

On the Complexity of the Realisability Problem for Visit Events in Trajectory Sample Databases

Arthur Jansen¹  

Hasselt University, Databases and Theoretical Computer Science Group and
Data Science Institute (DSI), Agoralaan, Building D, 3590 Diepenbeek, Belgium

Bart Kuijpers  

Hasselt University, Databases and Theoretical Computer Science Group and
Data Science Institute (DSI), Agoralaan, Building D, 3590 Diepenbeek, Belgium

Abstract

Trajectory sample databases store finite sequences of measured space-time locations of moving objects, along with a speed bound for each object. These databases can be seen as uncertain databases. We propose a language that allows the formulation of queries about the uncertainty in trajectory sample databases. As part of that language, we introduce the notion of *visit events*, which are used to describe certain constraints on the movement of an object. In our language, an atomic query asks whether a moving object can, given its limitations, realise such an event. We give complexity results for this realisability problem, in various settings.

2012 ACM Subject Classification Information systems → Spatial-temporal systems; Information systems → Query languages

Keywords and phrases Trajectory sample databases, uncertain databases, query languages, complexity

Digital Object Identifier 10.4230/LIPIcs.TIME.2025.12

Funding *Arthur Jansen*: Bijzonder Onderzoeksfonds (BOF22OWB06) from UHasselt

1 Introduction

Due to the proliferation of location-aware devices (such as GPS receivers) in the past two decades, one of the use-cases of moving object databases [8] is the storage of time-stamped measured locations of moving objects [7]. Such a sequence of spatio-temporal measurements of a single moving object is called a *trajectory sample*. Given this type of partial information on a moving object, we do not know the precise space-time path (or trajectory) which the object has followed, but we do know that the trajectory must have passed these measured spatio-temporal locations. However, without making further assumptions, there are no theoretical limits to the movement of the object in between two measurements. An assumption that originates from the area of time geography, where the moving object's accessibility to an environment is studied, is that we know a bound on the speed of the moving object [5, 9, 13]. Therefore, it is common to associate a maximal speed to each moving object, alongside a trajectory sample. With this additional knowledge, the actual trajectory of a moving object is guaranteed to be contained in a spatio-temporal region known as a “lifeline necklace” in spatio-temporal and moving object databases [6, 10, 18], or simply as a “chain of space-time prisms” in the fields of time geography [9] and Geographical Information Systems (GIS) [15, 12, 14].

¹ Corresponding author



12:2 On the Complexity of the Realisability Problem for Visit Events

We refer to a moving object database with the specific purpose of storing trajectory samples and speed bounds as a *trajectory sample database*. A trajectory sample database can be seen as an uncertain (or incomplete) database [1, 11]. In general, an uncertain database represents a set of “possible worlds”, where each possible world is a concrete instantiation of the data. In our setting, a possible world corresponds to an assignment of a trajectory to every object, satisfying the known limitations of that object. In this paper, we propose a *query language for trajectory sample databases* that is based on events that are possibly semantically interesting for some application and that may occur during a trajectory of a moving object. We introduce *visit events* as part of our language, which are used to describe constraints on trajectories. The most basic visit event expresses that a trajectory visits a particular region during a particular period of time. To allow for composition, the class of visit events is closed under Boolean combinations (using negation, conjunction and disjunction). For example, we could express the complex event that states that an object has visited a museum in the morning, some restaurant at lunch and has not been in a particular church in the afternoon. For a dataset of tourists visiting some city, this event may have occurred during the movement of some of the tourists. This example shows that our proposed query language is related to the field of *semantic trajectories* and *places of interest* (POIs) in such trajectories and could be used to match trajectories with event patterns [4, 17, 21]. The queries appearing in [20], where a cylinder model of uncertainty is used, are similar to what our languages can express.

The atomic query in our language asks whether a given visit event is *realisable* by the trajectory of some moving object, that is, whether there exists a trajectory that satisfies the constraints described by that event. We call the evaluation of this atomic query the *realisability problem*. To allow for composition on the query level, the query expressions are also closed under Boolean combinations. An important detail is that our query language is defined relative to two parameters: $\mathcal{R}$, denoting the class of spatial regions that can occur in visit events, and $\mathcal{T}$, denoting the periods of time that can occur in visit events. We study the complexity of the realisability problem for different choices of $\mathcal{R}$ and $\mathcal{T}$. For $\mathcal{R}$, we consider the class of singleton points (**Point**) and of semi-algebraic sets (**SemiAlg**). For $\mathcal{T}$, on the other hand, we consider the class of singleton moments (**Moment**) and intervals of time (**Interval**). Because the realisability problem already becomes NP-hard for very restrictive classes of events, and because we believe it is of interest to measure the influence that the size of the database has on the complexity of query evaluation, we distinguish between combined complexity, data complexity and query complexity [1]. A summary of our complexity results is displayed in Table 1.

