

A Theory of Spatio-Temporal Database Queries

Floris Geerts, Sofie Haesevoets and Bart Kuijpers

University of Limburg
Department WNI, B-3590 Diepenbeek, Belgium
{floris.geerts, sofie.haesevoets, bart.kuijpers}@luc.ac.be

Abstract. We address a fundamental question concerning spatio-temporal database systems: “What are exactly spatio-temporal queries?” We define spatio-temporal queries to be computable mappings that are also *generic*, meaning that the result of a query may only depend to a limited extent on the actual internal representation of the spatio-temporal data. Genericity is defined as invariance under transformations that preserve certain characteristics of spatio-temporal data (e.g., collinearity, distance, velocity, acceleration, ...) that are relevant to a database user. These transformations also respect the monotone nature of time. We investigate different genericity classes relative to the constraint database model for spatio-temporal databases and we identify sound and complete languages for the first-order, respectively the computable, queries in these genericity classes.

1 Introduction

Since the early 1990s, various database systems have been developed to handle spatial data [1, 5, 10, 14, 16, 26] and solid theories for such systems have been proposed and studied [21, 23]. Conceptually, spatial databases are possibly infinite sets of points in a real space \mathbf{R}^n . In more recent years, we have seen the emergence of database systems and applications that are dealing with *spatio-temporal data* [4, 7, 12, 15, 25]. Conceptually, spatio-temporal data can be modeled as infinite spatial sets that move or change in time, i.e., sets in $\mathbf{R}^n \times \mathbf{R}$.

A much acclaimed method for effectively representing infinite geometrical figures is provided by the *constraint database model*, that was introduced in 1990 by Kanellakis, Kuper and Revesz [18] (recently an overview of the area of constraint databases appeared [24]). Until recently this model has been used mainly in the area of spatial databases, but it provides an equally elegant and efficient way to model spatio-temporal data [7–9, 13, 20]. In the setting of the constraint model, a spatio-temporal database in $\mathbf{R}^n \times \mathbf{R}$ is finitely represented as a Boolean combination of polynomial equalities and inequalities. Figure 1 depicts the spatio-temporal database $\{(x, y; t) \mid x^2 + y^2 + t^2 \leq 1 \vee (x^2 + y^2 + (t - 2)^2 = 1 \wedge t \leq 5/2) \vee (x^2 + y^2 + (t - 3)^2 = 1 \wedge t > 5/2)\}$ in $\mathbf{R}^2 \times \mathbf{R}$.

A number of theoretical studies have appeared on the status of time and its relation with space in systems that model moving objects. Erwig et al. [11] give a taxonomy of applications ranging from those that rely on a step-wise

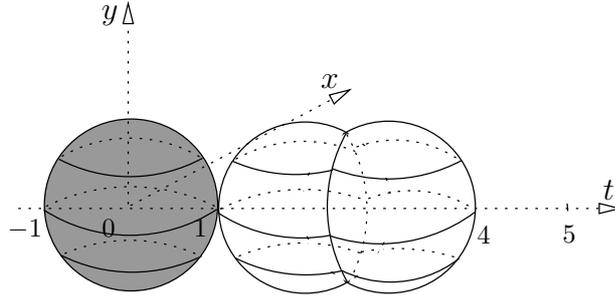


Fig. 1. An example of a spatio-temporal database in $\mathbf{R}^2 \times \mathbf{R}$.

constant geometry to applications which need more complete integration of space and time (like for instance a continuous description of a trajectory). MOST, an example of the latter category, relies on a strong interaction of the space and time components (since the space variables are described by linear polynomials in time) and provides a query language that is a combination of a spatial query language and a temporal logic. On the other range of the spectrum, variable independence (defined in terms of orthographic dimension) gives rise to a less expressive data model which has the advantage of a lower complexity of query evaluation [13, 22].

We study spatio-temporal queries from the perspective of expressive power, and do this against the background of the full modeling and querying power of the constraint database model and the first-order and computationally complete languages it offers. We ask which expressions in these languages may be considered as *reasonable* spatio-temporal queries. In database theory it is usually required that the result of queries should only to a certain limited extent depend on the actual internal representation of databases and that queries should only ask for properties that are shared by “isomorphic” encodings of the same data. The meaning of “isomorphic” may be influenced by the actual database application and by which notions are relevant in it. In the context of the relational database model, Chandra and Harel [6] formalized this independence of the actual encoding in terms of the notion of *genericity*. Paredaens, Van den Bussche and Van Gucht [23] identified a hierarchy of genericity classes for spatial database applications. The generic queries in the different classes focus on different geometrical and topological aspects of the spatial data. On a technical level, generic queries are defined as being invariant under those transformations of the data that preserve the relevant aspects of the data. Whereas Chandra and Harel considered the group of the isomorphisms (that possibly fix some elements of the domain) in the case of relational databases, Paredaens, Van den Bussche and Van Gucht identified different geometrical and topological transformation groups (affinities, isometries, translations, homeomorphisms ...) for spatial database applications.

We investigate which notions of genericity are appropriate for spatio-temporal databases and which transformation groups express them. We observe that the transformations should first and foremost respect the monotone nature of time, i.e., leave the temporal order of events unchanged. It follows that the relevant transformation groups are the product of a group of time-(in)dependent spatial transformations and a group of monotone transformations of the time-component of the spatio-temporal data. Next, we focus on the former groups and study which of them leave different spatial and spatio-temporal properties (like collinearity, distance and orientation) unchanged. We also focus on physical properties of spatio-temporal data (like velocity and acceleration). The transformation groups that we consider are all subgroups of the time-dependent or time-independent affinities of $\mathbf{R}^n \times \mathbf{R}$.

We study the notion of spatio-temporal genericity relative to two popular query languages in the constraint model: first-order logic over the reals (FO) and an extension of this logic with a while-loop (FO + While). Queries in both languages are known to be effectively computable (given termination in the case of FO + While-programs) and FO + While is known to be a computationally complete language on spatio-temporal databases [28]. First, we show that all the genericity classes are undecidable. We show that the considered classes of generic first-order queries are recursively enumerable, however. Hereto, we define first-order point-based languages in which variables range over points in $\mathbf{R}^n \times \mathbf{R}$ and which contain certain point predicates. These point-based languages are shown to be sound and complete for the first-order queries in the considered genericity classes. We have also shown that extensions of these point-based logics with a While-loop give sound and complete languages for the computable queries in the different genericity classes. Our results are inspired by similar results that were obtained by Gyssens, Van den Bussche and Van Gucht in the context of spatial databases [17]. However, mainly our results for genericity notions described by time-dependent transformations require new proof techniques.

