a semialgebraic set is regular in a point is first-order expressible. Hence the next lemma follows.

LEMMA 5.7. *The queries $Q_{k\text{-}reg}$, $k = 0, 1, \ldots, n$ are expressible in* FO+POLY.

Regular decompositions of semilinear sets are fully treated by Dumortier et al. [10] and Vandeurzen [46]. These authors showed that on semilinear databases, the $n+1$

queries $Q_{k\text{-reg}}$ are already expressible in FO+LIN. There is, however, a great difference between the semilinear and semialgebraic cases. Indeed, in the semialgebraic case, regularity implies that the set is a $C^1$-smooth algebraic variety, while in the semilinear case, regularity implies that the set is a $C^\infty$-smooth algebraic variety. One could ask if it is possible to define a regularity condition in first-order logic such that it also induces $C^\infty$-smoothness of semialgebraic sets, but this is impossible [49].

However, we still can generalize the regular decompositions defined above to $C^k$-regular decompositions by demanding $C^k$-smoothness instead of $C^1$-smoothness (regularity). Using again results from Rannou [39, Proposition 3], we have first-order expressibility of the corresponding query in this case too.

An interesting question is which extensions of FO+POLY can express $C^\infty$-regular decompositions. A useful observation in this context might be that for every semialgebraic set, there exists a natural number $K$ such that for all $k > K$, a $C^k$-regular decomposition is already a $C^\infty$-regular decomposition. Unfortunately, it is not known how to find $K$ for a given semialgebraic set [40] and we might have to compute $C^k$-regular decompositions for increasing values of $k$ until two consecutive decompositions are identical. This indicates that recursion is needed for the computation of $C^\infty$-regular decompositions. We leave open whether the recursion in FO+POLY+TC or FO+POLY+TCS is sufficient for this purpose.

**5.4. Transversality.** In computational geometry [9], a convenient assumption is the hypothesis of "general position," which dispenses with the detailed consideration of special cases. In the description of our linearization algorithm in section 6, we would like to assume this hypothesis. However, we need to make precise what we will mean by general position and see if this hypothesis may indeed be assumed.

Let $A$ and $B$ be two regular semialgebraic sets in $\mathbf{R}^n$. From differential topology [23], we recall that $A$ and $B$ are said to *intersect transversally* at $\vec{p} \in A \cap B$ if [3]

$$\mathrm{T}_{\vec{p}}\, A + \mathrm{T}_{\vec{p}}\, B = \mathbf{R}^n. \tag{5.8}$$

The sets $A$ and $B$ are *in general position* if they intersect transversally in every point of $A \cap B$. We denote this by $A \pitchfork B$. This is illustrated in Figure 5.6 and Figure 5.7, where some examples of transversal and nontransversal intersections in $\mathbf{R}^2$ and $\mathbf{R}^3$ are depicted.

Let $\mathcal{A} = \{A_1, \ldots, A_n\}$ and $\mathcal{B} = \{B_1, \ldots, B_m\}$ be finite sets of regular semialgebraic sets in $\mathbf{R}^n$ such that $A_i \cap A_j = \emptyset$ and $B_i \cap B_j = \emptyset$ for $i \neq j$. We say that $\mathcal{A}$ and $\mathcal{B}$ are in general position if $A_i$ and $B_j$ are in general position for every $i = 1, \ldots, n$ and every $j = 1, \ldots, m$. We denote this by $\mathcal{A} \pitchfork \mathcal{B}$.

Let $\mathcal{S} = \{S_1, S_2\}$, with $S_1$ and $S_2$ two $n$-ary relation names. We define the Boolean query $Q_\pitchfork$ such that for every polynomial constraint database $D$ over $\mathcal{S}$,

$$Q_\pitchfork(D) = \textit{true} \text{ if and only if } S_1^D \text{ and } S_2^D \text{ are regular and } S_1^D \pitchfork S_2^D.$$

Condition (5.8) can be readily expressed in FO+POLY, and by Lemma 5.7, regularity is expressible in FO+POLY. Hence Lemma 5.8 follows.

LEMMA 5.8. *The Boolean query $Q_\pitchfork$ is expressible in* FO+POLY.

Given two arbitrary regular semialgebraic sets $A$ and $B$ in $\mathbf{R}^n$ not in general position, we can ask how to force them to be in general position. The following

[3] Let $U$ and $V$ be two subspaces of a vector space $X$; then the *sum* $U + V$ is the set of all vectors $\vec{u} + \vec{v}$, where $\vec{u} \in U$ and $\vec{v} \in V$. Besides, $U + V$ is a subspace of $X$.

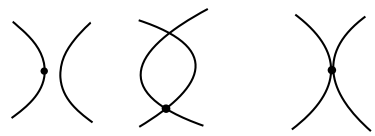


FIG. 5.6. *Curves in* $\mathbf{R}^2$.

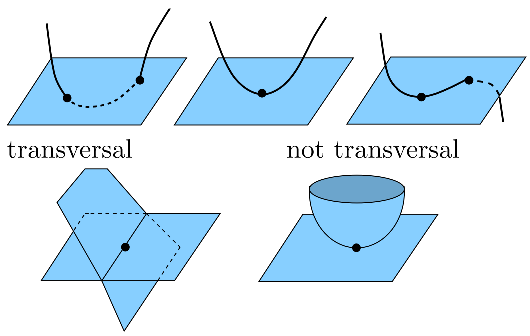


FIG. 5.7. *Curves and surfaces in* $\mathbf{R}^3$.

theorem answers this question. A translation of a set $X \subseteq \mathbf{R}^n$ is a set of the form $X + \boldsymbol{\tau} := \{\vec{x} + \boldsymbol{\tau} \in \mathbf{R}^n \mid \vec{x} \in X\}$, where $\boldsymbol{\tau} \in \mathbf{R}^n$.

THEOREM 5.9. *Let $A$ and $B$ two regular semialgebraic sets in $\mathbf{R}^n$. For almost all $\boldsymbol{\tau} \in \mathbf{R}^n$, we have that $A + \boldsymbol{\tau}$ and $B$ are in general position.*

*Proof.* This theorem is a direct consequence of the transversality theorem of differential topology. A proof of the transversality theorem given by Guillemin and Pollack [23] for $C^\infty$-smooth varieties in $\mathbf{R}^n$ literally remains valid in this case, except that the $C^1$-version of Sard's theorem given by Wilkie [50] needs to be used instead of the standard $C^\infty$-smooth version. $\square$

Here, "almost all" means that the set of translation vectors $\boldsymbol{\tau}$ for which $A+\boldsymbol{\tau}$ and $B$ are not in general position has *measure zero*.[4] Since a set of measure zero cannot contain an open set in $\mathbf{R}^n$, the set of translation vectors $\boldsymbol{\tau}$ for which $A+\boldsymbol{\tau}$ and $B$ are in general position is dense in $\mathbf{R}^n$.

Moreover, Theorem 5.9 can be easily generalized as follows.

COROLLARY 5.10. *Let $\mathcal{A} = \{A_1, \dots, A_n\}$ and $\mathcal{B} = \{B_1, \dots, B_m\}$ be sets of regular semialgebraic sets in $\mathbf{R}^n$ such that $A_i \cap A_j = \emptyset$ ($B_i \cap B_j = \emptyset$) for $i \neq j$. Then for almost all $\boldsymbol{\tau} \in \mathbf{R}^n$, $\mathcal{A} + \boldsymbol{\tau} \pitchfork \mathcal{B}$.*

We mention three useful properties of sets in general position. Let $\mathcal{A}$ and $\mathcal{B}$ be as above; then if $\mathcal{A} \pitchfork \mathcal{B}$, there exists an $\varepsilon > 0$ such that $\mathcal{A} + \boldsymbol{\tau} \pitchfork \mathcal{B}$ for any $\boldsymbol{\tau} \in \mathbf{R}^n$ of norm less than $\varepsilon$. Therefore, one says that transversality is a *stable* property. A second useful property is that the intersection of two regular sets in general position is again regular. A third property is that the tangent space in a point of the intersection of two sets in general position is the intersection of the tangent spaces of these sets in this point [23].

[4]A set in $\mathbf{R}^n$ has *measure zero* if it can be covered by a countable number of $n$-dimensional boxes with arbitrary small volume.

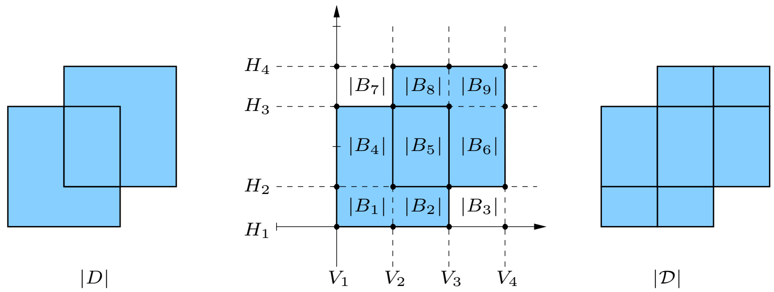


FIG. 5.8. *A two-dimensional example of the construction of a box collection for two boxes in the* $\mathbf{R}^2$.

**5.5. Box collections.** We need one more ingredient before we can start explaining the linearization algorithm: box collections.

We define an *$n$-dimensional box collection* $\mathcal{B}$ in $\mathbf{R}^n$ as a finite set of $n$-dimensional boxes satisfying an intersection condition. Let $B_1$ and $B_2$ be two arbitrary boxes in $\mathcal{B}$. Then, if $|B_1|$ and $|B_2|$ intersect, the intersection is included in their boundaries $\partial|B_1|$ and $\partial|B_2|$. By the *geometric realization* $|\mathcal{B}|$ of $\mathcal{B}$, we mean the union of the geometric realizations of all boxes in $\mathcal{B}$. If $X \subseteq \mathbf{R}^n$ is a semialgebraic set and $\mathcal{B}$ an $n$-dimensional box collection in $\mathbf{R}^n$, then $\mathcal{B} \cap X$ is the set of all boxes $B \in \mathcal{B}$ such that $B \cap X \neq \emptyset$.

Let $D$ be a set of $n$-dimensional boxes, which does not necessarily satisfy the above intersection condition. In the following, we show how in FO+POLY to split the boxes in $D$ into smaller boxes such that the collection of these smaller boxes is a box collection. We call this the *box collection of* $D$ and denote it by $\mathcal{D}$. By construction, the geometric realization of each box in $D$ is the union of the geometric realizations of certain boxes of $\mathcal{D}$.

We first give an example of the construction and then present the general construction more formally.

*Example* 5.4. Fix the dimension $n = 2$ and consider the set $D$ consisting of two boxes $(0, 2, 0, 3)$ and $(1, 3, 1, 4)$. The geometric realization $|D|$ of $D$ is depicted in Figure 5.8. In this figure, two sets of lines, $\mathcal{H}_{D,x} = \{H_1, H_2, H_3, H_4\}$ and $\mathcal{H}_{D,y} = \{V_1, V_2, V_3, V_4\}$, are drawn. Denote the intersection $\bigcup \mathcal{H}_{D,x} \cap \bigcup \mathcal{H}_{D,y}$ by $I$. In this example, $I$ consists of 16 points $\{\vec{p}_1, \ldots, \vec{p}_{16}\}$. From these points, we construct the set $\mathcal{P}$ which contains the 9 two-dimensional boxes denoted by $B_i$, $i = 1, \ldots, 9$. The geometric realizations of these boxes are shown in the figure. As can be seen, these boxes intersect only at their boundaries, and hence form a two-dimensional box collection. Finally, we define the box collection $\mathcal{D}$ of $D$ as the boxes included in $|D|$, i.e., $\mathcal{D} = \{B_1, B_2, B_4, B_6, B_6, B_8, B_9\}$. $\diamond$

In general, we define $n$ unions of $(n-1)$-dimensional hyperplanes,

$$\mathcal{H}_{D,i} := \{(x_1, \ldots, x_n) \in \mathbf{R}^n \mid \exists (a_1, b_1, \ldots, a_n, b_n) \in D \wedge (x_i = a_i \vee x_i = b_i)\},$$

for $i = 1, \ldots, n$. Let $I \subseteq \mathbf{R}^n$ be the set of points $\mathcal{H}_{D,1} \cap \cdots \cap \mathcal{H}_{D,n}$.

It is easily shown that $I$ is a finite set of points. Indeed, a proof by induction shows that $\dim(\mathcal{H}_{D,1} \cap \cdots \cap \mathcal{H}_{D,k}) = n - k$ for any $k = 1, \ldots, n$. In particular, $\dim(I) = n - n = 0$, or in other words, $I$ is a finite set.

Next, we construct an $n$-dimensional box collection, which we denote by $\mathcal{P}$, such that the geometric realization of each box in $D$ is the union of the geometric

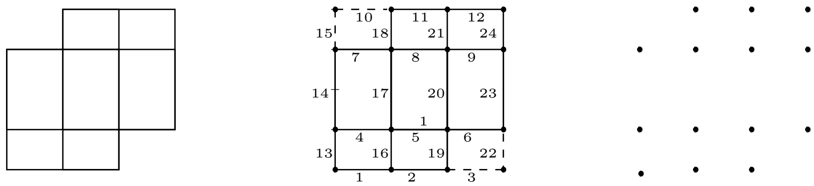


FIG. 5.9. *The set $|\mathcal{D}| - |\mathcal{D}|_2$ (left). The one-dimensional box collection $\mathcal{P}_x \cup \mathcal{P}_y$, where the line segment $L_i$ is labeled with the number $i$ (center). The set $|\mathcal{D}|_0$ (right).*

realizations of boxes in $\mathcal{P}$. More specifically,

$$\mathcal{P} := \Bigg\{ (a_1, b_1, \ldots, a_n, b_n) \in \mathbf{R}^{2n} \mid \exists \vec{p}_1 \exists \vec{q}_1 \cdots \exists \vec{p}_n \exists \vec{q}_n \in I$$
$$\bigwedge_{i=1}^{n} (a_i = (\vec{p}_i)_i \wedge b_i = (\vec{q}_i)_i \wedge a_i < b_i)$$
$$\wedge \left( \forall \vec{r} \in I \bigwedge_{i=1}^{n} \neg (a_i < (\vec{r})_i < b_i) \right) \Bigg\}.$$

Finally, we define $\mathcal{D}$ as those $n$-dimensional boxes $B$ in $\mathcal{P}$ such that $|B|$ is included in the geometric realization of any of the boxes in $B$. By construction, $\mathcal{D}$ is a box collection, and the geometric realization of any box in $D$ is the union of the geometric realizations of certain boxes in $\mathcal{D}$. The construction of $\mathcal{D}$ for a given $D$ can be expressed in FO+POLY, as is clear from the above expressions for $\mathcal{H}_{D,i}$ and $\mathcal{P}$.

Let $\mathcal{S} = \{S\}$, with $S$ a $2n$-ary relation name. We define the *box collection query* $Q_{\mathrm{bc}}$ such that for any polynomial constraint database $D$ over $\mathcal{S}$ representing a set of $n$-dimensional boxes in $\mathbf{R}^n$,

$$Q_{\mathrm{bc}}(D) = \mathcal{D},$$

where $\mathcal{D}$ is the box collection of $D$. From the constructions above, the following result is immediate.

LEMMA 5.11. *The query $Q_{\mathrm{bc}}$ is expressible in* FO+POLY.

When applied to the union of two box collections $D$ and $D'$, we will denote the box collection $Q_{\mathrm{bc}}(D \cup D')$ by $\mathcal{D} \sqcup \mathcal{D}'$.

We next define a useful decomposition of box collections. We again give first an example.

*Example* 5.5 (see Figure 5.8 and Figure 5.9). Let us continue the previous example. Let $|\mathcal{D}|_2$ be the set in $\mathbf{R}^2$ defined by $\bigcup_{i \in \{1,2,4,5,6,8,9\}} \mathrm{int}(|B_i|)$. Consider the set $|\mathcal{D}| - |\mathcal{D}|_2$ and define $\mathcal{P}_x$ to be the set of horizontal line segments $L_i$, with $i = 1, \ldots, 12$, and let $\mathcal{P}_y$ be the set of vertical line segments $L_i$, with $i = 13, \ldots, 24$. The line segments $L_i$ can easily be defined from the points in $I$ and from a one-dimensional box collection. We define $\mathcal{D}^1$ to be the box collection consisting of boxes in $\mathcal{P}_x \cup \mathcal{P}_y$, which are contained in $|\mathcal{D}|$. Next, define $|\mathcal{D}|_1$ to be the set $\bigcup_{i \in \{1,\ldots,24\} - \{3,10,22,15\}} \mathrm{int}(|L_i|)$. Here, when taking the interior, we regard each $|L_i|$ as a space on itself, so the result is an open line segment without the endpoints (as opposed to the empty set, where we would regard each $|L_i|$ as a set in $\mathbf{R}^2$). Now, $|\mathcal{D}| - |\mathcal{D}|_2 - |\mathcal{D}|_1$ is a subset of $I$, which we denote by $|\mathcal{D}|_0$. Hence, we have obtained a decomposition of $|\mathcal{D}|$. $\diamond$

This decomposition is important for two reasons. First, the geometric realization of each box of $D$ is the disjoint union of the interiors of the geometric realizations of certain boxes in $\mathcal{D}^2$, $\mathcal{D}^1$, and $\mathcal{D}^0$. Second, the interiors of boxes in $\mathcal{D}$ are open subsets of $Reg_2(|\mathcal{D}|)$, the interiors of boxes in $\mathcal{D}^1$ are open subsets of $Reg_1(|\mathcal{D}| - |\mathcal{D}|_2)$, and finally, $|\mathcal{D}|_0$ equals $Reg_0(|\mathcal{D}| - |\mathcal{D}|_2 - |\mathcal{D}|_1)$.

In general, the construction of this decomposition goes as follows. For $k = 0, 1, \ldots, n$ and any combination of $k$ different elements $i_1, \ldots, i_k$ in $\{1, \ldots, n\}$, we define the following set of $(n-k)$-dimensional boxes in $\mathbf{R}^n$:

$$
\begin{aligned}
\mathcal{P}_{\{i_1,\ldots,i_k\}} := \Bigg\{ & (a_1, b_1, \ldots, a_n, b_n) \in \mathbf{R}^{2n} \mid \exists \vec{p}_1 \exists \vec{q}_1 \cdots \exists \vec{p}_n \exists \vec{q}_n \in I \\
& \bigwedge_{i \in \{1,\ldots,n\}} (a_i = (\vec{p}_i)_i \wedge b_i = (\vec{q}_i)_i) \wedge \forall \vec{r} \in I \bigwedge_{i=1}^{n} \neg(a_i < (\vec{r})_i < b_i) \\
& \wedge \bigwedge_{i \in \{1,\ldots,n\} - \{i_1,\ldots,i_k\}} a_i < b_i \wedge \bigwedge_{i \in \{i_1,\ldots,i_k\}} a_i = b_i \Bigg\}.
\end{aligned} \tag{5.9}
$$

Note that $\mathcal{P}_{\{1,\ldots,n\}} = I$ and $\mathcal{P}_\emptyset = \mathcal{P}$. It is clear that these sets are expressible in FO+ Poly. We also define for $k = 0, 1, \ldots, n$ and any combination of $k$ different elements $i_1, \ldots, i_k$ in $\{1, \ldots, n\}$ the following $(n-k)$-dimensional box collection in $\mathbf{R}^n$:

$$
\begin{aligned}
\mathcal{D}_{\{i_1,\ldots,i_k\}} := \Bigg\{ & (a_1, b_1, \ldots, a_n, b_n) \in \mathcal{P}_{\{i_1,\ldots,i_k\}} \mid \exists (a'_1, b'_1, \ldots, a'_n, b'_n) \in \mathcal{D} \\
& \wedge \bigwedge_{i=1}^{n} (a'_i \leqslant a_i \wedge b_i \leqslant b'_i) \Bigg\}.
\end{aligned}
$$

We then define

$$
\mathcal{D}^{n-k} := \bigcup_{\{i_1,\ldots,i_k\}} \mathcal{D}_{\{i_1,\ldots,i_k\}}.
$$

Finally, for $k = 0, 1, \ldots, n$, we define $|\mathcal{D}|_{n-k}$ as the union of the interiors of the geometric realizations of boxes in $\mathcal{D}^{n-k}$. Here, when taking the interior, we regard each geometric realization of a box as a space on itself, so the result is an open box. By construction, we have the following properties:

1. $$|\mathcal{D}| = |\mathcal{D}|_n \cup \cdots \cup |\mathcal{D}|_0; \tag{5.10}$$
2. each geometric realization of a box in $\mathcal{D}$ is the union of the geometric realizations of boxes in $|\mathcal{D}|_k$ for $k = 0, 1, \ldots, n$; and
3. the interiors of the geometric realizations of boxes in $\mathcal{D}^k$ are open subsets of $Reg_k(|\mathcal{D}| - |\mathcal{D}|_n - \cdots - |\mathcal{D}|_{k+1})$.