■ **Table 1** A summary of the complexity results: data, query, and combined complexity.

Class of events	Data	Query	Combined
(Point , Moment)-event in DNF	linear	polynomial	polynomial
Positive (Point , Moment)-events	linear	NP-hard	NP-hard
(SemiAlg , Moment)-events	linear	NP-hard	NP-hard
Conjunctive (Point , Interval)-events	polynomial	NP-hard	NP-hard
Positive (SemiAlg , Interval)-events	polynomial	NP-hard	NP-hard

The paper is organised as follows. In Section 2, we give the definitions that are necessary to formalize the notion of a trajectory sample database. In Section 3, we define the syntax and the semantics of our query languages. In Section 4, we study the complexity of the realisability problem in different settings. Finally, we conclude the paper with Section 5.

2 Definitions and preliminaries on trajectory sample databases

In this section, we give definitions of the concepts needed to introduce the notion of a *trajectory sample database*.

We consider the space in which objects move to be the plane $\mathbb{R}^2$, and we use the letters $p, q, \dots$ (with or without indices) to denote locations in space. Time is modelled as the real line $\mathbb{R}$, and we use t (with or without indices) to refer to the temporal points (or moments). We also use x and y to refer to the real coordinates of spatial locations. A spatio-temporal point (p, t) is then an element of $\mathbb{R}^2 \times \mathbb{R}$ and if $p = (x, y)$, we also write (x, y, t) for the spatio-temporal point (p, t) . Furthermore, we use d to denote the Euclidean distance in $\mathbb{R}^2$.

The movement of an object is captured by the notion of “trajectory” and it corresponds to a function mapping (all possible) moments in time to locations in space, as is expressed by the following definition.

► **Definition 1.** A trajectory is a continuous function from $\mathbb{R}$ to $\mathbb{R}^2$.

We use the Greek letter γ (with or without indices) to refer to trajectories. In practice trajectories are only observed or measured at discrete moments in time and we call these partial views on trajectories “trajectory samples”.

► **Definition 2.** A trajectory sample (or sample, for short) is a finite sequence of space-time points $\langle (p_1, t_1), \dots, (p_n, t_n) \rangle$ that is ordered by the temporal component (that is, $t_1 < \dots < t_n$).

We use the letter S (with or without indices) to refer to trajectory samples. The following definition captures the notion of trajectories matching trajectory samples.

► **Definition 3.** We say a trajectory γ visits a space-time point (p, t) when $\gamma(t) = p$ and we say that a trajectory γ visits a subset of space-time if γ visits one of its elements.

A trajectory sample $\langle (p_1, t_1), \dots, (p_n, t_n) \rangle$ matches the trajectory γ if γ visits all the space-time points (p_i, t_i) , for $i = 1, \dots, n$.

Trajectory sample databases do not only store a trajectory sample for each moving object, but also a maximal speed for each moving object. Because we assume that every moving object has such speed bound, these bounds further restrict trajectories (as trajectory samples do). This is captured in the following definition.

► **Definition 4.** A trajectory γ is called $v_{\max}$ -bounded when for all $t, t' \in \mathbb{R}$ we have $d(\gamma(t), \gamma(t')) \leq v_{\max} \cdot |t - t'|$.

We use $\Gamma_{(S, v_{\max})}$ to denote the set of all $v_{\max}$ -bounded trajectories matching the sample S . Obviously, some trajectory samples have no $v_{\max}$ -bounded trajectories that match them. The consistency of a sample with a speed bound $v_{\max}$ is expressed in the following definition.

► **Definition 5.** A trajectory sample $\langle (p_1, t_1), \dots, (p_n, t_n) \rangle$ is called $v_{\max}$ -consistent when $d(p_i, p_{i+1}) \leq v_{\max} \cdot (t_{i+1} - t_i)$, for $i = 1, \dots, n - 1$.

