Linear Approximation of Semi-Algebraic Spatial Databases Using Transitive Closure Logic, in Arbitrary Dimension

Floris Geerts

University of Limburg, Belgium
floris.geerts@luc.ac.be

Abstract. We consider n-dimensional semi-algebraic spatial databases. We compute in first-order logic extended with a transitive closure operator, a linear spatial database which characterizes the semi-algebraic spatial database up to a homeomorphism. In this way, we generalize our earlier results to semi-algebraic spatial databases in arbitrary dimensions, our earlier results being true for only two dimensions.
Consequently, we can prove that first-order logic with a transitive closure operator extended with stop conditions, can express all Boolean topological queries on semi-algebraic spatial databases of arbitrary dimension.

1 Introduction

Conceptually, spatial databases are possibly infinite sets of points in the n-dimensional Euclidean space $\mathbf{R}^n$. The framework of constraint databases, introduced in 1990 by Kanellakis, Kuper and Revesz [14, 17], provides an elegant and powerful model for spatial databases. One distinguishes between *semi-algebraic spatial databases*, which store semi-algebraic sets, and *linear spatial databases*, which store semi-linear sets.

First-order logic FO over these databases yields a query language with rather poor expressive power. Its inability to define queries relying on recursion, suggests the extension of these query languages with a recursion mechanism. In this paper we consider the extension of first-order logic with transitive closure operators. Other attempts to introduce recursion mechanisms to first-order logics can be found in [9] and [15, 16]. A less traditional extension of FO which can express some recursive queries is the Path Logic [3]. This logic is able to express recursive queries like the query which ask whether a database is connected, or whether two points are path connected, but the exact expressive power of this logic is not known.

First-order logic extended with a transitive closure operator, denoted by FO+TC, shares with most programming languages the disadvantage that the evaluation of its formulas is not guaranteed to terminate. However, whenever the evaluation of a formula terminates, it evaluates to an output within the constraint model.

It is known that FO+TC with some kind of stopping condition is computationally complete on linear spatial databases where only rational coefficients are involved [8]. Indeed, this logic, denoted by FO+TCS, can easily seen to be complete on finite databases. The completeness of this logic on all linear spatial databases, is then obtained using a finite representation of these databases given by Vandeurzen et al. [20]. Since both the encoding and decoding between a linear spatial database and this finite representation are expressible in FO+TCS, we may conclude the completeness of this logic on linear spatial databases.

The use of this finite representation to obtain expressibility results for linear spatial database is ubiquitous [4, 9, 15, 16, 21]. We show that we can construct in FO+TC, a linear spatial database (and hence also a finite representation) which is homeomorphic to a given semi-algebraic spatial database input. We call this the *linearization* of a semi-algebraic spatial database. We prove that this construction terminates on all spatial database inputs. This was only known in the plane [8], and the generalization to arbitrary dimensions is far from trivial and uses results of differential topology [12], of Shiota [19] and of Rannou [18].

A direct consequence is that FO+TCS is computationally complete for Boolean topological queries on semi-algebraic spatial database. This is rather remarkable because a transitive closure, used as a recursion mechanism, is weaker than, e.g., a while-loop.

As alternative approach for expressing all topological properties of semi-algebraic spatial databases, one could add a generalized quantifier for each of these properties to FO. The query language obtained in this way is closed, i.e., every query evaluates to an output within the constraint model [3]. However, from a programming language point of view, it is more desirable to extend FO with a single programming feature (like a transitive closure operator), then to extend FO with uncountably many new features (since there are uncountably many topological properties).

The paper is organized as follows. In Sect. 2, we introduce the definitions of spatial databases, queries, and define the transitive closure logics. Section 3 shows that first-order logic is able to extract a large amount of topological information from a spatial database. This information is then put into use in Sect. 4 where we construct the linearization of a semi-algebraic spatial database. The completeness of the transitive closure logic for Boolean topological queries, is then obtained in Sect. 5. We conclude the paper with some remarks in Sect. 6

2 Preliminaries

A *semi-algebraic set in* $\mathbf{R}^n$ is a set of points that can be defined as a Boolean combination (union, intersection and complement) of sets of the form

$$\{(x_1, \ldots, x_n) \mid P(x_1, \ldots, x_n)\ \sigma\ 0\},$$

where $P(x_1, \ldots, x_n)$ is a multi-variate polynomial in the variables $x_1, \ldots, x_n$ with algebraic coefficients and $\sigma \in \{>, \leq\}$. A *database schema* $\mathcal{S}$ is a finite set of relation names, each with a given arity. A *semi-algebraic spatial database* over

$\mathcal{S}$ assigns to each $S \in \mathcal{S}$ a semi-algebraic set S^D in $\mathbf{R}^k$, where k is the arity of S. If only linear polynomials are involved, one speaks about *semi-linear sets* and *linear spatial databases*. A *k-ary query over $\mathcal{S}$* is a function mapping each database over $\mathcal{S}$ to a semi-algebraic set in $\mathbf{R}^k$.

As query language we use *first-order logic* (FO) over the vocabulary $(+, \cdot, 0, 1, <)$ expanded with the relation names in $\mathcal{S}$. A formula $\varphi(x_1, \ldots, x_k)$ expresses a k-ary query defined by

$$\varphi(D) := \{(x_1, \ldots, x_k) \mid \langle \mathbf{R}, D\rangle \models \varphi(x_1, \ldots, x_k)\},$$

for any database D. Note that $\varphi(D)$ is always semi-algebraic because all relations in D are; indeed, by Tarski's theorem, relations that are first-order definable on the real ordered field are precisely the semi-algebraic sets.

As example of a query expressed in FO is the following:

$$(\exists \varepsilon > 0)(\forall x'_1) \cdots (\forall x'_n)(\|\boldsymbol{x} - \boldsymbol{x}'\| < \varepsilon \rightarrow S(x'_1, \ldots, x'_n)),$$

where $\boldsymbol{x} = (x_1, \ldots, x_n)$, and $\boldsymbol{x}' = (x'_1, \ldots, x'_n)$. This query maps the set S to its interior. However, not every query is first-order expressible: the query which asks whether a spatial database is connected is not expressible in FO. This result and other results related to spatial databases have recently been collected in a single volume [17, Chapters 3 and 4].

