# A triangle-based logic for affine-invariant querying of spatial and spatio-temporal data

*Sofie Haesevoets*, *Bart Kuijpers**
Hasselt University

November 3, 2008

**Abstract**

In spatial databases, incompatibilities often arise due to different choices of origin or unit of measurement (e.g., centimeters versus inches). By representing and querying the data in an affine-invariant manner, we can avoid these incompatibilities.

In practice, spatial (resp., spatio-temporal) data is often represented as a finite union of triangles (resp., moving triangles). As two arbitrary triangles are equal up to a unique affinity of the plane, they seem perfect candidates as basic units for an affine-invariant query language.

We propose a so-called "triangle logic", a query language that is affine-generic and has triangles as basic elements. We show that this language has the same expressive power as the affine-generic fragment of first-order logic over the reals on triangle databases. We illustrate that the proposed language is simple and intuitive. It can also serve as a first step towards a "moving-triangle logic" for spatio-temporal data.

## 1 Introduction and summary of the results

In the area of spatial database research, a lot of attention has been focused on affine invariance of both data and queries. The main purpose of studying affine invariance is to obtain methods and techniques that are not affected by affine transformations of the input spatial data. This means that a particular choice of origin or some artificial choice of unit of measure (e.g., inches versus centimeters) has no effect on the final result of the method or query.

In computer vision, the so-called *weak perspective assumption* [25] is widely adopted. When an object is repeatedly photographed under different camera angles, all the different images are assumed to be affine transformations of each other. This assumption led to the need for affine-invariant similarity measures between pairs of pictures [13, 14, 18]. In computer graphics, affine-invariant norms and triangulations have been studied [20]. In the field of spatial and spatio-temporal constraint databases [23, 24], affine-invariant query languages [10, 11] have been proposed. In Section 2, we will go into more detail about the affine-invariant language for spatial constraint data proposed by Gyssens, Van den Bussche

*Corresponding author: Hasselt University, Theoretical Computer Science, B-3590 Diepenbeek, Belgium, `bart.kuijpers@uhasselt.be`

and Van Gucht [11]. Affinities are one of the transformation groups proposed at the introduction of the concept of "genericity of query languages" applied to constraint databases [22]. Also various subgroups of the affinities [22] such as isometries, similarities, ... and supergroups of the affinities such as topology preserving transformations [16, 21] have been studied in the same context.

If we now focus on the representation of two-dimensional spatial data, we see that, in practice, two-dimensional figures are approximated often as a finite union of triangles. In geographic information systems, Triangulated Irregular Networks (TIN) [19] are often used. In computer graphics, data is approximated by triangular meshes (e.g., [4]). Also, for spatio-temporal databases, "parametric moving triangle"-based models have been proposed and studied [6, 7].

Remark that two arbitrary triangles are indistinguishable up to an affinity of the plane. Indeed, each triangle in the plane can be mapped to each other triangle in the plane by a unique affinity.[1]

The combination of the need for affine-invariance, the representation of data by means of triangles in practice, and the fact that triangles itself are an affine-invariant concept, led to the idea of introducing a query language based on triangles. If the data is represented as a collection of triangles, why should one reason about it as a collection of points [11], or, even indirectly, by means of coordinates (as is the case for the classical spatial constraint language, first-order logic over the reals)? We consider first-order languages, in which variables are interpreted to range over triangles, both spatial and spatio-temporal.

We propose a new, first-order query language that has triangles as basic elements. We show that this language has the same expressive power as the affine-invariant segment of the queries in first-order logic over the reals on triangle databases. Afterwards, we give some examples illustrating the expressiveness of our language. We also address the notion of safety of triangle queries. We show that it is undecidable whether a specific triangle query returns a finite output on finite input. It is, however, decidable whether the output of a query on a particular finite input database can be represented as a finite union of triangles. We show that we can express this finite representation in our triangle language. Afterwards, we extend our results to the case of spatio-temporal triangles, *i.e.*, triples of co-temporal points in $(\mathbb{R}^2 \times \mathbb{R})$.

## 2 Related work and preliminaries

The idea that the result of a query on some spatial input database should be invariant under some group of spatial transformations, was first introduced by Paredaens, Van den Bussche and Van Gucht [22]. In a follow-up article, Gyssens, Van den Bussche and Van Gucht [11] proposed several first-order query languages, invariant under group of the affinities or some subgroup thereof. In these languages, variables are assumed to range over points in some real space $\mathbb{R}^n$ ($\mathbb{R}$ is the set of real numbers), rather than over real numbers (coordinates of such points). For the group of the affinities, the point language with only one predicate that expresses *betweenness* of points, was shown to have the same expressivity as the affine-invariant fragment of first-order logic over the reals, on point databases. We will use this

[1] This is only true if the triangle is not degenerated, i.e., no corner points coincide. Otherwise, there are more such affinities.

result to prove the expressiveness of our triangle-based logic. Therefore, we will recall some definitions from the article from Gyssens, Van den Bussche and Van Gucht [11]. All definitions listed in this section can be found there.

We start with the well-known definition of a constraint database, or semi-algebraic database, as this is the general setting which we will be working in.

**Definition 2.1** A *semi-algebraic relation in* $\mathbb{R}^n$ is a subset of $\mathbb{R}^n$ that can be described as a Boolean combination of sets of the form

$$\{(x_1, x_2, \ldots, x_n) \in \mathbb{R}^n \mid p(x_1, x_2, \ldots, x_n) > 0\},$$

with $p$ a polynomial with integer coefficients in the real variables $x_1, x_2, \ldots, x_n$. □

In mathematical terms, semi-algebraic relations are known as *semi-algebraic sets* [5].

We also call a semi-algebraic relation in $\mathbb{R}^n$ *a semi-algebraic relation of arity* $n$. A semi-algebraic database is essentially a finite collection of semi-algebraic relations. We give the definition next.

**Definition 2.2** A *(semi-algebraic) database schema* $\sigma$ is a finite set of relation names, where each relation name $R$ has an arity associated to it, which is a natural number and which is denoted by $ar(R)$.

Let $\sigma$ be a database schema. A *semi-algebraic database over* $\sigma$ is a structure $\mathcal{D}$ over $\sigma$ with domain $\mathbb{R}$ such that, for each relation name $R$ of $\sigma$, the associated relation $R^{\mathcal{D}}$ in $\mathcal{D}$ is a semi-algebraic relation of arity $ar(R)$. □

**Example 2.1** Let $\sigma = \{R, S\}$, with $ar(R) = 2$ and $ar(S) = 1$ be a semi-algebraic database schema. Then the structure $\mathcal{D}$ given by

$$(\mathbb{R}, R^{\mathcal{D}} = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1^2 + x_2^2 < 1\}, S^{\mathcal{D}} = \{x \in \mathbb{R} \mid 0 \leq x \leq 1\})$$

is an example of a semi-algebraic database over $\sigma$ that contains the open unit disk and the closed unit interval. □

**Definition 2.3** Let $\sigma$ be a $n$-dimensional semi-algebraic database schema. The language FO(+, ×, <, 0, 1, $\sigma$) (or FO(+, ×, <, 0, 1) , if $\sigma$ is clear from the context), first-order logic over the real numbers with polynomial constraints, is the first-order language with variables that are assumed to range over real numbers, where the atomic formulas are either of the form $p(x_1, x_2, \ldots, x_n) > 0$, with $p$ a polynomial with integer coefficients in the real variables $x_1, x_2, \ldots, x_n$, or the relation names from $\sigma$ applied to real terms. Atomic formulas are composed using the operations $\wedge$, $\vee$ and $\neg$ and the quantifiers $\forall$ and $\exists$. □

**Example 2.2** Consider the semi-algebraic database from Example 2.1. The expression

$$R(x, y) \wedge y > 0$$

is a FO(+, ×, <, 0, 1, $\{R, S\}$)-formula selecting the part of the open unit disk that lies strictly above the $x$-axis. □

We restrict all further definitions and results to dimension $n = 2$, as this is the dimension we will be working with in the rest of this text, although they were originally proved to hold for arbitrary $n$, $n \geq 2$.

Now we give the definition of a geometric database, a special type of constraint database that contains a possibly infinite number of points.

**Definition 2.4** Let $\sigma$ be a geometric database schema. A *geometric database* over $\sigma$ in $\mathbb{R}^2$ is a structure $\mathcal{D}$ over $\sigma$ with domain $\mathbb{R}^2$ such that, for each relation name $R$ of $\sigma$, the associated relation $R^{\mathcal{D}}$ in $\mathcal{D}$ is semi-algebraic. □

A geometric database $\mathcal{D}$ over $\sigma$ in $\mathbb{R}^2$ can be viewed naturally as a semialgebraic database $\overline{\mathcal{D}}$ over the schema $\overline{\sigma}$, which has, for each relation name $R$ of $\sigma$, a relation name $\overline{R}$ with arity $2k$, where $k$ is the arity of $R$ in $\sigma$. For each relation name $R$, of arity $k$, $\overline{R}^{\overline{\mathcal{D}}}$ is obtained from $R^{\mathcal{D}}$ by applying the *canonical bijection*[2] between $(\mathbb{R}^2)^k$ and $\mathbb{R}^{2k}$.

**Definition 2.5** Let $\sigma$ be a geometric database schema. A *$k$-ary geometric query* $Q$ over $\sigma$ in $\mathbb{R}^2$ is a partial computable function on the set of geometric databases over $\sigma$. Furthermore, for each geometric database $\mathcal{D}$ over $\sigma$ on which $Q$ is defined, $Q(\mathcal{D})$ is a geometric relation of arity $k$. □

Queries that are invariant under some transformation group $G$ of $\mathbb{R}^2$, are also called *$G$-generic* [22]. We define this next:

**Definition 2.6** Let $\sigma$ be a geometric database schema and Q a geometric query over $\sigma$ in $\mathbb{R}^2$. Let $G$ be a group of transformations of $\mathbb{R}^2$. Then $Q$ is called *$G$-generic* if, for any two geometric databases $\mathcal{D}$ and $\mathcal{D}'$ over $\sigma$ in $\mathbb{R}^2$ for which $\mathcal{D}' = g(\mathcal{D})$, for some $g \in G$, we have that $Q(\mathcal{D}') = g(Q(\mathcal{D}))$. □

In the remainder of this text, we will only focus on the group $G$ of *affinities*. The affinities of $\mathbb{R}^2$ form the group of linear transformations having a regular matrix, i.e., their matrix has a determinant different from zero. Affinities of the plane have the following form:

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix},$$

where $ad - bc$ is different from zero.

We now give the definition of the first-order point logic FO({**Between**}), a first-order language where the variables are not interpreted as real numbers, as in FO(+, ×, <, 0, 1) , but as 2-dimensional points.

We first introduce the point predicate **Between**.

**Definition 2.7** Let $p = (p_x, p_y)$, $q = (q_x, q_y)$ and $r = (r_x, r_y)$ be points in the plane. The expression **Between**$(p, q, r)$ is true if and only if either $q$ lies on the closed line segment between $p$ and $r$ or $p$ and/or $q$ and/or $r$ coincide. □

In Figure 1, **Between**$(p, t, q)$, **Between**$(p, p, q)$ and **Between**$(t, s, r)$ are true. On the other hand, but **Between**$(t, q, p)$ and **Between**$(p, q, r)$ are not true.

[2] The canonical bijection between $(\mathbb{R}^2)^k$ and $\mathbb{R}^{2k}$ associates with each $k$-tuple $(\mathbf{x_1}, \ldots, \mathbf{x_k})$ of $(\mathbb{R}^2)^k$ the $2k$-tuple $(x_1^1, x_1^2, \ldots, x_k^1, x_k^2)$, where for $1 \leq i \leq k$ $\mathbf{x_i} = (x_i^1, x_i^2)$.

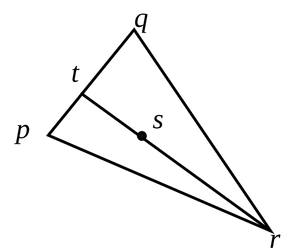


Figure 1: The predicate **InTriangle** can be expressed using **Between**.

**Definition 2.8** Let $\sigma$ be a 2-dimensional geometric database schema. The first-order point language over $\sigma$ and $\{$**Between**$\}$, denoted by FO($\{$**Between**$\}$, $\sigma$) (or, if $\sigma$ is clear from the context, by FO($\{$**Between**$\}$)), is a first-order language with variables that range over points in $\mathbb{R}^2$, (denoted $\hat{p}, \hat{q}, \ldots$), where the atomic formulas are equality constraints on point variables, the predicate **Between** applied to point variables, and the relation names from $\sigma$ applied to point variables. $\square$

A FO($\{$**Between**$\}$)-formula $\varphi(\hat{p_1}, \hat{p_2}, \ldots, \hat{p_l})$ over the relation names of $\sigma$ and the predicate **Between** defines on each geometric database $\mathcal{D}$ over $\sigma$ a subset $\varphi(\mathcal{D})$ of $(\mathbb{R}^2)^l$ in the standard manner.

Gyssens, Van den Bussche and Van Gucht have shown that the language FO($\{$**Between**$\}$) expresses exactly all affine-generic geometric queries expressible in FO(+, $\times$, <, 0, 1).

# 3 Notations

In this Section, we introduce triangle variables and constants. We work in $\mathbb{R}^2$. Spatial triangle variables will be denoted $\triangle_1, \triangle_2, \ldots$. Constants containing such triples of points will be denoted $T_1, T_2, \ldots$, or $T_{\mathbf{abc}}$ when we want to emphasize the relationship between a triangle and its corner points **a**, **b**, **c** $\in \mathbb{R}^2$. We remark that triangles can be modelled as triples of points in $\mathbb{R}^2$. Occasionally, we will need to refer to the area of a triangle. The area of a triangle $T$ will be abbreviated $A(T)$.

We also introduce spatio-temporal triangles, which can be modelled as triples of moving points in $\mathbb{R}^2$. Variables referring to spatio-temporal triangles are distinguished from spatial triangle variables by a superscript: $\triangle_1^{st}, \triangle_2^{st}, \ldots$. The same holds for constants, which are denoted $T_1^{st}, T_2^{st}, T_{pqr}^{st} \ldots$. We will also define triangle databases. For the spatial and spatio-temporal case respectively, we will use the symbols $\mathcal{D}$ and $\mathcal{D}^{st}$ to indicate triangle database instances.

The names of (spatio-temporal) triangle relations and database schemas containing such relation names will be recognizable by their hat: $\hat{R}$, $\hat{\sigma}$ and $\hat{R}^{st}$, $\hat{\sigma}^{st}$, respectively. Spatial and spatio-temporal point relation names and schemas are denoted $\dot{R}$ and $\dot{\sigma}$, $\dot{R}^{st}$ and $\dot{\sigma}^{st}$, respectively.

## 4 Definitions

We start with the definition of a *triangle database*, *i.e.*, a database that contains a (possibly infinite) collection of triangles. We define both spatial triangle databases and spatio-temporal triangle databases. We model triangles by triples of points of $\mathbb{R}^2$, *i.e.*, by elements of $(\mathbb{R}^2)^3$. Moving or changing (*i.e.*, spatio-temporal) triangles are modelled by sets of triples of co-temporal points in $(\mathbb{R}^2 \times \mathbb{R})$,*i.e.*, by sets of elements of $(\mathbb{R}^2 \times \{\tau_0\})^3$, for some $\tau_0 \in \mathbb{R}$. Triangles can degenerate, *i.e.*, corner points are allowed to coincide. For the remainder of this text, the term triangle refers to a triple of points. We refer to the set of points that is represented by a triangle as the *drawing* of that triangle.

**Definition 4.1 (Drawing of a triangle)** • Let $T = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3) \in (\mathbb{R}^2)^3$ be a spatial triangle. The *drawing* of $T$ is the subset of $\mathbb{R}^2$ that is the convex closure of the points $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{a}_3$.
• Let $T^{st} = (p_1, p_2, p_3) \in (\mathbb{R}^2 \times \mathbb{R})^3$ be a spatio-temporal triangle. The *drawing* of $T^{st}$ is the subset of co-temporal points of $(\mathbb{R}^2 \times \mathbb{R})$ that is the convex closure of the points $p_1$, $p_2$ and $p_3$. □

We now introduce four bijections.

- *can* : $(\mathbb{R}^n)^k \to \mathbb{R}^{nk}$ maps tuples $(\mathbf{a}_1, \ldots, \mathbf{a}_k)$ to $(a_{1,1}, \ldots, a_{1,n}, \ldots, a_{k,1}, \ldots, a_{k,n})$, where for $1 \leq i \leq k$ and $1 \leq j \leq n$, $a_{i,j}$ denotes the $j$th real coordinate of $\mathbf{a}_i$;
- $can_{ST}$ : $(\mathbb{R}^n \times \mathbb{R})^k \to \mathbb{R}^{(n+1)\times k}$ maps tuples $((\mathbf{a}_1, \tau_1), \ldots, (\mathbf{a}_k, \tau_k))$ to $(a_{1,1}, \ldots, a_{1,n}, \tau_1, \ldots, a_{k,1}, \ldots, a_{k,n}, \tau_k)$, where for $1 \leq i \leq k$ and $1 \leq j \leq n$, $a_{i,j}$ denotes the $j$th real coordinate of $\mathbf{a}_i$;
- $can_{tr}$ : $((\mathbb{R}^2)^3)^k \to (\mathbb{R}^2)^{3k}$ maps $k$-tuples of triangles to $(3k)$ tuples of points in $\mathbb{R}^2$; and
- $can_{trST}$ : $((\mathbb{R}^2 \times \mathbb{R})^3)^k \to (\mathbb{R}^2 \times \mathbb{R})^{3k}$ maps $k$-tuples of spatio-temporal triangles to $(3k)$-tuples of points in $(\mathbb{R}^2 \times \mathbb{R})$.

**Definition 4.2 (Triangle relations and databases)** A *(triangle) database schema* $\hat{\sigma}$ is a finite set of relation names, where each relation name $\hat{R}$ has a natural number $ar(\hat{R})$, called its arity, associated to it.

• A subset $\mathcal{C}$ of $((\mathbb{R}^2)^3)^k$ is a *spatial triangle relation of arity* $k$ if

(i) its image under the canonical bijection $can \circ can_{tr} : ((\mathbb{R}^2)^3)^k \to \mathbb{R}^{6k}$ is a semi-algebraic relation of arity $6k$, and

(ii) for each element $c = ((\mathbf{a}_{1,1}, \mathbf{a}_{1,2}, \mathbf{a}_{1,3}), (\mathbf{a}_{2,1}, \mathbf{a}_{2,2}, \mathbf{a}_{2,3}), \ldots, (\mathbf{a}_{k,1}, \mathbf{a}_{k,2}, \mathbf{a}_{k,3})) \in \mathcal{C}$, also the elements $((\mathbf{a}_{1,j_{1,1}}, \mathbf{a}_{1,j_{1,2}}, \mathbf{a}_{1,j_{1,3}}), (\mathbf{a}_{2,j_{2,1}}, \mathbf{a}_{2,j_{2,2}}, \mathbf{a}_{2,j_{2,3}}), \ldots, (\mathbf{a}_{k,j_{k,1}}, \mathbf{a}_{k,j_{k,2}}, \mathbf{a}_{k,j_{k,3}}))$ are in $\mathcal{C}$, where $\sigma_i(1,2,3) = (j_{i,1}, j_{i,2}, j_{i,3})$ with $1 \leq i \leq k$ and $\sigma_i \in \mathcal{S}_3$ where $\mathcal{S}_3$ is the set of all permutations of $\{1, 2, 3\}$.

Let $\hat{\sigma}$ be a triangle database schema. A *spatial triangle database* over $\hat{\sigma}$ in $(\mathbb{R}^2)^3$ is a structure $\mathcal{D}$ over $\hat{\sigma}$ with domain $(\mathbb{R}^2)^3$ such that, for each relation name $\hat{R}$ of $\hat{\sigma}$, the associated triangle relation $\hat{R}^{\mathcal{D}}$ in $\mathcal{D}$ is a spatial triangle relation of arity $ar(\hat{R})$.

- A subset $\mathcal{C}$ of $((\mathbb{R}^2 \times \mathbb{R})^3)^k$ is a *spatio-temporal triangle relation of arity $k$* if

  (i) its image under the canonical bijection $can_{trST} \circ can_{ST} : ((\mathbb{R}^2 \times \mathbb{R})^3)^k \to \mathbb{R}^{9k}$ is a semi-algebraic relation of arity $9k$, and

  (ii) for each element $c = ((p_{1,1}, p_{1,2}, p_{1,3}), (p_{2,1}, p_{2,2}, p_{2,3}), \ldots, (p_{k,1}, p_{k,2}, p_{k,3})) \in \mathcal{C}$, also $((p_{1,j_{1,1}}, p_{1,j_{1,2}}, p_{1,j_{1,3}}), (p_{2,j_{2,1}}, p_{2,j_{2,2}}, p_{2,j_{2,3}}), \ldots, (p_{k,j_{k,1}}, p_{k,j_{k,2}}, p_{k,j_{k,3}}))$ are in $\mathcal{C}$, where $\sigma_i(1,2,3) = (j_{i,1}, j_{i,2}, j_{i,3})(1 \leq i \leq k; \sigma_i \in \mathcal{S}_3)$. Here, $\mathcal{S}_3$ is the set of all permutations of $\{1,2,3\}$.