This paper is organized as follows. In Section 2, we define spatio-temporal databases, spatio-temporal queries, and the constraint query languages FO and FO + While. In Section 3, we define a number of genericity notions. In Section 4 and 5, we present sound and complete first-order and computationally complete query languages for the different notions of genericity. In Section 6, we end with a discussion of some open problems.

2 Definitions and preliminaries

We denote the set of the real numbers by \mathbf{R} .

2.1 Spatio-temporal databases

In the following, we will consider n -dimensional spatial figures that change in time ($n \geq 2$). A moving figure is described by means of an (often infinite) set of tuples $(x_1, x_2, \dots, x_n; t)$ in $\mathbf{R}^n \times \mathbf{R}$, where (x_1, x_2, \dots, x_n) represent the spatial

coordinates of a point in the n -dimensional real space \mathbf{R}^n and t is the time coordinate in \mathbf{R} . We will work with spatio-temporal data that can be modeled in the constraint model.

Definition 1. An (n -dimensional) *spatio-temporal database*¹ is a set $\{(x_1, x_2, \dots, x_n; t) \in \mathbf{R}^n \times \mathbf{R} \mid \varphi(x_1, x_2, \dots, x_n; t)\}$, where $\varphi(x_1, x_2, \dots, x_n; t)$ is a formula built with the logical connectives \wedge, \vee, \neg from atomic formulas of the form $p(x_1, x_2, \dots, x_n, t) > 0$, with $p(x_1, x_2, \dots, x_n, t)$ a polynomial with integer coefficients and real variables x_1, x_2, \dots, x_n, t . \square

Figure 1 in the Introduction gives an illustration of a 2-dimensional spatio-temporal database. It shows at its beginning (i.e., at $t = -1$) a single point in the origin of \mathbf{R}^2 . Then it shows a disk whose radius increases and later decreases and ends in a point at moment $t = 1$, followed by a circle whose radius increases, decreases, increases and then shrinks to a point.

2.2 Spatio-temporal database queries

Here, we give a first definition of a query. In the next section, we will impose further conditions on the nature of these mappings.

Definition 2. A *spatio-temporal database query* is a computable function that maps spatio-temporal databases to spatio-temporal databases. \square

2.3 Constraint query languages

In this paper, we will consider two popular constraint query languages: first-order logic and an extension of this logic with a while-loop.

First-order logic over the reals (in other words, the relational calculus augmented with polynomial inequality constraints $p(x_1, x_2, \dots, x_m) > 0$), FO for short, has been well-studied as a query language in the context of spatial databases [18, 23]. In the setting of spatio-temporal databases it can be used similarly as a query language. For instance, the calculus formula $\mathfrak{F}(x, y; t) \wedge (\exists x_0)(\exists y_0)(\exists r > 0)(\forall x)(\forall y)((x - x_0)^2 + (y - y_0)^2 = r^2 \leftrightarrow \mathfrak{F}(x, y; t))$ selects those snapshots from a spatio-temporal database \mathfrak{F} where it shows a circle. It is well-known that FO-formulas can be effectively evaluated on spatio-temporal databases in the constraint model and that the output can be represented in the same constraint formalism [28].

It is known that the extension of first-order logic over the reals with a while-loop, FO + While for short, yields a computationally complete language for constraint databases [28]. An FO + While-program is a finite sequence of *statements*

¹ The results in this paper can be extended straightforwardly to the situation where a spatio-temporal database consists of more such sets and where these sets are accompanied by classical thematic information. However, because the complete problem that is discussed here is captured by this simplified model, we stick to it for reasons of simplicity of exposition.

and *while-loops*. Each statement has the form

$$R := \{(x_1, \dots, x_k) \mid \varphi(x_1, \dots, x_k)\},$$

with R a relation variable of arity k and φ a formula in FO augmented with previously introduced relation variables. A while-loop has the form

$$\mathbf{while} \ \varphi \ \mathbf{do} \ P,$$

where P is a program and φ is a sentence in FO augmented with previously introduced relation variables. Semantically, a FO + While-program expresses a spatio-temporal query in the obvious way as soon as one of its relation variables has been designated as the output variable.

3 Spatio-temporal genericity

For simplicity we consider from now on only queries that take an n -dimensional spatio-temporal database as input and also output an n -dimensional spatio-temporal database (variations are possible but straightforward). As stated in the Introduction, we are interested in spatio-temporal database queries that are invariant under the elements of a certain spatio-temporal transformation group

$$\mathcal{F} = \{f \mid f = (f_1, f_2, \dots, f_n, f_t) : \mathbf{R}^n \times \mathbf{R} \rightarrow \mathbf{R}^n \times \mathbf{R}\}.$$

In the remainder of this section, we will impose two further conditions on these transformations. The first condition is a purely temporal one (it concerns the order of events), whereas the second is a purely spatial or spatio-temporal condition that reflects the nature of the queries one is interested in.

3.1 Temporal condition

An *event* is a subset of $\mathbf{R}^n \times \mathbf{R}$. The projection of an event A on the time-axis is denoted by $\pi_t(A)$ and called the *time-interval* of A .

Let A and B be events. In Allen's terminology [2, 3], A and B are called *co-temporal* if $\pi_t(A) = \pi_t(B)$ (we denote this by $A =_t B$). Allen says A is *before* B if $t_A < t_B$ for all $t_A \in \pi_t(A)$ and all $t_B \in \pi_t(B)$ (we denote this by $A <_t B$). Remark that $A \leq_t B := A =_t B$ or $A <_t B$ is an order on time-intervals. We say that a transformation $f : \mathbf{R}^n \times \mathbf{R} \rightarrow \mathbf{R}^n \times \mathbf{R}$ *preserves the order of events* if for all events A and B , $A =_t B$ implies $f(A) =_t f(B)$ and $A <_t B$ implies $f(A) <_t f(B)$.

Proposition 1. *A transformation $f = (f_1, f_2, \dots, f_n, f_t) : \mathbf{R}^n \times \mathbf{R} \rightarrow \mathbf{R}^n \times \mathbf{R}$ preserves the order of events if and only if f_t is a strictly monotone function of t alone. \square*

Proof. The only-if direction is clear. To prove the other direction, let $f = (f_1, f_2, \dots, f_n, f_t)$ be a transformation of $\mathbf{R}^n \times \mathbf{R}$. Consider any two events $A = \{(x_1, x_2, \dots, x_n, t)\}$ and $B = \{(x'_1, x'_2, \dots, x'_n, t)\}$. Then $f_t(x_1, x_2, \dots, x_n, t) = f_t(x'_1, x'_2, \dots, x'_n, t)$. This shows that f_t is a function of t alone.