We now define two extensions of first-order logic FO, both able to compute the transitive closure of a spatial database. We already introduced these logics in the context of planar spatial databases [8], but their definition did not rely on the planarity condition. These logics are not appealing from a query language point of view, since one can easily define queries with non-semi-algebraic output, hence leaving the constraint database framework. This is why we look at these recursive extensions from a programming language point of view. As in almost every programming language, programs can be written that don't halt, but it is the programmer's task to write terminating programs.

It is for this reason that we will define our logics as subclasses of programs of FO+While, which is FO extended with the standard programming features of assignment statements, sequential compositions, and while-loops [13]. More specifically, a program in FO+While over a schema $\mathcal{S}$, is a sequence of *assignment statements* and *while-loops*. A sufficient supply of *relation variables* is assumed, which are interpreted as new relation names not present in the given schema $\mathcal{S}$.

An assignment statement is an expression of the form

$$X := \{\boldsymbol{x} \mid \varphi(\boldsymbol{x})\},$$

where X is a relation variable of arity equal to the length of the vector $\boldsymbol{x}$ of variables, and φ is a formula in FO extended with relation variables introduced in previously occurring assignment statements.

A while-loop is an expression of the form

$$\textsc{While} \quad \varphi \quad \textsc{Do} \quad P \quad \textsc{Od}.$$

where φ is as above, but *without* free variables, and P is in turn an FO+While program, which is called the *body* of the loop.

The semantics of the program applied to an input spatial database D is the operational, step-by-step execution. The effect of an assignment statement is to evaluate the query on the right-hand side on the input spatial database D augmented with the values of the previously assigned relation variables, and to assign the result of this evaluation to a relation variable on the left-hand side. The effect of a while-loop is to execute the body as long as φ evaluates to true.

It is a consequence of the closure of the logic FO, that every program expresses a computable query in FO, which however may be only partially defined because of non-terminating loops.

We now state the definitions of the programming languages FO+TC and FO+TCS.

Let R be a relation variable of even arity $2k$. The following program TC:

$X := R; Y := \emptyset;$ (Y has arity $2k$)
While $Y \neq X$ Do
$Y := X;$
$X := X \cup \{(x_1, \ldots, x_k, y_1, \ldots, y_k) \mid \exists z_1 \ldots \exists z_k (X(x_1, \ldots, x_k, z_1, \ldots, z_k) \wedge X(z_1, \ldots, z_k, y_1, \ldots, y_k)\}$
Od.

computes the transitive closure of the relation R. We shall use the notation $\mathrm{TC}(R)$ for the relation name to which we assign the result of executing the program TC on the relation R in case the while-loop halts, and let $\mathrm{TC}(R)$ be undefined otherwise.

Definition 1. *The transitive closure logic* FO+TC *is the class of* FO+While*-programs in which only programs of the above form are allowed on relation names in the schema $\mathcal{S}$ extended with relation names introduced in previously occurring assignment statements.*

Remark that inside the transitive closure no free variables are allowed! This makes FO+TC a rather weak logic, but it is sufficient for our purposes.

As an example, let $\mathcal{S} = (S)$ and let D be a linear database over $\mathcal{S}$. Define the $2n$-ary relation *Path*,

$$Path(D) = \{(\boldsymbol{r}, \boldsymbol{s}) \mid (\exists \lambda)(0 \leq \lambda \leq 1 \wedge (\forall t)(\boldsymbol{t} = \lambda \cdot \boldsymbol{r} + (1 - \lambda) \cdot \boldsymbol{s} \rightarrow S^D(\boldsymbol{t}))\},$$

where $\boldsymbol{s}, \boldsymbol{t} \in \mathbf{R}^n$. It is easy to prove that the program TC terminates on every linear database D. Moreover, a pair of points $\boldsymbol{p}$ and $\boldsymbol{q}$ is in the relation $\mathrm{TC}(Path(D))$ if and only if they belong to the same connected component of D. Hence the connectivity of a linear database is expressible in FO+TC.

We now introduce an extension of FO+TC which allows some general FO-formula without free variables σ in the condition in the while-loop. More specific,

consider the following program TCS:

$X := R; Y := \emptyset;$ (Y has arity $2k$)
WHILE $Y \neq X \wedge \neg\sigma(X)$ DO
$Y := X;$
$X := X \cup \{(x_1, \ldots, x_k, y_1, \ldots, y_k) \mid \exists z_1 \ldots \exists z_k (X(x_1, \ldots, x_k, z_1, \ldots, z_k) \wedge X(z_1, \ldots, z_k, y_1, \ldots, y_k)\}$
OD.

The condition σ is called the *stop condition*. This program computes either the transitive closure of the relation R, in case the condition σ is never satisfied, or it computes the first p-stages of the transitive closure of the relation of R, where p is the minimal number of cycles of the while-loop such that the condition σ is satisfied. We shall denote by $\mathrm{TCS}(R)$ the relation name to which we assign the result of executing the program TCS on the relation R, in case the while-loop halts, and let $\mathrm{TCS}(R)$ be undefined, otherwise. The relation name X is reserved for the current stage in the computation inside a while-loop.

Definition 2. *The transitive closure logic with stop condition* FO+TCS *is the class of* FO+WHILE*-programs in which only programs of the above form are allowed on relation names in the schema $\mathcal{S}$ extended with relation names introduced in previously occurring assignment statements.*

Note again that no free variables are allowed inside the transitive closure. It is then also surprising that this logic is already computationally complete for Boolean topological queries. As an example, let $R := \{(x, y) \mid y = 2x\}$ whose transitive closure $\mathrm{TC}(R)$ is not a semi-algebraic set. Take as stop condition the formula $\sigma(X) := X(1, 8)$. The while-loop then terminates after three cycles, and the result is a semi-linear set consisting of three lines through the origin.

3 Local Topological Characterization

A well-known property of semi-algebraic sets, is that locally around each point, a semi-algebraic set is homeomorphic to a cone. Let A be a semi-algebraic set in $\mathbf{R}^n$. A *cone radius of A in a point $\boldsymbol{p}$*, is a radius around $\boldsymbol{p}$ in which this behavior shows. A first question one can ask, is whether a *cone radius query* is expressible in FO. This query must return for a semi-algebraic set A, a set of pairs $(r, \boldsymbol{p})$ giving for every point $\boldsymbol{p}$ a cone radius r of A in $\boldsymbol{p}$. A second question one can ask is whether there exists a *uniform cone radius* of a semi-algebraic set A. By a uniform cone radius, we mean a real number which is a cone radius of A in every point of A. We will answer these questions in the following sections.