Let $\hat{\sigma}^{st}$ be a triangle database schema. A spatio-temporal *triangle database* over $\hat{\sigma}^{st}$ is a structure $\mathcal{D}^{st}$ over $\hat{\sigma}^{st}$ with domain $(\mathbb{R}^2 \times \mathbb{R})^3$ such that, for each relation name $\hat{R}^{st}$ of $\hat{\sigma}^{st}$, the associated triangle relation $\hat{R}^{st\mathcal{D}^{st}}$ in $\mathcal{D}^{st}$ is a spatio-temporal triangle relation of arity $ar(\hat{R}^{st})$. □

We want to remark two things about the definition of triangle relations (as given in Definition 4.2), one about the items (i) and one about the items (ii) of the definition of triangle relations. They are discussed in Remark 4.1 below and Remark 4.4, which is postponed until after the definition of triangle database queries.

**Remark 4.1** A triangle database $\mathcal{D}$ over $\hat{\sigma}$ in $(\mathbb{R}^2)^3$ can be viewed naturally as a geometric database $\mathcal{S}$ over the schema $\dot{\sigma}$, which has, for each relation name $\hat{R}$ of $\hat{\sigma}$, a relation name $\dot{R}$ with arity $3 \times ar(\hat{R})$. For each relation name $\hat{R}$, of arity $k$, $\dot{R}^{\mathcal{S}}$ is obtained from $\hat{R}^{\mathcal{D}}$ by applying the canonical bijection $can_{tr} : ((\mathbb{R}^2)^3)^k \to (\mathbb{R}^2)^{3k}$. Analogously, a spatio-temporal triangle database $\mathcal{D}^{st}$ over $\hat{\sigma}^{st}$ can be viewed naturally as a spatio-temporal database $\mathfrak{T}$ over the schema $\dot{\sigma}^{st}$, which has, for each relation name $\hat{R}^{st}$ of $\hat{\sigma}^{st}$, a relation name $\dot{R}^{st}$ with arity $3 \times ar(\hat{R}^{st})$. For each relation name $\hat{R}^{st}$, of arity $k$, $\dot{R}^{st\mathfrak{T}}$ is obtained from $\hat{R}^{st\mathcal{D}^{st}}$ by applying the canonical bijection $can_{trST} : ((\mathbb{R}^2 \times \mathbb{R})^3)^k \to (\mathbb{R}^2 \times \mathbb{R})^{3k}$. □

**Example 4.1** It follows from the definition of triangle relations that they can be finitely represented by polynomial constraints on the coordinates of the corner points of the triangles they contain.

For example, the unary spatial triangle relation containing all triangles with one corner point on the $x$-axis, one on the $y$-axis and a third corner point on the diagonal $y = x$, can be finitely represented as follows:

$$\begin{aligned}\{(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3) = ((z_{1,x}, z_{1,y}), (z_{2,x}, z_{2,y}), (z_{3,x}, z_{3,y})) \in (\mathbb{R}^2)^3 \ \mid \\ (z_{1,x} = 0 \wedge z_{2,y} = 0 \wedge z_{3,x} = z_{3,y}) \vee (z_{1,x} = 0 \wedge z_{3,y} = 0 \wedge z_{2,x} = z_{2,y}) \\ \vee (z_{2,x} = 0 \wedge z_{1,y} = 0 \wedge z_{3,x} = z_{3,y}) \vee (z_{2,x} = 0 \wedge z_{3,y} = 0 \wedge z_{1,x} = z_{1,y}) \\ \vee (z_{3,x} = 0 \wedge z_{2,y} = 0 \wedge z_{1,x} = z_{1,y}) \vee (z_{3,x} = 0 \wedge z_{1,y} = 0 \wedge z_{2,x} = z_{2,y})\}.\end{aligned}$$

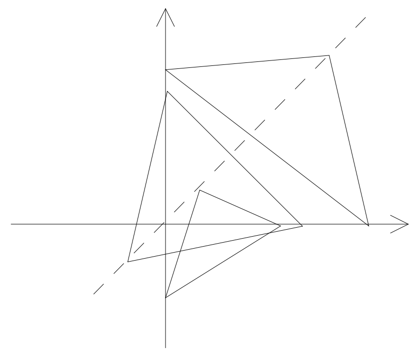

Figure 2: Some elements of the relation represented in Example 4.1.

Figure 2 gives some elements of this relation. Each triangle that is drawn is stored three times in the relation. □

**Remark 4.2** For the remainder of this text we assume that databases are finitely encoded by systems of polynomial equations and that a specific data structure is fixed (possible data structures are dense or sparse representations of polynomials). The specific choice of data structure is not relevant to the topic of this text, but we assume that one is fixed. When we talk about computable queries later on, we mean Turing computable with respect to the chosen encoding and data structures. □

We also remark the following.

**Remark 4.3** The data model and the query languages presented in this text can be extended straightforwardly to the situation where spatio-temporal relations are accompanied by classical thematic information. However, because the problem that is discussed here is captured by this simplified model, we stick to it for reasons of simplicity of exposition. □

We now define spatial and spatio-temporal triangle database queries.

**Definition 4.3 (Triangle database queries)** • Let $\hat{\sigma}$ be a triangle database schema and let us consider input spatial triangle databases over $\hat{\sigma}$. A *$k$-ary spatial triangle database query* $Q$ over $\hat{\sigma}$ is a computable partial mapping (in the sense of Remark 4.2) from the set of spatial triangle databases over $\hat{\sigma}$ to the set of $k$-ary spatial triangle relations.

• Let $\hat{\sigma}^{st}$ be a database schema and let us consider input spatio-temporal triangle databases over $\hat{\sigma}^{st}$. A *$k$-ary* spatio-temporal *triangle database query* $Q$ over $\hat{\sigma}^{st}$ is a computable partial mapping (in the sense of Remark 4.2) from the set of spatio-temporal triangle databases over $\hat{\sigma}^{st}$ to the set of $k$-ary spatio-temporal triangle relations. □

**Remark 4.4** In the (ii)-items of the definition of triangle relations, we require that, if a triangle $T$ is involved in a relation, that also all other triangles with the same drawing are stored in that relation. The reason for this is that we do not want the triangle queries to be dependent of the actual order and orientation used when enumerating the corner points of a triangle. When emphasizing property (ii) of a relation, we will call it *consistency* and talk about *consistent triangle relations*. Also, a database is said to be consistent, if all its relations are consistent. □

We illustrate the consistency property with some examples:

**Example 4.2** Let $\hat{\sigma} = \{\hat{R}\}$ be a database schema. First, we list some queries over $\hat{\sigma}$ that are not consistent:

- $Q_6$: *Give all triangles in $\hat{R}$ for which their first and second corner points coincide.*
- $Q_7$: *Give all triangles for which the segment defined by their first and second corner point is a boundary segment of one of the triangles in $\hat{R}$.*

Now some consistent queries follow:

- $Q_8$: *Give all triangles in $\hat{R}$ that are degenerated into a line segment.*
- $Q_9$: *Give all triangles that share a boundary segment with some triangle in $\hat{R}$.*

It is clear that the inconsistent queries are rather artificial. When a user specifies the triangles that should be in the result of a query, she intuitively thinks of the drawings of those triangles. The order of the corner points used in the construction of those triangles should not be important. □

**Remark 4.5** A spatio-temporal database $\mathcal{T}$ over $\sigma^{st}$ can be viewed in a natural way as a constraint database $D$ over the constraint schema $\sigma$, which has for each relation name $R^{st}$ of $\sigma^{st}$, a relation name $R$ of arity $(n+1)\times ar(R^{st})$. For each relation name $R^{st}$, $R^D$ is obtained from $R^{st^{\mathcal{T}}}$ by applying the canonical bijection $can_{ST} : (\mathbb{R}^n \times \mathbb{R})^{ar(R)} \to \mathbb{R}^{(n+1)\times ar(R)}$. We will use the notation introduced here, throughout this text. □

Analogously, spatial and spatio-temporal triangle database queries can be seen as constraint queries, and as spatial and spatio-temporal (point) database queries. We prefer the latter view, as we already developed affine-generic spatio-temporal point languages in [10], and there already exist affine-generic spatial point languages[11]. We define the equivalence between triangle queries and point queries formally:

**Definition 4.4 (Equivalence of point queries and triangle queries)** • Let $\hat{\sigma}$ be a triangle database schema and let us consider input spatial triangle databases over $\hat{\sigma}$. Let $\dot{\sigma}$ be the corresponding spatial point database schema (see Remark 4.1). Let $\hat{Q}$ be a $k$-ary spatial triangle database query over $\hat{\sigma}$ and let $\dot{Q}$ be a $(3k)$-ary spatial (point) database query over $\dot{\sigma}$. We say that $\hat{Q}$ *and* $\dot{Q}$ *are equivalent*, denoted $\hat{Q} \equiv_\triangle \dot{Q}$ if for every database $\mathcal{D}$ over $\hat{\sigma}$ we have

$$can_{tr}(\hat{Q}(\mathcal{D})) = \dot{Q}(can_{tr}(\mathcal{D})).$$

• Let $\hat{\sigma}^{st}$ be a triangle database schema and let us consider input spatio-temporal triangle databases over $\hat{\sigma}^{st}$. Let $\dot{\sigma}^{st}$ be the corresponding spatio-temporal point database schema (see Remark 4.1). Let $\hat{Q}$ be a $k$-ary spatio-temporal triangle database query over $\hat{\sigma}^{st}$ and let $\dot{Q}$ be a $(3k)$-ary spatio-temporal (point) database query over $\dot{\sigma}^{st}$. We say that $\hat{Q}$ *and* $\dot{Q}$ *are equivalent*, denoted $\hat{Q} \equiv_\triangle \dot{Q}$, if for every database $\mathcal{D}^{st}$ over $\hat{\sigma}^{st}$ we have

$$can_{trST}(\hat{Q}(\mathcal{D}^{st})) = \dot{Q}(can_{trST}(\mathcal{D}^{st})).$$

□

Since we have defined equivalence between triangle database queries and point database queries earlier, we can now discuss how the point languages FO($\{$**Between**$\{$) and FO($\{\mathbf{Between}^{\text{Cotemp}}, \mathbf{Before}, \mathbf{EqCr}^{ST}\}$) can be used to query triangle databases. We have to keep in mind that only spatial and spatio-temporal (point) databases can be considered that are the image under the bijections $can_{tr}$ and $can_{trST}$ of spatial and spatio-temporal triangle databases.

**Definition 4.5 (**FO($\{$**Between**$\}$) **as a triangle query language)** • Let $\hat{\sigma} = \{\hat{R}_1, \hat{R}_2, \ldots, \hat{R}_m\}$ be a spatial triangle database schema. Let $\dot{R}_i$ be the corresponding spatial point relation names of arity $3 \times ar(\hat{R}_i)$, for $i = 1 \ldots m$, and let $\dot{\sigma}$ be the spatial database schema $\{\dot{R}_1, \dot{R}_2, \ldots, \dot{R}_m\}$.

Let $\varphi(\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \mathbf{x}_{1,3}, \mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{x}_{2,3}, \ldots, \mathbf{x}_{k,1}, \mathbf{x}_{k,2}, \mathbf{x}_{k,3})$ be a FO($\{$**Between**$\}$)-formula expressing a spatial $(3k)$-ary query $\dot{Q}$ which is equivalent to a $k$-ary spatial triangle query $\hat{Q}$. For each input spatial triangle database $\mathcal{D}$ over $\hat{\sigma}$, $\hat{Q}(\mathcal{D})$ is defined as the set of points $(\mathbf{a}_{1,1}, \mathbf{a}_{1,2}, \mathbf{a}_{1,3}, \mathbf{a}_{2,1}, \mathbf{a}_{2,2}, \mathbf{a}_{2,3}, \ldots, \mathbf{a}_{k,1}, \mathbf{a}_{k,2}, \mathbf{a}_{k,3})$ in $(\mathbb{R}^6)^k$ such that

$$(\mathbb{R}^2, =, \mathbf{Between}, \dot{R}_1^{\mathcal{S}}, \dot{R}_2^{\mathcal{S}}, \ldots, \dot{R}_m^{\mathcal{S}}) \models \varphi[\mathbf{a}_{1,1}, \mathbf{a}_{1,2}, \mathbf{a}_{1,3}, \mathbf{a}_{2,1}, \mathbf{a}_{2,2}, \mathbf{a}_{2,3}, \ldots, \mathbf{a}_{k,1}, \mathbf{a}_{k,2}, \mathbf{a}_{k,3}].$$

Here, $\mathcal{S}$ is the image of $\mathcal{D}$ under the canonical bijection $can_{tr}$.

• Let $\hat{\sigma}^{st} = \{\hat{R}_1^{st}, \hat{R}_2^{st}, \ldots, \hat{R}_m^{st}\}$ be a spatio-temporal triangle database schema. Let $\dot{R}_i^{st}(1 \leq i \leq m)$ be the corresponding spatio-temporal point relation names of arity $3 \times ar(\hat{R}_i^{st})$ and let $\dot{\sigma}^{st}$ be the spatio-temporal database schema $\{\dot{R}_1^{st}, \dot{R}_2^{st}, \ldots, \dot{R}_m^{st}\}$.

Let $\varphi(u_{1,1}, u_{1,2}, u_{1,3}, u_{2,1}, u_{2,2}, u_{2,3}, \ldots, u_{k,1}, u_{k,2}, u_{k,3})$ be a FO($\{$**Between**$\}$)-formula, expressing a spatio-temporal $(3k)$-ary query $\dot{Q}$ which is equivalent to a $k$-ary spatial triangle query $\hat{Q}$. For each input spatio-temporal triangle database $\mathcal{D}^{st}$ over $\hat{\sigma}^{st}$, $\hat{Q}(\mathcal{D}^{st})$ is defined as the set of points $(p_{1,1}, p_{1,2}, p_{1,3}, p_{2,1}, p_{2,2}, p_{2,3}, \ldots, p_{k,1}, p_{k,2}, p_{k,3})$ of $(\mathbb{R}^9)^k$ such that

$$((\mathbb{R}^2 \times \mathbb{R}), =, \mathbf{Between}^{\text{Cotemp}}, \mathbf{Before}, \mathbf{EqCr}^{ST}, \dot{R}_1^{st\mathcal{D}^{st}}, \dot{R}_2^{st\mathcal{D}^{st}}, \ldots, \dot{R}_m^{st\mathcal{D}^{st}}) \models \varphi[p_{1,1}, p_{1,2}, p_{1,3}, p_{2,1}, p_{2,2}, p_{2,3}, \ldots, p_{k,1}, p_{k,2}, p_{k,3}].$$

Here, $\mathscr{T}$ is the image of $\mathcal{D}^{st}$ under the canonical bijection $can_{trST}$. □

The languages FO($\{$**Between**$\}$, $\dot{\sigma}$) and FO($\{$**Between**, **Before**, $qcrsts\}$, $\dot{\sigma}^{st}$) were designed to formulate queries on spatial and spatio-temporal point databases over some input schema $\dot{\sigma}$, resp. $\dot{\sigma}^{st}$. Using those languages to query triangle databases, involves expressing relations between the point sets that compose the triangles. This is a rather indirect way of expressing triangle relations. In the spirit of [10], we now construct affine-generic query languages based on triangle variables. As they directly express relations between the triangles, this results in a more intuitive way of querying spatial and spatio-temporal triangle databases. We define triangle-based logics next. Afterwards, we propose a specific spatial triangle logic in Section 5, and a spatio-temporal triangle logic in Section 6.

**Definition 4.6 (Triangle logics)** • Let $\hat{\sigma} = \{\hat{R}_1, \hat{R}_2, \ldots, \hat{R}_m\}$ be a triangle database schema and let $\mathbf{\Delta}$ be a set of predicates of a certain arity over triangles in $\mathbb{R}^2$. The first-order logic over $\hat{\sigma}$ and $\mathbf{\Delta}$, denoted by FO($\mathbf{\Delta}$, $\hat{\sigma}$), can be used as a spatial triangle query language when

variables are interpreted to range over triangles in $\mathbb{R}^2$. The atomic formulas in FO($\mathbf{\Delta}$, $\hat{\sigma}$) are equality constraints on triangle variables, the predicates of $\mathbf{\Delta}$, and the relation names $\hat{R}_1, \hat{R}_2, \ldots, \hat{R}_m$ from $\hat{\sigma}$, applied to triangle variables.
• Let $\hat{\sigma}^{st} = \{\hat{R}_1^{st}, \hat{R}_2^{st}, \ldots, \hat{R}_m^{st}\}$ be a database schema and let $\mathbf{\Delta}$ be a set of predicates of a certain arity over spatio-temporal triangles in $(\mathbb{R}^2 \times \mathbb{R})$. The first-order logic over $\hat{\sigma}^{st}$ and $\mathbf{\Delta}$, denoted by FO($\mathbf{\Delta}$, $\hat{\sigma}^{st}$), can be used as a spatio-temporal triangle query language when variables are interpreted to range over spatio-temporal triangles in $(\mathbb{R}^2 \times \mathbb{R})$. The atomic formulas in FO($\mathbf{\Delta}$, $\hat{\sigma}^{st}$) are equality constraints on spatio-temporal triangle variables, the predicates of $\mathbf{\Delta}$, and the relation names $\hat{R}_1^{st}, \hat{R}_2^{st}, \ldots, \hat{R}_m^{st}$ from $\hat{\sigma}^{st}$, applied to spatio-temporal triangle variables. □

A FO($\mathbf{\Delta}$, $\hat{\sigma}$)-formula $\varphi(\triangle_1, \triangle_2, \ldots, \triangle_k)$ (resp., FO($\mathbf{\Delta}$, $\hat{\sigma}^{st}$)-formula $\varphi(\triangle_1^{st}, \triangle_2^{st}, \ldots, \triangle_k^{st})$) defines for each spatial (resp., spatio-temporal) database $\mathcal{D}$ (resp., $\mathcal{D}^{st}$) over $\hat{\sigma}$ (resp. $\hat{\sigma}^{st}$) a subset $\varphi(\mathcal{D})$ (resp., $\varphi(\mathcal{D}^{st})$) of $\left((\mathbb{R}^2)^3\right)^k$ (resp., $\left((\mathbb{R}^2 \times \mathbb{R})^3\right)^k$) defined as

$$\{(T_1, T_2, \ldots, T_k) \in (\mathbb{R}^2)^{3k} \mid (\mathbb{R}^2, \mathbf{\Delta}^{\mathbb{R}^2}, \hat{R}_1^{\mathcal{D}}, \hat{R}_2^{\mathcal{D}}, \ldots, \hat{R}_m^{\mathcal{D}}) \models \varphi[T_1, T_2, \ldots, T_k]\ \},$$

respectively,

$$\{(T_1^{st}, T_2^{st}, \ldots, T_k^{st}) \in (\mathbb{R}^2 \times \mathbb{R})^{3k} \mid ((\mathbb{R}^2 \times \mathbb{R}), \mathbf{\Delta}^{(\mathbb{R}^2\times\mathbb{R})}, \hat{R}_1^{st\mathcal{D}^{st}}, \hat{R}_2^{st\mathcal{D}^{st}}, \ldots, \hat{R}_m^{st\mathcal{D}^{st}}) \models \varphi[T_1^{st}, T_2^{st}, \ldots, T_k^{st}]\ \}.$$

**Remark 4.6** We use the symbol $=_\triangle$ to indicate equality of triangle variables, as opposed to equality of point variables. If it is clear from the context of a formula which type of variables is used, we will omit the index. □

In Section 5 (resp., Section 6), we will develop languages that have the same expressive power as FO({**Between**}) and FO({$\mathbf{Between}^{\mathrm{Cotemp}}, \mathbf{Before}, \mathbf{EqCr}^{ST}$}) on spatial triangle databases and on spatio-temporal triangle databases, respectively. We will prove this by showing both soundness and completeness of those triangle languages with respect to FO({**Between**}) and FO({$\mathbf{Between}^{\mathrm{Cotemp}}, \mathbf{Before}, \mathbf{EqCr}^{ST}$}).

The concepts of soundness and completeness are introduced as follows:

**Definition 4.7 (Soundness and completeness)** • A query language $\mathcal{L}$ is said to be *sound* for the $\mathcal{G}$-generic FO(+, ×, <, 0, 1, $\sigma$)-queries on spatial (resp., spatio-temporal) databases, if formulas in $\mathcal{L}$ only express $\mathcal{G}_{st}$-generic FO(+, ×, <, 0, 1, $\sigma$)-queries on spatial (resp., spatio-temporal) databases.

• A query language $\mathcal{L}$ is said to be *complete* for the $(\mathcal{F}_{st}, \mathcal{F}_t)$-generic FO(+, ×, <, 0, 1, $\sigma$)-queries on spatio-temporal databases, if all $(\mathcal{F}_{st}, \mathcal{F}_t)$-generic FO(+, ×, <, 0, 1, $\sigma$)-queries on spatio-temporal databases can be expressed in $\mathcal{L}$. □

# 5 Affine-invariant Spatial Triangle Queries

In this section, we propose a spatial triangle logic that captures exactly the class of first-order affine-generic queries on spatial triangle databases. First, we remark the following:

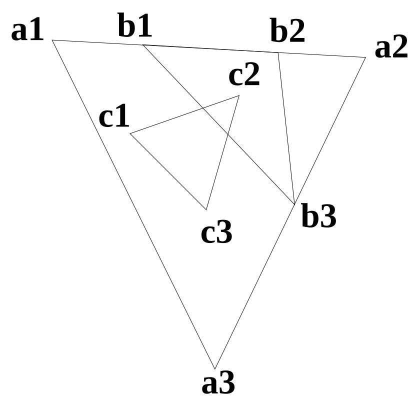


Figure 3: An illustration of the predicate **PartOf**. Let $T_1 = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3)$, $T_2 = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$ and $T_3 = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ The expressions $\mathbf{PartOf}(T_2, T_1)$ and $\mathbf{PartOf}(T_3, T_1)$ are true, the expression $\mathbf{PartOf}(T_3, T_2)$ is not true.