Let $\mathcal{S} = \{S\}$, with $S$ a $2n$-ary relation name. We define the $n+1$ queries $Q_{k\text{-box}}$ such that for any polynomial constraint database $D$ over $\mathcal{S}$ representing a box collection $\mathcal{D}$,

$$
Q_{k\text{-box}}(D) = \mathcal{D}^i
$$

for $k = 0, 1, \ldots, n$ with $\mathcal{D}^i$ the $i$-dimensional box collection in $\mathbf{R}^n$ defined above. The following trivially holds.

Lemma 5.12. *The queries $Q_{k\text{-box}}$, $k = 0, 1, \ldots, n$, are expressible in* FO+Poly.

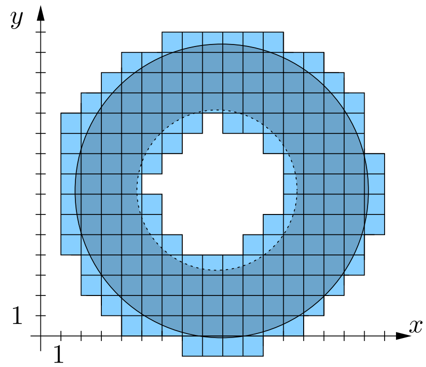


FIG. 5.10. *The $\delta$-cover of a semiopen annulus for $\delta = 1$.*

**5.6. Expressing the box covering query.** Let $\delta > 0$ be a real number. We define the $n$-dimensional standard grid of size $\delta$, called the $\delta$-grid, as the $n$-dimensional box collection $\delta$-grid consisting of all boxes of the form $(k_1\delta, (k_1+1)\delta, \ldots, k_n\delta, (k_n+1)\delta)$, where $k_1, \ldots, k_n \in \mathbf{Z}$. We define the *box covering of size* $\delta$ of a semialgebraic set $A$, denoted by $\delta$-cover$(A)$, as those boxes in the $\delta$-grid that intersect the closure of $A$ (see Figure 5.10). Let $\mathcal{S} = \{S\}$, with $S$ an $n$-ary relation name. For each $\delta > 0$ we define the *box covering query* $Q_{\delta\text{-cover}}$ such that for every constraint database $D$ over $\mathcal{S}$,

$$Q_{\delta\text{-cover}}(D) := \delta\text{-cover}(S^D).$$

PROPOSITION 5.13. *Let $\delta > 0$. The query $Q_{\delta\text{-cover}}$ is not expressible in* FO+POLY.

*Proof.* Let $\mathcal{S} = \{S\}$, with $S$ a binary relation name. We consider the following FO+POLY formula over $\mathcal{S}$: a formula `circle` such that for any database $D$ over $\mathcal{S}$, either $\texttt{circle}(D)$ is the circle through the points of $S^D$, if $S^D$ consists of three noncollinear points, or $\texttt{circle}(D) = S^D$.

Assume that the query $Q_{\delta\text{-cover}}$ is expressible in FO+POLY. Let $\delta$-cover be the formula which expresses $Q_{\delta\text{-cover}}$. Then the formula

$$\varphi \equiv \delta\text{-}\texttt{cover}(\texttt{circle})$$

is also expressible in FO+POLY. However, the number of 4-tuples in $\varphi(D)$ can be made arbitrarily large by choosing $D$ to be a database over $\mathcal{S}$ such that $S^D$ consists of three points far enough apart. This contradicts the dichotomy theorem of Benedikt and Libkin [4], which guarantees the existence of a polynomial $p_\varphi$ such that $|\varphi(D)| < p_\varphi(|S^D|) = p_\varphi(3)$ in the case when $|\varphi(D)|$ is finite. □

However, in FO+POLY+TC we can express the box covering query as follows.

PROPOSITION 5.14. *For each $\delta > 0$, the query $Q_{\delta\text{-cover}}$ is expressible in* FO+POLY+TC *when restricted to bounded input databases.*

*Proof.* Let $\mathcal{S} = \{S\}$, with $S$ an $n$-ary relation name. We define the bounding box query $Q_{\text{bb}}$ as the query such that for every polynomial constraint database $D$, such that $S^D$ is bounded, $Q_{\text{bb}}(D) := \{M\}$, with $M$ a real number such that $\text{cl}(S^D) \subseteq [-M, M]^n$. This query is clearly FO+POLY expressible by a formula over $\mathcal{S}$ which we denote by $\texttt{boundingbox}(x)$. Let

$$\texttt{grid}(u) \equiv [\text{TC}_{x;x'}\, \exists M(\texttt{boundingbox(M)} \wedge x \geqslant 0 \\ \wedge x' = x + \delta \wedge x' \leqslant M)](0, u) \vee u = 0.$$

Let

$$\delta\text{-}\mathtt{cover}(u_1, v_1, \ldots, u_n, v_n) \equiv \bigwedge_{i=1}^{n} (v_i = u_i + \delta \wedge \mathtt{grid}(u_i)) \\ \wedge \exists \vec{x} \left( \mathrm{cl}(S)(\vec{x}) \wedge \bigwedge_{i=1}^{n} u_i < x_i < v_i \right).$$

Then $Q_{\delta\text{-cover}}(D) = \delta\text{-}\mathtt{cover}(D)$ for any database $D$ over $\mathcal{S}$ such that $S^D$ is bounded. ◻

**6. Linearization and approximation of semialgebraic sets.** In this section, we give a construction of both an algebraic linearization and an $\varepsilon$-approximation of semialgebraic sets which are implementable in FO+Poly+TC. This implementation is based on the construction of a box collection satisfying some special properties.

More specifically, it is shown in section 6.1 how to construct such a box collection $\mathcal{R}$ for a semialgebraic set $A$. In section 6.2 we derive a box collection $\mathcal{U}$ from $\mathcal{R}$ and take a closer look at $A$ on the boundaries of $\mathcal{U}$. We show that we can apply the construction in section 6.1 again for $A$ on the lower-dimensional box collections on the boundaries of $\mathcal{U}$. This inductive process is the basis of the algorithm Linearize in section 6.3 which builds an algebraic linearization and an $\varepsilon$-approximation of bounded semi-algebraic sets. In the same section, we prove the correctness of the algorithm Linearize and show that the algorithm can be expressed by a query in FO+Poly+TC.

We also show how to extend this algorithm such that it also builds algebraic linearizations of possibly unbounded semialgebraic sets. Finally, in section 6.4 we show that after some minor changes, the algorithm Linearize can be used to build a rational linearization and an $\varepsilon$-approximation of semialgebraic sets.

**6.1. Construction of a special box collection.** Let $\mathcal{B}$ be an $n$-dimensional box collection in $\mathbf{R}^n$, and let $\mathcal{X} = \{X_1, \ldots, X_k\}$ be a finite set of pairwise disjoint semialgebraic sets in $\mathbf{R}^n$. We now define when $\mathcal{B}$ and $\mathcal{X}$ are in general position. We decompose $|\mathcal{B}|$ and $\mathcal{X}$ into a finite number of regular sets and then define "being in general position" in terms of these decompositions as follows.

In (5.10), we defined a decomposition of a box collection into regular sets. Applied to $|\mathcal{B}|$, this results in the decomposition $|\mathcal{B}|_n, \ldots, |\mathcal{B}|_0$, where $|\mathcal{B}|_i$ is a union of interiors of $i$-dimensional boxes in $\mathbf{R}^n$.

For each $X_i$, let $R_{i0}, \ldots, R_{in_i}$ be a regular decomposition of $X_i$. We say that $\mathcal{B}$ and $\mathcal{X}$ are in general position if and only if $\{|\mathcal{B}|_n, \ldots, |\mathcal{B}|_0\}$ and $\{R_{1,0}, \ldots, R_{1,n_1}, \ldots, R_{k,0}, \ldots, R_{k,n_k}\}$ are in general position.

We now describe the construction of an $n$-dimensional special box collection (the properties of this box collection will become clear later on). The construction takes as input

- a bounded semialgebraic set $A$ in $\mathbf{R}^n$;
- a uniform cone radius collection $U_0, \ldots, U_m$ of $\mathrm{cl}(A)$ (as defined in section 5.2); and
- a fixed $n$-dimensional box collection $\mathcal{F}$ in $\mathbf{R}^n$, which is in general position with $\{U_0, \ldots, U_m\}$.

The result of the construction will be

- a set of box collections $\mathcal{R} = \{\mathcal{R}_0, \ldots, \mathcal{R}_m\}$ and
- a positive real number $\delta$

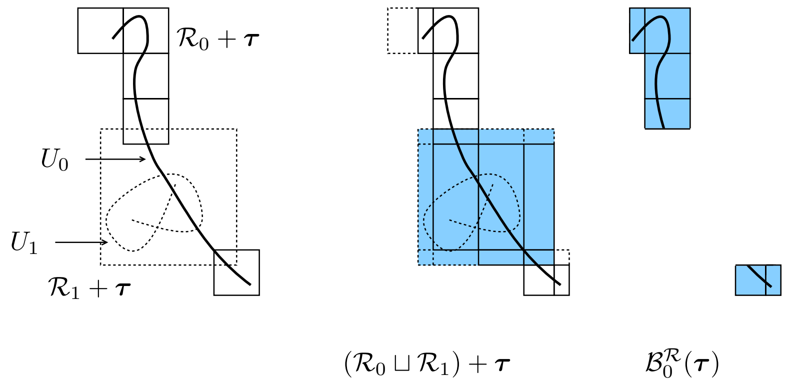


FIG. 6.1. *Illustration of the construction of the box collection $\mathcal{B}_0^{\mathcal{R}}(\boldsymbol{\tau})$ for $\mathcal{R} = \{\mathcal{R}_0, \mathcal{R}_1\}$ and $U = U_0 \cup U_1$ as explained in Example* 6.1. *The picture shows $\mathcal{R} + \boldsymbol{\tau}$ (right), the intermediate result $(\mathcal{R}_0 \sqcup \mathcal{R}_1) + \boldsymbol{\tau}$ (middle), and the end result $\mathcal{B}_0^{\mathcal{R}}(\boldsymbol{\tau})$ (right).*

satisfying some properties. Before we can state these properties, we need to define for $k = m, \ldots, 0$ and $\boldsymbol{\tau} \in \mathbf{R}^n$ the box collections

$$\begin{aligned}\mathcal{B}_k^{\mathcal{R}}(\boldsymbol{\tau}) := {} & (((\mathcal{R}_k \sqcup \cdots \sqcup \mathcal{R}_m) + \boldsymbol{\tau} \sqcup \mathcal{F}) \cap U_k) \\ & \setminus \{B' \in ((\mathcal{R}_k \sqcup \cdots \sqcup \mathcal{R}_m) + \boldsymbol{\tau} \sqcup \mathcal{F}) \cap U_k \mid |B'| \subseteq |\mathcal{B}_{k+1}^{\mathcal{R}}(\boldsymbol{\tau}) \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}}(\boldsymbol{\tau})|\}.\end{aligned}$$

In the following, we will write $\mathcal{B}_i^{\mathcal{R}}$ for $\mathcal{B}_i^{\mathcal{R}}(\mathbf{0})$ and let $U = U_0 \cup \cdots \cup U_m$. The definition of $\mathcal{B}_k^{\mathcal{R}}(\boldsymbol{\tau})$ basically tells how to fit together all the box collections in $\mathcal{R}$ and specifies which boxes should be disregarded. We illustrate the definition of $\mathcal{B}_k^{\mathcal{R}}$ by the following example.

*Example* 6.1. Assume we have a box collection $\mathcal{R} = \{\mathcal{R}_0, \mathcal{R}_1\}$ covering $U = U_0 \cup U_1$. In Figure 6.1 (left) we have depicted $\mathcal{R}_0$ and $\mathcal{R}_1$ with solid and dotted lines, respectively. Moreover, the set $U_1$ consists of the dotted curve, while $U_0$ is shown as a thick solid line. In this example, we assume that no fixed box collection $\mathcal{F}$ is present.

Then by definition, $\mathcal{B}_1^{\mathcal{R}}(\boldsymbol{\tau}) = (\mathcal{R}_1 + \boldsymbol{\tau}) \cap U_1$. This box collection (in this example consisting of a single box only) corresponds to the large shaded box in Figure 6.1 (middle). For the construction of $\mathcal{B}_0^{\mathcal{R}}(\boldsymbol{\tau})$, we first compute the box collection $(\mathcal{R}_0 \sqcup \mathcal{R}_1) + \boldsymbol{\tau}$, which consists of all the boxes shown in Figure 6.1 (middle). Solid-lined boxes intersect $U_0$; dotted-lined boxes do not. In order to obtain $\mathcal{B}_0^{\mathcal{R}}(\boldsymbol{\tau})$, all dotted-lined boxes are removed as well as those solid-lined boxes, which are included in $\mathcal{B}_1^{\mathcal{R}}(\boldsymbol{\tau})$ (the shaded area). The resulting box collection $\mathcal{B}_0^{\mathcal{R}}(\boldsymbol{\tau})$ is shown in Figure 6.1 (right). $\diamond$

We now continue with the statement of the desired properties of the box collection $\mathcal{R}$ and real number $\delta$. They must satisfy the properties

(i) $\mathrm{cl}(U)^\delta \subseteq \mathrm{int}(|\mathcal{B}_0^{\mathcal{R}} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}}|)$;

(ii) for all $i = 0, \ldots, m$ and for all $\boldsymbol{\tau} \in \mathbf{R}^n$ of norm less than $\delta$, $(\mathcal{R}_i + \boldsymbol{\tau}) \sqcup \mathcal{F} \pitchfork U_i$; and

(iii) for all $i = 0, \ldots, m$ and for all $\boldsymbol{\tau} \in \mathbf{R}^n$ of norm less than $\delta$, and for each $n$-dimensional box $B \in \mathcal{B}_i^{\mathcal{R}}(\boldsymbol{\tau})$, there exists a point $\vec{p} \in \mathrm{int}(|B| \cap U_i)$ such that $\gamma_{\mathrm{cone},A}(\vec{p}) > \mathrm{diam}(B)$.

*Construction algorithm.* The construction of the box collection is done inductively on the number of parts $m$ in the uniform cone radius collection $\{U_0, \ldots, U_m\}$.

For the base case, when the uniform cone radius collection is empty, we define

$\mathcal{R}_{-1} = \emptyset$ and take $\delta = \infty$. Properties (i), (ii), and (iii) are then trivially satisfied.

Suppose now that $U$ is nonempty and consists of $m$ parts. By the induction hypothesis, there exist $n$-dimensional box collections $\mathcal{R}' = \{\mathcal{R}'_1, \ldots, \mathcal{R}'_m\}$ and a positive real number $\delta'$ such that

(i)$'$ $\mathrm{cl}(U \setminus U_0)^{\delta'} \subseteq \mathrm{int}(|\mathcal{B}_1^{\mathcal{R}'} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}'}|)$;

(ii)$'$ for all $i = 1, \ldots, m$ and for all $\boldsymbol{\tau} \in \mathbf{R}^n$ of norm less than $\delta'$, $(\mathcal{R}'_i + \boldsymbol{\tau}) \sqcup \mathcal{F} \pitchfork U_i$; and

(iii)$'$ for all $i = 1, \ldots, m$ and for all $\boldsymbol{\tau} \in \mathbf{R}^n$ of norm less than $\delta'$, and for each $n$-dimensional box $B \in \mathcal{B}_i^{\mathcal{R}'}(\boldsymbol{\tau})$, there exists a point $\vec{p} \in \mathrm{int}(|B| \cap U_i)$ such that $\gamma_{\mathrm{cone},A}(\vec{p}) > \mathrm{diam}(B)$.

The construction consists of two steps:

*Step* 1. Cover the part of $U_0$ which may become uncovered by translations of the box collection $\mathcal{R}' + \boldsymbol{\tau}$, for $\|\boldsymbol{\tau}\| < \delta'$, with a box covering of a certain size. This size is determined by the uniform cone radius of the part of $U_0$ possibly uncovered by the translates of $\mathcal{R}'$.

*Step* 2. Some of the boxes in the above box covering might be in a degenerate position and in this way prevent the box collection from satisfying the required properties. This can be easily resolved, however, by translating all boxes with a small translation vector $\boldsymbol{\tau}$. Lemma 6.3 shows that it is possible to bring all boxes into general position; Lemma 6.4 shows that translating the boxes indeed results in a box collection with the desired properties.

We describe the two steps now in more detail. An example of the construction can be seen in Figure 6.2 and is described in the following example.

*Example* 6.2. We consider the case that no fixed box collection $\mathcal{F}$ is present. Let $\{A_0, A_1\}$ be the uniform cone radius decomposition of $\mathrm{cl}(A)$ (see Figure 6.2(a)). The set $A_1$ consists of the horizontal circle and point $\vec{p}$ in Figure 6.2(a). The set $A_0$ is equal to the remainder $\mathrm{cl}(A) \setminus A_1$.