3.1 Expressibility of the Cone Radius in FO

Let $A \subseteq \mathbf{R}^n$ be a semi-algebraic set and $\boldsymbol{p} \in \mathbf{R}^n$. We define the *cone with base A and top $\boldsymbol{p}$* as the union of all closed line segments between $\boldsymbol{p}$ and points in A. We

denote this set by $\mathrm{Cone}(A, \boldsymbol{p})$. For a point $\boldsymbol{p} \in \mathbf{R}^n$ and $\varepsilon > 0$, denote the closed ball centered at $\boldsymbol{p}$ with radius ε by $B^n(\boldsymbol{p}, \varepsilon)$, and denote its boundary sphere by $S^{n-1}(\boldsymbol{p}, \varepsilon)$. We denote the closure of a set A by $\mathrm{clo}(A)$, and the interior of A by $\mathrm{int}(A)$. The following theorem formalizes the property of semi-algebraic set mentioned above. The boundary of A, $\mathrm{clo}(A) - A$, will be denoted by ∂A.

Theorem 1 ([1, 5]). *Let $A \subseteq \mathbf{R}^n$ be a semi-algebraic set and $\boldsymbol{p}$ a point of A. Then there is a real number $\varepsilon > 0$ such that the intersection $A \cap B^n(\boldsymbol{p}, \varepsilon)$ is homeomorphic to the set* $\mathrm{Cone}(A \cap S^{n-1}(\boldsymbol{p}, \varepsilon), \boldsymbol{p})$.

A naive way of checking in first-order logic whether a real number r is a cone radius of A in $\boldsymbol{p}$, is just testing if the sets $\mathrm{Cone}(A \cap S^{n-1}(\boldsymbol{p}, \varepsilon), \boldsymbol{p})$ and $A \cap B^n(\boldsymbol{p}, \varepsilon)$ are homeomorphic. However, it is known that the query which tests whether two semi-algebraic sets in $\mathbf{R}^n$ (with $n > 1$) are homeomorphic, is not in FO [2, 10, 11].

We showed rather ad hoc, that a cone radius query is first-order expressible for planar spatial databases [7]. The following result extends this to n-dimensional semi-algebraic spatial databases.

Theorem 2. *Let A be a semi-algebraic set in $\mathbf{R}^n$. There exists a cone radius query*

$$\varphi_{radius} : A \mapsto \{(r, \boldsymbol{p}) \mid r \text{ is a cone radius of } A \text{ in } \boldsymbol{p}\},$$

which is expressible in FO.

Proof (sketch). We first sketch the prove in case A is a closed semi-algebraic set which has a tangent space in every of its points, and then sketch how the general case can be treated.

Case 1. A is closed and A has a tangent space in every of its points.
Let $\boldsymbol{p} \in \mathbf{R}^n$ and let $f_{\boldsymbol{p}} : A \to \mathbf{R} : \boldsymbol{x} \mapsto \|\boldsymbol{x} - \boldsymbol{p}\|^2$, where $\|\cdot\|$ denote the Euclidean distance. The tangent space of A in $\boldsymbol{x}$, denoted by $T_{\boldsymbol{x}}A$, is the secant limits set

$$T_{\boldsymbol{x}}A := \bigcap_{\eta > 0} \mathrm{clo}(\{\lambda(\boldsymbol{u} - \boldsymbol{v}) \in \mathbf{R}^n \mid \lambda \in \mathbf{R} \wedge \boldsymbol{u}, \boldsymbol{v} \in A \cap B^n(\boldsymbol{x}, \eta)\}),$$

and hence the query $\varphi_{\mathrm{tangent}} : A \mapsto (\boldsymbol{x}, T_{\boldsymbol{x}}A)$ is expressible in FO (see [18] for more details). The mapping $f : A \to \mathbf{R}$ induces a linear mapping, called the *differential* between tangent spaces, $df_{\boldsymbol{p}} : T_{\boldsymbol{x}}A \to \mathbf{R}$ which maps (as it can be easily verified) a tangent vector $\boldsymbol{v} \in T_{\boldsymbol{x}}A$ to the scalar product $2\boldsymbol{v} \cdot (\boldsymbol{x} - \boldsymbol{p}) \in \mathbf{R}$.

We now use Thom's First Isotopy Lemma [19, Theorem II.6.2], which says (adapted to our setting) that **if** (i) in every point of A the tangent space exists (which is the case by assumption), (ii) the mapping $f_{\boldsymbol{p}} : A \to \mathbf{R}$ is continuous and has continuous derivatives (this is the case), (iii) the set $f_{\boldsymbol{p}}^{-1}([c, d])$ is a compact set for any closed interval $[c, d]$ of $\mathbf{R}$ (this is true since A is closed), and (iv)

$$f_{\boldsymbol{p}} | (A \cap \mathrm{int}(B^n(\boldsymbol{p}, b) - \{\boldsymbol{p}\})) \longrightarrow (0, b) \subseteq \mathbf{R} \tag{1}$$

has no critical points (this will shortly be explained), **then** there exists a homeomorphism

$$h' : (A \cap S^{n-1}(\boldsymbol{p}, c)) \times (0, c] \to A \cap B^n(\boldsymbol{p}, c) - \{\boldsymbol{p}\}, \tag{2}$$

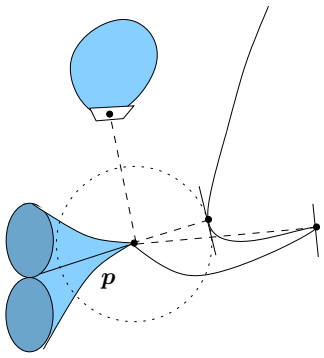

Fig. 1. Illustration of the cone radius query of Theorem 2.

where $c \in (0, b)$. Moreover, there exists such a homeomorphism h' for any $c \in (0, b)$. Since the cylindrical set at the left in (2) is homeomorphic to the cone set $\mathrm{Cone}(A \cap S^{n-1}(\boldsymbol{p}, c), \boldsymbol{p}) - \{\boldsymbol{p}\}$ in a straightforward way, we obtain a homeomorphism h between

$$h : A \cap B^n(\boldsymbol{p}, c) - \{\boldsymbol{p}\} \to \mathrm{Cone}(A \cap S^{n-1}(\boldsymbol{p}, c), \boldsymbol{p}) - \{\boldsymbol{p}\}.$$