Consider any two events $A = \{(x_1, x_2, \dots, x_n, t_A)\}$ and $B = \{(x_1, x_2, \dots, x_n, t_B)\}$ with $t_A < t_B$. Then $f_t(t_A) < f_t(t_B)$. This shows that f_t is a strictly monotone function of t . \square

Since we require that transformations preserve the order of events, we can write the transformation groups of interest as a product of groups:

$$\mathcal{F} = (\mathcal{F}_{st}, \mathcal{F}_t) = \{(f_{st}, f_t) \mid f_{st} = (f_1, f_2, \dots, f_n) : \mathbf{R}^n \times \mathbf{R} \rightarrow \mathbf{R}^n, f_t : \mathbf{R} \rightarrow \mathbf{R}\}.$$

The groups \mathcal{F}_t that we will consider are:

- $\mathcal{A}_t = \{t \mapsto a \cdot t + b \mid a, b \in \mathbf{R} \text{ and } a > 0\}$, i.e., the monotone affinities of the time-line;
- $\mathcal{T}_t = \{t \mapsto t + b \mid b \in \mathbf{R}\}$, i.e., the translations of the time-line; and
- $\text{Id}_t = \{\text{id}\}$, i.e., the identity.

3.2 Spatial and spatio-temporal conditions

For what concerns the spatial (or spatio-temporal) part, we consider transformations of the form:

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ t \end{pmatrix} \mapsto \begin{pmatrix} a_{11}(t) & a_{12}(t) & \cdots & a_{1n}(t) \\ a_{21}(t) & a_{22}(t) & \cdots & a_{2n}(t) \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1}(t) & a_{n2}(t) & \cdots & a_{nn}(t) \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} b_1(t) \\ b_2(t) \\ \vdots \\ b_n(t) \end{pmatrix},$$

where the a_{ij} and b_i are functions from \mathbf{R} to \mathbf{R} .

We will consider the following groups \mathcal{F}_{st} of transformations:

- \mathcal{A}_{st} , the group of transformations of the above form where the $a_{ij}(t)$ and $b_i(t)$ are arbitrary functions of t such that the matrix of the $a_{ij}(t)$ has an inverse for each value of t (these are the time-dependent affinities);
- \mathcal{A}_{st}^f , the subgroup of \mathcal{A}_{st} consisting of transformations for which the functions $a_{ij}(t)$ and $b_i(t)$ only take a finite number of values;
- \mathcal{A}_{st}^c , the subgroup of \mathcal{A}_{st} consisting of transformations for which the functions $a_{ij}(t)$ are constants and $b_i(t)$ are linear functions of t ;
- \mathcal{S}_{st} , \mathcal{S}_{st}^f , \mathcal{S}_{st}^c , the subgroups of the above where the matrix of the $a_{ij}(t)$ represents at each moment a similarity;
- \mathcal{I}_{st} , \mathcal{I}_{st}^f , \mathcal{I}_{st}^c , the subgroups of the above where the determinant of the matrix consisting of the $a_{ij}(t)$ equals 1 at each moment (i.e., the isometries);
- \mathcal{T}_{st} , \mathcal{T}_{st}^f , \mathcal{T}_{st}^c , the subgroups of the above where the matrix consisting of the $a_{ij}(t)$ is the identity matrix (i.e., the translations).

3.3 Physical conditions

The following groups are of interest when notions such as velocity, acceleration and force are important in an application. These transformation groups can be found by solving the differential equations that express that these notions are preserved. We consider these notions for arbitrary and rigid motions, respectively.

- \mathcal{V}_{st} , the subgroup of \mathcal{A}_{st}^c where the b_i are constants. This group of transformations preserves the *velocity vector*.
- $\mathcal{V}(\mathcal{R})_{st}$, the subgroup of \mathcal{I}_{st}^c where the b_i are constants. This group of transformations preserves the *velocity vector* of a moving object in rigid motion.
- \mathcal{AC}_{st} , is the group \mathcal{A}_{st}^c . This group of transformations preserves the *acceleration vector* of a moving object.
- $\mathcal{AC}(\mathcal{R})_{st}$, is the group \mathcal{I}_{st}^c . This group of transformations preserves the *acceleration vector* of a moving object in rigid motion.

In physics it is custom to consider only translations (i.e., \mathcal{T}_t) for what concerns the time-dimension.

3.4 Spatio-temporal genericity: definition

Finally, we define genericity of spatio-temporal database queries.

Definition 3. Let \mathcal{F}_{st} and \mathcal{F}_t be two of the above groups. Let Q be a spatio-temporal database query. We call Q $(\mathcal{F}_{st}, \mathcal{F}_t)$ -generic if and only if for all spatio-temporal databases \mathfrak{F}_1 and \mathfrak{F}_2 for which there exists a $f = (f_{st}, f_t) \in (\mathcal{F}_{st}, \mathcal{F}_t)$ such that $f(\mathfrak{F}_1) = \mathfrak{F}_2$ also $f(Q(\mathfrak{F}_1)) = Q(\mathfrak{F}_2)$ holds. \square

4 Sound and complete languages for the generic first-order spatio-temporal queries

In this section, we study the $(\mathcal{F}_{st}, \mathcal{F}_t)$ -generic queries that are expressible in FO. To start with, we give a general undecidability result. It can be proven using standard techniques from constraint databases [19].

Theorem 1. *For all non-trivial groups \mathcal{F}_{st} , \mathcal{F}_t mentioned in the previous section, $(\mathcal{F}_{st}, \mathcal{F}_t)$ -genericity of spatio-temporal FO queries is undecidable.* \square

We will show that the first-order queries that are $(\mathcal{F}_{st}, \mathcal{F}_t)$ -generic are recursively enumerable, however. We will show this by giving sound and complete languages for the FO queries of the different genericity classes. These languages are point-based logics of the following form.

Point-based logics. We define a number of first-order point-based languages that we shall denote by $\mathcal{P}(\{P_1, P_2, \dots, P_m\})$, where the P_i are point-predicates. The variables p, q, r, \dots in $\mathcal{P}(\{P_1, P_2, \dots, P_m\})$ represent points in $\mathbf{R}^n \times \mathbf{R}$. The atomic formulas of $\mathcal{P}(\{P_1, P_2, \dots, P_m\})$ are $\mathfrak{F}(p)$, meaning that the point p

belongs to the input database \mathfrak{F} , and $P_i(p, q, r, \dots)$. The formulas of $\mathcal{P}(\{P_1, P_2, \dots, P_m\})$ are built from atomic formulas using the connectives \wedge, \vee, \neg and quantification of point variables ($\exists p$), ($\forall p$). \square

In the remainder of this section, we first discuss notions of genericity determined by time-independent transformations with applications to physics and next focus on time-dependent transformations.