1. *Base case* (not shown in Figure 6.2): $U = \emptyset$, $U_0 = \emptyset$. By definition, $\mathcal{R}_{-1} = \{\emptyset\}$, $\delta = \infty$.
2. *Case* $m = 1$, $U = A_1$, $U_0 = A_1$.
   *Covering* $U_0$ : Since in Step 1 nothing is yet constructed, we have that $V = U_0$, $W = \emptyset$, and $\zeta = \infty$. Hence, $\mathcal{R}'' = \frac{\varepsilon_V}{4\sqrt{3}}$-cover$(V)$. This box covering is depicted by the dashed boxes in Figure 6.2(a). By definition, $\delta'' = \min\{\frac{\delta'}{3} = \infty, \eta, \zeta = \infty\} = \eta$, where $\eta$ is such that $\mathrm{cl}(V)^{\eta} \subseteq \mathrm{int}(|\mathcal{R}''|)$.
   *Translating* $\mathcal{R}''$ : As can be seen in Figures 6.2(a), (b), the point $\vec{p}$ lies on a side of one of the boxes at the bottom. In other words, $\vec{p}$ is not in general position with the box collection. A simple small translation, however, resolves this situation and brings $\vec{p}$ into general position with the box collection (see Figure 6.2(b)) while keeping the other points $U_0$ in general position as well. The resulting box collection is denoted by $\mathcal{R}$.
   From $\mathcal{R}$ we get $\mathcal{B}_0^{\mathcal{R}}$, as shown in Figure 6.2(c), by removing, in this case, a single box which no longer intersect $U_0$.
3. *Case* $m = 2$, $U = A_0 \cup A_1$, $U_0 = A_0$, $\mathcal{R}''_1 = \mathcal{R}$, *and* $\delta' = \delta$ (*obtained in Step* 2).
   *Covering* $U_0$ : We focus on a region around the box $B$ in $\mathcal{R}''_1$ containing $\vec{p}$ (See Figure 6.2(d)). For expository reasons, the position of $U$ with respect to $B$ is slightly simplified.
   We have depicted the set $V$ (dark shaded area) of points in $U_0$, which might be outside $|B|$ when $B$ is slightly translated, and show the remaining set $W$

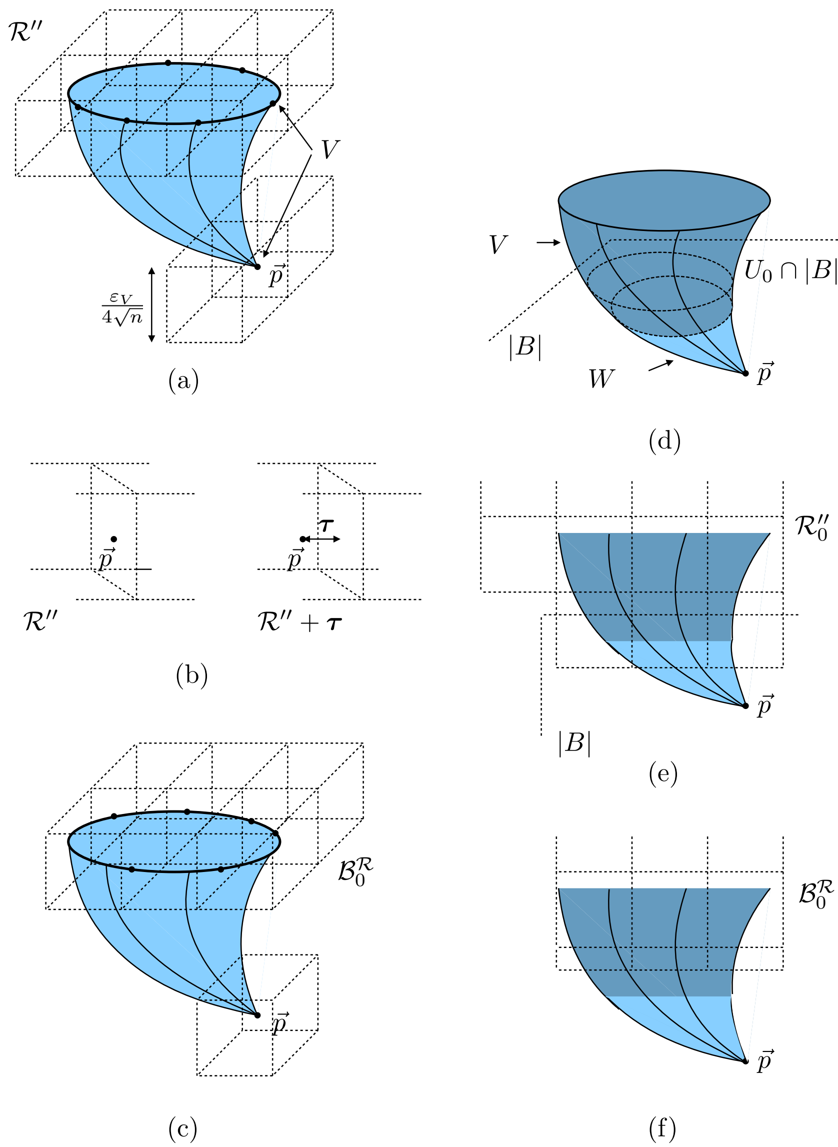


FIG. 6.2. *Construction of the special box collection $\mathcal{R}$.*

(light shaded area) as well. The new box collection $\mathcal{R}''_0$ will be $\frac{\varepsilon_V}{4\sqrt{3}}$-cover$(V)$. In order to not overload the figure, we have depicted the box collection from a sideways point of view (See Figure 6.2(e)). Let $\mathcal{R}'' = \{\mathcal{R}''_0, \mathcal{R}''_1\}$.

The constraint $\delta''$ on the norm of translation vectors is given by $\delta'' = \min\{\frac{\delta'}{3}, \eta, \zeta\}$. It takes into account the distance between $W$ and the boundary of the boxes constructed in Step 2 ($\zeta$), the distance between $V$ and the boundary of boxes in $\mathcal{R}''_0$ ($\eta$), and the constraint given in Step 2 ($\delta'$).

*Translating* $\mathcal{R}''$ : If necessary, slightly translate $\mathcal{R}''$ to bring it in general position such that it satisfies the desired properties. This results in the final box collection $\mathcal{R}$.

We also show part of $\mathcal{B}_0^{\mathcal{R}}$ (See Figure 6.2(f)). We refer to Example 6.1 for a discussion of its construction. The collection $\mathcal{B}_1^{\mathcal{R}}$ is equal to $\mathcal{B}_0^{\mathcal{R}}$ constructed in Step 2. $\diamond$

We now continue with the general description of the construction.

*First step: Covering* $U_0$. We will define a set $\mathcal{R}_0''$ and define $\mathcal{R}_i'' = \mathcal{R}_i'$ for $i = 1, \dots, m$ such that for $\mathcal{R}'' = \{\mathcal{R}_0'', \dots, \mathcal{R}_m''\}$, $\mathrm{cl}(U)^{\delta''} \subseteq \mathrm{int}(|\mathcal{B}_0^{\mathcal{R}''}(\boldsymbol{\tau}) \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}''}(\boldsymbol{\tau})|)$ for some $\delta'' > 0$.

All points of $U_0$ that can become uncovered by varying the vector $\boldsymbol{\tau}$ in $|\mathcal{B}_1^{\mathcal{R}'}(\boldsymbol{\tau}) \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}'}(\boldsymbol{\tau})|$ with $\|\boldsymbol{\tau}\| < \frac{\delta'}{3}$ are included in the set

$$V := U_0 - (|\mathcal{B}_1^{\mathcal{R}'} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}'}| - (\partial|\mathcal{B}_1^{\mathcal{R}'} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}'}|)^{\frac{\delta'}{3}}).$$

By (i)$'$, the minimal distance from any point in $U \setminus U_0$ to the boundary $\partial(|\mathcal{B}_1^{\mathcal{R}'} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}'}|)$ is greater than or equal to $\delta'$. This implies that

$$\mathrm{cl}(U \setminus U_0)^{\frac{\delta'}{3}} \subseteq |\mathcal{B}_1^{\mathcal{R}'} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}'}| - (\partial|\mathcal{B}_1^{\mathcal{R}'} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}'}|)^{\frac{\delta'}{3}},$$

and hence, because $U_0, \dots, U_m$ is a uniform cone radius collection, there exists a uniform cone radius, $\varepsilon_V$, of $A$ for the set $V$. Let $\mathcal{R}_0''$ be $\frac{\varepsilon_V}{4\sqrt{n}}$-cover$(V)$. Note that

$$\mathrm{diam}(B) = \frac{\varepsilon_V}{2} \tag{6.1}$$

for any box $B \in \mathcal{R}_0''$. The reason why we take this specific box covering is that the box collection, which we are constructing, must satisfy property (iii).

We now show that there exists a positive real number $\delta''$ such that (i) holds for $\mathcal{R}'' = \{\mathcal{R}_0'', \dots, \mathcal{R}_m''\}$ and $\delta''$.

We partition $U_0 \cup \cdots \cup U_m$ into three parts: $U \setminus U_0$, $V$, and

$$W := U_0 \cap (|\mathcal{B}_0^{\mathcal{R}''} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}''}| - (\partial|\mathcal{B}_0^{\mathcal{R}''} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}''}|)^{\frac{\delta'}{3}}).$$

By (i)$'$,

$$\mathrm{cl}(U \setminus U_0)^{\frac{\delta'}{3}} \subseteq \mathrm{int}(|\mathcal{B}_1^{\mathcal{R}'} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}'}|) \subseteq \mathrm{int}(|\mathcal{B}_0^{\mathcal{R}''} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}''}|). \tag{6.2}$$

We shall need the following lemma, which is readily verified.

LEMMA 6.1. *Let $X$ and $Y$ be two sets in $\mathbf{R}^n$. If $X$ is bounded, then* $\mathrm{cl}(X) \subseteq \mathrm{int}(Y)$ *implies that there exists a positive real number $\varepsilon$ such that* $\mathrm{cl}(X)^{\varepsilon} \subseteq \mathrm{int}(Y)$.

By the definition of a box covering, $\mathrm{cl}(V) \subseteq \mathrm{int}(|\mathcal{B}_0^{\mathcal{R}''}|) \subseteq \mathrm{int}(|\mathcal{B}_0^{\mathcal{R}''} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}''}|)$. Since $A$ is bounded, $V$ is also bounded. By Lemma 6.1, there exists a positive real number $\eta$ such that

$$\mathrm{cl}(V)^{\eta} \subset \mathrm{int}(|\mathcal{B}_0^{\mathcal{R}''} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}''}|). \tag{6.3}$$

We now prove that Lemma 6.1 can also be used for $W$.

LEMMA 6.2. $\mathrm{cl}(W) \subseteq \mathrm{int}(|\mathcal{B}_1^{\mathcal{R}'} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}'}|)$.

*Proof of Lemma* 6.2. Suppose that there exists a point $\vec{p} \in \mathrm{cl}(W)$ such that $\vec{p} \notin \mathrm{int}(|\mathcal{B}_1^{\mathcal{R}'} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}'}|)$. Let $(\vec{p}_m)$ for $m > 0$ be a sequence of points in $W$ such that $\|\vec{p} - \vec{p}_m\| < 1/m$. By the definition of $W$, for all points in $\vec{r} \in \partial|\mathcal{B}_1^{\mathcal{R}'} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}'}|$, $\|\vec{r} - \vec{p}_m\| \geqslant \frac{\delta'}{3}$ for every $m$.

Now, every line segment $\{\lambda\vec{p}_m + (1-\lambda)\vec{p} \mid 0 \leqslant \lambda \leqslant 1\}$, intersects $\partial|\mathcal{B}_1^{\mathcal{R}'} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}'}|$ in a point $\vec{r}_m$. However, since $\|\vec{p}_m - \vec{p}\| < 1/m$, also $\|\vec{p}_m - \vec{r}_m\| < 1/m$. Thus, we obtain a contradiction for $m$ large enough such that $\frac{1}{m} < \frac{\delta'}{3}$. $\square$

Hence, by Lemma 6.1 and Lemma 6.2 there exists a positive real number $\zeta$ such that

$$W^{\zeta} \subseteq \mathrm{int}(|\mathcal{B}_1^{\mathcal{R}'} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}'}|) \subseteq \mathrm{int}(|\mathcal{B}_0^{\mathcal{R}''} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}''}|). \tag{6.4}$$

From the inclusions (6.2), (6.3), and (6.4), it follows that property (i) is satisfied for $\mathcal{R}''$ and $\delta''$, with $\delta'' = \min\{\frac{\delta'}{3}, \eta, \zeta\}$.

*Second step: translating $\mathcal{R}''$.* The box collections in $\mathcal{R}''$ already satisfy property (i) for $\delta''$. However, properties (ii) and (iii) are not necessarily satisfied. This can be seen in Figure 6.2 (a), (b). We now show that a little translation of the box collection is all that is needed so that all properties are satisfied by the translated box collections.

LEMMA 6.3. *For each $i = 0, \dots, m$, there exists a translation $\boldsymbol{\tau} \in \mathbf{R}^n$ of norm $\|\boldsymbol{\tau}\| < \delta''$ such that*

$$(\mathcal{R}''_i + \boldsymbol{\tau}) \sqcup \mathcal{F} \pitchfork U_i.$$

*Proof of Lemma* 6.3. Consider the decomposition of $|(\mathcal{R}''_i + \boldsymbol{\tau}) \sqcup \mathcal{F}|$ into the sets $|(\mathcal{R}''_i + \boldsymbol{\tau}) \sqcup \mathcal{F}|_j$ for $i = 0, \dots, m$ and for $j = 0, \dots, n$. Recall from section 5.5 that $|(\mathcal{R}''_i + \boldsymbol{\tau}) \sqcup \mathcal{F}|_j$ is the union of the geometric realizations of boxes in $((\mathcal{R}''_0 + \boldsymbol{\tau}) \sqcup \mathcal{F})^j$.

We need to prove that there exists a translation $\boldsymbol{\tau} \in \mathbf{R}^n$, $\|\boldsymbol{\tau}\| < \delta''$, such that for each $i = 0, \dots, m$, for each $r \in \{0, \dots, n_i\}$, for each $j \in \{0, \dots, n\}$, and for each $B \in ((\mathcal{R}''_i + \boldsymbol{\tau}) \sqcup \mathcal{F})^j$, we have that

$$|B| \pitchfork R_{i,r}. \tag{6.5}$$

Let $T$ denote the set of all possible translations: $T := \{\boldsymbol{\tau} \in \mathbf{R}^n \mid \|\boldsymbol{\tau}\| < \delta''\}$. Note that case $i > 0$ of (6.5) holds for any $\boldsymbol{\tau} \in T$ by induction. Hence, we can focus on the case $i = 0$. Take an arbitrary $B$ as in (6.5), take $r$ arbitrary in $\{0, \dots, n\}$, and consider a point $\vec{x} \in |B| \cap R_{0,r}$. We are going to impose several conditions on $T$ such that if $\boldsymbol{\tau} \in T$ and $\boldsymbol{\tau}$ satisfies these conditions, then (6.5) holds for $\boldsymbol{\tau}$. By definition of the union operator $\sqcup$, there exists a neighborhood $W$ of $\vec{x}$ such that one of the following three cases holds:

1. $|B| \cap W = |B'| \cap W$ for some $B' \in \mathcal{F}^p$ for some $p$. Note that
$$\mathrm{T}_{\vec{x}}\,|B| = \mathrm{T}_{\vec{x}}(|B| \cap W) = \mathrm{T}_{\vec{x}}(|B'| \cap W) = \mathrm{T}_{\vec{x}}\,|B'|. \tag{6.6}$$
Given that $\mathcal{F} \pitchfork U_0$, $|B'|$ and $R_{0,r}$ are transversal in $\vec{x}$ for all $\boldsymbol{\tau} \in T$. By (6.6), we may conclude that $|B|$ and $R_{0,r}$ are transversal in $\vec{x}$ for all $\boldsymbol{\tau} \in T$.
2. $|B| \cap W = |B''| \cap W$ for some $B'' \in (\mathcal{R}''_0 + \boldsymbol{\tau})^q$ for some $q$. Note that
$$\mathrm{T}_{\vec{x}}\,|B| = \mathrm{T}_{\vec{x}}(|B| \cap W) = \mathrm{T}_{\vec{x}}(|B''| \cap W) = \mathrm{T}_{\vec{x}}\,|B''|. \tag{6.7}$$
Suppose that
$$(\mathcal{R}''_0 + \boldsymbol{\tau}) \pitchfork U_0. \tag{T1}$$
Then, $|B''| \pitchfork U_0$ and hence, $|B''|$ and $R_{0,r}$ are transversal in $\vec{x}$ for all $\boldsymbol{\tau} \in T$ such that condition (T1) is satisfied. By (6.7), we may conclude that $|B|$ and $R_{0,r}$ are transversal in $\vec{x}$ for all $\boldsymbol{\tau} \in T$ such that condition (T1) is satisfied.
3. $|B| \cap W = |B'| \cap |B''| \cap W$ for some $B' \in \mathcal{F}^p$ for some $p$, and for some $B'' \in (\mathcal{R}''_0 + \boldsymbol{\tau})^q$ for some $q$. Suppose that
$$(\mathcal{R}''_0 + \boldsymbol{\tau}) \pitchfork \mathcal{F}. \tag{T2}$$
Because the intersection of regular sets in general position is regular, the tangent space $\mathrm{T}_{\vec{x}}(|B'| \cap |B''|)$ exists. Note that
$$\mathrm{T}_{\vec{x}}\,|B| = \mathrm{T}_{\vec{x}}(|B| \cap W) = \mathrm{T}_{\vec{x}}(|B'| \cap |B''| \cap W) = \mathrm{T}_{\vec{x}}(|B'| \cap |B''|). \tag{6.8}$$

Furthermore, suppose that

$$\text{(T3)} \qquad |B''| \pitchfork (|B'| \cap R_{0,r}).$$

When two regular sets intersect transversally in a point, the tangent space of the intersection in this point is the intersection of the tangent spaces of the regular sets in this point [23]. Hence, by (T2) and given that $\mathcal{F} \pitchfork U_0$, we have that $\mathrm{T}_{\vec{x}}\,|B'| \cap \mathrm{T}_{\vec{x}}\,|B''| = \mathrm{T}_{\vec{x}}(|B'| \cap |B''|)$ and $\mathrm{T}_{\vec{x}}\,|B'| \cap \mathrm{T}_{\vec{x}}(R_{0,r}) = \mathrm{T}_{\vec{x}}(|B'| \cap R_{0,r})$. Moreover, $\mathrm{T}_{\vec{x}}(|B'| \cap R_{0,r}) \subseteq \mathrm{T}_{\vec{x}}(R_{0,r})$. By (T3) we have that $\mathrm{T}_{\vec{x}}(|B'| \cap |B''|) + \mathrm{T}_{\vec{x}}(|B'| \cap R_{0,r}) = \mathrm{T}_{\vec{x}}\,|B'|$. Hence,

$$\begin{aligned}
\mathrm{T}_{\vec{x}}\,|B| + \mathrm{T}_{\vec{x}}(R_{0,r}) &= \mathrm{T}_{\vec{x}}(|B'| \cap |B''|) + \mathrm{T}_{\vec{x}}(R_{0,r}) \\
&= \mathrm{T}_{\vec{x}}(|B'| \cap |B''|) + \mathrm{T}_{\vec{x}}(|B'| \cap R_{0,r}) + \mathrm{T}_{\vec{x}}(R_{0,r}) \\
&= \mathrm{T}_{\vec{x}}(|B'|) + \mathrm{T}_{\vec{x}}(R_{0,r}) \\
&= \mathbf{R}^n.
\end{aligned}$$

Hence, we may conclude that $|B|$ and $R_{0,r}$ are transversal in $\vec{x}$ for all $\boldsymbol{\tau} \in T$ such that conditions (T2) and (T3) are satisfied.