This homeomorphism can easily be extended to $A \cap B^n(\boldsymbol{p}, c)$ by defining $h(\boldsymbol{p}) = \boldsymbol{p}$. In this way, we have shown that any $c \in (0, b)$ is in fact a cone radius of A in the point $\boldsymbol{p}$. To show that there exists a cone radius query which is expressible in FO, we need to show that for each point $\boldsymbol{p}$, an interval $(0, b)$ is expressible in FO, satisfying condition (1), the other conditions being trivially satisfied. The *critical points* of $f_{\boldsymbol{p}}|A$ are defined as the points $\boldsymbol{x}$ in A such that the differential $df_{\boldsymbol{p}}|A$ is not surjective. From this we can deduce that $\boldsymbol{x}$ is a critical point of $f_{\boldsymbol{p}}|A$ if and only if $2\boldsymbol{v} \cdot (\boldsymbol{x} - \boldsymbol{p}) = 0$ for all tangent vectors $\boldsymbol{v} \in T_{\boldsymbol{x}}A$. In other words, $\boldsymbol{x}$ is a critical point of $f_{\boldsymbol{p}}|A$ if and only if $T_{\boldsymbol{x}}A$ is orthogonal to the vector $\boldsymbol{x} - \boldsymbol{p}$. This can clearly be expressed in FO. Define the set of *critical values* of $f_{\boldsymbol{p}}|A$ as the image by $f_{\boldsymbol{p}}$ of the critical points. It is a consequence of Sard's Theorem [1, Theorem 2.5.11] that this is a finite set. We therefore, define $\varphi_{\mathrm{radius}}$ as

$$\varphi_{\mathrm{radius}} : A \mapsto \{(r', \boldsymbol{p}) \mid \forall \text{ critical values } r \text{ of } f_{\boldsymbol{p}} \Rightarrow r' < r\}.$$

This concludes the proof for the case that A is closed and in all points of A the tangent space exists. We have illustrated the construction in the proof in Figure 1. In this figure, we have drawn a spatial database centered around the point $\boldsymbol{p}$, and also identified points $\boldsymbol{q}$ whose tangent lines are orthogonal to the vector $\boldsymbol{p} - \boldsymbol{q}$. We have drawn dotted circles through these points. Note that the topology of the intersection of a circle with center at the point $\boldsymbol{p}$, and the spatial database, is unchanged between two consecutive dotted circles, as is predicted by Thom's Isotopy Lemma.

Case 2. A is an arbitrary semi-algebraic set in $\mathbf{R}^n$.

We then consider a Whitney stratification (decomposition into a finite number of sets, called strata) of the closure of A, such that A is the union of connected components of these strata. The strata are semi-algebraic sets satisfying Whitney's condition and in each point of a stratum the tangent space exists [18]. We then find an interval $(0, b)$ such that condition (iv) is satisfied simultaneously for all strata. Thom's First Isotopy Lemma for sets admitting a Whitney stratification (like semi-algebraic sets), guarantees again the correctness of this procedure. To show the expressibility in FO, it is sufficient to prove that a Whitney stratification is expressible in first-order logic. This can be shown using results from Rannou [18] by not expecting the strata to be connected and not expecting the so-called frontier condition on the stratification. □

3.2 Uniform Cone Radius Decomposition

Although every point of a semi-algebraic set has a cone radius which is strictly greater than zero (Theorem 1), we are now interested in finding a *uniform cone radius* of a semi-algebraic set. We define the uniform cone radius of semi-algebraic set $A \subseteq \mathbf{R}^n$ as a real number $\varepsilon_A > 0$ such that ε_A is a cone radius of A in all its points. As shall be clear from the next Section, this radius offers the right information for constructing a *linearization of* A, i.e., a linear set homeomorphic to A.

A first observation is that the uniform cone radius of a semi-algebraic set, or even a semi-linear set not always exists. Consider for example two lines in $\mathbf{R}^3$ which intersect in a point $\boldsymbol{p}$. It is clear that the cone radius of points approaching the point $\boldsymbol{p}$ converge to zero. We define the ε-neighborhood of a semi-algebraic set $A \subset \mathbf{R}^n$ as

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

Theorem 3. *Let A be a semi-algebraic set in $\mathbf{R}^n$. Then there exists a finite decomposition into semi-algebraic sets* $\mathrm{clo}(A) = A_d \cup A_{d-1} \cdots A_\ell$, *satisfying* $\dim(A_\ell) < \cdots < \dim(A_{d-1}) < \dim(A_d) = \dim(A)$, *and such that for any tuple $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)$ of positive real numbers, the sets*

$$A_i - \bigcup_{j=\ell}^{i-1} A_j^{\varepsilon_j}, \quad \textit{for } i = \ell, \ldots, d, \tag{3}$$

all have a uniform cone radius if they are not empty.

Proof (sketch). We first define a semi-algebraic mapping $\gamma : \mathrm{clo}(A) \to \mathbf{R}$ which associates to each point $\boldsymbol{p}$ of A a cone radius of A in $\boldsymbol{p}$, and to each point of $\mathrm{clo}(A) - A$ a cone radius of $\mathbf{R}^n - A$ in $\boldsymbol{p}$. In view of Theorem 2, this mapping is FO+Poly-definable. Define the set

$$\Gamma(A) := \{\boldsymbol{p} \in \mathrm{clo}(A) \mid \gamma|A \text{ is not continuous in } \boldsymbol{p}\}.$$

It can be shown that the dimension of the semi-algebraic set $\Gamma(A)$, is strictly less than the dimension of A.

We shall denote the closure in $\mathrm{clo}(A)$ of the set $\Gamma(A)$, by $\Sigma(A)$. Furthermore, let $\Sigma^d(A) := \mathrm{clo}(A)$, $\Sigma^{d-1}(A) := \Sigma(A)$, and let $\Sigma^k(A) := \Sigma(\Sigma^{k+1}(A))$, for $k = \ell, \ell+1, \ldots, d$, where ℓ is minimal integer such that $\Sigma^\ell(A) \neq \emptyset$.