4.1 Genericity for time-independent transformations

In this section, we give a general result concerning $(\mathcal{F}_{st}, \mathcal{F}_t)$ -generic queries where \mathcal{F}_{st} is a group of time-independent affinities of $\mathbf{R}^n \times \mathbf{R}$.

Let $(O, E_1, E_2, \dots, E_{n+1})$ denote the standard coordinate system of $\mathbf{R}^n \times \mathbf{R}$. Let the point-predicate **Between** ^{$n+1$} be such that **Between** ^{$n+1$} (p, q, r) expresses that the points p, q, r in $\mathbf{R}^n \times \mathbf{R}$ are collinear and that q is between p and r . Let the predicate **Before** be such that **Before** (p, q) expresses that the time coordinate of p is less than or equal to the time coordinate of q .

The following meta-theorem can be proven using techniques introduced by Gyssens, Van den Bussche and Van Gucht [17].

Theorem 2. *Let \mathcal{F}_{st} be a subgroup of \mathcal{A}_{st}^c and let \mathcal{F}_t be a subgroup of \mathcal{A}_t . Let $\{P_1, P_2, \dots, P_m\}$ be a set of point-predicates that contains **Between** ^{$n+1$} and **Before**. If the predicates in $\{P_1, P_2, \dots, P_m\}$ are FO expressible and invariant under the transformations of $(\mathcal{F}_{st}, \mathcal{F}_t)$ and if the fact “ $(p_0, p_1, p_2, \dots, p_{n+1})$ is the image of the standard coordinate system $(O, E_1, E_2, \dots, E_{n+1})$ under some element f of $(\mathcal{F}_{st}, \mathcal{F}_t)$ ” is expressible in $\mathcal{P}(\{P_1, P_2, \dots, P_m\})$, then $\mathcal{P}(\{P_1, P_2, \dots, P_m\})$ is sound and complete for the $(\mathcal{F}_{st}, \mathcal{F}_t)$ -generic FO queries. \square*

We apply the previous theorem to some groups $(\mathcal{F}_{st}^c, \mathcal{F}_t)$. Further applications can be found in the next subsection. We need to introduce some further point predicates:

- **UnitTime** (p, q) expresses that the points $p, q \in \mathbf{R}^n \times \mathbf{R}$ have time-coordinates p_t and q_t such that $|p_t - q_t| = 1$;
- $+_t(p, q, r)$ holds for spatio-temporal points p, q, r with time-coordinates p_t, q_t and r_t if and only if $p_t + q_t = r_t$; $*_t$ is defined similarly;
- the unary predicates $\mathbf{0}_t$ and $\mathbf{1}_t$ express that the time-coordinate of a point equals zero and one respectively;
- the predicates $\leq_i(p, q)$ ($1 \leq i \leq n$) express that the i th spatial coordinate of p is less or equal than the i th spatial coordinate of q ;
- **EqDist** (p, q, r, s) is true if the distance between two co-temporal points p and q equals the distance between the two co-temporal points r and s ;
- **UnitDist** (p, q) expresses that p, q are co-temporal and that the spatial distance between p and q equals one; and finally
- **Pos** ^{$n+1$} $(p_0, p_1, p_2, \dots, p_{n+1})$ expresses that $(p_0, p_1, p_2, \dots, p_{n+1})$ form a positively oriented $(n + 1)$ -dimensional coordinate system.

Corollary 1. *Let $(\mathcal{F}_{st}, \mathcal{F}_t)$ be a group from Table 1 and let $\Pi(\mathcal{F}_{st}, \mathcal{F}_t)$ be as in Table 1. The point language $\mathcal{P}(\Pi(\mathcal{F}_{st}, \mathcal{F}_t))$ is sound and complete for the $(\mathcal{F}_{st}, \mathcal{F}_t)$ -generic FO queries. \square*

4.2 Applications to Physics

Here, we focus on the transformation groups $(\mathcal{V}_{st}, \mathcal{T}_t)$, $(\mathcal{V}(\mathcal{R})_{st}, \mathcal{T}_t)$, $(\mathcal{AC}_{st}, \mathcal{T}_t)$ and $(\mathcal{AC}(\mathcal{R})_{st}, \mathcal{T}_t)$. To formulate our results we need to define one more point-predicate, namely $=_{\text{space}}$. If $p = (p_1, \dots, p_n, p_t)$, $q = (q_1, \dots, q_n, q_t) \in \mathbf{R}^n \times \mathbf{R}$, then $=_{\text{space}}(p, q)$ if and only if $p_i = q_i$ for $1 \leq i \leq n$.

The following results can be proven using Theorem 2.

Theorem 3. *Let $(\mathcal{F}_{st}, \mathcal{T}_t)$ be a group from Table 2 and let $\Pi(\mathcal{F}_{st}, \mathcal{T}_t)$ be as in Table 2. The point language $\mathcal{P}(\Pi(\mathcal{F}_{st}, \mathcal{T}_t))$ is sound and complete for the $(\mathcal{F}_{st}, \mathcal{T}_t)$ -generic FO queries. \square*

4.3 Genericity for time-dependent transformations

Here, we focus on notions of genericity determined by time-dependent transformations. Our first result in this context shows that we can restrict our attention, without loss of generality, to piece-wise constant transformations. The proof of this proposition is postponed until the end of this section.

Proposition 2. *Let Q be a spatio-temporal FO query and let \mathcal{F}_{st} be \mathcal{A}_{st} , \mathcal{S}_{st} , \mathcal{I}_{st} or \mathcal{T}_{st} and \mathcal{F}_t be \mathcal{A}_t , \mathcal{T}_t or Id_t . Then Q is $(\mathcal{F}_{st}, \mathcal{F}_t)$ -generic if and only if it is $(\mathcal{F}_{st}^f, \mathcal{F}_t)$ -generic. \square*

We first focus on the group $(\mathcal{A}_{st}^f, \mathcal{A}_t)$. To formulate our result we need to define some additional point predicates:

- **Between** ^{n} (p, q, r) expresses that three points $p, q, r \in \mathbf{R}^n \times \mathbf{R}$ that are co-temporal are collinear and that q is between p and r ;