In the remainder of this paper, we assume, when given a trajectory sample S and a speed bound $v_{\max}$, that S is $v_{\max}$ -consistent.

► **Definition 6.** The linear interpolation trajectory of a trajectory sample $S = \langle (p_1, t_1), \dots, (p_n, t_n) \rangle$, denoted by $\text{LIT}(S)$, is defined as the trajectory γ , with

$$\gamma(t) = \begin{cases} p_1 & \text{if } t \leq t_1 \\ \frac{t_{i+1}-t}{t_{i+1}-t_i} \cdot p_i + \frac{t-t_i}{t_{i+1}-t_i} \cdot p_{i+1} & \text{if } t_i < t \leq t_{i+1} \\ p_n & \text{if } t_n < t. \end{cases} \quad (\text{for } 1 \leq i < n)$$

We note that if S is $v_{\max}$ -consistent, then $\text{LIT}(S)$ is a $v_{\max}$ -bounded trajectory.

In trajectory sample databases, we use natural numbers as identifiers for moving objects. Therefore, such database is defined relative to a finite subset Obj of $\mathbb{N}$, called the *object identifiers*.

► **Definition 7.** A trajectory sample database D over object identifiers $\text{Obj} \subset \mathbb{N}$ is a function mapping each identifier $i \in \text{Obj}$ to a pair $D(i) = (S_D(i), v_D(i))$, where $S_D(i)$ is a trajectory sample and $v_D(i)$ is a speed bound.

This means that, given a database D , the set $\Gamma_{D(i)} = \Gamma_{(S_D(i), v_D(i))}$ contains all trajectories that the object with identifier i may have followed (given the sample and the speed bound).

3 Syntax, semantics and evaluation of $(\mathcal{R}, \mathcal{T})$ -queries

In this section, we describe a family of query languages for trajectory sample databases. Our languages consists of two “tiers”: in the inner tier, we have the events, which are used as subexpressions in the outer tier, where we have the query expressions. We define the language relative to two parameters, $\mathcal{R}$ and $\mathcal{T}$, where $\mathcal{R}$ is a collection of spatial regions (or subsets of $\mathbb{R}^2$), and $\mathcal{T}$ is a collection of temporal periods (or subsets of $\mathbb{R}$).

In the following subsections, we define the syntax of the query languages, their semantics, and we end with a remark on the evaluation of query expressions.

3.1 The syntax of $(\mathcal{R}, \mathcal{T})$ -queries

We start by defining the inner tier of our language, which is a calculus of *events*. These events occur as subexpressions in the query expressions defined later on. Events express visits that may occur during the movement of an object.

► **Definition 8.** We define the $(\mathcal{R}, \mathcal{T})$ -events as follows:

1. $\text{visits}(R, T)$, with $R \in \mathcal{R}$ and $T \in \mathcal{T}$, is an atomic $(\mathcal{R}, \mathcal{T})$ -event;
2. If e, e_1 and e_2 are $(\mathcal{R}, \mathcal{T})$ -events, then so are $(\neg e)$, $(e_1 \wedge e_2)$ and $(e_1 \vee e_2)$.

In other words, $(\mathcal{R}, \mathcal{T})$ -events can be considered as propositional formulas over the propositional symbols $\text{visits}(R, T)$, for every $R \in \mathcal{R}$ and $T \in \mathcal{T}$. Now, we are ready to define $(\mathcal{R}, \mathcal{T})$ -queries. Their expression is given in the following definition.

► **Definition 9.** Let Var be a set of object identifier variables (or variables, for short). The $(\mathcal{R}, \mathcal{T})$ -query expressions are defined as follows:

1. If e is an $(\mathcal{R}, \mathcal{T})$ -event, i a natural number and $x \in \text{Var}$, then $\text{realisable}(i, e)$ and $\text{realisable}(x, e)$ are atomic $(\mathcal{R}, \mathcal{T})$ -query expressions;
2. If q, q_1 and q_2 are $(\mathcal{R}, \mathcal{T})$ -query expression, then so are $(\neg q)$, $(q_1 \wedge q_2)$ and $(q_1 \vee q_2)$.

3.2 The semantics of $(\mathcal{R}, \mathcal{T})$ -queries

To define the semantics of query expressions, we first define what it means for an event to be *realised* by a trajectory.

► **Definition 10.** Let e be an $(\mathcal{R}, \mathcal{T})$ -event and let γ be a trajectory.