We may conclude that $|(\mathcal{R}''_0 + \boldsymbol{\tau}) \sqcup \mathcal{F}| \pitchfork U_0$ if $\boldsymbol{\tau} \in T$ and if $\boldsymbol{\tau}$ is such that, for each box $B \in ((\mathcal{R}''_0 + \boldsymbol{\tau}) \sqcup \mathcal{F})^j$ for $j = 0, \ldots, n$, either no extra condition holds, the condition (T1) holds, or both conditions (T2) and (T3) hold. Hence, we obtain a finite number of conditions on the translations in $T$. By Corollary 5.10, the set of translations $\boldsymbol{\tau} \in T$ for which a single transversality condition, like (T1), (T2), or (T3), is not satisfied, has measure zero. Since a finite union of sets of measure zero also has measure zero, this implies that for almost all translations in $T$, all conditions can be satisfied simultaneously. This concludes the proof of the lemma. □

Let $\boldsymbol{\tau}_0$ be a translation, as specified in Lemma 6.3. We now define for $i = 0, \ldots, m$, $\mathcal{R}_i = \mathcal{R}''_i + \boldsymbol{\tau}_0$ and consider $\mathcal{R} = \{\mathcal{R}_0, \ldots, \mathcal{R}_m\}$ and $\delta''' < \delta'' - \|\boldsymbol{\tau}_0\|$.

LEMMA 6.4. *There exists a $\delta > 0$ such that $\mathcal{R}_0, \ldots, \mathcal{R}_m$ and $\delta$ satisfy properties* (i), (ii)*, and* (iii).

*Proof of Lemma* 6.4. We first prove that there exists a $\delta > 0$ such that property (ii) is satisfied. Indeed, the proof of Lemma 6.3 shows that for $i = 0, \ldots, m$, $(\mathcal{R}''_i + \boldsymbol{\tau}) \sqcup \mathcal{F} \pitchfork U_i$ holds for any $\boldsymbol{\tau}$ which satisfies a finite number of transversality conditions. Recall from section 5.4 that transversality is a stable property. Hence, if $\boldsymbol{\tau}$ is a translation vector satisfying these transversality conditions, then there exists an $\varepsilon > 0$ such that any $\boldsymbol{\tau'} \in \mathbf{R}^n$, for which $\|\boldsymbol{\tau'} - \boldsymbol{\tau}\| < \varepsilon$, also satisfies these transversality conditions.

Since $\mathcal{R}_i = \mathcal{R}''_i + \boldsymbol{\tau}_0$, and $\boldsymbol{\tau}_0$ is such that Lemma 6.3 holds, there exists a $\varepsilon > 0$ such that for $\boldsymbol{\tau} \in \mathbf{R}^n$, $\|\boldsymbol{\tau}\| < \varepsilon$,

$$(\mathcal{R}_i + \boldsymbol{\tau}) \sqcup \mathcal{F} \pitchfork U_i$$

for $i = 0, \ldots, m$. Hence, property (ii) is satisfied for $\mathcal{R}_0, \ldots, \mathcal{R}_m$ and $\delta = \min\{\delta''', \varepsilon\}$.

We now prove that $\mathcal{R}_0, \ldots, \mathcal{R}_m$ and $\delta$ also satisfy property (i). We will need the following properties which can be readily verified. Let $X$ and $Y$ be semialgebraic sets in $\mathbf{R}^n$. Then

(1) $X^\varepsilon \subseteq Y \Rightarrow X \subseteq Y + \boldsymbol{\tau}$ for any $\boldsymbol{\tau} \in \mathbf{R}^n$ such that $\|\boldsymbol{\tau}\| < \varepsilon$, and

(2) $(X^{\varepsilon_1})^{\varepsilon_2} = X^{\varepsilon_1 + \varepsilon_2}$.

We already know $\mathrm{cl}(U)^{\delta''} \subseteq \mathrm{int}(|\mathcal{B}_0^{\mathcal{R}''} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}''}|)$. Let $\varepsilon = \delta'' - \|\boldsymbol{\tau}_0\| - \delta$. Since $\delta < \delta'' - \|\boldsymbol{\tau}_0\|$, we have $\varepsilon > 0$, and by property (2),

$$\mathrm{cl}(U)^{\delta''} = (\mathrm{cl}(U)^{\delta})^{\|\boldsymbol{\tau}_0\| + (\delta'' - \|\boldsymbol{\tau}_0\| - \delta)} \subseteq \mathrm{int}(|\mathcal{B}_0^{\mathcal{R}''} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}''}|).$$

By property (1), we have that

$$\mathrm{cl}(U)^\delta \subseteq \mathrm{int}(|\mathcal{B}_0^{\mathcal{R}''} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}''}|) + \boldsymbol{\tau} \qquad \forall \boldsymbol{\tau} : \|\boldsymbol{\tau}\| < \|\boldsymbol{\tau}_0\| + \varepsilon.$$

In particular, $\mathrm{cl}(U)^\delta \subseteq \mathrm{int}(|\mathcal{B}_0^{\mathcal{R}''} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}''}|) + \boldsymbol{\tau}_0 = \mathrm{int}(|\mathcal{B}_0^{\mathcal{R}} \cup \cdots \cup \mathcal{B}_m^{\mathcal{R}}|)$, and property (i) is satisfied for $\mathcal{R}$ and $\delta$.

We now prove that property (iii) is satisfied. Let $B \in \mathcal{B}_i^{\mathcal{R}}(\boldsymbol{\tau})$ for any $\boldsymbol{\tau} \in \mathbf{R}^n$, $\|\boldsymbol{\tau}\| < \delta$. We distinguish between the following two cases:

1. $i > 0$. Since $\mathcal{B}_i^{\mathcal{R}}(\boldsymbol{\tau}) \subseteq \mathcal{B}_i^{\mathcal{R}'}(\boldsymbol{\tau}_0 + \boldsymbol{\tau})$ and $\|\boldsymbol{\tau} - \boldsymbol{\tau}_0\| < \delta'$, we have by induction that there exists a $\vec{p} \in \mathrm{int}(|B|) \cap U_i$ such that $\gamma_{\mathrm{cone},A}(\vec{p}) > \mathrm{diam}(B)$.
2. $i = 0$. Since $|B| \cap U_0 \neq \emptyset$, we need to prove that there exists a $\vec{p} \in \mathrm{int}(|B|) \cap U_0$ such that $\gamma_{\mathrm{cone},A}(\vec{p}) > \mathrm{diam}(B)$.

   Thus, let $\vec{x} \in |B| \cap U_0$. If $\vec{x} \in \mathrm{int}(|B|)$, we are done. If $\vec{x} \in \partial|B|$, then $\vec{x} \in |B'| \cap U_0$ for some $|B'| \in (((\mathcal{R}_0 \sqcup \cdots \sqcup \mathcal{R}_m) + \boldsymbol{\tau}) \sqcup \mathcal{F})^p$ and some $p$. Let $D = (x_1 - \varepsilon, x_1 + \varepsilon, \ldots, x_n - \varepsilon, x_n + \varepsilon)$ be an $n$-dimensional box centered around $\vec{x}$, with $\varepsilon \in \mathbf{R}$. For $\varepsilon$ sufficiently small, $|B'| \cap \mathrm{int}(|D|)$ has the form

   $$(x_1 - \varepsilon, x_1 - \varepsilon) \times \cdots \times (x_p - \varepsilon, x_p + \varepsilon) \times \{x_{p+1}\} \times \cdots \times \{x_n\},$$

   or a permutation of this form, which is handled analogously. Hence, $\mathrm{int}(|B|) \cap \mathrm{int}(|D|)$ has the form

   $$(x_1 - \varepsilon, x_1 - \varepsilon) \times \cdots \times (x_p - \varepsilon, x_p + \varepsilon) \times (x_{p+1}, x_{p+1} + \varepsilon) \times \cdots \times (x_n, x_n + \varepsilon),$$

   or a permutation of this form, which is handled analogously, or even a variant of this form, where some of the $n - p$ intervals $(x_i, x_i + \varepsilon)$ are replaced by $(x_i - \varepsilon, x_i)$, which again is handled analogously.

   By property (ii),

   $$\mathrm{T}_{\vec{x}}\,|B'| + \mathrm{T}_{\vec{x}}\,U_0 = \mathbf{R}^n. \tag{6.9}$$

   Now, any $\vec{v} \in \mathrm{T}_{\vec{x}}\,|B'|$ is of the form $\vec{v} = (v_1, \ldots, v_p, x_{p+1}, \ldots, x_n)$; hence, by (6.9) there exists a tangent vector $\vec{w} \in \mathrm{T}_{\vec{x}}\,U_0$ such that $x_{p+1} < w_{p+1}, \ldots, x_n < w_n$. By definition of the tangent space, if $\|\vec{w} - \vec{x}\|$ is small enough, there exists a point $\vec{q}$ in $U_0$ arbitrarily close to $\vec{w}$. This point $\vec{q}$ is also arbitrarily close to $\vec{x}$ and also has $n - p$ last coordinates, which are strictly greater than the $n - p$ last coordinates of $\vec{x}$. Hence, $\vec{q}$ is in $\mathrm{int}(|B|) \cap \mathrm{int}(|D|)$, and we have found a point in $\mathrm{int}(|B|) \cap U_0$.

   We now show that for any $\vec{p} \in \mathrm{int}(|B|) \cap U_0$, $\gamma_{\mathrm{cone},A}(\vec{p}) > \mathrm{diam}(B)$. Indeed, any box in $\mathcal{B}_0^{\mathcal{R}}(\boldsymbol{\tau})$ is included in a box in $\mathcal{R}_0'' + \boldsymbol{\tau}_0 + \boldsymbol{\tau}$. By (6.1), $\mathcal{R}_0''$ consists of boxes which have a diameter that is strictly smaller than the uniform cone radius of $\mathrm{int}(|B|) \cap U_0$. Hence, $\gamma_{\mathrm{cone},A}(\vec{p}) > \mathrm{diam}(B)$ for any point $\vec{p} \in \mathrm{int}(|B|) \cap U_0$.

As a result, property (iii) is satisfied for $\mathcal{R}$ and $\delta$. ◻

This concludes the construction of the box collection $\mathcal{R}$ and $\delta > 0$.

**6.2. A first glance at the linearization algorithm.** In this section we describe how the special box collection $\mathcal{R}$, constructed in the previous section, helps us in achieving our goal of linearizing a semialgebraic set $A \subseteq \mathbf{R}^n$.

First, using the box collection $\mathcal{R}$, we define

$$\mathcal{U} = \mathcal{B}_0^{\mathcal{R}} \cup \cdots \cup \mathcal{B}_n^{\mathcal{R}}. \tag{6.10}$$

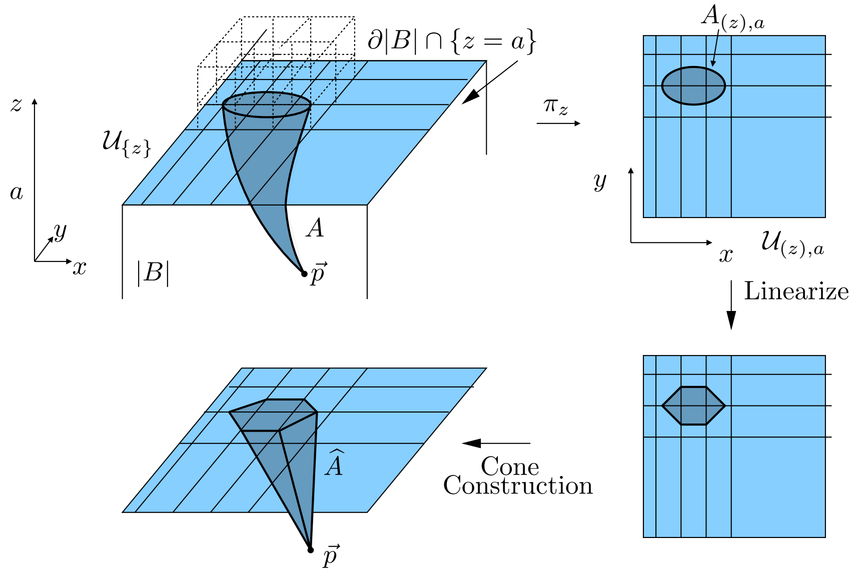


FIG. 6.3. *Illustration of the linearization $\widehat{A}$ inside $|B|$. The top side of $\partial|B|$ is shown together with that part of $\mathcal{U}_{(z)}$ and $A$ lying on it. The top side has $z$-coordinate $a$ (top left). The two-dimensional projected sets $\mathcal{U}_{(z),a}$ and $A_{(z),a}$ are shown (top right). The linearization algorithm is called inductively on these lower-dimensional sets (bottom right). The three-dimensional linearization consists of building a cone with top $\vec{p}$ and base the previously constructed linearization on the boundary of $B$ (bottom left).*

Recall that $\mathcal{B}_i^{\mathcal{R}}$ stands for $\mathcal{B}_i^{\mathcal{R}}(\mathbf{0})$. Since each $\mathcal{B}_i^{\mathcal{R}}$ is a box collection and $\mathrm{int}(|\mathcal{B}_i^{\mathcal{R}}|) \cap \mathrm{int}(|\mathcal{B}_j^{\mathcal{R}}|) = \emptyset$ for any $i \neq j$, $\mathcal{U}$ is a box collection too. It is clear that $\mathcal{U}$ inherits some of the properties of $\mathcal{R}$. Indeed, by property (i) of $\mathcal{R}$, we know that $\mathcal{U}$ is a box covering of $\mathrm{cl}(A)$ and, by property (iii) of $\mathcal{R}$, we know that for each box $B \in \mathcal{U}$ there exists a point $\vec{p} \in \mathrm{int}(|B|) \cap A$ such that $\gamma_{\mathrm{cone},A}(\vec{p}) > \mathrm{diam}(B)$.

The linearization algorithm, which will be described in more detail in section 6.3, works inductively on the boundaries of the boxes in $\mathcal{U}$. For each box $B \in \mathcal{U}$, the linearization algorithm replaces $|B| \cap A$ with a semilinear set in two steps. In the induction step, it replaces the intersection $\partial|B| \cap A$ with a semilinear set $\partial\widehat{|B| \cap A}$ on $\partial|B|$, which is homeomorphic to $\partial|B| \cap A$. Then, for each box $B \in \mathcal{U}$, it replaces $|B| \cap A$ with the semilinear set

$$\mathrm{Cone}(\partial\widehat{|B| \cap A}, \vec{p}),$$

where $\vec{p} \in \mathrm{int}(|B|) \cap A$ such that $\gamma_{\mathrm{cone},A}(\vec{p}) > \mathrm{diam}(B)$. It is shown in Lemma 6.5 that in this way we end up with a linearization of $A$. An illustration of the linearization algorithm is given in Figure 6.3.

In order to construct the linearization $\partial\widehat{|B| \cap A}$ on $\partial|B|$ of boxes $B \in \mathcal{U}$, we will need to construct again a box collection $\mathcal{R}$, but this time on the boundaries of the boxes in $\mathcal{U}$.

We will decompose the boundaries of the boxes in $\mathcal{U}$ according to the direction of their supporting hyperplanes and according to the coordinate value of the fixed coordinate of these hyperplanes.

These coordinates can be computed as

$$\begin{aligned}\mathrm{Coord}(\mathcal{U}_{\{i\}}) = \{a \in \mathbf{R} \mid & \exists a_1, \exists b_1, \ldots, \exists a_{i-1}, \exists b_{i-1}, \exists a_{i+1}, \exists b_{i+1}, \ldots, \exists a_n, \exists b_n \\ & (a_1, b_1, \ldots, a_{i-1}, b_{i-1}, a, a, a_{i+1}, b_{i+1}, \ldots, a_n, b_n) \in \mathcal{U}_{\{i\}}\}\end{aligned}$$

for $i = 1, \ldots, n$ and where $\mathcal{U}_{\{i\}}$ are the $n$-dimensional box collections defined in (5.9). Recall that $\mathcal{U}_{\{i\}}$ contains all $n$-dimensional boxes on the boundaries of boxes in $\mathcal{U}$, whose $i$th coordinates are all equal.

For each $a \in \mathrm{Coord}(\mathcal{U}_{\{i\}})$, we will need all the points in $\mathrm{cl}(A)$ with the $i$th coordinated fixed to $a$, i.e.,

$$\begin{aligned}\mathrm{cl}(A)_{(i),a} := \{(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \in \mathbf{R}^{n-1} \mid \\ (x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_n) \in \mathrm{cl}(A)\}\end{aligned}$$

for $i = 1, \ldots, n$.

Similarly, we define the $(n-1)$-dimensional box collections

$$\begin{aligned}\mathcal{U}_{(i),a} := \{(a_1, b_1, \ldots, b_{i-1}, a_{i+1}, \ldots, a_n, b_n) \in \mathbf{R}^{2(n-1)} \mid \\ (a_1, b_1, \ldots, b_{i-1}, a, a, a_{i+1}, \ldots, a_n, b_n) \in \mathcal{U}_{\{i\}}\}\end{aligned}$$

for $i = 1, \ldots, n$.

Since $\mathrm{cl}(A) = C_0 \cup \cdots \cup C_m$, and $C_m = R_{m,n} \cup \cdots \cup R_{m,0}$, we have that

$$\begin{aligned}\mathrm{cl}(A)_{(i),a} &= (C_0)_{(i),a} \cup \cdots \cup (C_m)_{(i),a}, \\ (C_j)_{(i),a} &= (R_{j,n})_{(i),a} \cup \cdots \cup (R_{j,0})_{(i),a}.\end{aligned}$$

For each $i = 0, \ldots, n$ and each $a \in \mathrm{Coord}(\mathcal{U}_{\{i\}})$, we now show that we can construct an $(n-1)$-dimensional box collection $\mathcal{R}$, as described in section 6.1, for $\mathrm{cl}(A)_{(i),a}$ in the role of $\mathrm{cl}(A)$, $(C_0)_{(i),a}, \ldots, (C_m)_{(i),a}$ in the role of, respectively, $U_0, \ldots, U_m$, and $\mathcal{U}_{(i),a}$ in the role of $\mathcal{F}$.

However, for the construction to be successful, we need to verify that we start with valid input data. In other words, we need to show that $(C_0)_{(i),a}$ is a uniform cone radius with a regular decomposition given by $(R_{j,n})_{(i),a}$ and that $\mathcal{F}$ (which is $\mathcal{U}_{(i),a}$) is in general position with $(C_0)_{(i),a}$ for the regular decomposition $(R_{j,n})_{(i),a}$.

CLAIM 6.1. *The sets* $(C_0)_{(i),a}, \ldots, (C_m)_{(i),a}$ *form a uniform cone radius decomposition of* $\mathrm{cl}(A)_{(i),a}$.

*Proof.* By definition, the sets $(C_0)_{(i),a}, \ldots, (C_m)_{(i),a}$ form a decomposition of $\mathrm{cl}(A)_{(i),a}$, so we need only show that each of the sets $(C_0)_{(i),a}, \ldots, (C_m)_{(i),a}$ form a uniform cone radius collection.

We will need the following property, which is readily verified. Let $X$ and $Y$ be semialgebraic sets in $\mathbf{R}^n$. Then

(1) if $Y$ is closed and bounded, then for all $\varepsilon'$ there exists an $\varepsilon$ such that $X^\varepsilon \cap Y \subseteq (X \cap Y)^{\varepsilon'}$.