$(\mathcal{F}_{st}, \mathcal{F}_t)$	Sets of point predicates $\Pi(\mathcal{F}_{st}, \mathcal{F}_t)$
$(\mathcal{A}_{st}^c, \mathcal{A}_t)$	{ Between ^{$n+1$} , Before }
$(\mathcal{A}_{st}^c, \mathcal{T}_t)$	{ Between ^{$n+1$} , Before , UnitTime }
$(\mathcal{A}_{st}^c, \text{Id}_t)$	{ Between ^{$n+1$} , Before , UnitTime , +_t , *_t , 0_t , 1_t }
$(\mathcal{S}_{st}^c, \mathcal{F}_t)$	$\Pi(\mathcal{A}_{st}^c, \mathcal{F}_t) \cup \{\mathbf{EqDist}\}$
$(\mathcal{T}_{st}^c, \mathcal{F}_t)$	$\Pi(\mathcal{A}_{st}^c, \mathcal{F}_t) \cup \{\mathbf{EqDist}, \mathbf{UnitDist}\}$
$(\mathcal{T}_{st}^c, \mathcal{F}_t)$	$\Pi(\mathcal{A}_{st}^c, \mathcal{F}_t) \cup \{\mathbf{EqDist}, \mathbf{UnitDist}, \leq_i (1 \leq i \leq n), \mathbf{Pos}^{n+1}\}$

Table 1. The different sets of point predicates for a number of spatio-temporal genericity notions. In the last three cases, we have $\mathcal{F}_t \in \{\mathcal{A}_t, \mathcal{T}_t, \text{Id}_t\}$.

- $\mathbf{EqCR}(p, q, r, p', q', r')$ expresses that the cross ratio² of three co-temporal and collinear points p, q, r equals the cross ratio of three co-temporal and collinear points p', q', r' ;
- $\mathbf{EqCR}^{\text{st}}(p, q, r, p', q', r')$ expresses that the cross ratio of three co-temporal and collinear points p, q, r equals the cross ratio of the time coordinates of the points p', q', r' .

Theorem 4. *The point language $\mathcal{P}(\{\mathbf{Between}^n, \mathbf{Before}, \mathbf{EqCR}, \mathbf{EqCR}^{\text{st}}\})$ is sound and complete for the $(\mathcal{A}_{st}^f, \mathcal{A}_t)$ -generic FO queries. \square*

Proof sketch. It is easily verified that the predicates $\mathbf{Between}^n$, \mathbf{Before} , \mathbf{EqCR} and $\mathbf{EqCR}^{\text{st}}$ are expressible in the language FO and that all these predicates are invariant under transformations in the group $(\mathcal{A}_{st}^f, \mathcal{A}_t)$. We can then prove by induction on the formulas in $\mathcal{P}(\{\mathbf{Between}^n, \mathbf{Before}, \mathbf{EqCR}, \mathbf{EqCR}^{\text{st}}\})$ that they are $(\mathcal{A}_{st}^f, \mathcal{A}_t)$ -generic. This proves soundness.

For completeness, it suffices to show that every FO formula φ can be simulated in $\mathcal{P}(\{\mathbf{Between}^n, \mathbf{Before}, \mathbf{EqCR}, \mathbf{EqCR}^{\text{st}}\})$ by a formula $\bar{\varphi}$ in the sense that $\mathfrak{F} \models \varphi$ if and only if $f(\mathfrak{F}) \models \bar{\varphi}$ with f some transformation from the group $(\mathcal{A}_{st}^f, \mathcal{A}_t)$. For $(\mathcal{A}_{st}^f, \mathcal{A}_t)$ -generic FO formulas this implies that they can be equivalently expressed in $\mathcal{P}(\{\mathbf{Between}^n, \mathbf{Before}, \mathbf{EqCR}, \mathbf{EqCR}^{\text{st}}\})$.

Let φ be a FO formula that we assume to be in prenex normal form. We now sketch the translation of an FO formula φ into a formula $\bar{\varphi}$ of $\mathcal{P}(\{\mathbf{Between}^n, \mathbf{Before}, \mathbf{EqCR}, \mathbf{EqCR}^{\text{st}}\})$. In this translation, first the atomic formulas of φ are translated and the connectives and quantifiers are later added in an almost natural way. First of all, in the description of $\bar{\varphi}$ two moments in time p_{t_O} and p_{t_E} are chosen such that $\neg\mathbf{Before}(p_{t_E}, p_{t_O})$ (time moments are represented in $\bar{\varphi}$ by spatio-temporal points). Next, in the hyperplane of points that are co-temporal with p_{t_O} an affine coordinate system $(O, E_1, E_2, \dots, E_n)$ is chosen (this can be expressed using the predicate $\mathbf{Between}^n$ only [17]). The arithmetic operations on real variables (which are translated into point variables on the line OE_1) will be simulated, using only $\mathbf{Between}^n$, in the *computation plane* (O, E_1, E_2) (using techniques similar to those used by Tarski [27]; see also [17]).

An appearance of $\mathfrak{F}(x_1, \dots, x_n; t)$ in φ is translated into a conjunction of three formulas. The first is $\mathfrak{F}(p)$. For the second, we chose an affine coordinate system

² The cross ratio of three collinear points p, q, r is $\frac{|pq|}{|pr|}$. It is known that the cross ratio is an affine invariant.

$(\mathcal{F}_{st}, \mathcal{I}_t)$	Set of point predicates $\Pi(\mathcal{F}_{st}, \mathcal{I}_t)$
$(\mathcal{V}_{st}, \mathcal{I}_t)$	$\{\mathbf{Between}^{n+1}, \mathbf{Before}, \mathbf{UnitTime}, =_{\text{space}}\}$
$(\mathcal{V}(\mathcal{R})_{st}, \mathcal{I}_t)$	$\{\mathbf{Between}^{n+1}, \mathbf{Before}, \mathbf{UnitTime}, =_{\text{space}}, \mathbf{EqDist}, \mathbf{UnitDist}\}$
$(\mathcal{AC}_{st}, \mathcal{I}_t)$	$\{\mathbf{Between}^{n+1}, \mathbf{Before}, \mathbf{UnitTime}\}$
$(\mathcal{AC}(\mathcal{R})_{st}, \mathcal{I}_t)$	$\{\mathbf{Between}^{n+1}, \mathbf{Before}, \mathbf{UnitTime}, \mathbf{EqDist}, \mathbf{UnitDist}\}$

Table 2. The different point-predicate sets for the physical genericity notions.

$(O^p, E_1^p, E_2^p, \dots, E_n^p)$ in the plane of points that are co-temporal with p ; determine in this plane the spatial coordinates p_1, p_2, \dots, p_n of p with respect to the coordinate system $(O^p, E_1^p, E_2^p, \dots, E_n^p)$; and, using the point-predicate **EqCR**, communicate these coordinates to the line OE_1 in the computation plane. The third conjunct communicates the time coordinate of p to the computation plane as p_t , using the expression **EqCRst** $(O, E_1, p_t, p_{t_O}, p_{t_E}, p)$.