Define for $k = \ell, \ell+1, \ldots, d$, the semi-algebraic sets $A_k := \Sigma^{k+1}(A) - \Sigma^k(A)$. In this way we obtain a decomposition of $\mathrm{clo}(A)$. Now, A_ℓ is the set for which the set $\Sigma(A_\ell)$ is empty, and A_ℓ is closed. Hence, $\gamma(A_\ell)$ is closed because $\gamma|A$ is continuous, and we define the uniform cone radius of A_ℓ as

$$\varepsilon_\ell := \min\{\gamma(\boldsymbol{p}) \mid \boldsymbol{p} \in A_\ell\}.$$

We now prove the theorem for $A_{i+1} - \bigcup_{j=\ell}^{i} A_j^{\varepsilon_j}$. Let $\eta = \min\{\varepsilon_\ell, \ldots, \varepsilon_i\}$, then we have,

$$\Sigma^i(A)^\eta \subseteq \bigcup_{j=\ell}^{i} A_j^{\varepsilon_j}.$$

Similarly as for A_ℓ, $\Sigma^{i+1}(A) - \Sigma^i(A)^\eta$ is closed and the restriction of γ to $(\Sigma^{i+1}(A) - \Sigma^i(A)^\eta)$ is continuous. Hence, the following minimum exists

$$\varepsilon_{i+1} := \min\{\gamma(\boldsymbol{p}) \mid \boldsymbol{p} \in A_{i+1} - \bigcup_{j=\ell}^{i} A_j^{\varepsilon_j}\}$$

and is a uniform cone radius of $A_{i+1} - \bigcup_{j=\ell}^{i} A_j^{\varepsilon_j}$. □

Define the queries $\varphi_{\mathrm{uniform},k} : A \mapsto \emptyset$ if $k > \dim(A)$, and $\varphi_{\mathrm{uniform},k} : A \mapsto A_k \cap A$ if $k \leqslant \dim(A)$. Note that for $k < \ell$, the result of the query $\varphi_{\mathrm{uniform},k}$ will be the empty set. The next corollary follows immediately from the construction in the proof of Theorem 3 and the fact that the query $\varphi_{\dim} : A \mapsto \dim(A)$ is expressible in FO (see [18]).

Corollary 1. *Let A be a semi-algebraic set in $\mathbf{R}^n$, then the queries $\varphi_{uniform,k}$, for $k = 1, \ldots, n$, are all expressible in* FO.

4 Linearization of a Semi-Algebraic Set

The aim of this section is to construct a semi-linear set $\hat{A} \subset \mathbf{R}^n$ which is homeomorphic to a given semi-algebraic set $A \subset \mathbf{R}^n$. We call the set $\hat{A}$ the *linearization* of A. We may assume that A is a bounded semi-algebraic set (if A is unbounded, we first apply the homeomorphism $\boldsymbol{x} \mapsto \boldsymbol{x}/(1 + \|\boldsymbol{x}\|)$)

Let A be a semi-algebraic set in $\mathbf{R}^n$, and let $\boldsymbol{p}$ be a point in A. Let $\varepsilon_{\boldsymbol{p}}$ be a cone radius of A in $\boldsymbol{p}$. Consider an n-dimensional box B, defined as $[a_1, a_1 + \delta] \times \cdots \times [a_n, a_n + \delta] \subset \mathbf{R}^n$, such that $\boldsymbol{p}$ is in the interior of the box B, and such that the diagonal of B (which equals $\sqrt{n}\delta$), is smaller than $\varepsilon_{\boldsymbol{p}}$. We know from Theorem 1, that with respect to the topology of A, inside a ball *Ball* of radius $\varepsilon_{\boldsymbol{p}}$ around $\boldsymbol{p}$, the semi-algebraic set A can be replaced with $\mathrm{Cone}(A \cap \partial \mathit{Ball}, \boldsymbol{p})$. We can prove an equivalent of Theorem 1, where n-dimensional boxes replace

the n-dimensional balls. Due to space limitations, we omit the details. Hence, with respect to the topology of A, inside B, the semi-algebraic set A can be replaced with $\mathrm{Cone}(A \cap \partial B, \boldsymbol{p})$. The whole idea of the algorithm is based on the observation that *a cone is semi-linear if its base set is semi-linear.*

Therefore, in order to linearize the set $B \cap A$, we could first try to linearize the set $\partial B \cap A$ on the boundary of the box in such a way that it remains on the boundary of the box, and then construct the cone.

This suggests the following algorithm which takes as input a semi-algebraic set A, and outputs a linearization $\hat{A}$ of A.Let $\mathcal{P}_k$ with $k = 0, \ldots, n$, denote finite relations consisting of pairs $(B, \boldsymbol{p})$, where B is a k-dimensional box and $\boldsymbol{p} \in \mathrm{int}(B)$.

LINEARIZE(A, n):

- If $n = 0$, this means that A is a finite set. We then add $(\boldsymbol{p}, \boldsymbol{p})$ to $\mathcal{P}_0$ for each $\boldsymbol{p} \in A$.
- Cover A with n-dimensional boxes $\{B_n^{(i)} \mid i \in I_n\}$ for some finite index set I_n, such that for each $i \in I_n$, there exists a point $\boldsymbol{p}_{B_n^{(i)}}$ in the interior of $B_n^{(i)}$, whose cone radius exceeds the diagonal of this box. For each $i \in I_n$, we add all pairs $(B_n^{(i)}, \boldsymbol{p}_{B_n^{(i)}})$ to the relation $\mathcal{P}_n$.
- If $n > 0$, then apply LINEARIZE$(A \cap B', n-1)$, where B' is an $(n-1)$-dimensional box on the boundary of some box $B_n^{(i)}$. Here, we interpret $A \cap B'$ as a semi-algebraic set in $\mathbf{R}^{n-1}$. This can easily be achieved since the $(n-1)$-dimensional boxes are parallel to one of the coordinate planes $H_i := \{\boldsymbol{x} \in \mathbf{R}^n \mid x_i = 0\}$.
- Inductively, the algorithm constructed a linear set $\hat{A}_{n-1}$ on the boundaries of each n-dimensional box B in $\mathcal{P}_n$ such that $\partial B \cap A$ is homeomorphic to $\partial B \cap \hat{A}_{n-1}$. We now construct the cone with top $\boldsymbol{p}_B$ and base $\partial B \cap \hat{A}_{n-1}$ for each pair $(B, \boldsymbol{p}_B) \in \mathcal{P}_n$. By definition of the points $\boldsymbol{p}_B$ this gives a set homeomorphic to $A \cap B$. When the cones are constructed for all boxes in $\mathcal{P}_n$, we have obtained a linear set $\hat{A}_n$ which is homeomorphic to the input set A.