**Remark 5.1** We defined a triangle database as a special type of geometric database. Accordingly, we take the affine image of a triangle for affinities of $\mathbb{R}^2$, and not of $\mathbb{R}^6$. This corresponds to our intuition. One triangle is an affine image of another triangle, if the drawing of the first one is the affine image of the drawing of the second one. Hence, the affine image of a triangle with corner points $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ under some affinity $\alpha$ of the plane, is the triangle with corner points $\alpha(\mathbf{x}_1)$, $\alpha(\mathbf{x}_2)$ and $\alpha(\mathbf{x}_3)$. □

We introduce one binary triangle predicate, *i.e.*, **PartOf**. Intuitively, when applied to two triangles, this predicate expresses that the drawing of the first triangle is a subset ($\subseteq$) of the drawing of the second triangle. We only consider $(\mathbb{R}^2)^3$ as the underlying domain. We show that the triangle predicate **PartOf** allows a natural extension to higher dimensions and other types of objects (instead of triangles).

We define the predicate **PartOf** and equality on triangles more precisely:

**Definition 5.1 (The triangle predicate PartOf)** Let $T_1 = (\mathbf{a}_{1,1}, \mathbf{a}_{1,2}, \mathbf{a}_{1,3})$ and $T_2 = (\mathbf{a}_{2,1}, \mathbf{a}_{2,2}, \mathbf{a}_{2,3})$ be two triangles. The binary predicate **PartOf**, applied to $T_1$ and $T_2$ expresses that the convex closure of the three points $\mathbf{a}_{1,1}$, $\mathbf{a}_{1,2}$ and $\mathbf{a}_{1,3}$ is a subset of the convex closure of the three points $\mathbf{a}_{2,1}$, $\mathbf{a}_{2,2}$ and $\mathbf{a}_{2,3}$. □

Figure 3 illustrates the predicate **PartOf**.

We also define triangle-equality, which differs from the standard equality operation.

**Definition 5.2 (Equality of triangles)** Let $T_1$ and $T_2$ be two triangles. The expression $T_1 =_\triangle T_2$ is true if and only if both $\mathbf{PartOf}(T_1, T_2)$ and $\mathbf{PartOf}(T_2, T_1)$ are tue. □

Before analyzing the expressiveness of the language FO($\{\mathbf{PartOf}\}$), we prove that the FO($\{\mathbf{PartOf}\}$)-queries are well-defined on consistent triangle databases. More concretely, given a triangle database schema $\hat{\sigma}$, we prove that the result of a $k$-ary FO($\Delta$, $\hat{\sigma}$) query on a consistent input database over $\hat{\sigma}$ is a consistent triangle relation of arity $k$.

**Lemma 5.1 (FO($\{$PartOf$\}$) is well-defined)** Let $\hat{\sigma} = \{\hat{R}_1, \hat{R}_2, \ldots, \hat{R}_m\}$ be a spatial triangle database schema. Let $\mathcal{D}$ be a consistent spatial triangle database over $\hat{\sigma}$. For each FO($\Delta$, $\hat{\sigma}$)-query $\hat{Q}$, $\hat{Q}(\mathcal{D})$ is a consistent triangle relation.

**Proof.** Let $\hat{\sigma} = \{\hat{R}_1, \hat{R}_2, \ldots, \hat{R}_m\}$ be a spatial triangle database schema. Let $\mathcal{D}$ be a consistent spatial triangle database over $\hat{\sigma}$.

We prove this lemma by induction on the structure of FO($\Delta$, $\hat{\sigma}$)-queries. The atomic formulas of FO($\{$**PartOf**$\}$) are equality expressions on triangle variables, expressions of the form **PartOf**$(\triangle_1, \triangle_2)$, and expressions of the form $\hat{R}_i(\triangle_1, \triangle_2, \ldots, \triangle_{ar(\hat{R}_i)})$, where $\hat{R}_i (1 \leq i \leq m)$ is a relation name from $\hat{\sigma}$. More complex formulas can be constructed using the Boolean operators $\wedge$, $\vee$ and $\neg$ and existential quantification.

For the atomic formulas, it is easy to see that, if two triangles $T_1$ and $T_2$ satisfy the conditions $T_1 =_{\triangle} T_2$ or **PartOf**$(T_1, T_2)$, that also $T_1' =_{\triangle} T_2'$ respectively **PartOf**$(T_1', T_2')$ are true if and only if $T_1 =_{\triangle} T_1'$ and $T_2 =_{\triangle} T_2'$ are true. As we assume the input database $\mathcal{D}$ to be consistent, the atomic formulas of the type $\hat{R}_i(\triangle_1, \triangle_2, \ldots, \triangle_{ar(\hat{R}_i)})$, where $(1 \leq i \leq m)$, trivially return consistent triangle relations.

Now we have to prove that the composed formulas always return consistent triangle relations. Let $\hat{\varphi}$ and $\hat{\psi}$ be two formulas in FO($\Delta$, $\hat{\sigma}$), of arity $k_\varphi$ and $k_\psi$ respectively, already defining consistent triangle relations. Then, the formula $(\hat{\varphi} \wedge \hat{\psi})$ (resp., $(\hat{\varphi} \vee \hat{\psi})$) also defines a triangle relation. This follows from the fact that the free variables of $(\hat{\varphi} \wedge \hat{\psi})$ (resp., $(\hat{\varphi} \vee \hat{\psi})$) are free variables in $\hat{\varphi}$ or $\hat{\psi}$. The universe of all triangles is trivially consistent. If a consistent subset is removed from this universe, the remaining part is still consistent. Therefor, $\neg\hat{\varphi}$ is well-defined. Finally, because consistency is defined argument-wise, the projection $\exists T_1\, \hat{\varphi}(T_1, T_2, \ldots, T_{k_\varphi})$ is consistent. □

After proving that the language FO($\{$**PartOf**$\}$) is well-defined, we can analyze its expressiveness.

## 5.1 Expressiveness of FO($\{$PartOf$\}$)

We now determine the expressiveness of the language FO($\{$**PartOf**$\}$). We prove that it is sound and complete for the affine-invariant fragment of first-order logic over the reals, on triangle databases. We prove this by comparing the languages FO($\{$**PartOf**$\}$) and FO($\{$**Between**$\}$).

From [11], we already know that FO($\{$**Between**$\}$) is sound and complete for the affine-invariant fragment of first-order logic over the reals, on spatial point databases.

The soundness and completeness of the query language FO($\{$**PartOf**$\}$) with respect to the language FO($\{$**Between**$\}$) is proved using two separate lemmas (Lemma 5.2 and Lemma 5.3). In both lemmas, formulas are translated from one language in the other, by using induction on the structure of FO($\{$**PartOf**$\}$) and FO($\{$**Between**$\}$)-formulas, respectively. This proof technique will be used several times in this text. Therefor, we explain the first such proofs in detail. Later on, we will only develop the crucial points in similar proofs.

**Lemma 5.2 (Soundness of FO($\{$PartOf$\}$)with respect to** FO($\{$**Between**$\}$)**)** Let $\hat{\sigma} = \{\hat{R}_1, \hat{R}_2, \ldots, \hat{R}_m\}$ be a spatial triangle database schema. Let $\dot{R}_i$ be the corresponding spatial point relation names of arity $3 \times ar(\hat{R}_i)$, for $(1 \leq i \leq m)$, and let $\dot{\sigma}$ be the spatial database schema $\{\dot{R}_1, \dot{R}_2, \ldots, \dot{R}_m\}$. Every FO($\Delta$, $\hat{\sigma}$)-expressible query can be expressed equivalently in FO($\{$**Between**$\}$, $\dot{\sigma}$).

**Proof.** Let $\hat{\sigma} = \{\hat{R}_1, \hat{R}_2, \ldots, \hat{R}_m\}$ be a spatial triangle database schema. Let $\dot{R}_i$ be the corresponding spatial point relation names of arity $3 \times ar(\hat{R}_i)$, for $(1 \leq i \leq m)$, and let $\dot{\sigma}$ be the corresponding spatial database schema $\{\dot{R}_1, \dot{R}_2, \ldots, \dot{R}_m\}$. We translate each formula of FO($\Delta$, $\hat{\sigma}$) into an equivalent formula in FO($\{$**Between**$\}$, $\dot{\sigma}$). We do this by induction on the structure of FO($\Delta$, $\hat{\sigma}$)-formulas.

First, we translate the variables of $\hat{\varphi}$. Each triangle variable $\triangle$ is naturally translated into three spatial point variables $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$. We allow one or more of the corner points of a triangle to coincide, so there are no further restrictions on the variables $\mathbf{x}_j, 1 \leq j \leq 3$.

The atomic formulas of FO($\Delta$, $\hat{\sigma}$) are equality expressions on triangle variables, expressions of the form **PartOf**$(\triangle_1, \triangle_2)$, and expressions of the form $\hat{R}_i(\triangle_1, \triangle_2, \ldots, \triangle_k)$, where $k = ar(\hat{R}_i)$ and $1 \leq i \leq m$. More complex formulas can be constructed using the Boolean operators $\wedge$, $\vee$ and $\neg$ and existential quantification.

*The translation of atomic formulas.*

We first show that all atomic formulas of FO($\Delta$, $\hat{\sigma}$) can be expressed in the language FO($\{$**Between**$\}$, $\dot{\sigma}$) .

(i) The translation of $(\triangle_1{=}\triangle_2)$, where $T_1$ is translated into $\mathbf{x}_{1,1}$, $\mathbf{x}_{1,2}$ and $\mathbf{x}_{1,3}$ and $T_2$ is translated into $\mathbf{x}_{2,1}$, $\mathbf{x}_{2,2}$ and $\mathbf{x}_{2,3}$, equalsLet $T_1^{st} = (p_{1,1}, p_{1,2}, p_{1,3})$ and $T_2^{st} = (p_{2,1}, p_{2,2}, p_{2,3})$ be two triangle snapshots. The binary predicate **PartOf**, applied to $T_1^{st}$ and $T_2^{st}$ expresses that $p_{1,1}$, $p_{1,2}$ and $p_{1,3}$ (resp., $p_{2,1}$, $p_{2,2}$ and $p_{2,3}$) are co-temporal and that the convex closure of the three points $p_{1,1}$, $p_{1,2}$ and $p_{1,3}$ is a subset of the convex closure of the three points $p_{2,1}$, $p_{2,2}$ and $p_{2,3}$.

$$\bigvee_{\sigma(1,2,3)=(j_1,j_2,j_3),\sigma\in\mathcal{S}_3} (\mathbf{x}_{1,1} = \mathbf{x}_{2,j_1} \wedge \mathbf{x}_{1,2} = \mathbf{x}_{2,j_2} \wedge \mathbf{x}_{1,3} = \mathbf{x}_{2,j_3}),$$

where $\mathcal{S}_3$ is the set of all permutations of $\{1, 2, 3\}$.

The correctness of this translation follows trivially from the definition of triangle equality (see Definition 5.2).

(ii) The translation of **PartOf**$(\triangle_1, \triangle_2)$, where $T_1$ is translated into $\mathbf{x}_{1,1}$, $\mathbf{x}_{1,2}$ and $\mathbf{x}_{1,3}$ and $T_2$ is translated into $\mathbf{x}_{2,1}$, $\mathbf{x}_{2,2}$ and $\mathbf{x}_{2,3}$, is

$$\bigwedge_{i=1}^{3} \mathbf{InTriangle}(\mathbf{x}_{1,i}, \mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{x}_{2,3}),$$

where the definition of **InTriangle** is:

$$\mathbf{InTriangle}(\mathbf{x}, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) := \exists \mathbf{x}_4(\mathbf{Between}(\mathbf{x}_1, \mathbf{x}_4, \mathbf{x}_2) \wedge \mathbf{Between}(\mathbf{x}_4, \mathbf{x}, \mathbf{x}_3)).$$

Figure 4 illustrates the corresponding geometric construction.

The correctness of this translation follows from the definition of the predicate **PartOf** (see Definition 5.1).

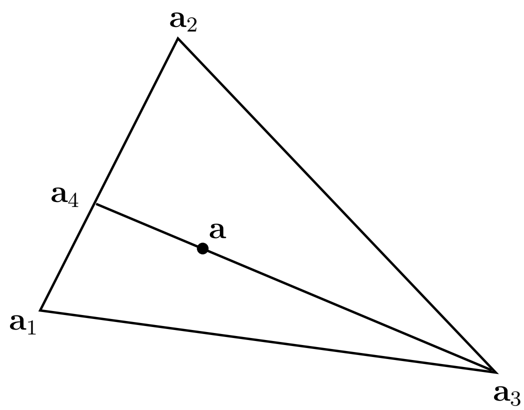


Figure 4: An illustration of the predicate **InTriangle**. The expression $\mathbf{InTriangle}(\mathbf{a}, \mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3)$ is true because there exists a point $\mathbf{a}_4$ between $\mathbf{a}_1$ and $\mathbf{a}_2$ such that $\mathbf{a}$ lies between $\mathbf{a}_4$ and $\mathbf{a}_3$.

(iii) The translation of $\hat{R}_i(\triangle_1, \triangle_2, \ldots, \triangle_k)$, where $T_i$ is translated into $\mathbf{x}_{i,1}$, $\mathbf{x}_{i,2}$ and $\mathbf{x}_{i,3}$ for $1 \leq i \leq k$, is

$$\dot{R}_i(\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \mathbf{x}_{1,3}, \mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{x}_{2,3}, \ldots, \mathbf{x}_{k,1}, \mathbf{x}_{k,2}, \mathbf{x}_{k,3}).$$

The correctness of this translation follows from Definition 4.2 and Remark 4.1.

*The translation of composed formulas.*

Assume that we already correctly translated the FO($\Delta$, $\hat{\sigma}$)-formulas $\hat{\varphi}$ and $\hat{\psi}$ into the FO({**Between**}, $\dot{\sigma}$)-formulas $\dot{\varphi}$ and $\dot{\psi}$. Suppose that the number of free variables in $\hat{\varphi}$ is $k_\varphi$ and that of $\hat{\psi}$ is $k_\psi$. Therefor, we can assume that, for each triangle database $\mathcal{D}$ over the input schema $\hat{\sigma}$, and for each $k_\varphi$-tuple of triangles $(T_1, T_2, \ldots, T_{k_\varphi})$ given as $((\mathbf{a}_{1,1}, \mathbf{a}_{1,2}, \mathbf{a}_{1,3}), (\mathbf{a}_{2,1}, \mathbf{a}_{2,2}, \mathbf{a}_{2,3}), \ldots, (\mathbf{a}_{k_\varphi,1}, \mathbf{a}_{k_\varphi,2}, \mathbf{a}_{k_\varphi,3}))$ that

$$\mathcal{D} \models \hat{\varphi}(T_1, T_2, \ldots, T_{k_\varphi}) \text{ if and only if}$$
$$\mathcal{S} \models \dot{\varphi}(\mathbf{a}_{1,1}, \mathbf{a}_{1,2}, \mathbf{a}_{1,3}, \mathbf{a}_{2,1}, \mathbf{a}_{2,2}, \mathbf{a}_{2,3}, \ldots, \mathbf{a}_{k_\varphi,1}, \mathbf{a}_{k_\varphi,2}, \mathbf{a}_{k_\varphi,3})$$

is true when $\mathcal{S}$ is the spatial (point) database over the input schema $\dot{\sigma}$, obtained from $\mathcal{D}$ by applying the canonical bijection $can_{tr}$ between $\left((\mathbb{R}^2)^3\right)^{k_\varphi}$ and $(\mathbb{R}^2)^{3k_\varphi}$, on $\mathcal{D}$. For the formula $\hat{\psi}$ the analog holds.

In the following, we omit the $k_\varphi$-tuples (resp., $k_\psi$-tuples) of triangles and $3k_\varphi$-tuples (resp., $3k_\psi$-tuples) of points the formulas are applied on, to make the proofs more readable.

(i) The translation of $\hat{\varphi} \wedge \hat{\psi}$ is $\dot{\varphi} \wedge \dot{\psi}$. Indeed,

$\mathcal{S} \models (\dot{\varphi} \wedge \dot{\psi})$
iff. $\mathcal{S} \models \dot{\varphi}$ and $\mathcal{S} \models \dot{\psi}$
iff. $\mathcal{D} \models \hat{\varphi}$ and $\mathcal{D} \models \hat{\psi}$
iff. $\mathcal{D} \models (\hat{\varphi} \wedge \hat{\psi})$.

(ii) The translation of $\hat{\varphi} \vee \hat{\psi}$ is $\dot{\varphi} \vee \dot{\psi}$. Indeed,

$\mathcal{S} \models (\dot{\varphi} \vee \dot{\psi})$
iff. $\mathcal{S} \models \dot{\varphi}$ or $\mathcal{S} \models \dot{\psi}$
iff. $\mathcal{D} \models \hat{\varphi}$ or $\mathcal{D} \models \hat{\psi}$
iff. $\mathcal{D} \models (\hat{\varphi} \vee \hat{\psi})$.

(iii) The translation of $\neg\hat{\varphi}$ is $\neg\dot{\varphi}$. Indeed,

$\mathcal{S} \models \neg\dot{\varphi}$
iff. it is not true that $\mathcal{S} \models \dot{\varphi}$
iff. it is not true that $\mathcal{D} \models \hat{\varphi}$
iff. $\mathcal{D} \models \neg\hat{\varphi}$.

(iv) Assume that $\hat{\varphi}$ has free variables $\triangle, \triangle_1, \ldots, \triangle_k$ and $\triangle$ is translated into $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ and $\triangle_i$ is translated into $\mathbf{x}_{i,1}$, $\mathbf{x}_{i,2}$, $\mathbf{x}_{i,3}$. The translation of

$$\exists \triangle \ \hat{\varphi}(\triangle, \triangle_1, \triangle_2, \ldots, \triangle_k)$$

is $\exists \mathbf{x}_1 \, \exists \mathbf{x}_2 \, \exists \mathbf{x}_3 \, \dot{\varphi}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \mathbf{x}_{1,3}, \ldots, \mathbf{x}_{k,1}, \mathbf{x}_{k,2}, \mathbf{x}_{k,3})$.

Indeed,

$\mathcal{S} \models$
$\exists \mathbf{x}_1 \, \exists \mathbf{x}_2 \, \exists \mathbf{x}_3 \, \dot{\varphi}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)[\mathbf{a}_{1,1}, \mathbf{a}_{1,2}, \mathbf{a}_{1,3}, \ldots, \mathbf{a}_{k,1}, \mathbf{a}_{k,2}, \mathbf{a}_{k,3}]$
iff. there exist points $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$) in $\mathbb{R}^2$ such that
$\mathcal{S} \models \dot{\varphi}[\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_{1,1}, \mathbf{a}_{1,2}, \mathbf{a}_{1,3}, \ldots, \mathbf{a}_{k,1}, \mathbf{a}_{k,2}, \mathbf{a}_{k,3}]$
iff. there exists a triangle $T$ such that $\mathcal{D} \models \hat{\varphi}[T, T_1, \ldots, T_k]$, where
$T_i$ is the triangle with corner points $\mathbf{a}_{i,1}$, $\mathbf{a}_{i,2}$ and $\mathbf{a}_{i,3}$ for $1 \leq i \leq k$
iff. $\mathcal{D} \models \exists T \, \hat{\varphi}(T)[T_1, \ldots, T_k]$.

To summarize, let $\hat{\sigma} = \{\hat{R}_1, \hat{R}_2, \ldots, \hat{R}_m\}$ be a spatial triangle database schema. Let $\dot{\sigma} = \{\dot{R}_1, \dot{R}_2, \ldots, \dot{R}_m\}$ be the corresponding spatial point database schema. Each formula $\hat{\varphi}$ in FO($\Delta$, $\hat{\sigma}$), with free variables $\triangle_1, \triangle_2, \ldots, \triangle_k$ can be translated into a FO({**Between**}, $\dot{\sigma}$)-formula $\dot{\varphi}$ with free variables $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \mathbf{x}_{1,3}, \mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{x}_{2,3}, \ldots, \mathbf{x}_{k,1}, \mathbf{x}_{k,2}, \mathbf{x}_{k,3}$. This translation is such that, for all triangle databases $\mathcal{D}$ over $\hat{\sigma}$, $\mathcal{D} \models \hat{\varphi}$ iff. $\mathcal{S} \models \dot{\varphi}$. Here, $\mathcal{S}$ is the spatial point database over $\dot{\sigma}$ which is the image of $\mathcal{D}$ under the canonical bijection between $((\mathbb{R}^2)^3)^k$ and $(\mathbb{R}^2)^{3k}$. This completes the soundness proof. □

For completeness, we translate FO({**Between**})-formulas into FO({**PartOf**})-formulas. We again prove this by induction, on the structure of FO({**Between**})-formulas. This translation is not as straightforward as the translation in the other direction, however.

**Lemma 5.3 (Completeness of FO({PartOf}))** Let $\hat{\sigma} = \{\hat{R}_1, \hat{R}_2, \ldots, \hat{R}_m\}$ be a spatial triangle database schema and $\dot{\sigma}$ be the corresponding spatial database schema. Every FO({**Between**}, $\dot{\sigma}$)-expressible query can be expressed equivalently in FO($\Delta$, $\hat{\sigma}$).