When all atomic subformulas of φ have been translated, the logical connectives can be added almost naturally (care has to be taken that in connecting, e.g., the subformulas $\mathfrak{F}(x_1, \dots, x_n; t)$ and $\mathfrak{F}(x'_1, \dots, x'_n; t')$ it is important that if $t = t'$, then the same affine coordinate system is used in their translation; hereto all possible orders of the time variables in the formula φ are considered and the same affine coordinate system is used for co-temporal events). \square

Proof of Proposition 2. Note that we only consider a finite number of moments in time in the previous proof (there are only a finite number of time variables in any FO formula φ). This implies that the transformation groups \mathcal{A}_{st}^f and \mathcal{A}_{st} yield the same results. In between the moments of time that are considered, it is indeed not important which transformation function is used. \square

Theorem 4 has a number of corollaries. We need two extra point predicates, namely **EqDist^{cotemp}** and **Posⁿ**:

- **EqDist^{cotemp}** (p, q, r, s) is true if and only if for four co-temporal points p, q, r and s the distance between p and q equals the distance between r and s ; and
- **Posⁿ** (p_0, p_1, \dots, p_n) expresses that for $n + 1$ co-temporal points p_0, p_1, \dots, p_n , (p_0, p_1, \dots, p_n) forms a positively oriented coordinate system.

Corollary 2. Let $(\mathcal{F}_{st}, \mathcal{F}_t)$ and $\Pi(\mathcal{F}_{st}, \mathcal{F}_t)$ be taken from Table 3. The language $\mathcal{P}(\Pi(\mathcal{F}_{st}, \mathcal{F}_t))$ is sound and complete for the $(\mathcal{F}_{st}, \mathcal{F}_t)$ -generic FO queries. \square

$(\mathcal{F}_{st}, \mathcal{F}_t)$	Sets of point predicates $\Pi(\mathcal{F}_{st}, \mathcal{F}_t)$
$(\mathcal{A}_{st}^{(f)}, \mathcal{A}_t)$	{ Betweenⁿ , Before , EqCR , EqCRst }
$(\mathcal{A}_{st}^{(f)}, \mathcal{I}_t)$	{ Betweenⁿ , Before , EqCR , EqCRst , UnitTime }
$(\mathcal{A}_{st}^{(f)}, Id_t)$	{ Betweenⁿ , Before , EqCR , EqCRst , UnitTime , $+_t$, $*_t$, 0_t , 1_t }
$(\mathcal{S}_{st}^{(f)}, \mathcal{F}_t)$	$\Pi(\mathcal{A}_{st}^{(f)}, \mathcal{F}_t) \cup \{\mathbf{EqDist}^{\text{cotemp}}\}$
$(\mathcal{I}_{st}^{(f)}, \mathcal{F}_t)$	$\Pi(\mathcal{A}_{st}^{(f)}, \mathcal{F}_t) \cup \{\mathbf{EqDist}^{\text{cotemp}}, \mathbf{UnitDist}\}$
$(\mathcal{T}_{st}^{(f)}, \mathcal{F}_t)$	$\Pi(\mathcal{A}_{st}^{(f)}, \mathcal{F}_t) \cup \{\mathbf{EqDist}^{\text{cotemp}}, \mathbf{UnitDist}, \leq_i (1 \leq i \leq n), \mathbf{Pos}^n\}$

Table 3. The different sets of point predicate for some spatio-temporal genericity notions. In the last three cases, we have $\mathcal{F}_t \in \{\mathcal{A}_t, \mathcal{I}_t, Id_t\}$.

5 Sound and complete languages for the generic computable spatio-temporal queries

In this section, we show that most of the languages $\mathcal{P}(\{P_1, P_2, \dots, P_m\})$ of the previous section, when augmented with **While**, yield sound and computationally complete languages for the genericity classes characterized by the point-predicates P_1, P_2, \dots, P_m .

Point-based logics extended with While. Let $\{P_1, P_2, \dots, P_m\}$ be a finite set of point predicates. Syntactically, a *program* in the language $\mathcal{P}(\{P_1, P_2, \dots, P_m\}) + \text{While}$ is then a finite sequence of *statements* and *while-loops*. Each statement has the form

$$R := \{(p_1, \dots, p_k) \mid \varphi(p_1, \dots, p_k)\},$$

with R a relation variable of arity k (the variables p_i range over $\mathbf{R}^n \times \mathbf{R}$) and φ a formula in $\mathcal{P}(\{P_1, P_2, \dots, P_m\})$ in which relation names of input relations and of previously introduced relation variables may be used. A while-loop has the form

$$\text{while } \varphi \text{ do } P,$$

where P is a program and φ is a sentence in $\mathcal{P}(\{P_1, P_2, \dots, P_m\})$ augmented with previously introduced relation variables.

Semantically, a program in the query language $\mathcal{P}(\{P_1, P_2, \dots, P_m\}) + \text{While}$ expresses a spatio-temporal query in the obvious way as soon as one of its relation variables has been designated as the output variable. \square

We follow the same structure as in the previous section. We first discuss notions of genericity determined by time-independent transformations with applications to physics and then focus on time-dependent transformations.

5.1 Genericity for time-independent transformations

In this section, we prove a general result concerning computable $(\mathcal{F}_{st}, \mathcal{F}_t)$ -generic queries where $(\mathcal{F}_{st}, \mathcal{F}_t)$ is a time-independent affinity of $\mathbf{R}^n \times \mathbf{R}$.

Let $(O, E_1, E_2, \dots, E_{n+1})$ be the standard coordinate system of $\mathbf{R}^n \times \mathbf{R}$. The following meta-theorem can then be proven, using techniques introduced by Gyssens, Van den Bussche and Van Gucht [17].

Theorem 5. *Let \mathcal{F}_{st} be a subgroup of \mathcal{A}_{st}^c and let \mathcal{F}_t be a subgroup of \mathcal{A}_t . Let $\{P_1, P_2, \dots, P_m\}$ be a set of point predicates that contains **Between** ^{$n+1$} and **Before**. If the predicates in $\{P_1, P_2, \dots, P_m\}$ are FO expressible and invariant under the transformations of $(\mathcal{F}_{st}, \mathcal{F}_t)$ and if the fact “ $(p_0, p_1, p_2, \dots, p_{n+1})$ is the image of the standard coordinate system $(O, E_1, E_2, \dots, E_{n+1})$ under some element f of $(\mathcal{F}_{st}, \mathcal{F}_t)$ ” is expressible in $\mathcal{P}(\{P_1, P_2, \dots, P_m\})$, then $\mathcal{P}(\{P_1, P_2, \dots, P_m\}) + \text{While}$ is sound and complete for the computable $(\mathcal{F}_{st}, \mathcal{F}_t)$ -generic queries. \square*

We apply the previous theorem to some groups $(\mathcal{F}_{st}^c, \mathcal{F}_t)$. Further applications can be found in the next subsection.