1. If e is the atomic event $\text{visits}(R, T)$, then e is realised by γ if $\gamma(t) \in R$ for some $t \in T$ (that is, γ visits $R \times T$).
2. If e is of the form $(\neg e_1)$, then e is realised by γ when e_1 is not realised by γ .
3. If e is of the form $(e_1 \wedge e_2)$, then e is realised by γ when e_1 and e_2 are realised by γ .
4. If e is of the form $(e_1 \vee e_2)$, then e is realised by γ when e_1 or e_2 is realised by γ .

In our definition of the semantics of $(\mathcal{R}, \mathcal{T})$ -queries, we distinguish between Boolean and non-Boolean queries. The first type of queries contain no variables and they evaluate to true or false. The second type of queries contain variables and they define a relations over Obj .

► **Definition 11.** Let D be a trajectory sample database over object identifiers Obj . We write $D \models q$ to express that a variable-free $(\mathcal{R}, \mathcal{T})$ -query expression q evaluates to true on D . This relation is defined as follows:

1. $D \models \text{realisable}(i, e)$ if $i \in \text{Obj}$ and there exists a trajectory in $\Gamma_{D(i)}$ that realises e .
2. $D \models \neg q$ if $D \models q$ does not hold.
3. $D \models q_1 \wedge q_2$ if $D \models q_1$ and $D \models q_2$.
4. $D \models q_1 \vee q_2$ if $D \models q_1$ or $D \models q_2$.

► **Definition 12.** Let D be a trajectory sample database over object identifiers Obj . If q is an $(\mathcal{R}, \mathcal{T})$ -query expression containing variables $x_1, \dots, x_k \in \text{Var}$, then the result of q evaluated in D is

$$q(D) = \left\{ (i_1, \dots, i_k) \in \text{Obj}_D^k \mid D \models q[i_1/x_1, \dots, i_k/x_k] \right\},$$

where $q[i_1/x_1, \dots, i_k/x_k]$ is obtained from q by instantiating the variable x_j in q by i_j , for $j = 1, \dots, k$.

3.3 Evaluation of $(\mathcal{R}, \mathcal{T})$ -queries

Having defined the semantics of our $(\mathcal{R}, \mathcal{T})$ -query languages, there is a “standard” way of evaluating query expressions with variables: given a query expression with k variables, we enumerate all k -tuples of object identifiers, consider all the instantiations associated with them and then evaluate the variable-free expression obtained by substituting the variable occurrences by the concrete object identifiers from this tuple.

However, it is not immediately clear from the definition of the semantics how an atomic query of the form $\text{realisable}(i, e)$ can be evaluated. In what follows, we restrict our attention to this decision problem.

► **Definition 13.** We define the $(\mathcal{R}, \mathcal{T})$ -realisability problem to be the following decision problem: given $(\mathcal{R}, \mathcal{T})$ -event e , sample S and speed bound v , does there exist a trajectory in $\Gamma_{(S,v)}$ that realises e ?

From now on, we use the notation $(S, v) \models e$ to express that there exists a trajectory in $\Gamma_{(S,v)}$ that realises the event e .

4 The complexity of the $(\mathcal{R}, \mathcal{T})$ -realisability problem

In this section, we give a number of complexity results on the above realisability problem. We recall that the input to this problem is a trajectory sample, a speed bound and an $(\mathcal{R}, \mathcal{T})$ -event. In order for the realisability problem to be a proper computational decision problem, these inputs must have finite representations. This means, for example, that we cannot take arbitrary real numbers as input and that we need to assume some finite encoding of our inputs. Here, we assume that the spatio-temporal points occurring in the trajectory sample have rational coordinates, and that the speed bound is rational. As for the $(\mathcal{R}, \mathcal{T})$ -event, its representation depends on the choice for $\mathcal{R}$ and $\mathcal{T}$ and the representation of their elements. The choices for $\mathcal{R}$ that we consider are **Point**, the collection of all singletons in $\mathbb{Q}^2$; and **SemiAlg**, the collection of all semi-algebraic sets in the plane. A semi-algebraic set

in the plane is a subset of $\mathbb{R}^2$ that can be defined using a Boolean combination of polynomial (in)equalities over two real variables (where the polynomials have integer coefficients) [3]. Elements of **Point** are simply represented by the coordinates of the point in question, while an element of **SemiAlg** is represented by some encoding of its defining formula. The choices for $\mathcal{T}$ we consider are **Moment**, containing all singletons in $\mathbb{Q}$, and **Interval**, containing all closed intervals of $\mathbb{R}$ with rational endpoints.