**Proof.** Let $\hat{\sigma} = \{\hat{R}_1, \hat{R}_2, \ldots, \hat{R}_m\}$ be a spatial triangle database schema and $\dot{\sigma}$ be the corresponding spatial database schema. We have to prove that we can translate every triangle

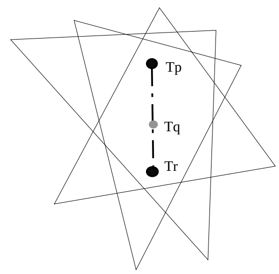


Figure 5: Illustration of the translation of the predicate **Between**. The (degenerated) triangle $T_q$ lies between the (degenerated) triangles $T_p$ and $T_r$ if and only if all triangles that contain both $T_p$ and $T_r$, also contain $T_q$.

database query, expressed in the language FO({**Between**}, $\dot{\sigma}$), into a triangle database query in the language FO($\mathbf{\Delta}$, $\hat{\sigma}$) over trangle databases.

We first show how we can simulate point variables by a degenerated triangle, and any FO({**Between**}, $\dot{\sigma}$)-formula $\dot{\varphi}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k)$ by a formula $\varphi(\triangle_1, \triangle_2, \ldots, \triangle_k)$, where $\triangle_1, \triangle_2$ $, \ldots, \triangle_k$ represent triangles that are degenerated into points. We prove this by induction on the structure of FO({**Between**}, $\dot{\sigma}$)-formulas. Initially, each FO({**Between**}, $\dot{\sigma}$)-formula $\dot{\varphi}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k)$ will be translated into a FO($\mathbf{\Delta}$, $\hat{\sigma}$)-formula $\hat{\varphi}(\triangle_1, \triangle_2, \ldots, \triangle_k)$ with the same number of free variables.

The translation of a point variable $\mathbf{x}$ is the triangle variable $\triangle$, and we add the condition **Point**($\triangle$) as a conjunct to the beginning of the translation of the formula. The definition of **Point**($\triangle$) is

$$\forall \triangle' \ (\mathbf{PartOf}(\triangle', \triangle) \rightarrow (\triangle = \triangle')).$$

In the following, we always assume that such formulas **Point**($\triangle$) are already added to the translation as a conjunct.

*The translation of atomic formulas.*

The atomic formulas of the language FO({**Between**}, $\dot{\sigma}$) are equality constraints on point variables, formulas of the form $\mathbf{Between}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$, and formulas of the type $\dot{R}_i(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k)$, where $k = 3 \times ar(\hat{R}_i)$. We show that all of those can be simulated into an equivalent FO($\mathbf{\Delta}$, $\hat{\sigma}$) formula.

(i) The translation of $(\mathbf{x}_1 = \mathbf{x}_2)$ is $(\triangle_1 =_{\triangle} \triangle_2)$.

(ii) The translation of $\mathbf{Between}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$, where $\triangle_1$, $\triangle_2$ and $\triangle_3$ (which as assumed are already declared points) are the translations of $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$, respectively, is expressed by saying that all triangles that contain both $\triangle_1$ and $\triangle_3$ should also contain $\triangle_2$. It then follows from the convexity of triangles (or line segments, in the degenerated case) that $\triangle_2$ lies on the line segment between $\triangle_1$ and $\triangle_3$. Figure 5 illustrates this principle. We now give the formula translating $\mathbf{Between}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$:

$$\forall \triangle_4 \ ((\mathbf{PartOf}(\triangle_1, \triangle_4) \wedge \mathbf{PartOf}(\triangle_3, \triangle_4)) \rightarrow \mathbf{PartOf}(\triangle_2, \triangle_4)).$$

The correctness of this translation follows from the fact that triangles are convex objects.

(iii) Let $\dot{R}_j$ be a relation name from $\dot{\sigma} = \{\dot{R}_1, \dot{R}_2, \ldots, \dot{R}_m\}$. Let $ar(\hat{R}_j) = k$ and thus $ar(\dot{R}_j) = 3k$, for $1 \leq j \leq m$. The translation of $\dot{R}_j(\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \mathbf{x}_{1,3}, \mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{x}_{2,3}, \ldots, \mathbf{x}_{k,1}, \mathbf{x}_{k,2}, \mathbf{x}_{k,3})$ is:

$$\exists \triangle_1 \, \exists \triangle_2 \ldots \exists \triangle_k \, (\hat{R}_j(\triangle_1, \triangle_2, \ldots, \triangle_k) \wedge \bigwedge_{i=1}^{k} \mathbf{CornerP}(\triangle_{i,1}, \triangle_{i,2}, \triangle_{i,3}, \triangle_i)).$$

The definition of **CornerP** is:

$$\mathbf{CornerP}(\triangle_1, \triangle_2, \triangle_3, \triangle) := \forall \triangle_4 \; ((\mathbf{Point}(\triangle_4) \wedge \mathbf{PartOf}(\triangle_4, \triangle)) \rightarrow \mathbf{Intriangle}_{\triangle}(\triangle_4, \triangle_1, \triangle_2, \triangle_3)).$$

The predicate $\mathbf{InTriangle}_{\triangle}$ is the translation of the predicate **InTriangle** of the language FO({**Between**}) as described in the proof of Lemma 5.2, into FO({**PartOf**}). The FO({**Between**}) formula expressing **InTriangle** only uses **Between**. In the previous item of this proof, we already showed how this can be translated into FO({**PartOf**}).

Given a $(3k)$-tuple of points $(\mathbf{a}_{1,1}, \mathbf{a}_{1,2}, \mathbf{a}_{1,3}, \mathbf{a}_{2,1}, \mathbf{a}_{2,2}, \mathbf{a}_{2,3}, \ldots, \mathbf{a}_{a,1}, \mathbf{a}_{k,2}, \mathbf{a}_{k,3})$ in $\mathbb{R}^2$. There will be $(6^k)$ $k$-tuples of triangles $(T_1, T_2, \ldots, T_k)$ such that, for each of the $T_i$, $1 \leq i \leq k$, the condition $\mathbf{CornerP}(T_{i,1}, T_{i,2}, T_{i,3}, T_i)$ is true. There will, however, only be one tuple of triangles that is the image of the $(3k)$-tuple of points $(\mathbf{a}_{1,1}, \mathbf{a}_{1,2}, \mathbf{a}_{1,3}, \mathbf{a}_{2,1}, \mathbf{a}_{2,2}, \mathbf{a}_{2,3}, \ldots, \mathbf{a}_{a,1}, \mathbf{a}_{k,2}, \mathbf{a}_{k,3})$ under the inverse of the canonical bijection $can_{tr}$. Therefor, the simulation is correct.

*The translation of composed formulas.*

Now suppose that we already simulated the FO({**Between**}, $\dot{\sigma}$) formulas $\dot{\varphi}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{k_\varphi})$ and $\dot{\psi}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{k_\psi})$ into formulas $\hat{\varphi}$ and $\hat{\psi}$ in FO($\mathbf{\Delta}$, $\hat{\sigma}$) with free variables $\triangle_1$, $\triangle_2$, ..., $\triangle_{k_\varphi}$ and $\triangle'_1$, $\triangle'_2$, ..., $\triangle'_{k_\psi}$, respectively. We can hence assume that, for each triangle database $\mathcal{D}$ over $\hat{\sigma}$ and for each $k_\varphi$-tuple of triangles $(T_1, T_2, \ldots, T_{k_\varphi}) = ((\mathbf{a}_1, \mathbf{a}_1, \mathbf{a}_1), (\mathbf{a}_2, \mathbf{a}_2, \mathbf{a}_2), \ldots, (\mathbf{a}_{k_\varphi}, \mathbf{a}_{k_\varphi}, \mathbf{a}_{k_\varphi}))$, which are required to be degenerated into points, that

$$\mathcal{D} \models \hat{\varphi}[T_1, T_2, \ldots, T_{k_\varphi}] \text{ iff. } \mathcal{S} \models \dot{\varphi}[\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{k_\psi}].$$

For $\hat{\psi}$ we have analogue conditions.

The composed formulas $\dot{\varphi} \wedge \dot{\psi}$, $\dot{\varphi} \vee \dot{\psi}$, $\neg\dot{\varphi}$ and $\exists \mathbf{x}\, \dot{\varphi}$, are translated into $\hat{\varphi} \wedge \hat{\psi}$, $\hat{\varphi} \vee \hat{\psi}$, $\neg\hat{\varphi}$ and $\exists \triangle \; (\hat{\varphi})$, respectively if we assume that $\mathbf{x}$ is translated into $\triangle$. The correctness proofs for these translations are similar to the proofs in Lemma 5.2. Therefor, we do not repeat them here. This concludes the proof of Lemma 5.3. □

**Remark 5.2** So far, we showed that we can simulate any FO({**Between**}, $\dot{\sigma}$) formula $\dot{\varphi}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k)$ by a formula $\varphi'(\triangle_1, \triangle_2, \ldots, \triangle_k)$, where $\mathbf{Point}(\triangle_i)$ is true for all $\triangle_i$ ($1 \leq i \leq k$). If $\dot{\varphi}$ expresses a $k$-ary triangle database query $Q$ however (*i.e.*, $\dot{\varphi}$ has $(3k)$ free variables), we can do better.

Let $\dot{\varphi}$ be the FO($\{\mathbf{Between}\}$, $\dot{\sigma}$)-formula expressing a $k$-ary triangle database query $\hat{Q}$. The free variables of $\dot{\varphi}$ are $\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \mathbf{x}_{1,3}, \mathbf{x}_{2,1}, \mathbf{x}_{2,2}, \mathbf{x}_{2,3}, \ldots, \mathbf{x}_{k,1}, \mathbf{x}_{k,2}, \mathbf{x}_{k,3}$.

We now construct the FO($\Delta$, $\hat{\sigma}$) formula $\hat{\varphi}$ expressing the query $\hat{Q}$ as follows:

$$\begin{aligned}
&\hat{\varphi}(\triangle_1, \triangle_2, \ldots, \triangle_k) \equiv \\
&\qquad \exists\, \triangle_{1,1}\ \exists\, \triangle_{1,2}\ \exists\, \triangle_{1,3}\ \exists\, \triangle_{2,1}\ \exists\, \triangle_{2,2}\ \exists\, \triangle_{2,3} \ldots \exists\, \triangle_{k,1}\ \exists\, \triangle_{k,2}\ \exists\, \triangle_{k,3}\ ( \\
&\qquad\qquad \bigwedge_{i=1}^{k} \mathbf{CornerP}(\triangle_{i,1}, \triangle_{i,2}, \triangle_{i,3}, \triangle_i) \wedge \\
&\qquad\qquad\quad \hat{\varphi}'(\triangle_{1,1}, \triangle_{1,2}, \triangle_{1,3}, \triangle_{2,1}, \triangle_{2,2}, \triangle_{2,3}, \ldots, \triangle_{k,1}, \triangle_{k,2}, \triangle_{k,3})),
\end{aligned}$$

For each triple of points, there are 6 different representations for the triangle having those points as its corner points. Therefor, for each tuple returned by $\hat{\varphi}'$, $6^k$ tuples will be returned by $\hat{\varphi}$. But, we know that $\dot{\varphi}$ is a well-defined triangle query. This means that, for each $(3k)$ tuple of points $((\mathbf{a}_{1,1}, \mathbf{a}_{1,2}, \mathbf{a}_{1,3}), (\mathbf{a}_{2,1}, \mathbf{a}_{2,2}, \mathbf{a}_{2,3}), \ldots, (\mathbf{a}_{k,1}, \mathbf{a}_{k,2}, \mathbf{a}_{k,3}))$ satisfying $\dot{\varphi}$, also the tuples $((\mathbf{a}_{1,j_{1,1}}, \mathbf{a}_{1,j_{1,2}}, \mathbf{a}_{1,j_{1,3}}), (\mathbf{a}_{2,j_{2,1}}, \mathbf{a}_{2,j_{2,2}}, \mathbf{a}_{2,j_{2,3}}), \ldots, (\mathbf{a}_{k,j_{k,1}}, \mathbf{a}_{k,j_{k,2}}, \mathbf{a}_{k,j_{k,3}}))$, where $\sigma_i(1,2,3) = (j_{i,1}, j_{i,2}, j_{i,3})(1 \leq i \leq k; \sigma_i \in \mathcal{S}_3)$ and $\mathcal{S}_3$ is the set of all permutations of $\{1,2,3\}$, satisfy $\dot{\varphi}$. Therefor, $\hat{\varphi}$ and $\dot{\varphi}$ are equivalent according to definition 4.4. □

We now combine the soundness and completeness lemmas, and use them to prove our main theorem for this section:

**Theorem 5.1 (Expressiveness of FO($\boldsymbol{\Delta}$))** Let $\hat{\sigma} = \{\hat{R}_1, \hat{R}_2, \ldots, \hat{R}_m\}$ be a spatial triangle database schema. Let $\overline{R}_i$ be the corresponding constraint relation names of arity $6 \times ar(\hat{R}_i)$, for $1 \leq i \leq m$, and let $\overline{\sigma}$ be the spatial database schema $\{\overline{R}_1, \overline{R}_2, \ldots, \overline{R}_m\}$. The language FO($\boldsymbol{\Delta}$, $\hat{\sigma}$) is sound and complete for the affine-generic FO($+$, $\times$, $<$, 0, 1, $\overline{\sigma}$)-queries on triangle databases.

**Proof.** Let $\hat{\sigma} = \{\hat{R}_1, \hat{R}_2, \ldots, \hat{R}_m\}$ be a spatial triangle database schema. Let $\dot{R}_i$ be the corresponding spatial point relation names of arity $3 \times ar(\hat{R}_i)$, for $1 \leq i \leq m$, and let $\dot{\sigma}$ be the spatial database schema $\{\dot{R}_1, \dot{R}_2, \ldots, \dot{R}_m\}$. Let $\overline{R}_i(1 \leq i \leq m)$ be the corresponding constraint relation names of arity $6 \times ar(\hat{R}_i)$ and let $\overline{\sigma}$ be the spatial database schema $\{\overline{R}_1, \overline{R}_2, \ldots, \overline{R}_m\}$.

From Lemma 5.2 and Lemma 5.3, we can conclude that FO($\Delta$, $\hat{\sigma}$) is sound and complete for the FO($\{\mathbf{Between}\}$, $\dot{\sigma}$)-queries on triangle databases.

Gyssens, Van den Bussche and Van Gucht showed that FO($\{\mathbf{Between}\}$, $\dot{\sigma}$) is sound and complete for the affine-generic FO($+$, $\times$, $<$, 0, 1, $\overline{\sigma}$)-queries on geometric databases [11].

From the definition of triangle databases, we know that they are geometric databases. This concludes the proof. □

The following remark is important, we will come back to it at the end of this section.

**Remark 5.3** In the proofs of Lemma 5.2 and Lemma 5.3, we only use the fact that triangles are convex objects having three corner points. We use no other properties of triangles. □

The following corollary follows from the fact that FO({**Between**}, $\dot{\sigma}$) +**While** is sound and complete for the computable affine-generic queries on geometric databases [11]. The language FO($\mathbf{\Delta}$, $\hat{\sigma}$) + **While** is a language in which FO($\mathbf{\Delta}$, $\hat{\sigma}$)-definable relations can be created and which has a while-loop with FO($\mathbf{\Delta}$, $\hat{\sigma}$) -definable stop conditions.

**Corollary 5.1 (Expressiveness of** FO($\mathbf{\Delta}$, $\hat{\sigma}$)**+ While)** Let $\hat{\sigma}$ be a spatial triangle database schema. The language FO($\mathbf{\Delta}$, $\hat{\sigma}$)+ **While** is sound and complete for the computable affine-generic queries on triangle databases. □

We now give some examples of FO($\mathbf{\Delta}$, $\hat{\sigma}$)-queries. We illustrate some geometrical constructions in Example 5.1. Afterwards, we formulate queries on an example spatial triangle database in Example 5.2.

**Example 5.1** We illustrate how to express that two triangles are similar, *i.e.*, each side of the first triangle is parallel to a side of the second triangle. We denote the formula expressing this by **Sim**.

We use the predicates **ColSeg** and **ParSeg**, expressing that two line segments are collinear and parallel respectively, to simplify the expression for **Sim**.

$$\begin{aligned}\mathbf{ColSeg}(\triangle_1, \triangle_2) := \mathbf{Seg}(\triangle_1) \wedge \mathbf{Seg}(\triangle_2) \wedge \\ \exists\, \triangle_3\ (\mathbf{Seg}(\triangle_3) \wedge \mathbf{PartOf}(\triangle_1, \triangle_3) \wedge \mathbf{PartOf}(\triangle_2, \triangle_3)).\end{aligned}$$

Here, $\mathbf{Seg}(\triangle_1)$ is a shorthand for

$$\begin{aligned}\exists\, \triangle_4\ \exists\, \triangle_5\ (\mathbf{Point}(\triangle_4) \wedge \mathbf{Point}(\triangle_5) \wedge \\ \forall\, \triangle_6\ ((\mathbf{Point}(\triangle_6) \wedge \mathbf{PartOf}(\triangle_6, \triangle_1)) \rightarrow (\mathbf{Between}_{\mathbf{\Delta}}(\triangle_4, \triangle_6, \triangle_5)))).\end{aligned}$$

The fact that two line segments are parallel is now defined as follows:

$$\begin{aligned}\mathbf{ParSeg}(\triangle_1, \triangle_2) := \mathbf{Seg}(\triangle_1) \wedge \mathbf{Seg}(\triangle_2) \wedge \forall\, \triangle_3\ \forall\, \triangle_4\ ( \\ (\mathbf{ColSeg}(\triangle_1, \triangle_3) \wedge \mathbf{ColSeg}(\triangle_2, \triangle_4)) \rightarrow \\ \neg\exists\, \triangle_5\ (\mathbf{PartOf}(\triangle_5, \triangle_3) \wedge \mathbf{PartOf}(\triangle_5, \triangle_4))).\end{aligned}$$

Now we can write the expression for **Sim**:

$$\begin{aligned}&\mathbf{Sim}(\triangle_1, \triangle_2) := \\ &\exists\, \triangle_{1,1}\ \exists\, \triangle_{1,2}\ \exists\, \triangle_{1,3}\ \exists\, \triangle_{1,4}\ \exists\, \triangle_{1,5}\ \exists\, \triangle_{1,6}\ \exists\, \triangle_{2,1}\ \exists\, \triangle_{2,2}\ \exists\, \triangle_{2,3}\ \exists\, \triangle_{2,4}\ \exists\, \triangle_{2,5}\ \exists\, \triangle_{2,6}\ ( \\ &\quad \bigwedge_{i=1}^{2} (\mathbf{CornerP}(\triangle_{i,1}, \triangle_{i,2}, \triangle_{i,3}, \triangle_i) \wedge \mathbf{CornerP}(\triangle_{i,1}, \triangle_{i,1}, \triangle_{i,2}, \triangle_{i,4}) \wedge \\ &\quad \mathbf{CornerP}(\triangle_{i,2}, \triangle_{i,2}, \triangle_{i,3}, \triangle_{i,5}) \wedge \mathbf{CornerP}(\triangle_{i,3}, \triangle_{i,3}, \triangle_{i,1}, \triangle_{i,6})) \wedge \\ &\quad \bigvee_{\sigma(1,2,3)=(i_1,i_2,i_3), \sigma\in\mathcal{S}_3} (\mathbf{ParSeg}(\triangle_{1,4}, \triangle_{2,(3+i_1)}) \wedge \mathbf{ParSeg}(\triangle_{1,5}, \triangle_{2,(3+i_2)}) \\ &\quad\quad \wedge \mathbf{ParSeg}(\triangle_{1,6}, \triangle_{2,(3+i_3)}))),\end{aligned}$$

where $\mathcal{S}_3$ is the set of all permutations of $\{1, 2, 3\}$. □

We proceed with an example of a spatial database containing information about butterflies, and some FO({**PartOf**})-queries that can be asked to such a database.

**Example 5.2** Consider a triangle database $\mathcal{D}$ over the schema $\hat{\sigma} = \{ButterflyB, PlantP, Rural\}$ that contains information about butterflies and flowers. The unary triangle relation $ButterflyB$ contains all regions where some butterfly $B$ is spotted. The unary triangle relation $PlantP$ contains all regions where some specific plant $P$ grows. We also have a unary triangle relation $Rural$, containing rural regions. It is known in biology that each butterfly appears close to some specific plant, as caterpillars only eat the leaves of their favorite plant. Suppose that it is also investigated that butterflies like to live in rural areas.

• $Q_{10}$ : *Are all butterflies $B$ spotted in regions where the plant $P$ grows?* This query can be used to see if it is possible that a butterfly was spotted in a certain region. The query $Q_{10}()$ can be expressed by the formula

$$\neg(\exists \triangle_1 \ \exists \triangle_2 \ (ButterflyB(\triangle_1) \wedge \mathbf{RealTriangle}(\triangle_2) \wedge \mathbf{PartOf}(\triangle_2, \triangle_1) \wedge \neg(\exists \triangle_3 \ (PlantP(\triangle_3) \wedge \mathbf{PartOf}(\triangle_2, \triangle_3))))).$$

Here, $\mathbf{RealTriangle}(\triangle)$ is a shorthand for $\neg\mathbf{Point}(\triangle) \wedge \neg\mathbf{Line}(\triangle)$.

• $Q_{11}$ : *Give the region(s) where we have to search if we want to see butterfly $B$.* The query $Q_{11}(\triangle)$ can be expressed by the formula

$$\exists \triangle_2 \ \exists \triangle_3 \ (PlantP(\triangle_2) \wedge Rural(\triangle_3) \wedge \mathbf{PartOf}(\triangle, \triangle_2) \wedge \mathbf{PartOf}(\triangle, \triangle_3)).$$

• $Q_{12}$ : *Give the region inside the convex hull of the search region for butterfly $B$.* It is much more convenient to search a convex region than having to deal with a very irregularly shaped region.