Corollary 3. *Let $(\mathcal{F}_{st}, \mathcal{F}_t)$ be a group from Table 1 and let $\Pi(\mathcal{F}_{st}, \mathcal{F}_t)$ be as in Table 1. The point language $\mathcal{P}(\Pi(\mathcal{F}_{st}, \mathcal{F}_t)) + \text{While}$ is sound and complete for the computable $(\mathcal{F}_{st}, \mathcal{F}_t)$ -generic queries. \square*

5.2 Applications to Physics

Here, we focus again on the transformation groups $(\mathcal{V}_{st}, \mathcal{T}_t)$, $(\mathcal{V}(\mathcal{R})_{st}, \mathcal{T}_t)$, $(\mathcal{AC}_{st}, \mathcal{T}_t)$ and $(\mathcal{AC}(\mathcal{R})_{st}, \mathcal{T}_t)$. The following results can be proven using Theorem 5.

Theorem 6. *Let $(\mathcal{F}_{st}, \mathcal{T}_t)$ be a group from Table 2 and let $\Pi(\mathcal{F}_{st}, \mathcal{T}_t)$ be as in Table 2. The point language $\mathcal{P}(\Pi(\mathcal{F}_{st}, \mathcal{T}_t)) + \text{While}$ is sound and complete for the computable $(\mathcal{F}_{st}, \mathcal{T}_t)$ -generic queries. \square*

5.3 Genericity for time-dependent transformations

Finally, we study notions of genericity determined by time-dependent transformations. Here, we only show results for the groups of arbitrary time-dependent transformations \mathcal{F}_{st} . For the groups \mathcal{F}_{st}^f the problem of identifying sound and complete languages is open.

Theorem 7. *The point language $\mathcal{P}(\{\mathbf{Between}^n, \mathbf{Before}, \mathbf{EqCR}, \mathbf{EqCR}^{\text{st}}\}) + \text{While}$ is sound and complete for the $(\mathcal{A}_{st}, \mathcal{A}_t)$ -generic computable queries. \square*

Proof sketch. We show that an $(\mathcal{A}_{st}, \mathcal{A}_t)$ -generic computable query Q can be simulated in the language $\mathcal{P}(\{\mathbf{Between}^n, \mathbf{Before}, \mathbf{EqCR}, \mathbf{EqCR}^{\text{st}}\}) + \text{While}$. This simulation is broken up into three steps: (1) an input database \mathcal{F} is encoded as a natural number $N(\mathcal{F})$; (2) Q is simulated by a computable function \bar{Q} on natural numbers; and (3) $\bar{Q}(N(\mathcal{F}))$ is decoded into a spatio-temporal database. Since all three steps can be expressed in $\mathcal{P}(\{\mathbf{Between}^n, \mathbf{Before}, \mathbf{EqCR}, \mathbf{EqCR}^{\text{st}}\}) + \text{While}$, as we will show, this gives us the desired result.

Steps (2) and (3) differ little from the ones described by Gyssens, Van den Bussche and Van Gucht [17], so we skip them in this sketch. The major difference is in step (1). Essentially, step (1) is a loop over the natural numbers, where for each number N it is tested whether or not N encodes an image of the database \mathcal{F} under a transformation $f \in (\mathcal{A}_{st}, \mathcal{A}_t)$. The fact that N encodes some spatio-temporal database \mathcal{F}_N can be expressed in $\mathcal{P}(\{\mathbf{Between}^n, \mathbf{Before}, \mathbf{EqCR}, \mathbf{EqCR}^{\text{st}}\}) + \text{While}$ by modifying the technique of Gyssens, Van den Bussche and Van Gucht [17] to the technique of the *computation plane* explained in the proof of Theorem 4. In $\mathcal{P}(\{\mathbf{Between}^n, \mathbf{Before}, \mathbf{EqCR}, \mathbf{EqCR}^{\text{st}}\})$ it can then be expressed that \mathcal{F}_N can be transformed into \mathcal{F} by an element of the group $(\mathcal{A}_{st}, \mathcal{A}_t)$. Indeed, in the language FO it can be expressed that for all moments in time t there exists an affinity A_t such that a spatial point p belongs to the snapshot of \mathcal{F}_N at moment t if and only if $A_t(p)$ belongs to the snapshot \mathcal{F} at

time $f_t(t)$, where f_t is an element of \mathcal{A}_t . Using Theorem 4, it can therefore also be expressed in $\mathcal{P}(\{\mathbf{Between}^n, \mathbf{Before}, \mathbf{EqCR}, \mathbf{EqCR}^{\text{st}}\})$. \square

The previous theorem has a number of corollaries.

Corollary 4. *Let $(\mathcal{F}_{st}, \mathcal{F}_t)$ be one of the groups $(\mathcal{A}_{st}, \mathcal{A}_t)$, $(\mathcal{A}_{st}, \mathcal{I}_t)$, $(\mathcal{A}_{st}, \text{Id}_t)$, $(\mathcal{S}_{st}, \mathcal{G}_t)$, $(\mathcal{I}_{st}, \mathcal{G}_t)$, or $(\mathcal{T}_{st}, \mathcal{G}_t)$ with $\mathcal{G}_t \in \{\mathcal{A}_t, \mathcal{T}_t, \text{Id}_t\}$ and let $\Pi(\mathcal{F}_{st}, \mathcal{F}_t)$ be as in Table 3. The point language $\mathcal{P}(\Pi(\mathcal{F}_{st}, \mathcal{F}_t)) + \text{While}$ is sound and complete for the $(\mathcal{F}_{st}, \mathcal{F}_t)$ -generic computable queries. \square*

6 Conclusion and discussion

We have investigated different genericity classes relative to the constraint database model for spatio-temporal databases and we have identified sound and complete languages for the first-order, respectively the computable, queries in (most of) these genericity classes. Some results were obtained by techniques introduced by Gyssens, Van den Bussche and Van Gucht [17], but for time-dependent transformations we have introduced new proof techniques.