We now explain two steps of the algorithm LINEARIZE in more detail.

4.1 Construction of the Box Covering

We define the n-dimensional standard grid of size δ as the set of n-dimensional boxes of the form $[k_1\delta, (k_1+1)\delta] \times \cdots [k_n\delta, (k_n+1)\delta]$, where $k_1, k_2, \ldots, k_n \in \mathbf{Z}$. We define a *box covering of size* δ of a semi-algebraic set A, denoted by A_δ, as a those boxes in the standard grid of size δ, which intersect the closure of A. By the Dichotomy Theorem [4], it is easy to show that the query which maps a semi-algebraic set to its box covering of size δ is not expressible in FO+POLY.

Lemma 1. *The query $\varphi_{cover} : A \mapsto A_\delta$ which maps a spatial database to its the box covering of size δ, is expressible in* FO+TC.

Proof (sketch). One represents boxes of size δ by means of $2n$-tuples $(a_1, a_1 + \delta, \ldots, a_n, a_n + \delta)$ of real numbers $a_i \in \mathbf{R}$. Let $\mathcal{B}$ be the set of all such tuples, and define an $4n$-ary adjacency relation Adj on this set. We have that $Adj(B_1, B_2)$ for two tuples B_1 and B_2 of $\mathcal{B}$ if the intersection of the boxes they represent is the union of lower (less than n) dimensional pieces of these boxes. As above, we may assume that A is bounded. We define the bounding box of A, as the set $[-M, M]^n$ such that 1/2-neighborhood $A^{1/2}$ is strictly included in $[-M, M]^n$. The computation of the transitive closure $\mathrm{TC}\{Adj(B_1, B_2) \mid B_i \cap [-M, M]^n \neq \emptyset\}$ then terminates, and the box covering A_δ of size δ are those boxes in this transitive closure which are in relation with the $(0, \delta, \ldots, 0, \delta)$ and intersect the closure of A. □

Suppose that a uniform cone radius of A exists. Let ε such radius, and let A_δ be the box covering of A of size δ, where $\delta^2 < (\varepsilon^2/n)$. We require for this covering that for each n-dimensional box $B \in A_\delta$ we have that $\mathrm{int}(B) \cap A \neq \emptyset$, so we can select a point $\boldsymbol{p}_B$ in this interior whose cone radius is larger than the diagonal δ of B.

Of course, this property is not automatically satisfied. Some of the boxes in A_δ may intersect A only in its boundary. To avoid this, we bring the covering A_δ in *general position* [12]. Formally, we require that A_δ and A are *transversal* in $\mathbf{R}^n$, or in symbols, that $A_\delta \pitchfork A$. The formal definition of transversality only makes sense when all points in A_δ and A have a tangent spaces. So we shall consider a decomposition of A_δ and A into sets such that the tangent space exists in every point of these sets. The construction of these decompositions is similar (but easier) to the decomposition in Theorem 2 and can also be performed in FO. So, let $A = \{A_0, \ldots, A_n\}$, and $A_\delta = \{A_{\delta,0}, \ldots, A_{\delta,n}\}$ the decompositions of A and A_δ. Then A_δ is called transversal to A in $\mathbf{R}^n$, if $A_{\delta,i}$ is transversal to A_j in $\mathbf{R}^n$ for any i and j. This means that in every point $\boldsymbol{x}$ in the intersection $A_\delta \cap A$, the following holds for any i and j,

$$T_{\boldsymbol{x}} A_{\delta,i} \oplus T_{\boldsymbol{x}} A_j = \mathbf{R}^n. \tag{4}$$

To get an idea of what this transversality condition means, we suppose that $n = 3$, that A_δ consist of a single box B and that the dimension of A equals two. Moreover, we assume that the tangent space exists in every point of A. We decompose the box B into its interior $A_{\delta,3}$, its 6 two-dimensional faces $A_{\delta,2}$, its 12 one-dimensional edges $A_{\delta,1}$, and its 8 vertices $A_{\delta,0}$. From condition 4 and the fact that $\dim(A) = 2$, it follows that A cannot intersect in one of vertices in $A_{\delta,0}$. Figure 2 shows one examples of a transversal (left) and two non-transversal intersections (right) of A with the edges and faces of B. It is easy to show that if a box B intersects A transversally, then $\mathrm{int}(B) \cap A \neq \emptyset$, so the question is how to force the transversality of A and boxes in A_δ. Fortunately, *almost all* coverings are already fine. More specifically, we can prove the following lemma

Lemma 2. *Let A be a semi-algebraic set in $\mathbf{R}^n$, and let B a n-dimensional box. Let b be a positive real number. Define the set of vectors*

$$T_b(B) := \{\boldsymbol{v} \in \mathbf{R}^n \mid \|\boldsymbol{v}\| < b \wedge B + \boldsymbol{v} \pitchfork A\},$$

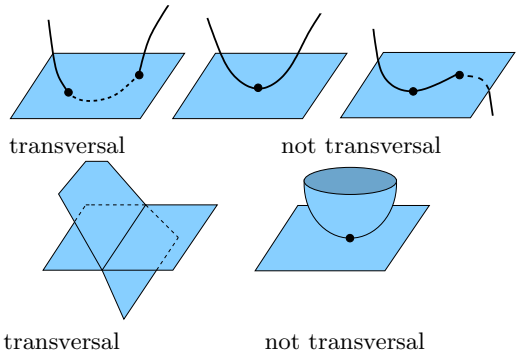

Fig. 2. Illustration of the notion of transversality.

where $B + \boldsymbol{v} = \{\boldsymbol{x} \mid \boldsymbol{x} - \boldsymbol{v} \in B\}$ *is a translation of* B*. Then* T_b *is dense in the set of vectors of norm smaller than* b*.*

Consider the covering A_δ and let d be the minimal distance from A to the boundary of A_δ. From the above lemma, we then can always select a translation $\tau \in T_{d/2}(B)$ such that $B + \tau \Cap A$, with $B \in A_\delta$. Since A_δ consists of a finite number of boxes, the set $\bigcap_i T_{d/2}(B_i)$ is also dense in the set of vectors of norm less than $d/2$. We can select a vector $\boldsymbol{v}$ in this intersection in FO. Indeed, transversality and hence also the set T_b for any $b > 0$ are expressible in FO. Since this translation is strictly smaller than d, $\mathbb{A} + \tau$ is still a covering of A, and we have found a covering which has the desired properties.