We first express how to test whether the region is convex ($Q'_{12}$), this will help understand the formula that computes the convex hull. The query $Q'_{12}()$ can be expressed by the formula

$$\forall \triangle_1 \ \forall \triangle_2 \ \forall \triangle_3 \ \forall \triangle_4 \ ((\bigwedge_{i=1}^{3} \mathbf{Point}(\triangle_i) \wedge \bigwedge_{i=1}^{3} Q_{11}(\triangle_i) \wedge \mathbf{CornerP}(\triangle_1, \triangle_2, \triangle_3, \triangle_4)) \Rightarrow (Q_{11}(\triangle_4))).$$

The expression

$$\exists \triangle_1 \ \exists \triangle_2 \ \exists \triangle_3 \ \exists \triangle_4 \ \exists \triangle_5 \ \exists \triangle_6 \ (\bigwedge_{i=1}^{3} \mathbf{Point}(\triangle_i) \wedge \bigwedge_{i=1}^{3} \mathbf{PartOf}(\triangle_i, \triangle_{i+3}) \wedge \bigwedge_{i=4}^{6} \mathbf{Q}_{11}(\triangle_i) \wedge \mathbf{CornerP}(\triangle_1, \triangle_2, \triangle_3, \triangle))$$

hence defines the query $Q_{12}(\triangle)$. For any three points in some triangles in $Q_{11}$, the triangle connecting them is added to $Q_{12}$. Figure 6 illustrates this. □

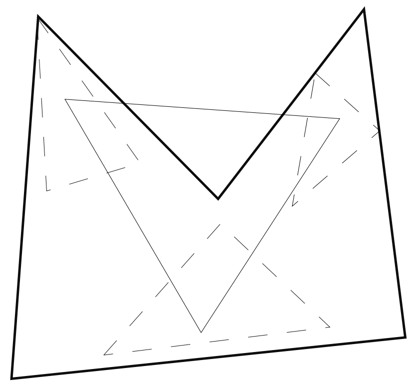

Figure 6: The convex hull of a set $S$ of triangles is computed by adding all triangles constructed from three points that are inside three triangles of $S$.

**Remark 5.4** The first two queries of Example 5.2 ask for relations between regions that can be expressed by the so-called 9-intersection model [9]. This model defines a relation between two regions by investigating the intersections between their boundaries, interiors and exteriors. As the boundary, interior and exterior of a region can be expressed in FO($+$, $\times$, $<$, 0, 1, $\overline{\sigma}$), and are affine invariant concepts [3], all relations that can be expressed by the 9-intersection model, can be expressed in FO($\mathbf{\Delta}$, $\hat{\sigma}$). □

**Remark 5.5** We now reconsider Remark 5.3. In the proofs of Lemma 5.2 and Lemma 5.3, we only used the fact that triangles are convex objects having three corner points. It is not difficult to prove that the predicate **PartOf** can be generalized to a predicate $\mathbf{PartOf}^{(n,k)}$, which arguments are $n$-dimensional convex objects with $k$ corner points (*(n,k)-objects*) and that the language FO($\{\mathbf{PartOf}^{(n,k)}\}$) is sound and complete for the first-order affine-generic queries on $(n, k)$-objects. □

In the context of this remark, we also want to refer to the work of Aiello and van Benthem [1, 2] on modal logics of space. They first propose a topological modal logic over regions, which can express "connectedness" and "parthood". By adding a "convexity" operator (expressed using a "betweenness" operator), they obtain an affine modal logic. Essentially, we do the same, as triangles are convex and connected sets, and we add the "parthood" operator **PartOf**.

In [2], the authors also motivate the use of finite unions of convex sets as basic elements for spatial reasoning. They argue that it is a very natural way for people to reason about objects. A fork, for example will be described as the union of its prongs and its handle.

## 5.2 Safety of Triangle Database Queries

Triangle relations can represent infinite sets of triangles. In practice, however, spatial databases will contain only finite sets of triangles. The $ButterflyB$ and $Rural$ triangle relations

[3]To be exact, they are topological concepts. The affinities of the plane are a subgroup of the homeomorphisms of the plane, so the invariance under the boundary and interior operations carry over naturally.

of Example 5.2, for instance, will be modelled in practice using a finite number of triangles.

The question that arises naturally is whether the language FO({**PartOf**}) returns a finite set of triangles when the input relations represent finite sets of triangles. The answer is "no" (see Example 5.3 below). In database theory this problem is usually referred to as the *safety problem*. Safety of FO(+, ×, <, 0, 1)-queries is undecidable in general [3], so we cannot decide *a priori* whether a triangle database query will return a finite output or not.

The following example illustrates the fact that the language FO({**PartOf**}) does not necessarily return finite output on finite input.

**Example 5.3** Let $\hat{\sigma} = \{\hat{R}\}$ be a spatial triangle database schema, with $\hat{R}$ a triangle relation containing a finite number of triangles. Consider the following spatial triangle database queries:

- $Q_{13}$ : *Give all triangles that are part of some triangle of* $\hat{R}$.
  The query $Q_{13}(\triangle)$ is expressed in FO($\Delta$, $\hat{\sigma}$) by the formula

$$\exists \triangle' \ (\hat{R}(\triangle') \wedge \mathbf{PartOf}(\triangle, \triangle')).$$

- $Q_{14}$ : *Give all triangles that intersect some triangle of* $\hat{R}$. The query $Q_{14}(\triangle)$ can be expressed by the formula

$$\exists \triangle' \ (\hat{R}(\triangle') \wedge \mathbf{Intersect}(\triangle, \triangle')).$$

- $Q_{15}$ : *Give all the corner points of all triangles of* $\hat{R}$. The query $Q_{15}(\triangle)$ can be expressed by the formula

$$\begin{aligned}\exists \triangle_1 \ \exists \triangle_{1,2} \ \exists \triangle_{1,3} \ (\hat{R}(\triangle_1) \wedge (&\mathbf{CornerP}(\triangle, \triangle_{1,2}, \triangle_{1,3}, \triangle_1) \\ &\vee \mathbf{CornerP}(\triangle_{1,2}, \triangle, \triangle_{1,3}, \triangle_1) \vee \mathbf{CornerP}(\triangle_{1,2}, \triangle_{1,3}, \triangle, \triangle_1))).\end{aligned}$$

The queries $Q_{13}$ and $Q_{14}$ return an infinite set of triangles. The query $Q_{15}$ returns a finite number of triangles on the condition that the input relation $\hat{R}$ is finite. □

As we cannot decide whether a given triangle database query will return a finite result, we turn to the question of determining whether the result of the query is finite or not, after executing the query. The answer is affirmative:

**Proposition 5.1 (Finiteness of triangle relations is decidable)** It is decidable whether a triangle relation consists of a finite number of triangles. Moreover, there exists a FO($\Delta$)$\{\hat{R}\}$ query that decides whether the triangle relation named $\hat{R}$ consists of a finite number of triangles.

**Proof.** A triangle relation of arity $k$ corresponds to a semi-algebraic set in $\mathbb{R}^{6k}$. The canonical bijection $can \circ can_{tr} : ((\mathbb{R}^2)^3)^k \to \mathbb{R}^{6k}$ establishes this correspondence. A triangle relation is finite if and only if the corresponding semi-algebraic set contains a finite number of points (in $\mathbb{R}^{(6k)}$). It is well known that there exists a FO(+, ×, <, 0, 1)-formula deciding whether a semi-algebraic set contains a finite number of points. Also, the fact that a triangle relation contains a finite number of $k$-tuples of triangle is affine-invariant. From the fact that the property is affine-invariant and expressible in FO(+, ×, <, 0, 1), it follows (from Theorem 5.1) that there is a a FO(Δ, $\{\hat{R}\}$)-formula expressing whether a triangle relation $\hat{R}$ is finite or not. □

We now have a means of deciding whether a triangle relation is finite, but it seems this requirement is too restrictive.

In Definition 4.1 in Section 4, we introduced the concept *drawing* of a triangle. We now straightforwardly extend this definition to spatial triangle databases.

**Definition 5.3 (Drawing of a triangle relation)** Let $\hat{R}$ be a triangle relation of arity one. The *drawing* of $\hat{R}$ is the two-dimensional figure that is the union of the drawings of all triangles in $\hat{R}$. □

For the remainder of this text, we restrict triangle relations (and triangle database queries) to be unary. It is not clear immediately if it would make sense to define drawings on relations or queries with an arity greater than one. For example, consider a binary relation containing only one tuple of line-adjacent non-degenerated triangles. If we draw this relation, we would like to draw both triangles participating in the relation. This gives the same result as the drawing of a unary relation containing two tuples. So the drawing apparently "wipes out" the relationship between the triangles.

We also remark the following.

**Remark 5.6** Different triangle relations can have the same drawing. Therefore, it seems natural to extend the strict notion of finiteness of a triangle relation to the existence of a finite triangle relation having the same drawing. Query $Q_1$ from Example 5.3, for instance, seems to be a query we would like to call "finite", because there exists a finite union of triangles with the same drawing. Indeed, the drawing of the union of all triangles that are part of a given triangle, is the same as the drawing of the given triangle itself. Query $Q_2$ clearly returns an infinite set of triangles that is cannot be represented as a finite union of triangles. This is the type of query we don't want to allow. □

Fortunately, given the output of a unary query, we can determine whether its drawing can be represented as a finite union of triangles.

**Proposition 5.2 (Finite triangle representation)** Let $\hat{\sigma}$ be a spatial triangle database schema. Given a unary triangle database query $\hat{Q}$ that is expressible in FO({**PartOf**}, $\hat{\sigma}$) and a spatial triangle database $\mathcal{D}$ over $\hat{\sigma}$, it is decidable whether the unary relation, named $\hat{R}_{\hat{Q}}$, containing $\hat{Q}(\mathcal{D})$ can be represented as a finite union of triangles. Furthermore, there exists a FO(Δ, $\hat{\sigma}'$)-formula deciding this for $\hat{\sigma}' = \hat{\sigma} \cup \{\hat{R}_{\hat{Q}}\}$.

**Proof.** It is clear that if the drawing of a triangle relation can be represented as a finite union of triangles, it can be represented by a FO(+, ×, <, 0, 1)-formula using only polynomials of degree at most one. A set that can be described using polynomials of at most degree one, is called a semi-linear set. It is well-known that the bounded semi-linear sets are the same as finite unions of bounded polytopes (which triangles are).

So, if we can check whether the drawing of a (possibly infinite) set of triangles is bounded and can be represented using polynomials of degree at most one, we know that the set can be represented by a finite number of triangles.

Checking whether the drawing of a triangle relation $\hat{R}$ is bounded can be done easily in FO({**PartOf**}){$\hat{R}$}. The following formula performs this check.

$$\mathbf{IsBounded}() := \exists \triangle_1 \ \forall \triangle_2 \ (\hat{R}(\triangle_2) \rightarrow \mathbf{PartOf}(\triangle_2, \triangle_1)).$$

Also, we can decide whether a two-dimensional[4] semi-algebraic set can be represented using polynomials of degree at most $k$, for any natural number $k$ [17]. There exists a FO(+, ×, <, 0, 1)-formula deciding this [17]. It is clear that the drawing of a unary triangle relation is a semi-algebraic set.

From the facts that (*i*) computing the drawing of a triangle relation is an affine-generic query that can be expressed in FO(+, ×, <, 0, 1) and that (*ii*) checking whether a triangle relation has a bounded drawing can be expressed in FO({**PartOf**}, $\hat{\sigma}$) and that (*iii*) there exists a FO(+, ×, <, 0, 1)-formula deciding whether the drawing of a triangle relation can be expressed by polynomials of degree at most one can be done in FO(+, ×, <, 0, 1) and, finally, that (*iv*) the fact that the drawing of a triangle relation can be expressed by polynomials of degree at most one is affine-invariant, we conclude that we can decide whether a triangle relation has a finite representation, and that we can construct a FO({**PartOf**}, $\hat{\sigma}$)-formula deciding this. □

We now show that, if the drawing of the output of a triangle database query is representable as a finite set of triangles, we can compute such a finite triangle representation in FO({**PartOf**}, $\hat{\sigma}$).

In [12], we proposed an algorithm that computes an affine invariant triangulation of a set of triangles. Recall that this algorithm computes the drawing of the input triangles, then partitions this drawing into a set of convex polygons according to the carriers of its boundary segments and finally triangulates convex polygons by connecting their center of mass to their corner points.

We assumed in [12] that the input set of triangles for the triangulation algorithm was finite. On an infinite collection of triangles for which there exists a finite collection of triangles with the same drawing, this algorithm would work also correctly, however. The triangulation described in [12] therefor seems a good candidate for representing infinite sets of triangles by finite sets of triangles. But, in [12], we conjectured that this triangulation cannot be expressed in FO({**PartOf**}, {$\hat{R}$}). The reason for this is the conjecture that the center of mass of a polygon, which is an affine-invariant, cannot be expressed in FO(+, ×, <, 0, 1), and therefore, also not in FO({**PartOf**}, {$\hat{R}$}).

**Conjecture 5.1** Let $P = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k\}$ be a set of corner points that represent a convex polygon. Assume that $k > 3$. The center of mass of the polygon represented by $P$ cannot be expressed in FO(+, ×, <, 0, 1). □

Remark that the center of mass of an arbitrary set of points is not expressible in FO(+, ×, <, 0, 1).

So, the triangulation algorithm from [12] cannot be used. But, this algorithm computes a *partition* of the input into triangles, which is not a requirement here. If we relax the requirement of having a partition of the original figure into triangles down to having a *finite union* of (possibly overlapping) triangles representing the figure, we can avoid the computation of the center of mass. The adapted algorithm $AfTr(S)$ in given in Figure 7.

[4] Note that this is not true for arbitrary dimensions.

**Require:** $S$ is a unary triangle relation that can be represented as a finite union of triangles.
1: Compute the boundary $B_S$ of $S$. $B_S$ is a finite set of line segments and points.
2: Compute the set of carriers for all line segments of $B_S$. Those carriers partition $S$ into a finite union of open convex polygons, points and open line segments. All closures of line segments that do not form a side of one of the convex polygons, together with all points that are not a corner point of one of the convex polygons are returned as degenerated triangles. Remark that we can return the closures of the line segments as $S$ originally is a union of closed triangles, closed line segments and points.
3: **for** each polygon **do**
4: output the finite set of triangles that connect three distinct corner points of the polygon
5: **end for**

Figure 7: The algorithm $AfTr(S)$.

Given an unary triangle relation $\hat{R}$, we denote the result of algorithm $AfTr(S)$ in Figure 7 on input $\hat{R}$ by the *affine finite triangle representation* of $\hat{R}$, or, abbreviated, $AfTr(\hat{R})$. Now we show that $AfTr(\hat{R})$ can be computed in FO(**Delta**, $\hat{R}$), provided that $\hat{R}$ can be represented as a finite union of triangles.

**Proposition 5.3 (Affine finite triangle representation)** Given a unary triangle relation $\hat{R}$ that can be represented as a finite union of triangles, then there exists an FO($\mathbf{\Delta}$, $\{\hat{R}\}$)-formula returning $AfTr(\hat{R})$.

**Proof.** We use the fact that all affine-generic semi-algebraic queries on triangle databases can be expressed in FO(**Delta**, $\hat{R}$). Therefor, we have to prove that, first, the affine finite triangle representation is affine-invariant and, second, that the affine finite representation is expressible in FO(+, ×, <, 0, 1).

*The affine finite representation is an affine invariant.*

We only have to prove this for Step 3 of algorithm $AfTr(S)$ in Figure 7. The rest follows from the analogous property in [12].

Let $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k\}$ be the set of corner points of a convex polygon $P$, where $k \geq 3$. Let $\alpha$ be an affinity of the plane. The set $\{\alpha(\mathbf{a}_1), \alpha(\mathbf{a}_2), \ldots, \alpha(\mathbf{a}_k)\}$ contains the corner points of the convex polygon $\alpha(P)$. It is clear that, for each triangle $(\mathbf{a}_h, \mathbf{a}_i, \mathbf{a}_j)$ (such that $h \neq i$, $i \neq j$, $h \neq j$ and $1 \leq h, i, j \leq k$) connecting three corner points of $P$, the triangle $\alpha(\mathbf{a}_h, \mathbf{a}_i, \mathbf{a}_j) = (\alpha(\mathbf{a}_h), \alpha(\mathbf{a}_i), \alpha(\mathbf{a}_j))$ is an element of the set of triangles connecting three corner points of $\alpha(P)$.

*The affine finite representation is computable in* FO(+, ×, <, 0, 1).

In FO(+, ×, <, 0, 1), it is possible to compute the boundary of a semi-linear set (Line 1 of the algorithm $AfTr(S)$ in Figure 7). It is also possible to compute the carriers of all boundary line segments, and their intersection points (Line 2). It can be expressed that two points belong to the same convex polygon, namely, by expressing that the line segment in between them is not intersected by a carrier. Finally, the set of all triples of intersection points between carriers that belong to the same convex polygon can be computed in FO(+, ×, <, 0, 1) (Lines 3 through 5). From the fact that the triangle representation is affine invariant and computable in FO(+, ×, <, 0, 1), it follows that it is computable in FO({**PartOf**}). □

This section on safety finishes the "spatial" part of this text. In the remaining part, we develop a query language for spatio-temporal triangle databases.

# 6 Spatio-temporal Triangle Queries

In this section, we will extend the spatial triangle logic FO({**PartOf**}) to a logic over spatio-temporal triangles, *i.e.*, triples of co-temporal points in ($\mathbb{R}^2 \times \mathbb{R}$). The genericity classes we consider in this section, are the group $(\mathcal{A}_{st}, \mathcal{A}_t)$ of time-dependent affinities, the group $(\mathcal{V}_{st}, \mathcal{A}_t)$ of velocity-preserving transformations and the group $(\mathcal{AC}_{st}, \mathcal{A}_t)$ of acceleration-preserving transformations. The first group is a natural spatio-temporal extension of the affinities of space. We also include the two other groups, because they are very relevant from a practical point of view, and because the point languages we previously identified as generic for those groups were not very intuitive.

Recall that $\mathcal{A}_t$ is the group of the affinities on the time line and that the elements of $\mathcal{A}_{st}$ are of the form

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ t \end{pmatrix} \mapsto \begin{pmatrix} \alpha_{11}(t) & \alpha_{12}(t) & \cdots & \alpha_{1n}(t) \\ \alpha_{21}(t) & \alpha_{22}(t) & \cdots & \alpha_{2n}(t) \\ \vdots & \vdots & \cdots & \vdots \\ \alpha_{n1}(t) & \alpha_{n2}(t) & \cdots & \alpha_{nn}(t) \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} \beta_1(t) \\ \beta_2(t) \\ \vdots \\ \beta_n(t) \end{pmatrix},$$

where the matrix of the $\alpha_{ij}(t)$ is an affinity for each value of $t$. The group $(\mathcal{AC}_{st}, \mathcal{A}_t)$ is the subgroup of $(\mathcal{A}_{st}, \mathcal{A}_t)$ in which the functions $\alpha_{ij}$ are constants and the functions $\beta_{ij}$ are linear functions of time. The group $(\mathcal{V}_{st}, \mathcal{A}_t)$ is the subgroup of $(\mathcal{AC}_{st}, \mathcal{A}_t)$ where the $\beta_{ij}$ are constants too.

In [10], we proposed point languages capturing exactly those genericity classes. Table 1 summarizes the point languages expressing all $(\mathcal{F}_{st}, \mathcal{F}_t)$-generic queries, for the above groups $(\mathcal{F}_{st}, \mathcal{F}_t)$. As we will always assume, in this section, that the underlying dimension is 2, we adapted the table accordingly. Now we propose spatio-temporal point languages that have the same expressivity as the languages listed in Table 1, but on spatio-temporal triangle databases.

| $(\mathcal{F}_{st}, \mathcal{T}_t)$ | Set of point predicates $\Pi(\mathcal{F}_{st}, \mathcal{T}_t)$ |
|---|---|
| $(\mathcal{A}_{st}, \mathcal{A}_t)$ | $\{\mathbf{Between}^{\text{Cotemp}}, \mathbf{Before}, \mathbf{EqCr}^{ST}\}$ |
| $(\mathcal{AC}_{st}, \mathcal{A}_t)$ | $\{\mathbf{Between}, \mathbf{Before}\}$ |
| $(\mathcal{V}_{st}, \mathcal{A}_t)$ | $\{\mathbf{Between}, \mathbf{Before}, \mathbf{EqSpace}\}$ |

Table 1: The point logics FO($\Pi(\mathcal{F}_{st}, \mathcal{T}_t)$) capturing the FO $(\mathcal{F}_{st}, \mathcal{T}_t)$-generic queries, for the classes $(\mathcal{A}_{st}, \mathcal{A}_t)$, $(\mathcal{AC}_{st}, \mathcal{A}_t)$ and $(\mathcal{V}_{st}, \mathcal{A}_t)$.

We will start with the most general transformation group, the group $(\mathcal{A}_{st}, \mathcal{A}_t)$ of time-dependent affinities.

## 6.1 Predicates Invariant under Time-dependent Affinities

In this section, we propose a set of spatio-temporal triangle predicates such that the spatio-temporal triangle logic with this predicate set, captures exactly the $(\mathcal{A}_{st}, \mathcal{A}_t)$-generic queries on spatio-temporal triangle databases that are expressible in FO($+$, $\times$, $<$, 0, 1). We can prove this by comparing the expressiveness of this spatio-temporal triangle logic with the language FO($\{\mathbf{Between}^{\mathrm{Cotemp}}, \mathbf{Before}, \mathbf{EqCr}^{ST}\}$), when used as a spatio-temporal triangle query language (see Definition 4.5). Recall also that we will have to make sure that the result of a spatio-temporal triangle query is a consistent spatio-temporal triangle relation.