For notational convenience, the atomic $(\text{Point}, \mathcal{T})$ -event $\text{visits}(\{p\}, T)$ will be written as $\text{visits}(p, T)$. And similarly, we will write $\text{visits}(R, t)$ for the atomic $(\mathcal{R}, \text{Moment})$ -event $\text{visits}(R, \{t\})$.

As we show below, the realisability problem is already NP-hard for quite restricted classes of events. Similar to the evaluation of queries in relational databases, the hardness of the realisability problem is caused by the size of the event, and not by the size of the trajectory sample. Therefore, we study the complexity of the realisability problem in three different settings [1], being:

- the *data complexity*, where we measure the complexity in terms of the size of the sample, and consider the event to be fixed,
- the *query complexity*, where we measure the complexity in terms of the size of event, and consider the sample to be fixed, and
- the *combined complexity*, where we measure the complexity in terms of both the size of the sample, and the size of the event.

We also consider several restrictions to the class of input events, such as *positive* events, containing no negations, *conjunctive* events, being conjunctions of atoms (or, not containing disjunctions and negations), and events in disjunctive normal form (DNF).

In the remainder of this section, we give various results on the complexity of the $(\mathcal{R}, \mathcal{T})$ -realisability problem, for different choices of $\mathcal{R}$ and $\mathcal{T}$ and in the three different settings.

For the complexity results mentioned below, we use a computational model in which operations (addition, multiplication, ...) and comparison relations ($=$, $<$, ...) on rational numbers are assumed to take unit time. That is, we measure the time complexity in terms of the number of spatio-temporal points in a trajectory sample and in terms of number of atoms (and their length) in an event-expression.

4.1 The query complexity of the $(\mathcal{R}, \mathcal{T})$ -realisability problem

Our first result shows that the realisability problem is already NP-hard for a relatively restricted class of events, namely the positive $(\text{Point}, \text{Moment})$ -events.

► **Theorem 14.** *In terms of query complexity, the $(\text{Point}, \text{Moment})$ -realisability problem for positive events is NP-hard.*

Proof. To prove NP-hardness, we describe a reduction from SAT, the satisfiability problem of propositional formulas. Because the statement concerns query complexity, we reduce SAT to the $(\text{Point}, \text{Moment})$ -realisability problem with fixed sample and speed bound. In this case, we choose the sample $S = \langle ((0, 0), 0), ((0, 0), 1) \rangle$ and the speed bound $v = 1$. The input of the reduction is a propositional formula ϕ . We can assume that ϕ is in negation normal form², because the satisfiability problem remains NP-hard under this restriction. The output is a positive $(\text{Point}, \text{Moment})$ -event e_ϕ such that the formula ϕ is satisfiable iff there exists a trajectory γ in $\Gamma_{(S, v)}$ that realises the event e_ϕ .

² A formula is said to be in negation normal form if negation operators are only applied to atoms.

Let $P_1, \dots, P_k$ be the propositional symbols occurring in ϕ . For $1 \leq i \leq k$, we define $t_i = \frac{i+1}{k+2}$, $p_i = (\frac{1}{2(k+2)}, 0)$ and $p'_i = (-\frac{1}{2(k+2)}, 0)$. Now, we take e_ϕ to be the result of substituting occurrences of $\neg P_i$ by $\text{visits}(p'_i, t_i)$ and non-negated occurrences of P_i by $\text{visits}(p_i, t_i)$ in ϕ . Clearly, e_ϕ is a (Point, Moment)-event, not containing negations.

Finally, we show that ϕ is satisfiable if and only if there exists a γ in $\Gamma_{(S,v)}$ that realises e_ϕ . The “if”-direction is straightforward. If there is some γ that realises e_ϕ , then ϕ must certainly be satisfiable. To be precise, the assignment that assigns P_i to true if and only if $\gamma(t_i) = p_i$ satisfies ϕ . For the “only if”-direction, assume that ϕ is satisfiable. Then there exists a truth assignment α that makes ϕ true. Now, we extend the sample S to a sample S' such that it contains (p_i, t_i) if P_i is assigned true by α , and otherwise contains (p'_i, t_i) . It is easily verified that S' is v -consistent, which means $\text{LIT}(S')$ is a v -bounded trajectory. Now, we have that $\text{LIT}(S')$ realises $\text{visits}(p_i, t_i)$ if P_i is true under α , and realises $\text{visits}(p'_i, t_i)$ if $\neg P_i$ is true under α . It follows from the way we constructed the event, that $\text{LIT}(S')$ realises e_ϕ . ◀

While the above result shows that the (Point, Moment)-realisability problem is NP-hard for positive events (not containing negations), the problem for (Point, Interval)-events already becomes NP-hard for conjunctions of atoms, as shown below.