Let $H_{(i),a} = \{\vec{x} \in \mathbf{R}^n \mid x_i = a\}$, and let $\pi_i : \mathbf{R}^n \to \mathbf{R}^{n-1}$ be defined by $\pi_i(x_1, \ldots, x_n) = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$. Let $j \in \{0, \ldots, m\}$, and let $\varepsilon'_0, \ldots, \varepsilon'_m$ be positive real numbers. We have that

$$(C_j)_{(i),a} \setminus \bigcup_{k=j+1}^{m} ((C_k)_{(i),a})^{\varepsilon'_k} = \pi_i\left((C_j \cap H_{(i),a}) \setminus \bigcup_{k=j+1}^{m} (C_k \cap H_{(i),a})^{\varepsilon'_k}\right).$$

By property (1), there exist $\varepsilon_0 > 0, \ldots, \varepsilon_m > 0$ such that

$$(C_j \cap H_{(i),a}) \setminus \bigcup_{k=j+1}^{m} (C_k \cap H_{(i),a})^{\varepsilon'_k} \subseteq (C_k \cap H_{(i),a}) \setminus \bigcup_{k=j+1}^{m} (C_k^{\varepsilon_k} \cap H_{(i),a}) = \left( C_j \setminus \bigcup_{k=j+1}^{m} C_k^{\varepsilon_k} \right) \cap H_{(i),a}. \tag{6.11}$$

Moreover, we have that $\mathrm{cl}(A) = C_0 \cup \cdots \cup C_m$ and, since $C_0, \ldots, C_m$ is a uniform cone radius collection, from the inclusion (6.11), it follows that

$$0 < \inf \left\{ \gamma_{\mathrm{cone},A} \left( \left( C_j \setminus \bigcup_{k=j+1}^{m} C_k^{\varepsilon_k} \right) \cap H_{(i),a} \right) \right\} \leqslant \inf \left\{ \gamma_{\mathrm{cone},A} \left( (C_j \cap H_{(i),a}) \setminus \bigcup_{k=j+1}^{m} (C_k \cap H_{(i),a})^{\varepsilon'_k} \right) \right\}.$$

We will next show that the following inequality holds:

$$\inf \left\{ \gamma_{\mathrm{cone},A} \left( (C_j \cap H_{(i),a}) \setminus \bigcup_{k=j+1}^{m} (C_k \cap H_{(i),a})^{\varepsilon'_k} \right) \right\} \leqslant \inf \left\{ \gamma_{\mathrm{cone},A \cap H_{(i),a}} \left( (C_j \cap H_{(i),a}) \setminus \bigcup_{k=j+1}^{m} (C_k \cap H_{(i),a})^{\varepsilon'_k} \right) \right\} = \inf \left\{ \gamma_{\mathrm{cone},\pi_i(A \cap H_{(i),a})} \left( \pi_i \left( (C_j \cap H_{(i),a}) \setminus \bigcup_{k=j+1}^{m} (C_k \cap H_{(i),a})^{\varepsilon'_k} \right) \right) \right\}.$$

Hence,

$$0 < \inf \left\{ \gamma_{\mathrm{cone},A_{(i),a}} \left( (C_j)_{(i),a} \setminus \bigcup_{k=j+1}^{m} ((C_k)_{(i),a})^{\varepsilon'_k} \right) \right\},$$

which proves that $(C_0)_{(i),a}, \ldots, (C_m)_{(i),a}$ is a uniform cone radius collection.

We still need to prove that for each $\vec{x} \in C_j \cap H_{(i),a}$,

$$\gamma_{\mathrm{cone},A}(\vec{x}) \leqslant \gamma_{\mathrm{cone},A \cap H_{(i),a}}(\vec{x}).$$

The proof is illustrated in Figure 6.4. The main ingredient is the construction of the cone radius, as described in the proof of Theorem 2 in [14]. As explained in the paragraph immediately following Theorem 5.3, the radius query produces for each point $\vec{x}$ an interval $(0, r)$ of cone radii, where $r$ is the minimal distance between $\vec{x}$ and each $\vec{s} \in \mathcal{S} \subseteq \mathbf{R}^n$, where $\mathcal{S}$ contains those points $\vec{s}$ which have a tangent space that is orthogonal to $\vec{x} - \vec{s}$ or parallel to one of the axes-parallel hyperplanes. Here, the tangent spaces are taken with respect to a Whitney-decomposition $\mathcal{Z}$ of $A$, which is compatible with the union of all axes-parallel hyperplanes (including $H_{i,a}$) through $\vec{x}$. An example of such a Whitney-decomposition is given in Figure 6.4 (top right). Also in this figure, we have depicted the set $\mathcal{S}$. The (maximal) cone radius of $A$ in $(a, b)$ is illustrated by the dashed circle centered around $(a, b)$.

Recall that we defined

$$\gamma_{\mathrm{cone},A}(\vec{x}) = \frac{1}{2} r = \frac{1}{2} \min_{\vec{s} \in \mathcal{S}} d(\vec{x}, \vec{s}),$$

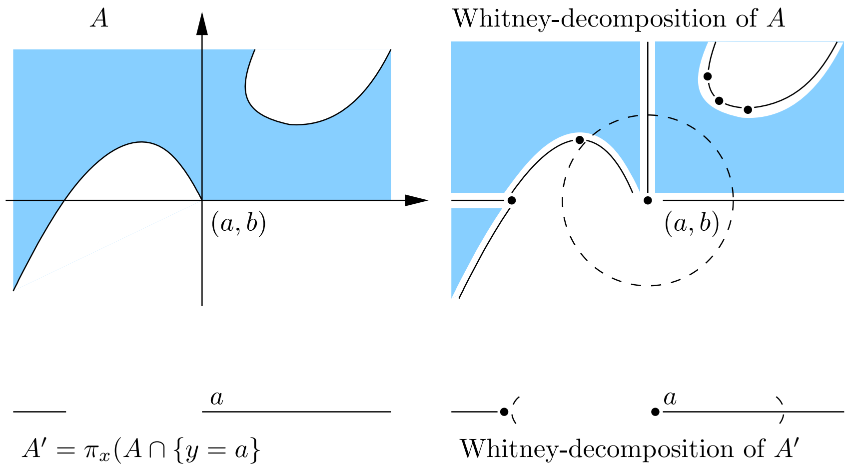


FIG. 6.4. *Semialgebraic set $A$ locally around $(a, b)$ (top left). Whitney-decomposition $\mathcal{Z}$ of $A$ compatible with axes-parallel hyperplanes through $(a, b)$ (top right). Intersection $A'$ of $A$ with horizontal hyperplane through $(a, b)$ and projected on the $x$-axis (bottom left). Whitney-decomposition $\mathcal{Z}'$ of $A'$ (bottom right). The isolated points (top and bottom right) denote the critical points, i.e., points $(c, d)$ with a horizontal or vertical tangent space, or a tangent space perpendicular to the vector $(c, d) - (a, b)$. Note that these tangent spaces are relative to the Whitney-decomposition. Moreover, by construction the set $\mathcal{S}$ of critical points for $A$ around $(a, b)$ shown as the isolated points (top right) includes the set $\mathcal{S}'$ of critical points of $A'$ around $a$ (bottom right). Consequently, $\gamma_{cone,A}(a, b) \leqslant \gamma_{cone,A'}(a)$.*

where $d$ denotes the ordinary distance function.

In the same way,

$$\gamma_{\text{cone},A\cap H_{(i),a}}(\vec{x}) = \frac{1}{2}\min_{\vec{s}\in\mathcal{S}'} d(\vec{x}, \vec{s}),$$

where $\mathcal{S}'$ contains those points $\vec{s}$ which have a tangent space that is orthogonal to $\vec{x} - \vec{s}$ or parallel to one of the axes-parallel hyperplanes. Here, the tangent spaces are taken with respect to a Whitney-decomposition $\mathcal{Z}'$ of $A \cap H_{(i),a}$. An example of such a Whitney-decomposition is given in Figure 6.4 (bottom right). Also in this figure we have depicted $\mathcal{S}'$. The (maximal) cone radius is illustrated by the interval bounded by the two dashed line segments centered around $a$.

Due to the requirement that $\mathcal{Z}$ is compatible with the axes-parallel hyperplanes through $\vec{x}$, the Whitney-decomposition $\mathcal{Z}'$ of $A \cap H_{(i),a}$ is equal to those strata $Z \in \mathcal{Z}$ such that $Z \subseteq H_{(i),a}$. In other words, $\mathcal{S}' \subseteq \mathcal{S}$, and hence,

$$\gamma_{\text{cone},A}(\vec{x}) = \frac{1}{2}\min_{\vec{s}\in\mathcal{S}} d(\vec{x}, \vec{s}) \leqslant \frac{1}{2}\min_{\vec{s}\in\mathcal{S}'} d(\vec{x}, \vec{s}) = \gamma_{\text{cone},A\cap H_{(i),a}}(\vec{x}),$$

as desired. ∎

CLAIM 6.2. *The sets $(R_{j,0})_{(i),a}, \ldots, (R_{j,n_j})_{(i),a}$ form a regular decomposition of $(C_j)_{(i),a}$.*

*Proof.* By definition, the sets $(R_{j,n})_{(i),a}, \ldots, (R_{j,0})_{(i),a}$ form a decomposition of $(C_j)_{(i),a}$, so we need only show that each of the sets $(R_{j,k})_{(i),a}$, for $k = 0, \ldots, n$, is regular. Let $H_{(i),a} = \{\vec{x} \in \mathbf{R}^n \mid x_i = a\}$, and let $\pi_i : \mathbf{R}^n \to \mathbf{R}^{n-1}$ be defined by $\pi_i(x_1, \ldots, x_n) = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$.

It is sufficient to show that $R_{j,k}$ and $H_{(i),a}$ are in general position. Indeed, by the observation at the end of section 5.4, the intersection of two regular sets in general position is again regular. Hence, $R_{j,k} \cap H_{(i),a}$ is regular. Thus, $(R_{j,k})_{(i),a} = \pi_i(R_{j,k} \cap H_{(i),a})$ is the image by the $C^1$-diffeomorphism $\pi_i$ of a regular set, and hence is regular itself.

We still need to show that $R_{j,k} \pitchfork H_{(i),a}$. By property (ii) of the constructed box collection $\mathcal{U}$, we know that $R_{j,k} \pitchfork \mathcal{U}$, and hence $R_{j,k} \pitchfork |\mathcal{U}|_\ell$. Let $\vec{x} \in R_{j,k} \cap H_{(i),a}$ and $B \in (\mathcal{U})^\ell$ such that $\vec{x} \in B \subset H_{(i),a}$. Note that such a $B$ always exists because $a \in \mathrm{Coord}(\mathcal{U}_{(i)})$ and $\mathcal{U}$ covers $A$. Hence, $R_{j,k} \pitchfork |B|$ or, in other words, $\mathrm{T}_{\vec{x}}\, R_{j,k} + \mathrm{T}_{\vec{x}}\,|B| = \mathbf{R}^n$. Since $|B| \subset H_{(i),a}$, we have that $\mathrm{T}_{\vec{x}}\,|B| \subseteq \mathrm{T}_{\vec{x}}\, H_{(i),a}$, and hence also $\mathrm{T}_{\vec{x}}\, R_{j,k} + \mathrm{T}_{\vec{x}}\, H_{(i),a} = \mathbf{R}^n$. ∎

CLAIM 6.3. *The box collections $\mathcal{U}_{(i),a}$ are in general position with $(C_0)_{(i),a}, \dots, (C_m)_{(i),a}$.*

*Proof.* We need to prove that $\{|\mathcal{U}_{(i),a}|_0, \dots, |\mathcal{U}_{(i),a}|_n\} \pitchfork \{(R_{j,k})_{(i),a} \mid j = 0, \dots, m, k = 0, \dots, n\}$. Let $H_{(i),a} = \{\vec{x} \in \mathbf{R}^n \mid x_i = a\}$, and let $\pi_i : \mathbf{R}^n \to \mathbf{R}^{n-1}$ be defined by $\pi_i(x_1, \dots, x_n) = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$.

We have that $|\mathcal{U}_{(i),a}|_\ell = \pi_i(|\mathcal{U}|_\ell \cap H_{(i),a})$. Thus, $B' \in (\mathcal{U}_{(i),a})^\ell$ if and only if $|B'| = \pi_i(|B|)$ with $B \in (\mathcal{U})^\ell$ and $|B| \subseteq H_{(i),a}$.

As already observed in the proof of Claim 6.2, $R_{j,k} \cap |\mathcal{U}|_\ell$ is a regular set. Hence, for $\vec{x} \in R_{j,k} \cap |\mathcal{U}|_\ell$ the tangent space $\mathrm{T}_{\vec{x}}(R_{j,k} \cap |\mathcal{U}|_\ell)$ exists. Moreover, $\mathrm{T}_{\vec{x}}(R_{j,k} \cap |\mathcal{U}|_\ell) = \mathrm{T}_{\vec{x}}(R_{j,k} \cap |B|)$ for some $B \in (\mathcal{U})^\ell$ and $|B| \subseteq H_{(i),a}$.

Let $|B'| = \pi_i(|B|)$. We need to prove that

$$\mathrm{T}_{\vec{x}_{(i),a}}\,|B'| + \mathrm{T}_{\vec{x}_{(i),a}}((R_{j,k})_{(i),a}) = \mathbf{R}^{n-1}. \tag{6.12}$$

We have that

$$\mathrm{T}_{\vec{x}_{(i),a}}\,|B'| \quad = \mathrm{d}\pi_i(\mathrm{T}_x\,|B|) \text{ and} \tag{6.13}$$

$$\mathrm{T}_{\vec{x}_{(i),a}}((R_{j,k})_{(i),a}) = \mathrm{d}\pi_i(\mathrm{T}_{\vec{x}}(R_{j,k} \cap |B|)), \tag{6.14}$$

where $d\pi_i$ is the differential of $\pi_i$ [23].

Moreover, because of property (ii) of the box collection $\mathcal{U}$ and the remark at the end of section 5.4 on the intersection of tangent spaces, we have

$$\mathrm{T}_{\vec{x}}\,|B| + \mathrm{T}_{\vec{x}}(R_{j,k}) = \mathbf{R}^n \text{ and} \tag{6.15}$$

$$\mathrm{T}_{\vec{x}}(R_{j,k} \cap |B|) = \mathrm{T}_{\vec{x}}\, R_{j,k} \cap \mathrm{T}_{\vec{x}}\,|B|. \tag{6.16}$$

Now, let $(v_1, \dots, v_{i-1}, v_{i+1}, \dots, v_n) \in \mathbf{R}^{n-1}$ and let $\vec{v} = (v_1, \dots, v_{i-1}, 0, v_{i+1}, \dots, v_n) \in \mathbf{R}^n$. By (6.15) there exists $\vec{b} \in \mathrm{T}_{\vec{x}}\,|B|$ and $\vec{r} \in \mathrm{T}_{\vec{x}}(R_{j,k})$ such that $\vec{v} = \vec{b} + \vec{r}$. Moreover, we may take $b_i = 0$ since vectors in $\mathrm{T}_{\vec{x}}\,|B|$ have no component in the $i$th coordinate. Hence $r_i$ has to be zero too. By (6.16), we have $\vec{r} \in \mathrm{T}_{\vec{x}}(R_{j,k} \cap |B|)$. Let $\vec{b'} = \mathrm{d}\pi_i(\vec{b})$ and $\vec{r'} = \mathrm{d}\pi_i(\vec{r})$. Then by (6.13), $\vec{b'} \in \mathrm{T}_{\vec{x}_{(i),a}}\,|B'|$, and by (6.14), $\vec{r'} \in \mathrm{T}_{\vec{x}_{(i),a}}((R_{j,k})_{(i),a})$. By construction, $(v_1, \dots, v_{i-1}, v_{i+1}, \dots, v_n) = \vec{b'} + \vec{r'}$, proving (6.12). ∎

**6.3. Putting everything together: The linearization algorithm.** The algorithm that constructs an **A**-linear set, which is homeomorphic to a given semialgebraic set, works inductively on the dimension of the surrounding space in which the semialgebraic set is embedded.

**6.3.1. The bounded case.** The algorithm consists of two parts. The first part is a preprocessing step, which takes as input a bounded semialgebraic set $A$ in $\mathbf{R}^n$ and returns the regular decomposition of each part of the uniform cone radius decomposition of $A$.

**Subroutine:** PREPROCESS

**Input:** A semialgebraic set $A$ in $\mathbf{R}^n$.

**Output:** A uniform cone radius decomposition $C_0, \dots, C_k$ of $A$ and for each $C_i$ a regular decomposition $R_{i,0}, \dots, R_{i,i}$ of $C_i$.

**Method:**

1. Compute the uniform cone radius decomposition of $A$:

$$A = C_0 \cup \cdots \cup C_k.$$

2. Compute the regular decomposition of $C_i$, for $i = 0, \dots, k$:

$$C_i = R_{i,0} \cup \cdots \cup R_{i,i}.$$

**Subroutine:** LINEARIZE-IN-$n$-DIMENSIONS

**Input:** $(\{C_i\}, \{R_{i,r}\}, \mathcal{F})$ with $C_0, \dots, C_k$ a uniform cone radius collection, $\{R_{i,r}\}$ a regular decomposition of $C_i$, and $\mathcal{F}$ an $n$-dimensional box collection in $\mathbf{R}^n$ which is in general position with $C_0, \dots, C_k$.

**Output:** An **A**-linear set $\widehat{C}$ in $\mathbf{R}^n$ which is homeomorphic to $C = C_0 \cup \cdots \cup C_k$.

**Method:**

- If $n > 1$, do the following:
  1. Compute the box collection $\mathcal{U}$ constructed in section 6.2.
  2. Compute a $(3n+1)$-ary relation $\mathcal{P}$ consisting of pairs $(B, \vec{p}_B, b)$, where $B$ is an $n$-dimensional box in $\mathcal{U}$, $\vec{p}_B \in \mathbf{R}^n$, and $b \in \{0,1\}$ such that:
     (a) $\vec{p}_B \in \mathrm{cl}(C) \cap \mathrm{int}(B)$ and is uniquely selected for each $B$;
     (b) $\gamma_{\mathrm{cone},C}(\vec{p}_B) > \mathrm{diam}(B)$; and
     (c) $b = 0$ in case $\vec{p}_B \in \mathrm{cl}(C) \setminus C$ and $b = 1$ in case $\vec{p}_B \in C$.
  3. Compute all $\mathcal{U}_{(i),a}$ with $a \in \mathrm{Coord}(\mathcal{U}_{\{i\}})$ and $i \in \{1, \dots, n\}$.
  4. Compute all $(C_j)_{(i),a} \subset \mathbf{R}^{n-1}$ with $a \in \mathrm{Coord}(\mathcal{U}_{\{i\}})$ and $i \in \{1, \dots, n\}$.
  5. Compute all $(R_{i,r})_{(i),a} \subset \mathbf{R}^{n-1}$ with $a \in \mathrm{Coord}(\mathcal{U}_{\{i\}})$ and $i \in \{1, \dots, n\}$.
  6. For all input triples $(\{(C_j)_{(i),a}\}, \{(R_{i,r})_{(i),a}\}, \mathcal{U}_{(i),a})$ with $a \in \mathrm{Coord}(\mathcal{U}_{\{i\}})$ and $i \in \{1, \dots, n\}$, apply LINEARIZE-IN-$(n-1)$-DIMENSIONS and embed the result in the corresponding hyperplane in $\mathbf{R}^n$, i.e., apply $(x_1, \dots, x_{n-1}) \mapsto (x_1, \dots, a, \dots, x_{n-1})$ where $a$ appears in the $i$th position.
  7. Initialize $\widehat{C}$ to the union of the results of the calls to LINEARIZE-IN-$(n-1)$-DIMENSIONS of step 6.
- If $n = 1$, then do the following:
  1. Initialize $\widehat{C}$ to $C_0 \cup \cdots \cup C_k$.
- Output

$$\begin{aligned} \widehat{C} := \widehat{C} &\cup \{\mathrm{Cone}(\widehat{C} \cap \partial B, \vec{p}_B) \mid (B, \vec{p}_B, b) \in \mathcal{P} \text{ and } b = 1\} \\ &\cup \{\mathrm{Cone}(\widehat{C} \cap \partial B, \vec{p}_B) \setminus \{\vec{p}_B\} \mid (B, \vec{p}_B, b) \in \mathcal{P} \text{ and } b = 0\}. \end{aligned}$$

**Algorithm:** LINEARIZE
**Input:** A bounded semialgebraic set $A$ in $\mathbf{R}^n$.
**Output:** An $\mathbf{A}$-linear set $\widehat{A}$ in $\mathbf{R}^n$ which is homeomorphic to $A$.
**Method:**
1. Call LINEARIZE-IN-$n$-DIMENSIONS(PREPROCESS$(A), \emptyset$).