The nature of the class $(\mathcal{A}_{st}, \mathcal{A}_t)$ is such that $(\mathcal{A}_{st}, \mathcal{A}_t)$-generic queries can describe snapshots of a spatio-temporal database in fairly much detail, *i.e.*, all affine-invariant properties of the snapshot can be expressed. In between snapshots, the expressive power of $(\mathcal{A}_{st}, \mathcal{A}_t)$-generic queries is more limited. This follows directly from the fact that an element of $(\mathcal{A}_{st}, \mathcal{A}_t)$ transforms each snapshot with another affinity. We now want to construct a $(\mathcal{A}_{st}, \mathcal{A}_t)$-generic query language for spatio-temporal triangle databases. This means we will be able to describe a spatio-temporal triangle database by means of its snapshots, which are collections of snapshots of spatio-temporal triangles in $(\mathbb{R}^2 \times \{\tau_0\})^3$, for some $\tau_0 \in \mathbb{R}$. The basic objects for our new language will be, accordingly, triples of co-temporal points. In this section, we will call these triples of points *triangle snapshots*. Triangle snapshot variables will be denoted $\triangle^{st}, \triangle_1^{st}, \triangle_2^{st}, \ldots$ and triangle snapshot constants by $T^{st}, T_1^{st}, T_2^{st}, \ldots$. If we want to emphasize the connection between a triangle snapshot and its corner points, we use the notation $T_{pqr}^{st}$.

In our search for a set of predicates on triangle snapshots for a $(\mathcal{A}_{st}, \mathcal{A}_t)$-generic query language, or, a language with the same expressive power as the language FO($\{\mathbf{Between}^{\mathrm{Cotemp}}$, $\mathbf{Before}, \mathbf{EqCr}^{ST}\}$) on spatio-temporal triangle databases, the following observations are helpful.

(i) In [10], we showed that we need the binary predicate **Before** on points to reflect the monotonicity of time, which is preserved by the transformation group $(\mathcal{A}_{st}, \mathcal{A}_t)$.

(ii) The predicate $\mathbf{Between}^{\mathrm{Cotemp}}$ is used to express affine-invariant properties of co-temporal points.

(iii) In Section 5, we showed that the predicate **PartOf** has the same expressive power as the predicate **Between**, on (spatial) triangles.

From observation (i) it follows that the query language we want to construct should be able to express the order on triangle snapshots. We introduce the triangle snapshot predicate $\mathbf{Before}_\triangle$, which, when applied to two triangle snapshots, expresses that the first one is strictly before or co-temporal with the second one. We will define this more formally later.

From observation (ii) and (iii), we conclude that we can use, slightly adapted, the predicate **PartOf** on co-temporal triangle snapshots, we will denote it $\mathbf{PartOf}^{\mathrm{Cotemp}}$. This will allow us to express snapshots of spatio-temporal triangle databases in an affine-invariant way. Concluding, the set of spatio-temporal triangle predicates we are looking for should contain the elements $\mathbf{PartOf}^{\mathrm{Cotemp}}$ and $\mathbf{Before}_\triangle$. Because, in the end, we want to express al queries expressible in FO($\{\mathbf{Between}^{\text{Cotemp}}, \mathbf{Before}, \mathbf{EqCr}^{ST}\}$), on spatio-temporal triangle databases, we still have to look for a (set of) triangle snapshot predicate(s) capturing the expressive power of the predicate $\mathbf{EqCr}^{ST}$.

We repeat the definition of the point predicate $\mathbf{EqCr}^{ST}$. For six spatio-temporal points $p_1, p_2, p_3, q_1, q_2, q_3 \in (\mathbb{R}^2 \times \mathbb{R})$, $\mathbf{EqCr}^{ST}(p_1, p_2, p_3, q_1, q_2, q_3)$ expresses that the cross-ratio of the three co-temporal and collinear points $p_1$, $p_2$ and $p_3$ equals the cross-ratio of the time coordinates $\tau_{q_1}$, $\tau_{q_2}$ and $\tau_{q_3}$ of the points $q_1$, $q_2$ and $q_3$. The expression $\mathbf{EqCr}^{ST}(p_1, p_2, p_3, q_1, q_2, q_3)$ implicitly refers to a movement. Indeed, the line segment defined by the points $p_1$ and $p_3$ and the interval $[\tau_{q_1}, \tau_{q_3}]$ can be interpreted as the spatial and temporal projection of a linear movement with constant speed and we can then interpret $\mathbf{EqCr}^{ST}(p_1, p_2, p_3, q_1, q_2, q_3)$ as an expression of the fact that when an object moves with constant speed from $p_1$ to $p_3$ during the interval $[\tau_{q_1}, \tau_{q_3}]$, it passes $p_2$ at time moment $\tau_{q_2}$.

There is one obvious way to define the speed of a moving point. For moving triangles, or moving objects in general, the definition of *speed* is somewhat ambiguous. Triangles can move by changing their position, but also by changing their shape. We define the speed (resp., acceleration) of a moving triangle as the speed (resp., acceleration) of is moving *center of mass*. Hence, a triangle that is growing or shrinking, but its center of mass remains in the same position, has zero speed. Based on that definition, we propose a spatio-temporal triangle database query language, with the triangle predicates $\mathbf{PartOf}^{\text{Cotemp}}$, $\mathbf{Before}_{\triangle}$ and **Cas** (which is an abbreviation of "Constant Average Speed"). The predicate **Cas** takes six arguments $\triangle_1^{st}, \triangle_2^{st}, \ldots, \triangle_6^{st}$. The first three triangle snapshots, $\triangle_1^{st}, \triangle_2^{st}$ and $\triangle_3^{st}$, are co-temporal and their barycenters are collinear. The last three triangle snapshots, $\triangle_4^{st}, \triangle_5^{st}$ and $\triangle_6^{st}$, indicate three different time moments. Furthermore, the cross-ratio of the barycenters of $\triangle_1^{st}, \triangle_2^{st}$ and $\triangle_3^{st}$ is the same as the cross-ratio of the time coordinates of $\triangle_4^{st}, \triangle_5^{st}$ and $\triangle_6^{st}$. Intuitively, this predicate, similar to the point predicate $\mathbf{EqCr}^{ST}$, approximates or estimates a linear movement. Given the time interval during which a triangle moves from the first position to the second one, it estimates, assuming the triangle moves with constant speed, how long it will take to reach the position of the third triangle.

It turns out, however, that the language with these three triangle predicates is not very intuitive to express properties of the shape of triangles, e.g., their relative areas. Therefor, we will also propose an alternative language. This language has exactly the same expressivity as the first one, but offers a more direct means to express shape properties of triangles. We propose to replace the predicate **Cas** by the predicate **Lex** (which is an abbreviation for "Linear Expansion"). This predicate also takes six arguments $\triangle_1^{st}$, $\triangle_2^{st}$, $\ldots$, $\triangle_6^{st}$. The first three triangle snapshots, $\triangle_1^{st}, \triangle_2^{st}$ and $\triangle_3^{st}$, are co-temporal and both $\mathbf{PartOf}^{\text{Cotemp}}(\triangle_1^{st}, \triangle_2^{st})$ and $\mathbf{PartOf}^{\text{Cotemp}}(\triangle_2^{st}, \triangle_3^{st})$ hold. The other three triangle snapshots exist at three different time moments. Finally, the cross-ratio of the time coordinates of $\triangle_4^{st}, \triangle_5^{st}$ and $\triangle_6^{st}$ equals the cross ratio of the areas of the three first triangles. More exactly,

$$\frac{|A(\triangle_2^{st}) - A(\triangle_1^{st})|}{|A(\triangle_3^{st}) - A(\triangle_1^{st})|} = \frac{|\tau_{\triangle_2^{st}} - \tau_{\triangle_1^{st}}|}{|\tau_{\triangle_3^{st}} - \tau_{\triangle_1^{st}}|},$$

where $\tau_{\triangle_i^{st}}$ denotes the time moment at which $\triangle_i^{st}$ exists and $A(\triangle_i^{st})$ denotes the area of the triangle $\triangle_i^{st}$. Intuitively, this predicate approximates or estimates a linear growth or expansion. Given the time interval during which the first triangle expanded into the second one, it estimates, assuming the triangle grows linearly, how long it will take to reach the area of the third triangle.

In applications where objects are not growing or shrinking, a language with the predicate **Cas** may be preferred, whereas in applications where objects do change their shape, the predicate **Lex** may be preferred. Of course, one can also include both predicates to make the language suitable for all types of applications.

We will prove that the both the languages FO($\{\mathbf{PartOf}^{\text{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{Cas}\}$) and FO($\{\mathbf{PartOf}^{\text{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{Lex}\}$) are sound and complete for the $(\mathcal{A}_{st}, \mathcal{A}_t)$-generic first-order spatio-temporal database queries.

### 6.1.1 Expressiveness of the Language FO($\{\mathbf{PartOf}^{\mathbf{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{Cas}\}$)

In this section, we first give the definitions of the triangle predicates $\mathbf{PartOf}^{\text{Cotemp}}$, $\mathbf{Before}_\triangle$ and **Cas**. Next, we show that the language FO($\{\mathbf{PartOf}^{\text{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{Cas}\}$) produces queries that are well-defined on spatio-temporal triangle databases. After that, we show its expressive power.

**Definition 6.1 (The triangle snapshot predicate $\mathbf{PartOf}^{\mathbf{Cotemp}}$)** Let $T_1^{st} = (p_{1,1}, p_{1,2}, p_{1,3})$ and $T_2^{st} = (p_{2,1}, p_{2,2}, p_{2,3})$ be two triangle snapshots. The binary predicate $\mathbf{PartOf}^{\text{Cotemp}}$, applied to $T_1^{st}$ and $T_2^{st}$ expresses that $p_{1,1}$, $p_{1,2}$ and $p_{1,3}$ (resp., $p_{2,1}$, $p_{2,2}$ and $p_{2,3}$) are co-temporal and that the convex closure of the three points $p_{1,1}$, $p_{1,2}$ and $p_{1,3}$ is a subset of the convex closure of the three points $p_{2,1}$, $p_{2,2}$ and $p_{2,3}$. □

**Definition 6.2 (The triangle snapshot predicate $\mathbf{Before}_\triangle$)** Let $T_1^{st} = (p_{1,1}, p_{1,2}, p_{1,3})$ and $T_2^{st} = (p_{2,1}, p_{2,2}, p_{2,3})$ be two triangle snapshots. The binary predicate $\mathbf{Before}_\triangle$, applied to $T_1^{st}$ and $T_2^{st}$ expresses that $p_{1,1}$, $p_{1,2}$ and $p_{1,3}$ (resp., $p_{2,1}$, $p_{2,2}$ and $p_{2,3}$) are co-temporal and that the time coordinate $\tau_{p_{1,1}}$ of $p_{1,1}$ is smaller than or equal to the time coordinate $\tau_{p_{2,1}}$ of $p_{2,1}$. □

**Definition 6.3 (The triangle snapshot predicate Cas)** Let $T_1^{st} = (p_{1,1}, p_{1,2}, p_{1,3}), T_2^{st} = (p_{2,1}, p_{2,2}, p_{2,3}), \ldots, T_6^{st} = (p_{6,1}, p_{6,2}, p_{6,3})$ be six triangle snapshots. Let $q_1$ (resp., $q_2$, $q_3$) be the barycenter of $T_1^{st}$ (resp., $T_2^{st}$, $T_3^{st}$). The 6-ary predicate **Cas**, applied to $T_1^{st}$, $T_2^{st}$, $\ldots$, $T_6^{st}$ expresses that $p_{i,1}$, $p_{i,2}$ and $p_{i,3}$ are co-temporal for $i = 1 \ldots 6$, that $q_1$, $q_2$ and $q_3$ are collinear and that the cross-ratio of the points $q_1$, $q_2$ and $q_3$ is the same as the cross-ratio of the time coordinates $\tau_{p_{4,1}}$, $\tau_{p_{5,1}}$ and $\tau_{p_{6,1}}$ of $p_{4,1}$, $p_{5,1}$ and $p_{6,1}$, respectively. □

We now show, by induction on their structure, that the FO($\{\mathbf{PartOf}^{\text{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{Cas}\}$)-queries are well-defined on spatio-temporal triangle databases.

**Lemma 6.1 (FO($\{\mathbf{PartOf}^{\mathbf{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{Cas}\}$) is well-defined)** Let $\hat{\sigma}^{st} = \{\hat{R}_1^{st}, \hat{R}_2^{st}, \ldots, \hat{R}_m^{st}\}$ be a spatio-temporal triangle database schema. Let $\mathcal{D}^{st}$ be a consistent spatio-temporal triangle database over $\hat{\sigma}^{st}$. For each FO($\{\mathbf{PartOf}^{\text{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{Cas}\}$, $\hat{\sigma}^{st}$)-query $\hat{Q}$, $\hat{Q}(\mathcal{D}^{st})$ is a consistent triangle relation.

**Proof.** Let $\hat{\sigma}^{st} = \{\hat{R}_1^{st}, \hat{R}_2^{st}, \ldots, \hat{R}_m^{st}\}$ be a spatio-temporal triangle database schema. Let $\mathcal{D}^{st}$ be a consistent spatial triangle database over $\hat{\sigma}^{st}$.

We prove this lemma by induction on the structure of FO($\{\textbf{PartOf}^{\text{Cotemp}}, \textbf{Before}_\triangle, \textbf{Cas}\}$, $\hat{\sigma}$)-queries. The atomic formulas of FO($\{\textbf{PartOf}^{\text{Cotemp}}, \textbf{Before}_\triangle, \textbf{Cas}\}$, $\hat{\sigma}$) are equality expressions on spatio-temporal triangle variables, expressions of the form $\textbf{PartOf}^{\text{Cotemp}}(\triangle_1^{st}, \triangle_2^{st})$, expressions of the form $\textbf{Before}_\triangle(\triangle_1^{st}, \triangle_2^{st})$, expressions of the form $\textbf{Cas}(\triangle_1^{st}, \triangle_2^{st}, \ldots, \triangle_6^{st})$, and expressions of the form $\hat{R}_i^{st}(\triangle_1^{st}, \triangle_2^{st}, \ldots, \triangle_{ar(\hat{R}_i)}^{st})$, where $\hat{R}_i^{st}(1 \leq i \leq m)$ is a relation name from $\hat{\sigma}^{st}$. More complex formulas can be constructed using the Boolean operators $\wedge$, $\vee$ and $\neg$ and existential quantification.

For the atomic formulas, it is easy to see that, if two triangles $T_1^{st}$ and $T_2^{st}$ satisfy the conditions $T_1^{st} =_\triangle T_2^{st}$, $\textbf{PartOf}^{\text{Cotemp}}(T_1^{st}, T_2^{st})$, or $\textbf{Before}_\triangle(T_1^{st}, T_2^{st})$ that also $T_3^{st} =_\triangle T_4^{st}$ respectively $\textbf{PartOf}^{\text{Cotemp}}(T_3^{st}, T_4^{st})$, $\textbf{Before}_\triangle(T_3^{st}, T_4^{st})$ are true if and only if $T_1 =_\triangle T_3$ and $T_2 =_\triangle T_4$ are true. As we assume the input database $\mathcal{D}$ to be consistent, the atomic formulas of the type $\hat{R}_i^{st}(\triangle_1^{st}, \triangle_2^{st}, \ldots, \triangle_{ar(\hat{R}_i^{st})}^{st})$, where $(1 \leq i \leq m)$, trivially return consistent triangle relations.

For the predicate **Cas**, the proof is less straightforward. First, it is true that any pair of triangles $T^{st}$ and $T^{st\prime}$ such that $T^{st} =_\triangle T^{st\prime}$ have the same center of mass. Note that this center of mass, which is represented by a degenerated triangle, only has one representation. Second, all corner points representing a spatio-temporal triangle are co-temporal. Therefor, we can conclude that the cross-ratio of the time coordinates of three triangles $T_1^{st}$, $T_2^{st}$ and $T_3^{st}$ is the same as the cross-ratio of the time coordinates of any triple of triangles $T_1^{st\prime}$, $T_2^{st\prime}$ and $T_3^{st\prime}$, such that $T_l^{st} =_\triangle T_l^{st\prime}(1 \leq l \leq 3)$. It now follows from the first and second statements, that given the spatio-temporal triangles $T_1^{st}$, $T_2^{st}$, $T_3^{st}$, $T_4^{st}$, $T_5^{st}$ and $T_6^{st}$,

$$\textbf{Cas}(T_1^{st}, T_2^{st}, T_3^{st}, T_4^{st}, T_5^{st}, T_6^{st}) \leftrightarrow \textbf{Cas}(T_1^{st\prime}, T_2^{st\prime}, T_3^{st\prime}, T_4^{st\prime}, T_5^{st\prime}, T_6^{st\prime}),$$

for any $T_l^{st\prime}$ such that $T_l^{st} =_\triangle T_l^{st\prime}(1 \leq l \leq 6)$.

Now we have to prove that the composed formulas always return consistent triangle relations. Let $\hat{\varphi}$ and $\hat{\psi}$ be two formulas in FO($\{\textbf{PartOf}^{\text{Cotemp}}, \textbf{Before}_\triangle, \textbf{Cas}\}$, $\hat{\sigma}^{st}$), of arity $k_\varphi$ and $k_\psi$ respectively, already defining consistent triangle relations. Then, the formula $(\hat{\varphi} \wedge \hat{\psi})$ (resp., $(\hat{\varphi} \vee \hat{\psi})$) also defines a triangle relation. This follows from the fact that the free variables of $(\hat{\varphi} \wedge \hat{\psi})$ (resp., $(\hat{\varphi} \vee \hat{\psi})$) are free variables in $\hat{\varphi}$ or $\hat{\psi}$. The universe of all triangles is trivially consistent. If a consistent subset is removed from this universe, the remaining part is still consistent. Therefor, $\neg\hat{\varphi}$ is well-defined. Finally, because consistency is defined argument-wise, the projection $\exists T_1^{st}\, \hat{\varphi}(T_1^{st}, T_2^{st}, \ldots, T_{k_\varphi}^{st})$ is consistent. □

**Theorem 6.1 (Expressiveness of FO($\{\textbf{PartOf}^{\textbf{Cotemp}}, \textbf{Before}_\triangle, \textbf{Cas}\}$))** Let $\hat{\sigma}^{st}$ be a database schema. The language FO($\{\textbf{PartOf}^{\text{Cotemp}}, \textbf{Before}_\triangle, \textbf{Cas}\}$, $\hat{\sigma}^{st}$) is sound and complete for the $(\mathcal{AC}_{st}, \mathcal{A}_t)$-generic FO-queries on spatio-temporal triangle databases.

As usual, we prove this theorem using the following two lemma's:

**Lemma 6.2 (Soundness** FO($\{\textbf{PartOf}^{\textbf{Cotemp}}, \textbf{Before}_\triangle, \textbf{Cas}\}$) **)** Let $\hat{\sigma}^{st}$ be a spatio-temporal triangle database schema. Then FO($\{\textbf{PartOf}^{\text{Cotemp}}, \textbf{Before}_\triangle, \textbf{Cas}\}$, $\hat{\sigma}^{st}$) is sound for the $(\mathcal{AC}_{st}, \mathcal{A}_t)$-generic FO-queries on spatio-temporal triangle databases.

**Proof.** Let $\hat{\sigma}^{st} = \{\hat{R}_1^{st}, \hat{R}_2^{st}, \ldots, \hat{R}_m^{st}\}$ be a spatio-temporal triangle database schema. Similar to the proof of Lemma 5.2, this proof consists of two parts.

First, let $\dot{\sigma}^{st} = \{\dot{R}_1^{st}, \dot{R}_2^{st}, \ldots, \dot{R}_m^{st}\}$ be a spatio-temporal point database schema where the arity of $\dot{R}_i^{st}$ is $3\times ar(\hat{R}_i^{st})$, for $i = 1, 2, \ldots, m$. We show that each formula of FO($\{\mathbf{PartOf}^{\mathrm{Cotemp}}$, $\mathbf{Before}_\triangle, \mathbf{Cas}\}$, $\hat{\sigma}^{st}$) can be translated in FO($\{\mathbf{Between}^{\mathrm{Cotemp}}, \mathbf{Before}, \mathbf{EqCr}^{ST}\}$, $\dot{\sigma}^{st}$) . We to this by induction on FO($\{\mathbf{PartOf}^{\mathrm{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{Cas}\}$, $\hat{\sigma}^{st}$)-formulas. Next, we have to prove that each FO($\{\mathbf{PartOf}^{\mathrm{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{Cas}\}$, $\hat{\sigma}^{st}$)-query defines a consistent spatio-temporal triangle relation.

We start with the first part of this proof. Let $\dot{R}_i^{st}(1 \leq i \leq m)$ be the corresponding spatio-temporal point relation names of arity $3 \times ar(\hat{R}_i^{st})$ and let $\dot{\sigma}^{st}$ be the spatio-temporal (point) database schema $\{\dot{R}_1^{st}, \dot{R}_2^{st}, \ldots, \dot{R}_m^{st}\}$. Let $\hat{\varphi}$ be a FO($\{\mathbf{PartOf}^{\mathrm{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{Cas}$, $\hat{\sigma}^{st}\}$)-formula.

Each triangle variable $\triangle^{st}$ in $\hat{\varphi}$ is translated naturally by three spatio-temporal point variables $u_1, u_2, u_3$. As we assume that all points composing a spatio-temporal triangle are co-temporal, we add the formula

$$\mathbf{Cotemp}(u_1, u_2) \wedge \mathbf{Cotemp}(u_2, u_3)$$

to the beginning of the translation of the sub-formula where $\triangle^{st}$ appears first. In the remainder of this proof we will omit these temporal constraints to keep formulas shorter and hence more readable, but always assume them.