► **Theorem 15.** *In terms of query complexity, the (Point, Interval)-realisability problem for conjunctive events is NP-hard.*

Proof. We give a reduction from the Euclidean travelling salesman problem (E-TSP for short), shown to be NP-hard in [16] (the problem we refer to here is called the Euclidean tour-TSP there). The Euclidean travelling salesman problem asks, given a finite set of locations $P \subseteq \mathbb{Q}^2$ and a positive number $\ell \in \mathbb{Q}$, whether there is a cycle through all locations of P whose length is at most ℓ . Formally, this means there is a permutation $p_1, \dots, p_n$ of the locations in P such that $\sum_{i=1}^{n-1} d(p_i, p_{i+1}) + d(p_n, p_1) \leq \ell$. Without loss of generality, we assume that P always contains the origin $(0, 0)$.

Again, we work with a fixed sample $S = \langle ((0, 0), 0) \rangle$ and speed bound $v = 1$. From an instance P, ℓ of E-TSP, we give a conjunction of (Point, Interval)-atoms C , such that $(S, v) \models C$ if and only if there is cycle through P of length at most ℓ . We define C as

$$\text{visits}((0, 0), [\ell, \ell]) \wedge \bigwedge_{p \in P \setminus \{(0, 0)\}} \text{visits}(p, [0, \ell]).$$

To prove that this reduction is correct, we first show that if $(S, v) \models C$, then there is cycle through P of length at most ℓ . Let γ be a v -bounded trajectory matching S and realising C . Because γ matches S , we have $\gamma(0) = (0, 0)$. And, because γ realises C , it reaches every location in P at some moment in $[0, \ell]$, and $\gamma(\ell) = (0, 0)$. This induces an order $p_1, \dots, p_n$ of the locations in P , where γ first reaches p_1 , then p_2 , and so on (we note that p_1 is always $(0, 0)$). Thus, if we let t_i be the first moment where $\gamma(t_i) = p_i$ for $i = 1, \dots, n$, then $0 = t_1 < \dots < t_n \leq \ell$. We claim the cycle $p_1, \dots, p_n, p_1$ has length at most ℓ . The length of this cycle is $\sum_{i=1}^{n-1} d(p_i, p_{i+1}) + d(p_n, p_1)$. For every i , γ visits (p_i, t_i) , and because γ is v -bounded, we have $d(p_i, p_{i+1}) \leq v \cdot |t_i - t_{i+1}| = t_{i+1} - t_i$. Similarly, γ visits (p_n, t_n) and (p_1, ℓ) , thus $d(p_n, p_1) \leq \ell - t_n$. It follows that the length of the cycle is at most $\sum_{i=1}^{n-1} (t_{i+1} - t_i) + \ell - t_n = t_n - t_1 + \ell - t_n = \ell$.

Finally, we show that if there is cycle through P of length at most ℓ , then $(S, v) \models C$. Let $p_1, \dots, p_n, p_1$ be such cycle, where we choose p_1 to be $(0, 0)$. Now, let $t_1 = 0$ and for $i = 2, \dots, n$, take $t_i = t_{i-1} + d(p_{i-1}, p_i)$. Then, $t_n = \sum_{i=1}^{n-1} d(p_i, p_{i+1})$, which, by assumption, is at most $\ell - d(p_n, p_1)$. Define the trajectory sample $S' = \langle (p_1, t_1), \dots, (p_n, t_n), (p_1, \ell) \rangle$. It is clear from the definition of t_i that the part of S' excluding (p_1, ℓ) is v -consistent. We have seen that $t_n \leq \ell - d(p_n, p_1)$, and thus $d(p_n, p_1) \leq \ell - t_n$, which implies that S' is v -consistent. From this follows that $\text{LIT}(S')$ is v -bounded, and it clearly matches S and realises C . ◀