Before we prove the correctness of the LINEARIZE algorithm, we want to point out the importance of the general position assumption made in the input of the algorithm. First of all, it allows us to treat all boxes in $\mathcal{U}$ in the same way. More specifically, for every box $B$ we are assured of having a point $\vec{p}_B \in \operatorname{int}(|B|)$ as described in step 2 of the algorithm (see Lemma 6.4). The existence of these points is essential for the linearization, as is clear from the last step in the algorithm. Second, the general position assumption ensures that the lower-dimensional sets defined in steps 3–5 are nice and are again in general position (see the three claims in section 6.2). This implies that we can apply LINEARIZE on the lower-dimensional sets, which is a key feature for the algorithm.

LEMMA 6.5. *For any semialgebraic set $A$ in $\mathbf{R}^n$, the set $\widehat{A} =$ LINEARIZE$(A)$ is indeed a linearization of $A$.*

*Proof.* The linearity of $\widehat{A}$ is immediate, so we focus on the existence of a homeomorphism $h : \mathbf{R}^n \to \mathbf{R}^n$, which maps $A$ to $\widehat{A}$.

The existence proof (which is also a constructive proof) is an inductive proof. Before we can state the induction hypothesis, we need to define some box collections in $\mathbf{R}^n$.

We define $\mathcal{U}_{[n]}$ to be the $n$-dimensional box collection $\mathcal{U}$ in $\mathbf{R}^n$ constructed in step 1 when LINEARIZE-IN-$n$-DIMENSIONS is called.

Let $k < n$. With each call of LINEARIZE-IN-$k$-DIMENSIONS during the linearization of $A$ we associate the pair $(i_{n-k}, a_{i-k}) \in \{1, \ldots, n\} \times \mathbf{R}$ such that $a_{n-k}$ is the value in $\operatorname{Coord}(\mathcal{U}_{\{i_{n-k}\}})$ used in step 6. Note that $\mathcal{U}$ is the box collection constructed in step 1 during the preceding call of LINEARIZE-IN-$(k+1)$-DIMENSIONS.

This sequence of pairs gives us a unique identifier for the box collection constructed in step 1 during each call of the algorithm. More specifically, we denote by $\mathcal{U}_{(i_1,a_1),\ldots,(i_{n-k},a_{n-k})}$ the box collection $\mathcal{U}$ constructed in step 1 of the call LINEARIZE-IN-$k$-DIMENSIONS corresponding to $(i_{n-k}, a_{n-k})$, which was called within LINEARIZE-IN-$(k+1)$-DIMENSIONS corresponding to $(i_{n-k-1}, a_{n-k-1})$, and so forth until LINEARIZE-IN-$(n-1)$-DIMENSIONS is called with $(i_1, a_1)$ within the initial call LINEARIZE-IN-$n$-DIMENSIONS. If $k = 1$, then no box collection $\mathcal{U}$ is constructed, since step 1 is skipped in the algorithm. However, for the purpose of this proof, we define $\mathcal{U}_{(i_1,a_1),\ldots,(i_{n-1},a_{n-1})}$ to be $\mathcal{U}_{\{i_{n-1}\},a_{n-1}}$, where $\mathcal{U}$ is the box collection constructed in step 1 of the preceding call to LINEARIZE-IN-2-DIMENSIONS corresponding to $(i_{n-2}, a_{n-2})$, and so forth.

At the same time, the sequence of pairs $(i_j, a_j)$ tells us how to correctly embed $\mathcal{U}_{(i_1,a_1),\ldots,(i_{n-k},a_{n-k})}$ into $\mathbf{R}^n$. Indeed, the embedding simply maps $\vec{x} \in \mathbf{R}^k$ to the vector $\vec{x}' \in \mathbf{R}^n$ obtained by putting $a_j$ at position $i_j$ and filling up the $k$ open slots with the values (in this order) $x_1, \ldots, x_k$. We will denote this embedding by $\rho_{(i_1,a_1),\ldots,(i_{n-k},a_{n-k})}$.

We now define the $k$-dimensional box collection $\mathcal{U}_{[k]}$ in $\mathbf{R}^n$ as

$$\mathcal{U}_{[k]} = \cup_{(i_1,a_1),\ldots,(i_{n-k},a_{n-k})} \rho_{(i_1,a_1),\ldots,(i_{n-k},a_{n-k})}(\mathcal{U}_{(i_1,a_1),\ldots,(i_{n-k},a_{n-k})}).$$

Let $\mathcal{U}_{[\leqslant k]}$ be the union of all boxes in $\mathcal{U}_{[k]}, \ldots, \mathcal{U}_{[1]}$. We shall construct homeomorphisms $h_k : |\mathcal{U}_{[\leqslant k]}| \to |\mathcal{U}_{[\leqslant k]}|$, such that

- $h_k(A \cap |\mathcal{U}_{[\leqslant k]}|) = \widehat{A} \cap |\mathcal{U}_{[\leqslant k]}|$, and
- for all boxes $B$ in $\mathcal{U}_{[k]}, \ldots, \mathcal{U}_{[1]}$, $h_k|_{|B|} : |B| \to |B|$ is a homeomorphism.

We shall construct the homeomorphisms $h_k$ by induction on $k$.

For the base case, $k = 1$, the linearization algorithm keeps $A$ intact (see the case $n = 1$ in the description of the Linearize-In-$n$-Dimensions algorithm). Hence, $\mathcal{U}_{[1]} \cap \widehat{A} = \mathcal{U}_{[1]} \cap A$ and we let $h_k$ be the identity mapping on $\mathcal{U}_{[1]}$. Both conditions are trivially satisfied for $h_1$.

Suppose we have constructed a homeomorphism $h_{k-1} : |\mathcal{U}_{[\leqslant k-1]}| \to |\mathcal{U}_{[\leqslant k-1]}|$ such that

- $h_{k-1}(A \cap |\mathcal{U}_{[\leqslant k-1]}|) = \widehat{A} \cap |\mathcal{U}_{[\leqslant k-1]}|$, and
- for all boxes $B$ in $\mathcal{U}_{[k]}, \ldots, \mathcal{U}_{[1]}$, $h_{k-1}|_{|B|} : |B| \to |B|$ is a homeomorphism.

Let $B' \in \mathcal{U}_{[k]}$; then we will define $h_k|_{|B'|} : |B'| \to |B'|$ as the composition of two homeomorphisms $f$ and $g$. Let us first describe the homeomorphism $g$. By definition, $|B'| = \rho_{(i_1,a_1),\ldots,(i_{n-k},a_{n-k})}(|B|)$ with $B \in \mathcal{U}_{(i_1,a_1),\ldots,(i_{n-k},a_{n-k})}$.

Let $\mathcal{P}$ be the relation computed in step 2 after $\mathcal{U}_{(i_1,a_1),\ldots,(i_{n-k},a_{n-k})}$ was computed. By the definition of the relation $\mathcal{P}$ and by Theorem 5.2, there exists a homeomorphism $g|_{|B|} : |B| \to |B|$ such that $g|_{\partial|B|}$ is the identity, and either

1. $g|_{|B|}(|B| \cap A) = \mathrm{Cone}(A \cap \partial|B|, \vec{p}_B)$ in case $(B, \vec{p}_B, 1) \in \mathcal{P}$, or
2. $g|_{|B|}(|B| \cap A) = \mathrm{Cone}(A \cap \partial|B|, \vec{p}_B) \setminus \{\vec{p}_B\}$ in case $(B, \vec{p}_B, 0) \in \mathcal{P}$.

Since the second case is completely analogous to the first case, we assume that the first case holds for $g$. This concludes the description of the homeomorphism $g$.

Before we explain the construction of the second homeomorphism $f$, we show how to partition $|B|$ using the boundary of boxes $|B_t|$ parametrized by $t \in [0,1]$. Suppose that $|B| = [a_1, b_1] \times \cdots \times [a_n, b_n]$, and suppose $\vec{p}_B = (c_1, \ldots, c_n)$ with $a_i < c_i < b_i$ for $i = 1, \ldots, n$. Then the following sets for $0 \leqslant t \leqslant 1$ partition $|B|$ such that $|B| = \cup_{t\in[0,1]} \partial|B_t|$:

$$\begin{aligned} |B_t| := [ta_1 + (1-t)c_1, tb_1 + (1-t)c_1] \times \cdots \\ \times [ta_n + (1-t)c_n, tb_n + (1-t)c_n] \qquad 0 \leqslant t \leqslant 1. \end{aligned}$$

Let $\vec{x} \in |B|$. To start with the construction of $f(\vec{x})$ for $\vec{x} \in |B|$, we define the unique $t_0$ such that $\vec{x} \in \partial|B_{t_0}|$. Then, let $L$ be the halfline from $\vec{p}_B$ to $\vec{x}$ and define

$$\vec{y} = L \cap \partial|B|.$$

Next, let $L'$ is the halfline from $\vec{p}_B$ to $h_{k-1}(\vec{y})$. Note that $h_{k-1}(\vec{y})$ still lies on the boundary $\partial|B|$. Finally, define $f|_{|B|} : |B| \to |B|$ in $\vec{x}$ as

$$f|_{|B|}(\vec{x}) = \partial|B_{t_0}| \cap L'.$$

The construction of $f$ is illustrated in Figure 6.5. It can easily be verified that $f|_{|B|}$ is a homeomorphism from $|B|$ to $|B|$ such that

$$f|_{|B|}(\mathrm{Cone}(A \cap \partial|B|, \vec{p}_B)) = \mathrm{Cone}(h_{k-1}(A \cap \partial|B|), \vec{p}_B). \tag{6.17}$$

Finally, we define $h_k|_{|B'|} : |B'| \to |B'|$ using the composition of the two homeomorphisms $f|_{|B|}$ and $g|_{|B|}$, i.e.,

$$h_k|_{|B'|} = \rho_{(i_1,a_1),\ldots,(i_{n-k},a_{n-k})} \circ f|_{|B|} \circ g|_{|B|} \circ \pi_{i_1,\ldots,i_{n-k}}.$$

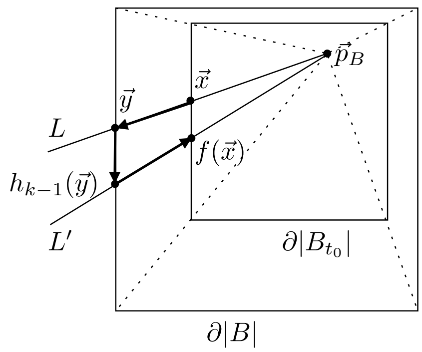


FIG. 6.5. *Construction of the homeomorphism $f : |B| \to |B|$. The figure shows the construction of $f(\vec{x})$ for a point $\vec{x} \in |B|$.*

We now define $h_k : |\mathcal{U}_{[\leqslant k]}| \to |\mathcal{U}_{[\leqslant k]}|$ as

$$h_k := \bigcup_{B \in \mathcal{U}_{[k]}} h_k|_{|B|}$$

and show that it has the desired properties. First, we prove that $h_k$ is a homeomorphism. By the gluing Lemma [35, Lemma 3.8], it is sufficient to show that for any two boxes $B$ and $B'$ in $\mathcal{U}_{[k]}$, we have that

$$h_k|_{|B| \cup |B'|} = h_k|_{|B|} \cup h_k|_{|B'|} : |B| \cup |B'| \to |B| \cup |B'|.$$

For this to hold, it is sufficient to show that for any $k$-dimensional box $B' \in \mathcal{U}_{[k]}$ in $\mathbf{R}^n$,

$$(h_k|_{|B|})|_{|B'|} = (h_k|_{|B'|})|_{|B|}. \tag{6.18}$$

This holds indeed. If $|B| \cap |B'| = \emptyset$, then we are done. Suppose that $\vec{x} \in |B| \cap |B'|$. Then by the definition of a box collection, $\vec{x} \in \partial|B| \cap \partial|B'|$. Now, for every box $B'' \in \mathcal{U}_{[k]}$, we have $h_k|_{\partial|B''|}(\vec{x}) = f|_{\partial|B''|}(\vec{x}) = h_{k-1}(\vec{x})$. Hence,

$$\begin{aligned}(h_k|_{|B|})|_{|B'|}(\vec{x}) &= h_k|_{\partial|B| \cap \partial|B'|}(\vec{x}) \\ &= h_{k-1}(\vec{x}) \\ &= h_k|_{\partial|B'| \cap \partial|B|}(\vec{x}) \\ &= (h_k|_{|B'|})|_{|B|}(\vec{x}).\end{aligned}$$

Hence, $h_k : |\mathcal{U}_{[\leqslant k]}| \to |\mathcal{U}_{[\leqslant k]}|$ is a homeomorphism.

Second, we show that for all boxes $B$ in $\mathcal{U}_{[k]}, \ldots, \mathcal{U}_{[1]}$, $h_{k-1}|_{|B|} : |B| \to |B|$ is a homeomorphism. By construction, this holds for any box $B \in \mathcal{U}_{[k]}$. For boxes $B'$ in $\mathcal{U}_{[i]}$ for $i < k$, it is sufficient to observe that such boxes $B'$ lie on the boundary of a box $B$ in $\mathcal{U}_{[k]}$, and on these boundaries $h_k$ coincides with $h_{k-1}$ for which the desired property holds by induction.

Finally, we still need to verify that $h_k(A \cap |\mathcal{U}_{[\leqslant k]}|) = \widehat{A} \cap |\mathcal{U}_{[\leqslant k]}|$. It is sufficient to show that $h_k(A \cap |B|) = \widehat{A} \cap |B|$ for any $B \in \mathcal{U}_{[k]}$. By (6.17), the induction hypothesis, and the definition of $\widehat{A}$ in the algorithm LINEARIZE-IN-$n$-DIMENSIONS, we have

$$\begin{aligned} h_k(A \cap |B|) &= \mathrm{Cone}(h_{k-1}(A \cap \partial|B|), \vec{p}_B) \\ &= \mathrm{Cone}(\widehat{A} \cap \partial|B|, \vec{p}_B) \\ &= \widehat{A} \cap |B|. \end{aligned}$$

Since $|\mathcal{U}|$ is closed, a standard result from topology [36] implies that the final homeomorphism $h_n$ can be extended to a homeomorphism $h : \mathbf{R}^n \to \mathbf{R}^n$. ◻

We are now ready to state the main result of this section.

THEOREM 6.6. *For each $n$, there exists an* FO+POLY+TC *formula* `linearize` *over the schema $\mathcal{S} = \{S\}$, with $S$ an $n$-ary relation name such that for any polynomial constraint database $D$ over $\mathcal{S}$,* `linearize`$(D)$ *is an algebraic linearization of $S^D$ if $S^D$ is bounded.*

*Proof.* The desired FO+POLY+TC formula `linearize` expresses the algorithm LINEARIZE described above. From Lemma 5.5 and Lemma 5.7, it follows that the algorithm PREPROCESS is FO+POLY-expressible.

Concerning the algorithm LINEARIZE-IN-$n$-DIMENSIONS, we have that in step 1, the box collection $\mathcal{U}$ is computed. In the construction of this box collection in section 6.1, we need to compute the following:

- The computation of a uniform cone radius. This is FO+POLY-expressible by Theorem 5.3.
- The computation of a finite number of box coverings, i.e., the $\frac{\varepsilon_V}{4\sqrt{n}}$-cover$(V)$ coverings of section 6.1. This is FO+POLY+TC-expressible by Proposition 5.14.
- A candidate $\boldsymbol{\tau} \in \mathbf{R}^n$ as specified in Lemma 6.3 needs to be found. Since this is essentially checking a finite number of transversality conditions, this is FO+POLY-expressible by Lemma 5.8.

Hence, we may conclude that the computation of $\mathcal{U}$ is in FO+POLY+TC. In step 2, the relation $\mathcal{P}$ is constructed. Given the box collection $\mathcal{U}$, we know by property (iii) of this collection that in each $B \in \mathcal{U}$ there exists a point $\vec{p} \in \mathrm{int}(|B|) \cap \mathrm{cl}(C)$ such that $\gamma_{\mathrm{cone},C}(\vec{p}) > \mathrm{diam}(B)$. The set of points in $\mathrm{int}(|B|)$ with this property is FO+POLY-expressible by Theorem 5.3. Hence, we can also select in FO+POLY for each $B \in \mathcal{U}$ a unique representant among these points. This will be $\vec{p}_B$. Hence, we may conclude that the computation of the relation $\mathcal{P}$ is FO+POLY-expressible. In steps 3, 4, and 5, we need to compute $\mathrm{Coord}(\mathcal{U}_{\{i\}})$, $\mathcal{U}_{(i),a}$, $(C_j)_{(i),a}$, and $(R_{i,r})_{(i),a}$. By definition these are all FO+POLY-expressible. In step 6 we call the algorithm $n$ times. We have to be careful of how the inductive step is translated in FO+POLY+TC. A straightforward translation would result in a parametrized call of the transitive closure operators in the computation of the box coverings in step 1. Observe, however, that the set of parameters $\mathrm{Coord}(\mathcal{U}_{(i)})$ for $i = \{1, \ldots, n\}$ can be computed inside the transitive closure operator and that these parameters can then be passed on outside the transitive closure operator by simply annotating the vectors inside the transitive closure with these parameters. Indeed, suppose that we want to compute the transitive closure of a parametrized set $X \in \mathbf{R}^{n+m}$, where the last $m$ coordinates are the parameters. Suppose that the set of parameters is FO+POLY+TC-definable from the database by a formula $\varphi$. We now define $Y = [\mathrm{TC}_{\vec{x},\vec{a};\vec{y},\vec{b}} X \wedge \vec{a} = \vec{b} \wedge \varphi(\vec{a})]$. We can then uniquely identify the result of this transitive closure computation for each parameter value by asking for all $(\vec{x}, \vec{a}) \in Y$, for which $\varphi(\vec{a})$ holds. By adapting the box covering formula constructed in Proposition 5.14, we can compute the box coverings for the parameter

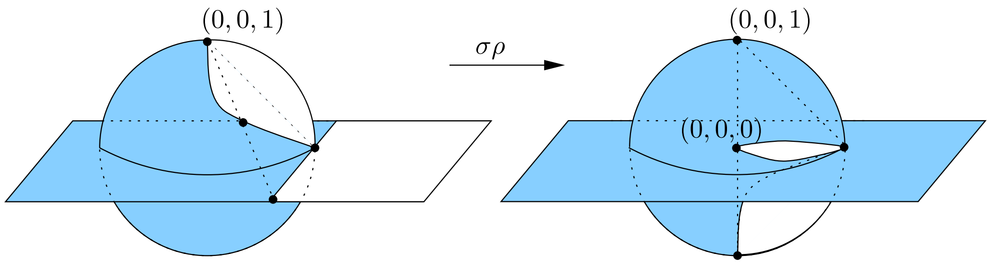


FIG. 6.6. *A semialgebraic set (shaded area) is mapped onto the sphere $S^2(\vec{0},1)$, flipped vertically, and projected back onto the sphere $S^2(\vec{0},1)$. This brings the point at infinity $\vec{p}_\infty$ to the origin $\vec{0}$.*

set $\mathrm{Coord}(\mathcal{U}_{(i)})$ in parallel and keep them apart afterwards. In this way, we do not need parametrized transitive closure, and hence step 6 is expressible in FO+POLY+TC.