For what concerns computationally complete languages these techniques seem to be insufficient to deal with the genericity notions that are expressed by the groups $(\mathcal{A}_{st}^f, \mathcal{A}_t)$, $(\mathcal{A}_{st}^f, \mathcal{I}_t)$, $(\mathcal{A}_{st}^f, \text{Id}_t)$, $(\mathcal{S}_{st}^f, \mathcal{G}_t)$, $(\mathcal{I}_{st}^f, \mathcal{G}_t)$, and $(\mathcal{T}_{st}^f, \mathcal{G}_t)$ with $\mathcal{G}_t \in \{\mathcal{A}_t, \mathcal{T}_t, \text{Id}_t\}$. The problem in adapting the proof technique of Theorem 7 to these groups is that it is not clear how we can express in the respective point-based logics that two spatio-temporal databases can be mapped to each other by some piece-wise constant affinity. Indeed, since the number of pieces is not defined *a priori*, this might not be expressible. This would imply that yet another new proof technique would be required to deal with the remaining cases.

References

1. D. Abel and B. C. Ooi (eds.), *Advances in spatial databases—3rd Symposium (SSD'93)*, Lecture Notes in Computer Science, vol. 692, Springer-Verlag, 1993.
2. J. F. Allen, *Maintaining knowledge about temporal intervals*, Communications of the ACM **26** (1983), no. 11, 832–843.
3. J. F. Allen and G. Ferguson, *Actions and events in interval temporal logic*, Journal of Logic and Computation **4** (1994), no. 5, 531–579.
4. M. H. Böhlen, Ch. S. Jensen, and M. Scholl (eds.), *Proceedings of the international workshop on spatio-temporal database management (STDBM'99)*, Lecture Notes in Computer Science, vol. 1678, Springer, 1999.
5. A. Buchmann (ed.), *Design and implementation of large spatial databases—1st Symposium (SSD'89)*, Lecture Notes in Computer Science, vol. 409, Springer-Verlag, 1989.
6. A. K. Chandra and D. Harel, *Computable queries for relational data bases*, Journal of Computer and System Sciences **21** (1980), no. 2, 156–178.
7. C. X. Chen and C. Zaniolo, *SQLST: A spatio-temporal data model and query language*, Conceptual Modeling (ER'00) (V. Storey, A. Laender, S. Liddle, eds.), Lecture Notes in Computer Science, vol. 1920, Springer-Verlag, 2000, pp. 96–111.

8. J. Chomicki and P. Revesz, *Constraint-based interoperability of spatio-temporal databases*, in [26], pp. 142–161.
9. ———, *A geometric framework for specifying spatiotemporal objects*, Proceedings of the 6th International Workshop on Temporal Representation and Reasoning, IEEE Computer Society, 1999, pp. 41–46.
10. M. J. Egenhofer and J. R. Herring (eds.), *Advances in spatial databases—4th Symposium (SSD'95)*, Lecture Notes in Computer Science, vol. 951, Springer-Verlag, 1995.
11. M. Erwig, R. H. Güting, M. Schneider, and M. Vazirgiannis, *Spatio-temporal data types: An approach to modeling and querying moving objects in databases*, *GeoInformatica* **3** (1999), no. 3, 269–296.
12. A. Frank, S. Grumbach, R. Güting, C. Jensen, M. Koubarakis, N. Lorentzos, Y. Manopoulos, E. Nardelli, B. Pernici, H.-J. Schek, M. Scholl, T. Sellis, B. Theodoulidis, and P. Widmayer, *ChoroChronos: A research network for spatiotemporal database systems*, *SIGMOD Record* **28** (1999), 12–21.
13. S. Grumbach, P. Rigaux, and L. Segoufin, *Spatio-temporal data handling with constraints*, Proceedings of the 6th international symposium on Advances in Geographic Information Systems (ACM-GIS'98) (R. Laurini, K. Makki, and N. Pissinou, eds.), 1998, pp. 106–111.
14. O. Gunther and H.-J. Schek (eds.), *Advances in spatial databases—2nd Symposium (SSD '91)*, Lecture Notes in Computer Science, vol. 525, Springer-Verlag, 1991.
15. R. H. Güting, M. H. Bohlen, M. Erwig, C. S. Jensen, N. A. Lorentzos, M. Schneider, and M. Vazirgiannis, *A foundation for representing and querying moving objects*, *ACM Transactions on Databases Systems* **25** (2000), 1–42.
16. R.H. Güting (ed.), *Advances in spatial databases—6th Symposium (SSD'99)*, Lecture Notes in Computer Science, vol. 1651, Springer-Verlag, 1999.
17. M. Gyssens, J. Van den Bussche, and D. Van Gucht, *Complete geometric query languages*, *Journal of Computer and System Sciences* **58** (1999), no. 3, 483–511.
18. P. C. Kanellakis, G. M. Kuper, and P.Z. Revesz, *Constraint query languages*, *Journal of Computer and System Sciences* **51** (1995), 26–52 (also in Proceedings of the 9th ACM Symposium on Principles of Database Systems (PODS'90), ACM Press, 1990, pp. 299–313).
19. B. Kuijpers and D. Van Gucht, *Genericity in spatial databases*, Chapter 12 in [24], pp. 293–304.
20. B. Kuijpers, J. Paredaens, and D. Van Gucht, *Towards a theory of movie database queries*, Proceedings of the 7th International Workshop on Temporal Representation and Reasoning, IEEE Computer Society, 2000, pp. 95–102.
21. G. Kuper and M. Scholl, *Geographic information systems*, Constraint databases (J. Paredaens, G. Kuper, and L. Libkin, eds.), Springer-Verlag, 2000, pp. 175–198.
22. L. Libkin, *Variable independence, quantifier elimination, and constraint representation*, Automata, Languages and Programming, 27th International Colloquium (ICALP 2000) (U. Montanari, J. D. P. Rolim, and E. Welzl, eds.), Lecture Notes in Computer Science, vol. 1853, Springer-Verlag, 2000, pp. 260–271.
23. J. Paredaens, J. Van den Bussche, and D. Van Gucht, *Towards a theory of spatial database queries*, Proceedings of the 13th ACM Symposium on Principles of Database Systems (PODS'94), ACM Press, 1994, pp. 279–288.
24. J. Paredaens, G. Kuper, and L. Libkin (eds.), *Constraint databases*, Springer-Verlag, 2000.
25. D. Pfoser and N. Tryfona, *Requirements, definitions and notations for spatiotemporal application environments*, Proceedings of the 6th International Symposium

- on Advances in Geographic Information Systems (ACM-GIS'98) (R. Laurini, K. Makki, and N. Pissinou, eds.), 1998, pp. 124–130.
26. M. Scholl and A. Voisard (eds.), *Advances in spatial databases—5th Symposium (SSD'97)*, Lecture Notes in Computer Science, vol. 1262, Springer-Verlag, 1997.
 27. W. Schwabhäuser, W. Szmielew, and A. Tarski, *Metamathematische methoden in der geometrie*, Springer-Verlag, 1983.
 28. J. Van den Bussche, *Constraint databases, queries and query languages*, Chapter 2 in [24], pp. 21–54.