Suppose that A has no uniform cone radius. Then we apply Theorem 3 and we get the decomposition $A = A_\ell \cup \ldots \cup A_d$. We start with a box covering A_δ^ℓ of A_ℓ, because this set has a uniform cone radius. Suppose that we already have constructed a box covering A_δ^{k-1} of the set $A_\ell \cup \ldots \cup A_{k-1}$. We then consider the set $A' = A_k - (A_\ell \cup \ldots \cup A_{k-1})^{d_{k-1}}$, where d_{k-1} is a positive real number satisfying certain conditions.

This numbers ensure that after translating the covering A_δ^{k-1} with a small translation, no new pieces of A_k get uncovered. So it is sufficient to cover the set A'. This is possible because by Theorem 3 this set has a uniform cone radius. We take the intersection $A_{\delta,\cap}$ of the coverings $(A')_\delta$ and A_δ^{k-1}, and define

$$A_\delta^k = A_\delta^{k-1} \cup ((A')_\delta - A_{\delta,\cap}).$$

We then show an equivalent of Lemma 2 which takes into account the multiple sets $A_\ell, \ldots, A_d$. In this way, we obtain a covering A_δ which satisfies our requirements.

To conclude, we construct a relation $R = \{(B, \boldsymbol{p}) \mid B \in A_\delta \wedge \boldsymbol{p} \in A \cap \mathrm{int}(B)\}$. Note that for every box $B \in A_\delta$ there exists at least a single point $\boldsymbol{p}$ in its interior. So, we can select a unique representative $\boldsymbol{p}_B$ in the interior of B. We define the relation $\mathcal{P}_n$ containing the pairs $(B, \boldsymbol{p}_B)$ for each $B \in A_\delta$. Of course, the points $\boldsymbol{p}_B$ will be the tops of the cones, when linearizing the set A.

4.2 Inductive Application of the Algorithm

Consider for the moment a single n-dimensional box B of the covering A_δ. We regard the intersection $\partial B \cap A$ as the union of $2n$ $(n-1)$-dimensional sets parallel to one of n-coordinate planes H_i. We identify these sets with $\mathbf{R}^{n-1}$ by means of an orthogonal projection. The box covering construction can then be applied to this set in $\mathbf{R}^{n-1}$. Finally we can then bring it back, using the inverse projection, to the boundary of the box.

Let us focus on a single hyperplane H which is identified with $\mathbf{R}^{n-1}$. We want to cover the intersection of $\partial B \cap A \cap H$ with $(n-1)$-dimensional boxes. In the last section, we saw that we sometimes needed to translate the covering a bit in order to satisfy our requirements. Moreover, we want the covering to be compatible with the $(n-1)$-dimensional boxes, induced by the intersection of the covering A_δ with the hyperplane H. In this way, we will be able to construct a homeomorphism between A and the linearization $\hat{A}$ in a patchwork-like way. However, some of the boundaries of the $(n-1)$-dimensional boxes will come

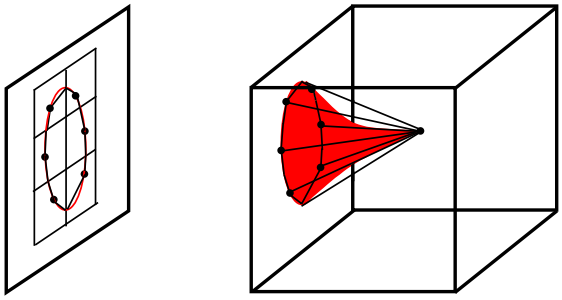

Fig. 3. Illustration of the algorithm LINEARIZE.

from this intersection, while others are from the new covering in the hyperplane H. While we may translate this last covering, we need to keep the intersection induced by A_δ fixed. Otherwise, we would have to construct a parameterized box covering, which is not possible in FO+TC since no free variables are allowed inside a transitive closure. Thanks to the following property, the boxes induced by this intersection are already in general position.

Proposition 1. *Let A_δ be an n-dimensional box covering which intersects a semi-algebraic set A in $\mathbf{R}^n$ transversally. Let H be an $(n-1)$-hyperplane in which $(n-1)$-dimensional boxes $B'_1, \ldots, B'_k$ on the boundaries of boxes of A_δ lie. Then the set of boxes $B'_1, \ldots, B'_k$ intersects the set $A \cap H$ transversally in H.*

So, we only need to bring the new box covering in general position.We then may apply the same construction to all boundaries of all boxes in A_δ. In this way we obtain a relation $\mathcal{P}_{n-1}$ consisting of pairs $(B, \boldsymbol{p}_B)$ where B is box in one of the $n-1$-dimensional coverings.

In Figure 3 we illustrated the idea of the algorithm on a simple three-dimensional example. The set inside the box is linearized, after linearizing the figure on the left face of the cube and then constructing the cone, as showed in the figure.

4.3 Expressing the Construction in Transitive Closure logic

From Lemma 1 we know that the query φ_{cover} is expressible in FO+TC. Corollary 1 shows that the queries $\varphi_{\text{uniform},k}$ are all expressible in FO. From the construction above, it is clear that a translation vector of Lemma 2 can be selected in FO. This shows that the basic ingredients of the algorithm LINEARIZE, are expressible in FO+TC. Since the dimension of the real space $\mathbf{R}^n$ is fixed, we only get a recursion depth of n in the algorithm, and this can be written as a single FO formula. In short,

Theorem 4. *It is possible in* FO+TC*, to construct a semi-linear set $\hat{A}$ which is homeomorphic to the original semi-algebraic set A.*

5 Completeness Result

We now refine the geometric construction such that the $n+1$ relations $\mathcal{P}_n, \ldots, \mathcal{P}_0$ consists of rational values only.