In step 7 a simple union is performed (which is trivially in FO+POLY), and finally, the cones are constructed, which is also clearly expressible in FO+POLY.

Since the recursion depth is bounded by the dimension, we can write the complete execution of the algorithm as a single FO+POLY+TC formula. ∎

If the linearization obtained in Theorem 6.6 also needs to be a good approximation from a metrical point of view, we can easily adapt the algorithms such that the approximation lies arbitrarily close to the original polynomial constraint database. Indeed, we can simply bound the diameter of the boxes used in the construction by a specified $\varepsilon$-value. We will see some applications of these *$\varepsilon$-approximations* in the next section.

THEOREM 6.7. *For each $n$ there exists an* FO+POLY+TC *query $\varepsilon$-`approx` over the schema $\mathcal{S} = \{S\}$ with $S$ an $n$-ary relation name such that for any polynomial constraint database $D$ over $\mathcal{S}$ such that $S^D$ is bounded, the set $\varepsilon$-`approx`$(D)$ is an algebraic $\varepsilon$-approximation of $S^D$.*

*Proof.* The proof follows at once from the fact that the homeomorphism $h$ constructed in the proof of Theorem 6.6 maps $A \cap |B|$ to $\widehat{A} \cap |B|$ for each box $B \in \mathcal{U}$. Thus, if $\vec{p} \in A \cap |B|$ then also $h(\vec{p}) \in |B|$. Because $\mathrm{diam}(B) < \varepsilon$, the distance between $\vec{p}$ and $h(\vec{p})$ is smaller than $\varepsilon$, so in this case $\widehat{A}$ will be an $\varepsilon$-approximation of $A$. ∎

**6.3.2. The general case.** Let $A$ be an unbounded semialgebraic set in $\mathbf{R}^n$. We reduce the construction of an algebraic linearization of $A$ to the construction for bounded semialgebraic sets as follows.

First, we need to define the *cone radius of $A$ in the point at infinity $\vec{p}_\infty$*. Consider the embedding $i : \mathbf{R}^n \to \mathbf{R}^{n+1} : (x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_n, 0)$. Let $\rho : \mathbf{R}^{n+1} \to \mathbf{R}^{n+1}$ be the reflection defined by $(x_1, \ldots, x_{n+1}) \mapsto (x_1, \ldots, x_n, -x_{n+1})$. Let $\mathbf{R}^n \cup \{\vec{p}_\infty\}$ be the one-point compactification of $\mathbf{R}^n$ [35]. Finally, consider the stereographic projection $\sigma : S^n((0, \ldots, 0), 1) \to i(\mathbf{R}^n) \cup \{\vec{p}_\infty\}$ defined by $\sigma(x_1, \ldots, x_{n+1}) = \frac{(x_1, \ldots, x_n)}{1 - x_{n+1}}$ and $\sigma(0, \ldots, 0, 1) = \vec{p}_\infty$.

We define a cone radius of $A$ at $\vec{p}_\infty$ as a cone radius of the semialgebraic set

$$i^{-1}(\sigma(\rho(\sigma^{-1}(i(A) \cup \{\vec{p}_\infty\}))))$$

in the origin of $\mathbf{R}^n$. Figure 6.6 illustrates the above transformation process. The local conic structure of semialgebraic sets implies that there exists an $m > 0$ such that $\{\vec{x} \in \mathbf{R}^n \mid \|\vec{x}\| \geqslant m\} \cap A$ is topologically equivalent to $\{\lambda \vec{x} \in \mathbf{R}^n \mid \vec{x} \in \partial([-m, m] \times \cdots \times [-m, m]) \cap A \wedge \lambda \geqslant 1\}$.

We now present the unbounded version of the algorithm LINEARIZE.

**Algorithm** LINEARIZE′
**Input:** A semialgebraic set $A$ in $\mathbf{R}^n$ $\mathbf{R}^n$.
**Output:** An $\mathbf{A}$-linear set $\widehat{A}$ in $\mathbf{R}^n$ which is homeomorphic to $A$.
**Method:**
1. Compute a cone radius $m$ of $A$ in $\vec{p}_\infty$. Let $M = [-m, m] \times \cdots \times [-m, m]$.
2. Call Linearize($A \cap M$).
3. Output

$$\widehat{A} := (\widehat{A \cap M}) \cup \{\lambda\vec{x} \in \mathbf{R}^n \mid \vec{x} \in A \cap \partial M \wedge \lambda \geqslant 1\}.$$

We obtain the following generalization of Theorem 6.6.

THEOREM 6.8. *For each $n$ there exists an* FO+POLY+TC *formula* `linearize` *over the schema $\mathcal{S} = \{S\}$, with $S$ an $n$-ary relation name such that for any polynomial constraint database $D$ over $\mathcal{S}$,* `linearize`$(D)$ *is an algebraic linearization of $S^D$.*

**6.4. Rational linearizations.** We now refine the previous theorems to *rational linearization*.

THEOREM 6.9. *For each $n$, there exists an* FO+POLY+TC *query* `ratlin` *over the schema $\mathcal{S} = \{S\}$, with $S$ $n$-ary, such that for any polynomial constraint database $D$ over $\mathcal{S}$ such that $S^D$ is bounded,* `ratlin`$(D)$ *is a rational linearization of $S^D$.*

*Proof.* We can obtain this result easily by modifying the construction of the special box collection in section 6.1 in the following way. When the box covering $\mathcal{V}$ of size $\frac{\varepsilon_\mathcal{V}}{\sqrt{n}}$ is computed in this construction, we compute a rational number that is smaller than $\frac{\varepsilon_\mathcal{V}}{\sqrt{n}}$ and take this as the size of the box covering $\mathcal{V}$ to be computed. By similar techniques as those in section 4, it is easy to show that there exists an FO+POLY+TC query, which returns a rational number smaller than the input number. In this way, all boxes in $\mathcal{R} \subset \mathbf{Q}^{2n}$. A second adaptation is that the relation $\mathcal{P}$ is replaced by the following relation

$$\mathcal{P}' = \{(B, \vec{c}_B, b) \in \mathcal{U} \times \mathbf{Q}^n \times \{0, 1\} \mid \exists \vec{p}_B(B, \vec{p}_B, b) \in \mathcal{P}\},$$

where $\vec{c}_B$ denote the center of the box $B$.

In this way the algorithm LINEARIZE-IN-$n$-DIMENSIONS will select points with rational coordinates. ∎

We also have a rational equivalent of Theorem 6.7.

THEOREM 6.10. *For each $n$ there exists an* FO+POLY+TC *query $\varepsilon$-*`ratlin` *over the schema $\mathcal{S} = \{S\}$, with $S$ an $n$-ary relation name, such that for any polynomial constraint database $D$ over $\mathcal{S}$ such that $S^D$ is bounded, the set $\varepsilon$-*`ratlin`$(D)$ *is a rational $\varepsilon$-approximation of $S^D$.*

**6.5. The connectivity query.** Although we know already that the connectivity query, which asks whether a polynomial constraint database is connected, is expressible in FO+POLY+TCS, we show in this section that the connectivity query is already expressible in FO+POLY+TC. Let $A$ be a semialgebraic set in $\mathbf{R}^n$. For semialgebraic sets, expressing the connectivity query is the same as expressing whether any two points can be connected by a path lying entirely in $A$ [6, Proposition 2.5.13]. One can even choose the paths to be semialgebraic, in case of a semialgebraic set, and semilinear, in case of a semilinear set [44, Chapter 6, Proposition 3.2].

We now show that this query can be expressed in FO+POLY+TC using the formula `linearize` given in Theorem 6.8.

Let $\mathcal{S} = \{S\}$, with $S$ an $n$-ary relation name. Consider the FO+POLY+TC formula $\texttt{lineconn}(\vec{r}, \vec{s})$ over $\mathcal{S}$ such that for any database $D$ over $\mathcal{S}$, $(\vec{p}, \vec{q}) \in \texttt{lineconn}(D)$ if and only if

$$\forall\lambda(0 \leqslant \lambda \leqslant 1), \quad \lambda\vec{p} + (1-\lambda)\vec{q} \in \texttt{linearize}(D).$$

Define now the FO+POLY+TC sentence `connected`, which tests for any database $D$ over $\mathcal{S}$ whether

$$\forall\vec{p} \in \texttt{linearize}(D), \forall\vec{q} \in \texttt{linearize}(D), \quad (\vec{p}, \vec{q}) \in [\mathrm{TC}_{\vec{x};\vec{y}}\texttt{lineconn}(D)].$$

PROPOSITION 6.11. *Let $\mathcal{S} = \{S\}$ with $S$ an $n$-ary relation name. The* FO+POLY+TC *formula* `connected` *always terminates and expresses the connectivity query.*

*Proof.* Since $\texttt{linearize}(D)$ is topologically equivalent to $S^D$, $S^D$ is connected if and only if $\texttt{linearize}(D)$ is. Since $\texttt{linearize}(D)$ is semilinear, two points $\vec{p}$ and $\vec{q}$ belong to the same connected component of $\texttt{linearize}(D)$ if and only if there exists a piecewise linear path from $\vec{p}$ to $\vec{q}$ lying entirely in $\texttt{linearize}(D)$. The formula `connected` expresses that all points of $\texttt{linearize}(D)$ belong to the same connected component, i.e., that $\texttt{linearize}(D)$ is connected.

To conclude that the evaluation of the transitive closure in the formula `connected` ends in finitely many steps, we need to show that there exists an upper bound on the number of line segments in `linearize(D)`, which is needed to connect any two points in the same connected component of `linearize(D)`. Now, any semilinear set can be decomposed into a finite number of convex sets [44]. The finiteness of this decomposition yields the desired bound. □

Since FO+POLY+TC is included in stratified DATALOG with polynomial constraints, Proposition 6.11 solves the question [15, 31, 33] of whether stratified DATALOG with polynomial constraints can express the connectivity query.

**6.6. Volume approximation.** In this section, we shall use the box covering and the $\varepsilon$-approximation to approximate the volume of semialgebraic sets with an FO+POLY+TC formula. We restrict our attention to bounded semialgebraic sets and require that the evaluation of this FO+POLY+TC formula is effective for all bounded semialgebraic inputs.

Let $\mathcal{S} = \{S\}$, with $S$ an $n$-ary relation name. Let $D$ be a polynomial constraint database over $\mathcal{S}$.

The *volume* of a database $D$ is defined as the Lebesgue-measure of the semialgebraic set $S^D \subseteq \mathbf{R}^n$ and is denoted by $\mathrm{VOL}(D)$.

Since we want an FO+POLY+TC formula whose evaluation is effective on all databases, it is impossible to define the *exact* volume of polynomial constraint databases in FO+POLY+TC. Indeed, consider the database consisting of the unit disk $D$ in $\mathbf{R}^2$. The volume of $D$ equals $\pi$. Since $\pi$ is not algebraic, this value cannot be the output of an effective FO+POLY+TC query.

Hence, as suggested by Koiran [28] and Benedikt and Libkin [5], we consider for each $\varepsilon > 0$ an *$\varepsilon$-volume approximation query* $\mathrm{VOL}^{\varepsilon}$, such that for any polynomial constraint database $D$ over $\mathcal{S}$, such that if $v \in \mathrm{VOL}^{\varepsilon}(D)$, then

$$|v - \mathrm{VOL}(S^D)| < \varepsilon.$$

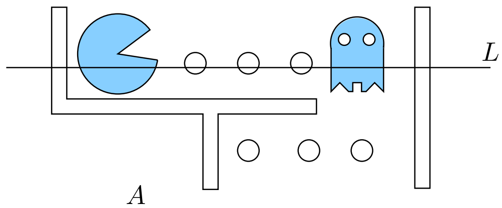


FIG. 6.7. *A semialgebraic set $A$ with $\kappa(A) = 12$.*

It is known that volume approximation is not expressible in FO+POLY [5]. We show that it is expressible in FO+POLY+TC.

We will use the following result.

THEOREM 6.12 (see [28]). *Let $A$ be a semialgebraic set in $\mathbf{R}^n$, and let $\delta$-cover$(A)$ be its box covering of size $\delta$. Then*

$$|\textsc{Vol}(A) - \textsc{Vol}(\delta\text{-}cover(A))| < \frac{1}{\delta}(\mathrm{diam}(A))^{n+1}\kappa(A)n, \tag{6.19}$$

*where $\kappa(A)$ is the maximal number of connected components of the intersection of $A$ with any axis-parallel line $L$ (see Figure 6.7), and where $\mathrm{diam}(A)$ is the diameter of $A$.*[5]

THEOREM 6.13. *For each $\varepsilon > 0$, there exists an $\varepsilon$-volume approximation query in* FO+POLY+TC.

*Proof.* We first show that the number $\kappa$ of Theorem 6.12 is expressible in FO+POLY+TC. Thereto, first we define $n$ sets $K_i$ which contain $(2n-1)$-tuples $(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n, \vec{p})$, where $a_j \in \mathbf{R}$ for $j = 1, \ldots, i-1, i+1, \ldots, n$, and where $\vec{p}$ is either an isolated point on the intersection of $A$ with $\{\vec{x} \mid \bigwedge_{j \neq i} x_j = a_j\}$, or in the middle of an interval in this intersection. Using similar techniques as in section 4, we compute for each $(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$ the number of points $\vec{p}$ such that $(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n, \vec{p}) \in K_i$. We then obtain $n$ sets $K'_i$ consisting of $n$-tuples $(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n, N)$ with $N \in \mathbf{N}$, and we define $M_i$ to be the maximum of all those $N$ which are in $K'_i$ for some $(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$. Finally, $\kappa = \max\{M_1, \ldots, M_n\}$.

Let $\delta = \frac{1}{\varepsilon}(\mathrm{diam}(S^D))^n\kappa(S^D)n + 1$. By Proposition 5.14, the box covering of $S^D$ of size $\delta$ is expressible in FO+POLY+TC. By Theorem 6.12, $\textsc{Vol}(\delta\text{-cover}(S^D))$ approximates the volume of $S^D$ within an $\varepsilon$-error margin.

Recall that $\delta$-cover$(S^D)$ is represented as a $2n$-ary relation. Each $2n$-tuple corresponds to an $n$-dimensional box of size $\delta$ (see section 5.5). Let $\texttt{nrofboxes}(y)$ be the formula

$$[\mathrm{TC}_{\vec{b},x;\vec{b}',x'}\texttt{lexicographic}(\vec{b}, \vec{b}') \wedge x' = x + 1](\vec{b}_{\min}, 1, \vec{b}_{\max}, y),$$

where $\texttt{lexicographic}(\vec{b}, \vec{b}')$ is an FO+POLY formula expressing that $\vec{b}$ is less than $\vec{b}'$ with respect to the lexicographical ordering on tuples in $\mathbf{R}^n$, and where $\vec{b}_{\min}$,$\vec{b}_{\max} \in \delta$-cover$(S^D)$ is the minimum (respectively, maximum) $n$-tuple in $\delta$-cover$(S^D)$ with respect to the lexicographical ordering. Finally, let $N \in \mathbf{R}$ such that $\texttt{nrofboxes}(N)$ holds. Then we define $\textsc{Vol}^\varepsilon(v)$ to be the FO+POLY+TC formula which expresses that $v = N\delta^n$. $\square$

[5]For $X \subseteq \mathbf{R}^n$ bounded, the *diameter* of $X$ is defined as the supremum of $\{\|\vec{x} - \vec{y}\| \mid \vec{x}, \vec{y} \in X\}$.

Since the $\delta$-approximation of $A$ is included in the box covering $\delta$-cover($A$), a better volume approximation can be obtained by using the volume of the $\delta$-approximation instead of the volume of $\delta$-cover($A$). By the next theorem, this also gives an FO+ Poly+TC expressible $\varepsilon$-approximation query.

It is known that taking the volume of a semilinear set does not take us out of the semialgebraic setting and that the volume of a semilinear set can be expressed in the aggregate language FO+Poly+Sum [5].

Theorem 6.14. *Let $\mathcal{S} = \{S\}$, with $S$ an $n$-ary relation name. There exists an* FO+Poly+TC *formula* `volume` *over $\mathcal{S}$ such that* $\texttt{volume}(S^D)$ *is the volume of $S^D$ for any linear constraint database $D$ over $\mathcal{S}$.*

*Proof.* If $\dim(S^D) < n$, then we define $\texttt{volume}(x) \equiv x = 0$. Suppose that $\dim(S^D) = n$. Since $\mathrm{Vol}(S^D) = \mathrm{Vol}(\mathrm{cl}(\mathrm{int}(S^D)))$, we actually may assume that $S^D$ is closed and consists entirely of $n$-dimensional pieces.

It is well known that $S^D$ is a finite union of convex sets $c_1, \ldots, c_r$ of a partition of $\mathbf{R}^n$ induced by a finite number of $(n-1)$-dimensional hyperplanes $H_1, \ldots, H_s$ [48]. Vandeurzen, Gyssens, and Van Gucht [48] show that there exists an FO+Poly formula $\texttt{hyperplanes}(v_1, \ldots, v_n, d)$ such that $\texttt{hyperplanes}(D)$ consists of $s$ tuples $(\vec{v}_1, d_1), \ldots, (\vec{v}_s, d_s)$ such that $H_i = \{\vec{x} \in \mathbf{R}^n \mid \vec{v}_i\vec{x} = d_i\}$. Moreover, there exists an FO+Poly formula `points` such that $\texttt{points}(D)$ is equal to the extremal points of the convex sets $c_1, \ldots, c_s$. Recall that the *extremal points* of a convex set are those points which cannot be written as a linear combination of two other points of the convex set [51].

We now want to retrieve the extremal points of the convex sets $c_1, \ldots, c_r$. In order to do so, we shall first select a unique point in the interior of each convex set. With each of these points we then associate all special points which are in the corresponding convex set. These will then be the extremal points.