The formulas in FO($\{\mathbf{PartOf}^{\mathrm{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{Cas}\}$, $\hat{\sigma}^{st}$) are build from atomic formulas, composed by the operators $\wedge$, $\wedge$ and $\neq$ and quantification. The atomic formulas of FO($\{\mathbf{PartOf}^{\mathrm{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{Cas}\}$, $\hat{\sigma}^{st}$) are equality constraints between spatio-temporal triangle variables, the triangle predicates $\mathbf{PartOf}^{\mathrm{Cotemp}}$, $\mathbf{Before}_\triangle$ and $\mathbf{Cas}$ applied to spatio-temporal triangle variables, and predicates of the form $\hat{R}_i^{st}(\triangle_1^{st}, \triangle_2^{st}, \ldots, \triangle_{ar(\hat{R}_i^{st})}^{st})(1 \leq i \leq m)$, where $\hat{R}_i^{st} \in \hat{\sigma}^{st}$. As this proof is analogous to the proof of Lemma 5.2, we only give the translation of the atomic formulas:

(i) The translation of $(\triangle_1^{st}{=}\triangle_2^{st})$ is

$$\bigvee_{\sigma(1,2,3)=(i_1,i_2,i_3),\sigma\in\mathcal{S}_3} (u_{1,1} = u_{2,i_1} \wedge u_{1,2} = u_{2,i_2} \wedge u_{1,3} = u_{2,i_3}),$$

where $\mathcal{S}_3$ is the set of all permutations of $\{1, 2, 3\}$.

(ii) In the proof of Lemma 5.2, we already showed that the predicate **PartOf** can be expressed in FO($\{\mathbf{Between}\}$).

(iii) Expressions of the form $\mathbf{Before}_\triangle(\triangle_1^{st}, \triangle_2^{st})$ are translated as follows:

$$\mathbf{Before}(u_{1,1}, u_{2,1}).$$

Recall that the formulas expressing that the corner points of each triangle should be co-temporal are already added to the translation.

(iv) For the predicate **Cas**, first we need to to express in FO($\{\mathbf{Between}^{\mathrm{Cotemp}}, \mathbf{Before}$, $\mathbf{EqCr}^{ST}\}$, $\dot{\sigma}^{st}$) that some point (in $(\mathbb{R}^2 \times \mathbb{R})$) is the center of mass of a triangle, represented by three other points, all co-temporal with the first point.

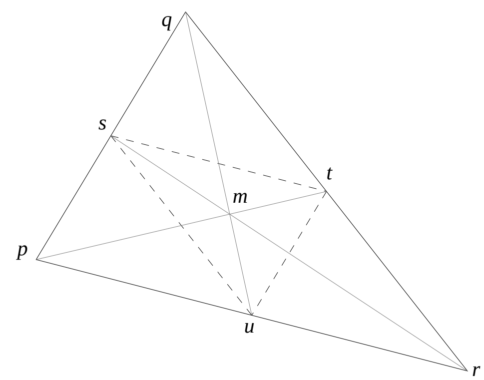


Figure 8: The center of mass of the triangle $pqr$ is the intersection of the medians $pt$, $qu$ and $rs$. Also, the lines $tu$, $us$ and $st$ are parallel to $pq$, $qr$ and $rp$, respectively.

Figure 8 illustrates the construction of the center of mass of a triangle. Given a triangle $T^{st}_{pqr}$. There is only one way of constructing a triangle $T^{st}_{stu}$ inscribed in $T^{st}_{pqr}$ such that each side of $T^{st}_{stu}$ is parallel to a side of $T^{st}_{pqr}$. The corner points of $T^{st}_{stu}$ are in the middle of the sides of $T^{st}_{pqr}$. Hence, the center of mass of $T^{st}_{pqr}$ is the intersection of the line segments connecting the corner points of $stu$ with the opposite corner point of $T^{st}_{pqr}$. The next formula expresses the predicate **CenterOM** in the language FO({$\mathbf{Between}^{\mathrm{Cotemp}}$, **Before**, $\mathbf{EqCr}^{ST}$}). The free variables are $v$ (representing the center of mass), $u_1$, $u_2$ and $u_3$ (representing the corner points of the triangle).

$$\begin{aligned}&\exists w_1\,\exists w_2\,\exists w_3\,(\mathbf{Between}^{\mathrm{Cotemp}}(u_1,w_1,u_2)\wedge\mathbf{Between}^{\mathrm{Cotemp}}(u_2,w_2,u_3)\wedge\\&\quad\mathbf{Between}^{\mathrm{Cotemp}}(u_3,w_3,u_1)\wedge\mathbf{Par}(u_1,u_2,w_2,w_3)\wedge\mathbf{Par}(u_2,u_3,w_1,w_3)\wedge\\&\qquad\mathbf{Par}(u_3,u_1,w_1,w_2)\wedge\mathbf{Between}^{\mathrm{Cotemp}}(u_1,v,w_2)\wedge\\&\qquad\quad\mathbf{Between}^{\mathrm{Cotemp}}(u_2,v,w_3)\wedge\mathbf{Between}^{\mathrm{Cotemp}}(u_3,v,w_1)).\end{aligned}$$

Here, $\mathbf{Par}(v_1,v_2,v_3,v_4)$ is an abbreviation for the sub-formula

$$\neg\exists w\,(\mathbf{Collinear}(w,v_1,v_2)\wedge\mathbf{Collinear}(w,v_3,v_4)).$$

We now give the expression translating $\mathbf{Cas}(\triangle^{st}_1,\triangle^{st}_2,\triangle^{st}_3,\triangle^{st}_4,\triangle^{st}_5,\triangle^{st}_6)$. The following formula has $(6\times 3)$ free point variables $u_{1,1}u_{1,2},u_{1,3},u_{2,1},u_{2,2},u_{2,3},\ldots,u_{6,1},u_{6,2},u_{6,3}$ that are the translation of the triangle variables $\triangle^{st}_1,\triangle^{st}_2,\ldots,\triangle^{st}_6$.

$$\begin{aligned}&\exists v_1\,\exists v_2\,\exists v_3\,(\\&\qquad\bigwedge_{i=1}^{3}\mathbf{CenterOM}(v_i,u_{i,1},u_{i,2},u_{i,3})\wedge\mathbf{EqCr}^{ST}(v_1,v_2,v_3,u_{4,1},u_{5,1},u_{6,1})).\end{aligned}$$

(v) The translation of a formula of the type $\hat{R}^{st}(\triangle^{st}_1,\triangle^{st}_2,\ldots,\triangle^{st}_k)$, where $\hat{R}^{st}\in\hat{\sigma}^{st}$ is

$$\dot{R}(u_{1,1},u_{1,2},u_{1,3},u_{2,1},u_{2,2},u_{2,3},\ldots,u_{k,1},u_{k,2},u_{k,3}).$$

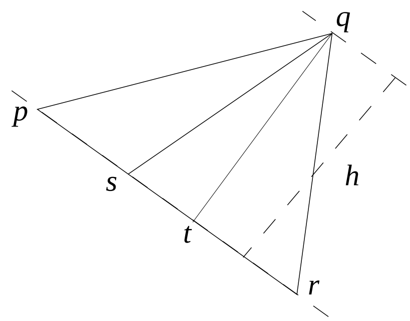


Figure 9: The area of $T_{pqr}$ is to the area of $T_{pqs}$ as the length of $pr$ to the length of $ps$.

The correctness of this translation follows from Definition 4.2 and Remark 4.1. □

We can also show the possibility of the translation in the other direction. As the proof of Lemma 6.3 is completely analogous to the proof of Lemma 5.3, we omit it. The only new items are the translations of the spatio-temporal point predicates **Before** and $\mathbf{EqCr}^{ST}$ into $\mathrm{FO}(\{\mathbf{PartOf}^{\mathrm{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{Cas}\})$. It is easy to see that these translations involve only replacing point variables by triangle variables that represent points.

**Lemma 6.3 (Completeness of $\mathbf{FO}(\{\mathbf{PartOf}^{\mathbf{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{Cas}\})$)** Let $\hat{\sigma}^{st}$ be a spatio-temporal triangle database schema. The language $\mathrm{FO}(\{\mathbf{PartOf}^{\mathrm{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{Cas}\}, \hat{\sigma}^{st})$ is complete for the $(\mathcal{AC}_{st}, \mathcal{A}_t)$-generic FO-queries on spatio-temporal triangle databases. □

We now propose an alternative language, with the same expressiveness as the language $\mathrm{FO}(\{\mathbf{PartOf}^{\mathrm{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{Cas}\})$, which allows us to talk about areas of triangles.

### 6.1.2 Expressiveness of the Language $\mathrm{FO}(\{\mathbf{PartOf}^{\mathbf{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{Lex}\})$

We start this subsection with some geometric constructions. We will use those to express the predicate **Lex** in the language $\mathrm{FO}(\{\mathbf{Between}^{\mathrm{Cotemp}}, \mathbf{Before}, \mathbf{EqCr}^{ST}\})$. For these constructions, we assume that all spatio-temporal points and triangles are co-temporal.

**Observation 6.1** Let two triangles $T^{st}_{pqr}$ and $T^{st}_{pqs}$ be given. If the point $s$ is chosen on the line segment $pr$ such that the cross ration of $p$, $s$ and $r$ equals $c$, then the areas of $T^{st}_{pqr}$ and $T^{st}_{pqs}$ have a ratio which is also equal to $c$. The correctness of this construction is easy to verify because the area of a triangle is half the length of its base line multiplied by its height. As $T^{st}_{pqr}$ and $T^{st}_{pqs}$ have both height $h$, their areas have the same relation as the lengths of their base lines $ps$ and $pq$. Figure 9 illustrates this observation.

Suppose we have three triangles $T^{st}_{pqr}$, $T^{st}_{pqs}$ and $T^{st}_{pqt}$, such that the points $q$, $r$, $s$ and $t$ are all collinear (suppose they are arranged as in Figure 9). Then it is true that

$$\frac{A(T^{st}_{pqt}) - A(T^{st}_{pqs})}{A(T^{st}_{pqr}) - A(T^{st}_{pqs})} = \frac{|\ st\ |}{|\ sr\ |}.$$

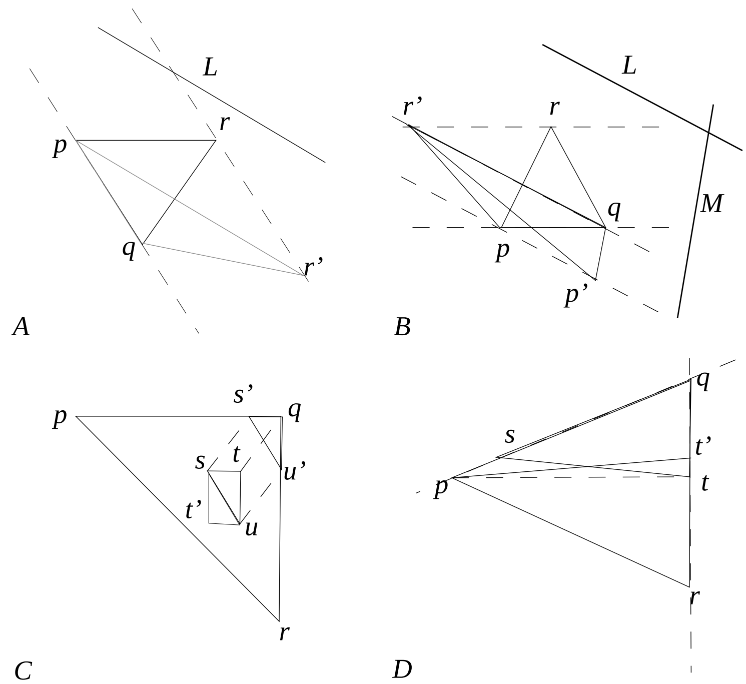


Figure 10: Area-preserving affine-invariant constructions.

So it turns out to be possible to convert area ratios to cross-ratios of collinear points, for triangles that have the special configuration as described in Observation 6.1. We will observe next that it is possible, given three triangles $T_1^{st}$, $T_2^{st}$ and $T_3^{st}$ such that $T_1^{st}$ is part of $T_2^{st}$ and $T_2^{st}$ part of $T_3^{st}$, to construct triangles $T_4^{st}$ and $T_5^{st}$ with the same area as $T_1^{st}$ and $T_2^{st}$, respectively, such that $T_4^{st}$, $T_5^{st}$ and $T_3^{st}$ have this special configuration.

**Observation 6.2** Given a pair of triangles $T_1^{st}$ and $T_2^{st}$ such that $T_1^{st}$ is part of $T_2^{st}$. Following the construction steps described below, we can construct a triangle $T_3^{st}$, with the same area as $T_1^{st}$. The triangle $T_3^{st}$ shares one side with $T_2^{st}$ and its third corner point is on one of the other sides of $T_2^{st}$.

*Construction step 1:*

Given a triangle $T_{pqr}^{st}$ and a line $L$, we construct a triangle with the same area as $T_{pqr}^{st}$, but one side parallel to $L$. We do this by moving the point $r$ over the line through $r$ and parallel with $pq$ until one of the line segments $pr$ or $qr$ is parallel to $L$. The resulting triangle $T_{pqr'}^{st}$ has the same area as $T_{pqr}^{st}$ because it has the same base line segment and the same height as $T_{pqr}^{st}$. Figure 10, part $A$, illustrates this construction.

If we apply this construction twice, we can construct a triangle with two sides parallel to two given (different) lines. This is shown in Figure 10, part $B$, where the triangle $T_{pqr}^{st}$ is first transformed into $T_{pqr'}^{st}$ and, in a second step, into $T_{p'qr'}^{st}$.

*Construction Step 2:*

Let a triangle $T^{st}_{pqr}$ and a smaller triangle, either $T^{st}_{stu}$ or $T^{st}_{st'u}$, which has two sides parallel to the sides $pq$ and $qr$ respectively of $T^{st}_{pqr}$, be given. There are two possible orientations for the smaller triangle. Either it is oriented in such a way that the corner point $t'$ is on the opposite side of $su$ than the point $q$, as is the case for triangle $T^{st}_{st'u}$ in Figure 10, part C, or it is oriented otherwise, as is the case for triangle $T^{st}_{stu}$. In the first case, we *flip* $T^{st}_{st'u}$ by constructing the parallelogram $st'ut$, and then considering the triangle $T^{st}_{stu}$.

Next, starting from a triangle $T^{st}_{stu}$ with the right orientation, we construct a triangle $T^{st}_{qs'u'}$ which has the same area as $T^{st}_{stu}$, but shares a corner point with $T^{st}_{pqr}$ and has its other corner points on the two sides of $T^{st}_{pqr}$, adjacent to the common corner point. This transformation involves only a translation, which can be carried out by constructing a set of parallel lines.

*Construction Step 3:*

Given a triangle $T^{st}_{pqr}$, and a triangle $T^{st}_{sqt}$ such that $s$ lies on the line through $pq$ and $t$ lies on the line through $qr$. We can construct a triangle $T^{st}_{pqt'}$ that has the same area as $T^{st}_{sqt}$ by making sure that the cross-ratio of the points $p$, $s$ and $q$ equals the cross-ratio of the points $t$, $t'$ and $q$. Figure 10, part $D$, illustrates this construction.

Using the above three steps, we constructed, starting from two arbitrary triangles, one being part of the other, two triangles that have the desired configuration.

We now can prove that our alternative language, FO($\{\mathbf{PartOf}^{\text{Cotemp}}, \mathbf{Before}, \mathbf{Lex}\}$) also is sound and complete for the $(\mathcal{AC}_{st}, \mathcal{A}_t)$-generic FO-queries on triangle databases. As the proof is completely analog as the proof of Theorem 6.1, except for the translations of the predicates **Lex** and $\mathbf{EqCr}^{ST}$, we only give those translations.

**Theorem 6.2 (Expressiveness of FO($\{\mathbf{PartOf}^{\mathbf{Cotemp}}, \mathbf{Before}_\Delta, \mathbf{Lex}\}$))** Let $\hat{\sigma}^{st}$ be a spatio-temporal triangle database schema. The language FO($\{\mathbf{PartOf}^{\text{Cotemp}}, \mathbf{Before}_\Delta, \mathbf{Lex}\}$, $\hat{\sigma}^{st}$) is sound and complete for the $(\mathcal{AC}_{st}, \mathcal{A}_t)$-generic FO$(+, \times, <, 0, 1)$-queries on spatio-temporal triangle databases.

**Proof.** First, let $\hat{\sigma}^{st} = \{\hat{R}^{st}_1, \hat{R}^{st}_2, \ldots, \hat{R}^{st}_m\}$ be a spatio-temporal triangle database schema and let $\dot{\sigma}^{st} = \{\dot{R}^{st}_1, \dot{R}^{st}_2, \ldots, \dot{R}^{st}_m\}$ be a spatio-temporal point database schema where the arity of $\dot{R}^{st}_i$ is $3 \times ar(\hat{R}^{st}_i)$, for $i = 1, 2, \ldots, m$.

We first show that the predicate **Lex** can be expressed in FO($\{\mathbf{Between}^{\text{Cotemp}}, \mathbf{Before}, \mathbf{EqCr}^{ST}\}$, $\dot{\sigma}^{st}$) . We verify that this predicate is invariant for transformations in $(\mathcal{AC}_{st}, \mathcal{A}_t)$. The proportion of the areas of two co-temporal triangles is invariant under affinities. This, together with the fact that cross-ratios of time moments are invariant under affine transformations of the time, shows that the predicate **Lex** is $(\mathcal{AC}_{st}, \mathcal{A}_t)$-invariant.

The constructions described in Observation 6.2 can all be expressed in the language FO($\{\mathbf{Between}^{\text{Cotemp}}, \mathbf{Before}, \mathbf{EqCr}^{ST}\}$). They mainly involve parallelism-constraints on points.

Let **SameRelArea** be the abbreviation for a predicate in FO($\{\mathbf{Between}^{\text{Cotemp}}, \mathbf{Before}, \mathbf{EqCr}^{ST}\}$) of arity 11. The first nine free variables represent the corner points of three co-temporal triangles, such that the first triangle is part of the second, which is again part of the third triangle. The two last point variables are located on one side of the third triangle, in such a way that the parts they define of the third triangle (denoted triangle four and five),

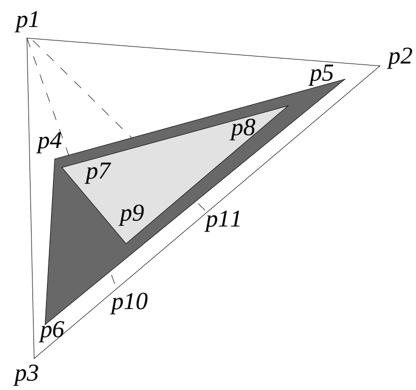


Figure 11: An illustration of the predicate **SameRelArea**. The expression **SameRelArea**$(p_1, p_2, \ldots, p_{11})$ will be true if and only if two conditions are met. First, the triangle with corner points $p_7$, $p_8$ and $p_9$ (the light shaded one) is part of the triangle with corner points $p_4p$, $p_5$ and $p_6$ (the dark shaded one), which is part of the triangle with corner points $p_1$, $p_2$ and $p_3$ (the white triangle). Second, The areas of the light and dark shaded triangles are to the area of the white triangle as the areas of the triangles with corner points $p_1$, $p_2$ and $p_{10}$, resp. $p_1$, $p_2$ and $p_{11}$ to the area of the white triangle.

are part of each other also. Finally, the proportion of the areas of the first three triangles is the same as the proportion of the areas of the fourth, fifth and third triangle. Fig 11 illustrates this predicate.

The translation of $\mathbf{Lex}(\triangle_1^{st}, \triangle_2^{st}, \ldots, \triangle_6^{st})$ then is the following expression:

$$\exists v_1\, \exists v_2\, (\mathbf{SameArea}(u_{1,1}, u_{1,2}, u_{1,3}, u_{2,1}, u_{2,2}, u_{2,3}, u_{3,1}, u_{3,2}, u_{3,3}, v_1, v_2) \wedge \\ \mathbf{EqCr}^{ST}(v_1, v_2, u_{3,3}, u_{4,1}, u_{5,1}, u_{5,2})),$$

if $\triangle_i^{st}$ is translated by $u_{i,1}$, $u_{i,2}$ and $u_{i,2}$ for $i = 1, \ldots, 6$.

The translation in the other direction is simpler. The formula $\mathbf{EqCr}^{ST}(u_1, u_2, u_3, u_4, u_5, u_6)$ can be expressed as

$$\exists\, \triangle_7^{st}\ \exists\, \triangle_8^{st}\ \exists\, \triangle_9^{st}\ \exists\, \triangle_{10}^{st}\ \exists\, \triangle_{11}^{st}\ ( \\ \mathbf{CornerP}(\triangle_7^{st}, \triangle_8^{st}, \triangle_1^{st}, \triangle_9^{st}) \wedge \mathbf{CornerP}(\triangle_7^{st}, \triangle_8^{st}, \triangle_2^{st}, \triangle_{10}^{st}) \\ \wedge \mathbf{CornerP}(\triangle_7^{st}, \triangle_8^{st}, \triangle_3^{st}, \triangle_{11}^{st}) \wedge \mathbf{Lex}(\triangle_9^{st}, \triangle_{10}^{st}, \triangle_{11}^{st}, \triangle_4^{st}, \triangle_5^{st}, \triangle_6^{st})).$$

□

## 6.2 Physics-based Classes

In the previous section, we investigated a triangle language for $(\mathcal{AC}_{st}, \mathcal{A}_t)$-generic triangle queries. Next, we focus on triangle languages for the physics-based queries, *i.e.*, those generic for the group $(\mathcal{V}_{st}, \mathcal{A}_t)$ of velocity-preserving transformations and the group $(\mathcal{AC}_{st}, \mathcal{A}_t)$ of acceleration-preserving transformations.