Theorem 5. *Rational linearizations of spatial databases are expressible in the transitive closure logic* FO+TC*.*

Proof. We can obtain this result easily by modifying the proof of Theorem 4 such that boxes are used which have corner points with rational coordinates. The selection of the top of the cones in these boxes then consists of selecting the center of the box, which has rational coordinates also. □

We now recall the definition of a complete query language. A logic $\mathcal{L}$ is called *complete* for a class of databases if for each computable query Q on this class of databases, there exists a formula $\varphi_Q \in \mathcal{L}$, such that for each database D in this class,

$$Q(D) := \{\boldsymbol{x} \mid \langle D, \boldsymbol{x}\rangle \models \varphi_Q(\boldsymbol{x})\}.$$

Nothing is said however on the evaluation of these formulas φ_Q. A logic $\mathcal{L}$ is said to be *computationally complete* if it is complete and if the query Q is defined on a database D, then the evaluation of $\varphi_Q(D)$ must be finite.

For arbitrary spatial databases we show that FO+TCS is computationally complete for Boolean topological queries. A query Q is said to be *topological*, if for any two spatial databases A and B for which there exists a homeomorphism h of $\mathbf{R}^n$ such that $h(A) = B$, then $Q(A) = Q(B)$ holds.

We will need the following result

Theorem 6 ([8]). FO+TCS *is computationally complete on linear spatial databases involving only rational coefficients.*

A direct consequence is then:

Theorem 7. *All computable Boolean topological queries on spatial databases can be expressed in* FO+TCS *in an effective way.*

Proof. by Theorem 5, there exists an FO+TC formula Q_{approx} that defines, for any given spatial database A, a linear spatial database $\hat{A}$ such that there exists a homeomorphism h from $\mathbf{R}^n$ to $\mathbf{R}^n$ such that $h(A) = \hat{A}$.

Let Q be a Boolean topological computable query. Since Q is computable, it is in particular computable on linear spatial databases involving only rational coefficients, and therefore, by Theorem 6 expressible on these databases by an FO+TCS formula φ_Q. It is clear that

$$Q(A) := \textsc{True} \text{ iff } \varphi_Q(Q_{\text{approx}}(A)) := \textsc{True},$$

which concludes the proof. □

6 Concluding Remarks

We showed that it is possible in first-order logic extended with a transitive closure operator to construct from a semi-algebraic spatial database, a homeomorphic linear spatial database. The existence of a finite representation of linear spatial database, makes it possible to obtain expressiveness results. It is an interesting question whether there exists natural extensions of FO, which can compute a finite representation of semi-algebraic spatial databases, without the deviation to the linear case.

References

1. R. Benedetti and J.J. Risler. *Real Algebraic and Semi-algebraic Sets.* Hermann, Paris, 1990.
2. M. Benedikt, G. Dong, L. Libkin, and L. Wong. Relational expressive power of constraint query languages. *Journal of the ACM*, 45(1):1–34, 1998.
3. M. Benedikt, M. Grohe, L. Libkin, and L. Segoufin. Reachability and connectivity queries in constraint databases. In *Proceedings of the 19th ACM Symposium on Principles of Database Systems*, pages 104–115. ACM Press, 2000.
4. M. Benedikt and L. Libkin. Safe constraint queries. *SIAM Journal of Computing*, 29(5):1652–1682, 2000.
5. J. Bochnak, M. Coste, and M.-F. Roy. *Real Algebraic Geometry*, volume 36 of *Ergebenisse der Mathematik und ihrer Grenzgebiete.* Springer-Verlag, 1998.
6. A. Chandra and D. Harel. Computable queries for relational data bases. *Journal of Computer and System Sciences*, 21(2):156–178, 1980.
7. F. Geerts and B. Kuijpers. Expressing topological connectivity of spatial databases. In *Research Issues in Structured and Semistructured Database Programming. Proceedings of the 8th International Workshop on Database Programming Languages*, volume 1949 of *Lecture Notes in Computer Science*, pages 224–238. Springer-Verlag, 1999.

8. 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*, pages 126–135. ACM Press, 2000.
9. S. Grumbach and G. Kuper. Tractable recursion over geometric data. In G. Smolka, editor, *Proceedings of the 3rd Conference on Principles and Practice of Constraint Programming*, volume 1330 of Lecture Notes in Computer Science, pages 450–462. Springer-Verlag, 1997.
10. S. Grumbach and J. Su. Finitely representable databases. *Journal of Computer and System Sciences*, 55(2):273–298, 1997.
11. S. Grumbach and J. Su. Queries with arithmetical constraints. *Theoretical Computer Science*, 173(1):151–181, 1997.
12. V. Guillemin and A. Pollack. *Differential topology*. Prentice-Hall, 1974.
13. M. Gyssens, J. Van den Bussche, and D. Van Gucht. Complete geometrical query languages. *Journal of Computer and System Sciences*, 58(1):483–511, 1999.
14. P.C. Kanellakis, G.M. Kuper, and P.Z. Revesz. Constraint query languages. *Journal of Computer and System Science*, 51(1):26–52, 1995.
15. S. Kreutzer. Fixed-point query languages for linear constraint databases. In *Proceedings of the 19th ACM Symposium on Principles of Database Systems*, pages 116–125. ACM Press, 2000.
16. S. Kreutzer. Query languages for constraint databases: First-order logic, fixed-points, and convex hulls. In J. Van den Bussche and V. Vianu, editors, *Proceedings of the 9th International Conference on Database Theory*, volume 1973 of *Lecture Notes in Computer Science*, pages 248–262. Springer-Verlag, 2001.
17. G.M. Kuper, J. Paredaens, and L. Libkin, editors. *Constraint Databases*. Springer-Verlag, 1999.
18. E. Rannou. The complexity of stratification computation. *Discrete and Computational Geometry*, 19:47–79, 1998.
19. M. Shiota. *Geometry of Subanalytic and Semialgebraic Sets*. Birkhäuser, 1997.
20. 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*, pages 109–118. ACM Press, 1998.
21. L. Vandeurzen, M. Gyssens, and D. Van Gucht. On query languages for linear queries definable with polynomial constraints. In E. F. Freuder, editor, *Proceedings of the 2nd Conference on Principles and Practice of Constraint Programming*, volume 1118 of *Lecture Notes in Computer Science*, pages 468–481, Springer-Verlag, 1996.