We thus define an FO+Poly+TC formula `unique` over $\mathcal{S}$ such that $\texttt{unique}(D)$ consists of points $\vec{p}_1, \ldots, \vec{p}_s$ such that $\vec{p}_i \in \mathrm{int}(c_i)$ for $i = 1, \ldots, s$. The formula `unique` makes use of the following formulas over $\mathcal{S}$:

- A formula over $\mathcal{S}$ which computes the barycenter of any $n$-dimensional simplex obtained as the convex hull of an $(n+1)$-tuple of points in `specialpoints` $(D)$, i.e.,

$$\texttt{barycenter}(\vec{x}) \equiv \exists \vec{y}_1 \cdots \exists \vec{y}_{n+1} \left( \bigwedge_{i=1}^{n} \texttt{points}(\vec{y}_i) \right. \\ \left. \wedge\, x_i = \frac{1}{n+1}((\vec{y}_1)_i + \cdots + (\vec{y}_{n+1})_i) \right).$$

- A formula `interiors` over $\mathcal{S}$ which computes the interiors of the sets $c_1, \ldots, c_s$, i.e.,

$$\texttt{interiors}(\vec{x}) \equiv S(\vec{x}) \wedge \neg(\exists\, \vec{v}\, \exists\, d(\texttt{hyperplanes}(\vec{v}, d) \wedge \vec{v} \cdot \vec{x} = d)).$$

- A formula over $\mathcal{S}$ which checks whether two barycenters are in the same convex set $c_i$ for some $i$, i.e.,

$$\texttt{samecell}(\vec{x}, \vec{y}) \equiv \texttt{barycenter}(\vec{x}) \wedge \texttt{barycenter}(\vec{y}) \\ \wedge\, \forall \lambda (0 \leqslant \lambda \leqslant 1) \rightarrow \texttt{interiors}(\lambda\vec{x} + (1-\lambda)\vec{y}).$$

We then define the formula $\texttt{unique}(\vec{x})$ as

$$\forall\, \vec{z} \texttt{samecell}(\vec{x},\vec{z}) \rightarrow \texttt{lexicographic}(\vec{x},\vec{z}),$$

where $\texttt{lexicographic}(\vec{x},\vec{z})$ is an FO+POLY formula expressing that $\vec{x}$ is less than or equal to $\vec{z}$ with respect to the lexicographical ordering on tuples in $\mathbf{R}^n$.

Define the formula

$$\begin{aligned}\texttt{extremal}(\vec{x},\vec{y}) \equiv\ & \texttt{points}(\vec{x}) \wedge \texttt{unique}(\vec{y}) \\ & \wedge \forall\lambda(0 < \lambda \leqslant 1) \rightarrow \texttt{interiors}(\lambda\vec{y} + (1-\lambda)\vec{x}).\end{aligned}$$

We can now identify each convex set $c_1,\dots,c_r$, so we may focus on a single convex set. We now show that, given the extremal points of a convex set $c$ in $\mathbf{R}^n$, a decomposition of $c$ in a finite number of $n$-simplices can be constructed in FO+POLY. The $n$-simplices will be represented by $n+1$ independent points.

We first identify the hyperplanes which have an $(n-1)$-dimensional intersection with the boundary of the convex set $c$. Let $\vec{e}_1,\dots,\vec{e}_k$ be the extremal points of $c$. Let $\texttt{onboundary}$ be the FO+POLY formula which selects the tuples in $\texttt{hyperplanes}(D)$ with this property. Next, let $\texttt{sameface}$ be an FO+POLY formula such that $\texttt{face}(\vec{e},\vec{v},d)$ if and only if $\vec{e}$ is an extremal point of $c$, $(\vec{v},d) \in \texttt{onboundary}(\vec{e}_1,\dots,\vec{e}_k)$, and $\vec{e} \in \{\vec{x} \in \mathbf{R}^n \mid \vec{v}\cdot\vec{x} = d\}$. In this way, we can group the extremal points of $c$ such that each group corresponds to a single face of the convex cell $c$.

For each face of $c$, we now project the extremal points corresponding to this face to $\mathbf{R}^{n-1}$ such that they are the extremal points of a convex set in $\mathbf{R}^{n-1}$. Thus, if $\texttt{face}(\vec{x}_1,\vec{v},d,\vec{e}_1,\dots,\vec{e}_k) \wedge \cdots \wedge \texttt{face}(\vec{x}_\ell,\vec{v},d,\vec{e}_1,\dots,\vec{e}_k)$, then we obtain extremal points of a convex set in $\mathbf{R}^{n-1}$ as follows: Let $i \in \{1,\dots,n\}$ be such that $\{\vec{x} \in \mathbf{R}^n \mid x_i = 0\}$ is not perpendicular to $\{\vec{x} \in \mathbf{R}^n \mid \vec{v}\cdots\vec{x} = d\}$ (this can be easily expressed in FO+POLY). Then consider the projection $\pi_i : \mathbf{R}^n \rightarrow \mathbf{R}^{n-1}$ defined as $\pi_i(x_1,\dots,x_n) \mapsto (x_1,\dots,x_{i-1},x_{i+1},\dots,x_n)$ and apply this map on $\vec{x}_1,\dots,\vec{x}_\ell$.

**Algorithm** TRIANGULATE-IN-$n$-DIMENSIONS
**Input:** The extremal points $\vec{e}_1,\dots,\vec{e}_k$ of a convex set $c$ in $\mathbf{R}^n$.
**Output:** A finite number of $n$-simplices forming a decomposition of $c$.
**Method:**

1. Compute the pairs $(\vec{v},d) \in \texttt{onboundary}(\vec{e}_1,\dots,\vec{e}_k)$.
2. For each $(\vec{v},d) \in \texttt{onboundary}(\vec{e}_1,\dots,\vec{e}_k)$ do the following:
   (a) Compute $\texttt{face}(\vec{x},\vec{v},d,\vec{e}_1,\dots,\vec{e}_k)$.
   (b) Find an $i$ as described above and call TRIANGULATE-IN-$(n-1)$-DIMENSIONS$(\pi_i(\texttt{face}(\vec{v},d,\vec{e}_1,\dots,\vec{e}_k)))$.
3. Select a point $\vec{p}_{n+1}$ in the interior of $c$.
4. Output the $(n+1)$-tuples $(\vec{p}_1,\dots,\vec{p}_n,\vec{p}_{n+1})$, where $(\vec{p}_1,\dots,\vec{p}_n)$ is an $n$-tuple in the result of the calls of TRIANGULATE-IN-$(n-1)$-DIMENSIONS in step 2(b).

We now define the FO+POLY formula $\texttt{simplexdecomp}$ over $\mathcal{S}$ such that $\texttt{simplexdecomp}(D)$ is a decomposition into $n$-simplices of $S^D$ for any polynomial constraint database $D$ over $\{S\}$. Let $\texttt{triang}$ be a formula which expresses the algorithm TRIANGULATE-IN-$n$-DIMENSIONS. Then

$$\begin{aligned}\texttt{simplexdecomp}(\vec{x}_1,\dots,\vec{x}_{n+1}) \equiv\ & \exists\vec{y}(\texttt{unique}(\vec{y}) \\ & \wedge \texttt{triang}(\texttt{extremal})(\vec{x}_1,\dots,\vec{x}_{n+1},\vec{y})).\end{aligned}$$

Let $(\vec{p}_1, \ldots, \vec{p}_{n+1})$ be an $n$-simplex points. Let $\vec{r}_i = \vec{p}_i - \vec{p}_1$ for $i = 2, \ldots, n+1$, and let $G$ be the $n \times n$ matrix whose rows contain the coordinates of the vectors $\vec{r}_j$ for $1 \leqslant j \leqslant n$. Then by the Gram determinant formula [37], the volume of $(\vec{p}_1, \ldots, \vec{p}_{n+1})$ is equal to

$$\frac{|\det(GG^{\mathrm{t}})|^{\frac{1}{2}}}{n!},$$

where $G^{\mathrm{t}}$ is the transpose of $G$. Hence, the volumes of the simplices are expressible by an FO+Poly formula, which we will denote by `volsimplex`.

Finally, define

$$\begin{aligned}\Psi(y) \equiv [&\mathrm{TC}_{x,s;x',s'} s = \exists \vec{p}_1, \ldots, \exists \vec{p}_{n+1}, \exists \vec{q}_1, \ldots, \exists \vec{q}_{n+1} \\ &\mathtt{volsimplex}(\vec{p}_1, \ldots, \vec{p}_{n+1}) \wedge s' = \mathtt{volsimplex}(\vec{q}_1, \ldots, \vec{q}_{n+1}) \\ &\wedge \mathtt{successor}(\vec{q}_1, \ldots, \vec{q}_{n+1}, \vec{p}_1, \ldots, \vec{p}_{n+1}) \\ &\wedge \mathtt{simplexdecomp}(\vec{p}_1, \ldots, \vec{p}_{n+1}) \wedge \mathtt{simplexdecomp}(\vec{q}_1, \ldots, \vec{q}_{n+1}) \\ &\wedge x' = x + s](0, v_1, y, v_\ell),\end{aligned}$$

where `successor` is a successor relation defined on the $n$-simplices in the decomposition into simplices $\mathtt{simplexdecomp}(D)$, and where $v_1$ and $v_\ell$ are, respectively, the volume of the first and last simplex according to this successor relation. The total volume of $S^D$ is then given by

$$\mathtt{volume}(v) \equiv \exists y \Psi(y) \wedge v = y + v_\ell,$$

with $v_\ell$ as above. $\square$

**Acknowledgments.** We would like to thank the referees for their comments which contributed significantly towards the readability of the paper.

## REFERENCES

[1] D. Abel and B. C. Ooi, eds., *Advances in Spatial Databases—3rd Symposium (SSD'93)*, Lecture Notes in Comput. Sci. 692, Springer-Verlag, Berlin, New York, 1993.
[2] S. Basu, R. Pollack, and M.-F. Roy, *On the combinatorial and algebraic complexity of quantifier elimination*, J. ACM, 43 (1996), pp. 1002–1046.
[3] S. Basu, R. Pollack, and M.-F. Roy, *Algorithms in Real Algebraic Geometry*, Algorithms Comput. Math. 10, Springer-Verlag, Berlin, New York, 2003.
[4] M. Benedikt and L. Libkin, *Safe constraint queries*, SIAM J. Comput., 29 (2000), pp. 1652–1682.
[5] M. Benedikt and L. Libkin, *Aggregate operators in constraint query languages*, J. Comput. System Sci., 64 (2002), pp. 626–654.
[6] J. Bochnak, M. Coste, and M.-F. Roy, *Real Algebraic Geometry*, Ergeb. Math. Grenzgeb. (3) 36, Springer-Verlag, Berlin, 1998.
[7] H. Brakhage, *Topologische Eigenschaften algebraischer Gebilde über einem beliebigen reell-abgeschlossenen Konstantenkörper*, Dissertation, Universität Heidelberg, Heidelberg, Germany, 1954.
[8] A. P. Buchmann, O. Günther, T. R. Smith, and Y.-F. Wang, eds., *Design and Implementation of Large Spatial Databases—First Symposium (SSD'89)*, Lecture Notes in Comput. Sci. 409, Springer-Verlag, Berlin, New York, 1989.
[9] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarkopf, *Computational Geometry*, Springer-Verlag, Berlin, New York, 1997.
[10] F. Dumortier, M. Gyssens, L. Vandeurzen, and D. Van Gucht, *On the decidability of semilinearity for semialgebraic sets and its implications for spatial databases*, J. Comput. System Sci., 58 (1999), pp. 535–571.

[11] H. D. Ebbinghaus and J. Flum, *Finite Model Theory*, Springer-Verlag, Berlin, 1995.

[12] M. J. Egenhofer and J. R. Herring, eds., *Advances in Spatial Databases—4th Symposium* (*SSD'*95), Lecture Notes in Comput. Sci. 951, Springer-Verlag, Berlin, New York, 1995.

[13] F. Geerts, *Linear approximation of semi-algebraic spatial databases using transitive closure logic, in arbitrary dimension*, in Proceedings of the 8th International Workshop on Databases and Programming Languages, Lecture Notes in Comput. Sci., Springer-Verlag, 2002, pp.

[14] F. Geerts, *Expressing the box cone radius in the relational calculus with real polynomial constraints*, Discrete Comput. Geom., 30 (2003), pp. 607–622.

[15] F. Geerts and B. Kuijpers, *Expressing topological connectivity of spatial databases*, in Research Issues in Structured and Semistructured Database Programming. Proceedings of the 7th International Workshop on Database Programming Languages, Lecture Notes in Comput. Sci. 1949, Springer-Verlag, Berlin, New York, 1999, pp. 224–238.

[16] F. Geerts and B. Kuijpers, *Linear approximation of planar spatial databases using transitive-closure logic*, in Proceedings of the 19th ACM Symposium on Principles of Database Systems, ACM, New York, 2000, pp. 126–135.

[17] F. Geerts and B. Kuijpers, *On the decidability of termination of query evaluation in transitive-closure logics for polynomial constraint databases*, Theoret. Comput. Sci., 336 (2005), pp. 125–151.

[18] S. Grumbach and G. Kuper, *Tractable recursion over geometric data*, in Proceedings of the 3rd Conference on Principles and Practice of Constraint Programming, G. Smolka, ed., Lecture Notes in Comput. Sci. 1330, Springer-Verlag, Berlin, New York, 1997, pp. 450–462.

[19] S. Grumbach, P. Rigaux, M. Scholl, and L. Segoufin, *DEDALE, a spatial constraint database*, in Proceedings of Database Programming Languages (DBPL '97), S. Cluet and R. Hull, eds., Lecture Notes in Comput. Sci. 1369, Springer-Verlag, Berlin, New York, 1998, pp. 38–59.

[20] S. Grumbach, P. Rigaux, and L. Segoufin, *The DEDALE system for complex spatial queries*, in Proceedings of the ACM International Conference on Management of Data (SIGMOD '98), L. M. Haas and A. Tiwary, eds., ACM, New York, 1998, pp. 213–224.

[21] S. Grumbach and J. Su, *Towards practical constraint databases*, in Proceedings of the 15th ACM Symposium on Principles of Database Systems (PODS '96), ACM, New York, 1996, pp. 28–39.

[22] S. Grumbach and J. Su, *Queries with arithmetical constraints*, Theoret. Comput. Sci., 173 (1997), pp. 151–181.

[23] V. Guillemin and A. Pollack, *Differential Topology*, Prentice-Hall, Englewood Cliffs, NJ, 1974.

[24] O. Gunther and H.-J. Schek, eds., *Advances in Spatial Databases—2nd Symposium* (*SSD*'91), Lecture Notes in Comput. Sci. 525, Springer-Verlag, Berlin, New York, 1991.

[25] R. H. Güting, ed., *Advances in Spatial Databases—6th Symposium* (*SSD*'99), Lecture Notes in Comput. Sci. 1651, Springer-Verlag, Berlin, New York, 1999.

[26] R. Hardt, *Triangulation of subanalytic sets and proper light subanalytic maps*, Invent. Math., 38 (1976/77), pp. 207–217.

[27] P. C. Kanellakis, G. M. Kuper, and P. Z. Revesz, *Constraint query languages*, J. Comput. System Sci., 51 (1995), pp. 26–52.

[28] P. Koiran, *Approximating the volume of definable sets*, in Proceedings of the 36th IEEE Symposium on Foundations of Computer Science (FOCS), IEEE, Los Alamitos, CA, 1995, pp. 134–141.

[29] S. Kreutzer, *Operational semantics for fixed-point logics on constraint databases*, in Proceedings of the 8th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning, R. Nieuwenhuis and A. Voronkov, eds., Lecture Notes in Comput. Sci. 2250, Springer, Berlin, 2002, pp. 470–484.

[30] S. Kreutzer, *Query languages for constraint databases: First-order logic, fixed-points, and convex hulls*, in Proceedings of the 9th International Conference on Database Theory, J. Van den Bussche and V. Vianu, eds., Lecture Notes in Comput. Sci. 1973, Springer-Verlag, Berlin, New York, 2001, pp. 248–262.

[31] B. Kuijpers, J. Paredaens, M. Smits, and J. Van den Bussche, *Termination properties of spatial datalog*, in Logic in Databases (LID '96), D. Pedreschi and C. Zaniolo, eds., Lecture Notes in Comput. Sci. 1154, Springer-Verlag, Berlin, New York, 1996, pp. 101–116.

[32] 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), pp. 1530–1555.

[33] B. Kuijpers and M. Smits, *On expressing topological connectivity in spatial datalog*, in Proceedings of the 2nd Workshop on Constraint Databases and Applications, V. Gaede, A. Brodsky, O. Gunter, D. Srivastava, V. Vianu, and M. Wallace, eds., Lecture Notes in Comput. Sci. 1191, Springer-Verlag, Berlin, New York, 1997, pp. 116–133.
[34] G. M. Kuper, J. Paredaens, and L. Libkin, eds., *Constraint Databases*, Springer-Verlag, Berlin, New York, 2000.
[35] J. M. Lee, *Introduction to Topological Manifolds*, Graduate Texts in Math. 202, Springer-Verlag, New York, 2000.
[36] E. E. Moise, *Geometrical Topology in Dimensions* 2 *and* 3, Springer-Verlag, Berlin, New York, 1977.
[37] J. O'Rourke, *Computational Geometry in C*, Cambridge University Press, Cambridge, UK, 1998.
[38] F. P. Preparata and M. I. Shamos, *Computational Geometry—An Introduction*, Springer-Verlag, Berlin, New York, 1985.
[39] E. Rannou, *The complexity of stratification computation*, Discrete Comput. Geom., 19 (1998), pp. 47–79.
[40] E. Rannou, *Personal communication*, 2000.
[41] J. Renegar, *On the computational complexity and geometry of the first-order theory of the reals*, J. Symbolic Comput., 13 (1992), pp. 255–352.
[42] M. Scholl and A. Voisard, eds., *Advances in Spatial Databases—5th Symposium* (*SSD*'97), Lecture Notes in Comput. Sci. 1262, Springer-Verlag, Berlin, New York, 1997.
[43] A. Tarski, *A Decision Method for Elementary Algebra and Geometry*, 2nd ed., University of California Press, Berkeley and Los Angeles, 1951.
[44] L. van den Dries, *Tame Topology and O-minimal Structures*, Cambridge University Press, Cambridge, UK, 1998.
[45] L. van den Dries and C. Miller, *Geometric categories and O-minimal structures*, Duke Math. J., 82 (1996), pp. 497–540.
[46] L. Vandeurzen, *Logic-Based Query Languages for the Linear Constraint Database Model*, Ph.D. thesis, Limburgs Universitair Centrum (LUC), Diepenbeek, Belgium, 1999.
[47] L. Vandeurzen, M. Gyssens, and D. Van Gucht, *On the desirability and limitations of linear spatial query languages*, in Advances in Spatial Databases—4th Symposium (SSD '95), M. J. Egenhofer and J. R. Herring, eds., Lecture Notes in Comput. Sci. 951, Springer-Verlag, Berlin, New York, 1995, pp. 14–28.
[48] L. Vandeurzen, M. Gyssens, and D. Van Gucht, *An expressive language for linear spatial database queries*, in Proceedings of the 17th ACM Symposium on Principles of Database Systems (PODS '98), ACM, New York, 1998, pp. 109–118.
[49] A. J. Wilkie, *On defining* $C^\infty$, J. Symbolic Logic, 59 (1994), p. 344.
[50] A. J. Wilkie, *A theorem of the complement and some new o-minimal structures*, Selecta Math. (N. S.), 5 (1999), pp. 397–421.
[51] G. M. Ziegler, *Lectures on Polytopes*, Graduate Texts in Math. 152, Springer-Verlag, Berlin, New York, 1998.