In [10], the query languages expressing queries generic for the physics-based transformation groups were found by starting with the languages expressing the affine-invariant spatial point queries. The reason was that the physics-based transformation groups of $(\mathbb{R}^2 \times \mathbb{R})$ are a subgroup of the affinities of $\mathbb{R}^3$, and that spatio-temporal points in $(\mathbb{R}^2 \times \mathbb{R})$ can be interpreted equally well as points in $\mathbb{R}^3$.

Here, it is not expedient to do so. We can see spatio-temporal triangles in $(\mathbb{R}^2 \times \mathbb{R})$ as convex objects in $\mathbb{R}^3$, but then the predicate **PartOf** would not make much sense, as spatio-temporal triangles can only overlap when they exist at the same moment in time. Another solution would be to choose other convex objects, that have a temporal extend of more than one time moment. But, these objects would make rather poor spatio-temporal objects. Indeed, even if all corner points of a triangle in $\mathbb{R}^2$ move with a linear function of time, this movement can result in a 3-dimensional object bounded by non-planar surfaces, and hence possibly not convex.

Therefor, we take another approach and start with the predicates $\mathbf{PartOf}^{\mathrm{Cotemp}}$ and $\mathbf{Before}_{\triangle}$, as in the previous section, and add other predicates until the resulting language is expressive enough. In concrete, this means that we have to be able to translate the point predicate **Between** in that language.

As $(\mathcal{V}_{st}, \mathcal{A}_t) \subset (\mathcal{AC}_{st}, \mathcal{A}_t)$, we start with the acceleration preserving transformations first, and later extend the language expressing all $(\mathcal{AC}_{st}, \mathcal{A}_t)$-generic queries in such a way we obtain a language expressing the $(\mathcal{V}_{st}, \mathcal{A}_t)$-generic queries.

### 6.2.1 $(\mathcal{AC}_{st}, \mathcal{A}_t)$-generic Queries

For the acceleration-preserving queries, we introduce the spatio-temporal triangle predicate **SAS** (which is an abbreviation for "Same Average Speed"). Let $\triangle_1^{st}, \triangle_2^{st}, \triangle_3^{st}$ and $\triangle_4^{st}$ be four triangles that have center of mass $p_i = (a_i, b_i, \tau_i), i = 1 \ldots 4$. Furthermore, $\tau_1 \leq \tau_2$ and $\tau_3 \leq \tau_4$. Then $\mathbf{SAS}(\triangle_1^{st}, \triangle_2^{st}, \triangle_3^{st}, \triangle_4^{st})$ is true if and only if

$$\frac{a_2 - a_1}{\tau_2 - \tau_1} = \frac{a_4 - a_3}{\tau_4 - \tau_3} \text{ and } \frac{b_2 - b_1}{\tau_2 - \tau_1} = \frac{b_4 - b_3}{\tau_4 - \tau_3}.$$

In other words, the movement from $\triangle_1^{st}$ to $\triangle_2^{st}$ has the same average speed, in both $x$- and $y$-direction, as the movement from $\triangle_3^{st}$ to $\triangle_4^{st}$.

We now show that the language $\mathrm{FO}(\{\mathbf{PartOf}^{\mathrm{Cotemp}}, \mathbf{Before}_{\triangle}, \mathbf{SAS}\})$ is sound and complete for the $(\mathcal{AC}_{st}, \mathcal{A}_t)$-generic FO(+, ×, <, 0, 1)-queries on triangle databases.

As soundness and completeness proof are completely analogous to those of the previous section, we only give the translations of the triangle predicates from the set $\{\mathbf{PartOf}^{\mathrm{Cotemp}}, \mathbf{Before}_{\triangle}, \mathbf{SAS}\}$ into $\mathrm{FO}(\{\mathbf{Between}\})$ and of the point predicate **Between** into the logic $\mathrm{FO}(\{\mathbf{PartOf}^{\mathrm{Cotemp}}, \mathbf{Before}_{\triangle}, \mathbf{SAS}\})$.

**Theorem 6.3 (Expressiveness of $\mathbf{FO}(\{\mathbf{PartOf}^{\mathbf{Cotemp}}, \mathbf{Before}_{\triangle}, \mathbf{SAS}\})$)** Let $\hat{\sigma}^{st}$ ba a triangle database schema. Let $\overline{\sigma}^{st}$ be the corresponding semi-algebraic database schema. The language $\mathrm{FO}(\{\mathbf{PartOf}^{\mathrm{Cotemp}}, \mathbf{Before}_{\triangle}, \mathbf{SAS}\}, \hat{\sigma}^{st})$ is sound and complete for the $(\mathcal{AC}_{st}, \mathcal{A}_t)$-generic FO(+, ×, <, 0, 1, $\overline{\sigma}^{st}$)-queries on triangle databases over $\hat{\sigma}^{st}$.

**Proof sketch.** Let $\hat{\sigma}^{st} = \{\hat{R}_1^{st}, \hat{R}_2^{st}, \ldots, \hat{R}_m^{st}\}$ be a spatial triangle database schema. Let $\dot{R}_i^{st}, 1 \leq i \leq m$ be the corresponding spatial point relation names of arity $3 \times ar(\hat{R}_i^{st})$

and let $\dot{\sigma}^{st}$ be the spatial database schema $\{\dot{R}_1^{st}, \dot{R}_2^{st}, \ldots, \dot{R}_m^{st}\}$. Let $\overline{R}_i^{st}, 1 \leq i \leq m$ be the corresponding constraint relation names of arity $6 \times ar(\dot{R}_i^{st})$ and let $\overline{\sigma}^{st}$ be the spatial database schema $\{\overline{R}_1^{st}, \overline{R}_2^{st}, \ldots, \overline{R}_m^{st}\}$.

In this proof sketch, we only give the translation of **SAS** into FO($\{$**Between**$\}$, $\dot{\sigma}^{st}$). For the translations of **PartOf**$^{\text{Cotemp}}$ and **Before**$_\triangle$, see Section 5 and Section 6.1 respectively.

Given the expression **SAS**($\triangle_1^{st}, \triangle_2^{st}, \triangle_3^{st}, \triangle_4^{st}$). The following formula is its translation into FO($\{$**Between**$\}$, $\dot{\sigma}^{st}$):

$$
\begin{aligned}
&\exists v_1\, \exists v_2\, \exists v_3\, \exists v_4\, ( \\
&\qquad \bigwedge_{i=1}^{4} \mathbf{CenterOM}(v_i, u_{i,1}, u_{i,2}, u_{i,3}) \wedge \mathbf{Before}(v_1, v_2) \wedge \mathbf{Before}(v_3, v_4) \\
&\quad \wedge \mathbf{CoPlanar}(v_1, v_2, v_3, v_4) \wedge \neg\exists w\, (\mathbf{Collinear}(w, v_1, v_2) \wedge \mathbf{Collinear}(w, v_3, v_4))).
\end{aligned}
$$

We have omitted the sub formulas expressing that the corner points of a triangle should be co-temporal. The predicate **CoPlanar** expresses that four 3-dimensional points are co-planar. It is clear that this is an affine invariant and FO-expressible.

For the definition of **CenterOM**, we refer to the proof of Lemma 6.2.

We next prove that the predicate **Between** can be expressed in FO($\{$**PartOf**$^{\text{Cotemp}}$, **Before**$_\triangle$, **SAS**$\}$, $\hat{\sigma}^{st}$). This translation is not complicated. If the expression **Between**($p, q, r$) holds for three points $p$, $q$ and $r$, then either they are all co-temporal or they all exist at a different time moment. In the first case, we can translate **Between** using **PartOf**, as we showed in the proof of Lemma 5.3. If they all have a different time coordinate, we can express that $q$ is between $p$ and $r$ using **SAS**:

$$
\begin{aligned}
&(\mathbf{CoTemp}(\triangle_1^{st}, \triangle_2^{st}) \wedge \mathbf{CoTemp}(\triangle_2^{st}, \triangle_3^{st}) \wedge \\
&\qquad \mathbf{Between}_\Delta(\triangle_p, \triangle_q, \triangle_r)) \vee (\mathbf{SAS}(\triangle_1^{st}, \triangle_2^{st}, \triangle_2^{st}, \triangle_3^{st})).
\end{aligned}
$$

In the previous formula, we have omitted the sub-formulas expressing that the triangles translating the point variables should be points. □

Since the group $(\mathcal{V}_{st}, \mathcal{A}_t)$ is a subgroup of the group $(\mathcal{AC}_{st}, \mathcal{A}_t)$, we use our knowledge from this subsection to extend the language FO($\{$**PartOf**$^{\text{Cotemp}}$, **Before**$_\triangle$, **SAS**$\}$, $\hat{\sigma}$), which we will do next.

### 6.2.2 $(\mathcal{V}_{st}, \mathcal{A}_t)$-generic Queries

In this subsection, we propose a language sound and complete of the first-order $(\mathcal{V}_{st}, \mathcal{A}_t)$-generic triangle queries. We add the element **NoSp** (an abbreviation for "No Speed") to the set $\{$**PartOf**$^{\text{Cotemp}}$, **Before**$_\triangle$, **SAS**$\}$.

Suppose two spatio-temporal triangles $T_1^{st}$ and $T_2^{st}$ have center of mass $p_i = (a_i, b_i, \tau_i)$, $i = 1, 2$. If we furthermore assume that $\tau_1 \leq \tau_2$, then **NoSp**($\triangle_1^{st}, \triangle_2^{st}$) is true if and only if $a_1 = a_2$ and $b_1 = b_2$. In other words, the average speed is zero, both triangles are on the same position.

We now show that the language FO($\{\mathbf{PartOf}^{\text{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{SAS}, \mathbf{NoSp}\}$) is sound and complete for the $(\mathcal{AC}_{st}, \mathcal{A}_t)$-generic FO(+, ×, <, 0, 1)-queries on triangle databases.

As soundness and completeness proof are completely analogous to those of the previous section, we only give the new translations.

**Theorem 6.4 (Expressiveness of FO($\{\mathbf{PartOf}^{\mathbf{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{SAS}, \mathbf{NoSp}\}$))** Let $\hat{\sigma}^{st}$ be a triangle database schema. Let $\overline{\sigma}^{st}$ be the corresponding semi-algebraic database schema. Then the language FO($\{\mathbf{PartOf}^{\text{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{SAS}, \mathbf{NoSp}\}$, $\hat{\sigma}^{st}$) is sound and complete for the $(\mathcal{AC}_{st}, \mathcal{A}_t)$-generic FO(+, ×, <, 0, 1, $\overline{\sigma}^{st}$)-queries on triangle databases over $\hat{\sigma}^{st}$.

**Proof sketch.** Let $\hat{\sigma}^{st} = \{\hat{R}_1^{st}, \hat{R}_2^{st}, \ldots, \hat{R}_m^{st}\}$ be a spatial triangle database schema. Let $\dot{R}_i^{st}$, $1 \leq i \leq m$ be the corresponding spatial point relation names of arity $3 \times ar(\hat{R}_i^{st})$ and let $\dot{\sigma}^{st}$ be the spatial database schema $\{\dot{R}_1^{st}, \dot{R}_2^{st}, \ldots, \dot{R}_m^{st}\}$. Let $\overline{R}_i^{st}, 1 \leq i \leq m$ be the corresponding constraint relation names of arity $6 \times ar(\hat{R}_i^{st})$ and let $\overline{\sigma}^{st}$ be the spatial database schema $\{\overline{R}_1^{st}, \overline{R}_2^{st}, \ldots, \overline{R}_m^{st}\}$.

In this proof sketch, we only give the translation of the predicate **NoSp** into the language FO($\{\mathbf{Between}, \mathbf{Before}, \mathbf{EqSpace}\}$) and of the predicate **EqSpace** into the language FO($\{\mathbf{PartOf}^{\text{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{SAS}, \mathbf{NoSp}\}$).

The next formula, with free variables $u_1, u_2, u_3, v_1, v_2, v_3$ is the translation of $\mathbf{NoSp}(\triangle_u, \triangle_v)$ into FO($\{\mathbf{Between}, \mathbf{Before}, \mathbf{EqSpace}\}$, $\hat{\sigma}$).

$$\exists w_1 \exists w_2 (\mathbf{CenterOM}(w_1, u_1, u_2, u_3) \wedge \mathbf{CenterOM}(w_2, v_1, v_2, v_3) \wedge \mathbf{EqSpace}(w_1, w_2)).$$

Finally, the formula

$$\mathbf{Point}(\triangle_u^{st}) \wedge \mathbf{Point}(\triangle_v^{st}) \wedge \mathbf{NoSp}(\triangle_u^{st}, \triangle_v^{st})$$

translates $\mathbf{EqSpace}(u, v)$ into FO($\{\mathbf{PartOf}^{\text{Cotemp}}, \mathbf{Before}_\triangle, \mathbf{SAS}, \mathbf{NoSp}\}$). Note that, if a triangle is degenerated into a point, its center of mass is equal to the triangle itself. □

We end with a note on safety of spatio-temporal triangle database queries.

## 6.3 Safety of Spatio-temporal Triangle Database Queries

In Section 5.2, we addressed the safety-problem for spatial triangle queries. In the spatial case, we defined a query to be safe when it returns a finite number of triangles on an input consisting of a finite number of triangles. Due to our choice of not considering convex objects in $(2+1)$-dimensional space but spatio-temporal triangles as basic objects for our language (see Remark 5.5 and the start of Section 6.2), this definition does not carry over to the spatio-temporal case. Indeed, it would be very unnatural to consider spatio-temporal databases containing a finite number of spatio-temporal triangles only.

It follows from a well-known property of semi-algebraic sets that there exists a finite partition of the time domain of a spatio-temporal database in points and open intervals such that within such an interval all snapshots are isotopic to each other and there exists a continuous family of homeomorphisms mapping these snapshots to each other (this is explained in more detail in [15]). So, spatio-temporal databases that are semi-algebraic sets can in fact be considered "finite" spatio-temporal databases in general. However, given a spatio-temporal relation $R$, a formula in FO($+$, $\times$, $<$, 0, 1, $R$) that expresses this partition for $R$ does not exist. The partition can be computed by performing a CAD (Cylindrical Algebraic Decomposition) [8].

A desirable property for a "finite" spatio-temporal triangle database, would be that every snapshot of the spatio-temporal database can be represented using a finite number of spatio-temporal triangles. This essentially is the requirement that each snapshot would be a finite spatial triangle relation. It is easy to see that we can express this requirement using $\mathbf{PartOf}^{\mathrm{Cotemp}}$, using the results of Section 5.2.

We can conclude that the safety problem for spatio-temporal triangle databases is strongly related to the safety problem for spatial triangle databases. Because we do not consider real spatio-temporal objects as basic objects for our language and as basic elements of spatio-temporal triangle databases, we can only ask that each snapshot of a spatio-temporal triangle database is finite.

# 7 Conclusion

In this article, we introduced the new triangle-based query language FO($\{\mathbf{PartOf}\}$). The use of triangles instead of points or real numbers is motivated by the spatial (spatio-temporal) practice, where data is often represented as a collection of (moving) triangles.

We showed that our query language has the same expressiveness as the affine-invariant FO($\{\mathbf{Between}\}$)-queries on triangle databases. We did this by showing that our language is sound and complete for the FO($\{\mathbf{Between}\}$)-queries on triangle databases.

Afterwards, we gave several examples to illustrate the expressiveness of the triangle-based language and the ease of use of manipulating triangles.

We then turned to the notion of safety. We showed that, although we cannot decide whether a particular Tquery returns a finite output given a finite input, we can decide whether the output is finite. We also extended this finiteness to the more intuitive notion of sets that have a finite representation. We proved that we can decide whether the output of a query has a finite representation and compute such a finite representation in FO($\{\mathbf{PartOf}\}$).

Besides the intuitive manipulation of spatial data represented as a collection of triangles, another motivation for this language is that it can serve as a first step towards a natural query language for spatio-temporal data that are collections of *moving triangles*.

Geerts, Haesevoets and Kuijpers [10] already proposed point-based languages for several classes of spatio-temporal queries. The data model used there represented a moving two-dimensional object as a collection of points in three-dimensional space. There exist however, data models that represent spatio-temporal data as a collection of moving objects (see for example [6, 7]), which is more natural. Hence, a *moving triangle*-based language with the same expressiveness as the spatio-temporal point languages mentioned above would be much more useful in practice.

# Acknowledgements

The authors would like to thank Jan Van den Bussche and Floris Geerts for discussions that have given rise to a more concise description of the proposed query languages.

# References


[1] M. Aiello and J. van Benthem. Logical patterns in space. In D. Barker-Plummer, D. Beaver, J. van Benthem, and P. Scotto di Luzio, editors, *Words, Proofs, and Diagrams*, pages 5–25. CSLI, 2002.

[2] M. Aiello and J. van Benthem. A modal walk through space. *Journal of Applied Non-Classical Logics*, 12(3-4):319–363, 2002.

[3] M. Benedikt and L. Libkin. Safe constraint queries. *SIAM Journal on Computing*, 29(5):1652–1682, 2000.

[4] M. De Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry*. Springer-Verlag, 2000.

[5] J. Bochnak, M. Coste, and M.F. Roy. *Géométrie Algébrique Réelle*. Springer-Verlag, Berlin, 1987.

[6] C. X. Chen and C. Zaniolo. SQLST: A spatio-temporal data model and query language. In A. H. F. Laender, S. W. Liddle, and V. C. Storey, editors, *Conceptual Modeling, 19th International Conference on Conceptual Modeling (ER'00)*, volume 1920 of *Lecture Notes in Computer Science*, pages 96–111. Springer-Verlag, 2000.

[7] J. Chomicki, S. Haesevoets, B. Kuijpers, and P. Revesz. Classes of spatiotemporal objects and their closure properties. *Annals of Mathematics and Artificial Intelligence*, 39(4):431–461, 2003.

[8] G.E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In H. Brakhage, editor, *Automata Theory and Formal Languages*, volume 33 of *Lecture Notes in Computer Science*, pages 134–183, Berlin, 1975. Springer-Verlag.

[9] M. Egenhofer and J. Herring. A mathematical framework for the definition of topological relationships. In K. Brassel and H. Kishimoto, editors, *Proceedings of the Fourth International Symposium on Spatial Data Handling*, pages 803–813, 1990.

[10] F. Geerts, S. Haesevoets, and B. Kuijpers. First-order complete and computationally complete query languages for spatio-temporal databases. *ACM Transactions on Computational Logic*, 9(2), 2008.

[11] M. Gyssens, J Van den Bussche, and D Van Gucht. Complete geometric query languages. *Journal of Computer and System Sciences*, 58(3):483–511, 1999.

[12] S. Haesevoets and B. Kuijpers. Time-dependent affine triangulation of spatio-temporal data. In D. Pfoser, I. F. Cruz, and M. Ronthaler, editors, *Proceedings of the 12th ACM International Workshop on Geographic Information Systems*, pages 57–66. ACM, 2004.

[13] M. Hagedoorn and R. C. Veldkamp. Reliable and efficient pattern matching using an affine invariant metric. *International Journal of Computer Vision*, 31:203–225, 1999.

[14] D.P. Huttenlocher, G.A. Klauderman, and W.J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:850–863, 1998.

[15] B. Kuijpers, J. Paredaens, and D. Van Gucht. Towards a theory of movie database queries. In *Proceedings of the 7th International Workshop on Temporal Representation and Reasoning*, pages 95–102. IEEE Computer Society Press, 2000.

[16] Bart Kuijpers, Jan Paredaens, and Jan Van den Bussche. On topological elementary equivalence of spatial databases. In Foto Afrati and Phokion Kolaitis, editors, *Proceedings of the 6th International Conference on Database Theory (ICDT'97)*, volume 1186 of *Lecture Notes in Computer Science*, pages 432–446. Springer-Verlag, 1997.

[17] Bart Kuijpers and Marc Smits. On expressing topological connectivity in spatial datalog. In Volker Gaede, Alexander Brodsky, Oliver Günther, Divesh Srivastava, Victor Vianu, and Mark Wallace, editors, *Constraint Databases and Their Applications (CDB'97)*, volume 1191 of *Lecture Notes in Computer Science*, pages 116–133. Springer-Verlag, 1997.

[18] Y. Lamdan, J.T. Schwartz, and H.J. Wolfson. Affine-invariant model-based object recognition. *IEEE Journal of Robotics and Automation*, 6:578–589, 1990.

[19] R. Laurini and D. Thompson. *Fundamentals of Spatial Information Systems*. Number 37 in APIC Series. Academic Press, 1992.

[20] G. Nielson. A characterization of an affine invariant triangulation. In G. Farin, H. Hagen, and H. Noltemeier, editors, *Geometric Modelling, Computing Supplementum 8*, pages 191–210, 1993.

[21] C.H. Papadimitriou, D. Suciu, and V. Vianu. Topological queries in spatial databases. In *Proceedings of the 15th ACM Symposium on Principles of Database Systems*, pages 81–92. ACM Press, 1996.

[22] J. Paredaens, J. Van den Bussche, and D. Van Gucht. Towards a theory of spatial database queries. In *Proceedings of the 13th ACM Symposium on Principles of Database Systems*, pages 279–288, New York, 1994. ACM Press.

[23] J. Paredaens, G. Kuper, and L. Libkin, editors. *Constraint databases*. Springer-Verlag, 2000.

[24] P. Revesz. *Introduction to Constraint Databases*. Springer-Verlag, 2002.

[25] L.G. Roberts. Machine perception of three-dimensional solids. *J.T. Tippet, editor, Optical and Electro-optical Information Processing*, 1965.