4.2 The data complexity of the $(\mathcal{R}, \mathcal{T})$ -realisability problem

In this section, we show that the (SemiAlg, Moment)-realisability problem has linear-time data complexity, and the (SemiAlg, Interval)-realisability problem has polynomial-time data complexity

We start with the result on (SemiAlg, Moment)-queries, but first we introduce some notation and we give two lemmas. Every region in SemiAlg is of the form $\{(x, y) \in \mathbb{R}^2 \mid \varphi(x, y)\}$, where $\varphi = \varphi(x, y)$ is a quantifier-free formula over the vocabulary $(+, \times, <, 0, 1)$, with x and y as free variables. The set $\{(x, y) \in \mathbb{R}^2 \mid \varphi(x, y)\}$ is called the region defined by φ , and we denote it by $R(\varphi)$.

► **Definition 16.** If $C = \text{visits}(R_1, t_1) \wedge \dots \wedge \text{visits}(R_k, t_k)$ is a conjunction of atomic $(\mathcal{R}, \text{Moment})$ -events, and $I \subseteq \mathbb{R}$ is an interval, the formula C_I is the conjunction of those $\text{visits}(R_i, t_i)$, with $1 \leq i \leq k$, for which $t_i \in I$.

► **Lemma 17.** Let C be a conjunction of atomic $(\mathcal{R}, \text{Moment})$ -events, let S be a trajectory sample $\langle (p_1, t_1), \dots, (p_n, t_n) \rangle$ and let v be a speed bound. Then, $(S, v) \models C$ if and only if all of the following are true:

- (1) $(\langle (p_1, t_1) \rangle, v) \models C_{(-\infty, t_1]}$,
- (2) for $i = 1, \dots, n-1$, we have $(\langle (p_i, t_i), (p_{i+1}, t_{i+1}) \rangle, v) \models C_{[t_i, t_{i+1}]}$, and
- (3) $(\langle (p_n, t_n) \rangle, v) \models C_{[t_n, +\infty)}$.

Proof. The “only if”-direction is obvious. We prove the “if”-direction. Assume (1), (2) and (3) are true. By (1), there exists a v -bounded trajectory γ_0 matching $\langle (p_1, t_1) \rangle$ that realises $C_{(-\infty, t_1]}$. By (2), for $i = 1, \dots, n-1$, there exists a v -bounded trajectory γ_i matching $\langle (p_i, t_i), (p_{i+1}, t_{i+1}) \rangle$ that realises $C_{[t_i, t_{i+1}]}$. And by (3), there exists a v -bounded trajectory γ_n matching $\langle (p_n, t_n) \rangle$ that realises $C_{[t_n, +\infty)}$. We define the trajectory γ as follows:

$$\gamma(t) = \begin{cases} \gamma_0(t) & \text{if } t \in (-\infty, t_1], \\ \gamma_i(t) & \text{if } t \in [t_i, t_{i+1}] \text{ and} \\ \gamma_n(t) & \text{if } t \in [t_n, +\infty). \end{cases}$$

We note that the intervals $[t_{i-1}, t_i]$ and $[t_i, t_{i+1}]$ both contain t_i . However, this does not pose a problem for the definition of γ , because both γ_{i-1} and γ_i visit (p_i, t_i) , which means $\gamma_{i-1}(t_i) = \gamma_i(t_i) = p_i$. It only requires a simple application of the triangle inequality (of Euclidean distance) to show that γ is a v -bounded trajectory. For $i = 1, \dots, n$, we have $\gamma(t_i) = p_i$, thus γ matches S . The only thing left to prove is that γ realises C . Let $A = \text{visits}(R, t)$ be an arbitrary conjunct of C . Depending on t , there are three cases to be considered. First, if $t \in (-\infty, t_1]$, then A is a conjunct of $C_{(-\infty, t_1]}$. This implies γ_0 realises A , so $\gamma(t) = \gamma_0(t) \in R$, which means γ realises A . The other two cases, when t is contained in $[t_i, t_{i+1}]$ or in $[t_n, +\infty)$, are similar. Because γ realises all the conjuncts of C , it also realises C itself. ◀

► **Lemma 18.** If $C = \text{visits}(R_1, t_1) \wedge \dots \wedge \text{visits}(R_k, t_k)$ is a conjunction of atomic $(\mathcal{R}, \text{Moment})$ -events, S is a trajectory sample and v a speed bound, then $(S, v) \models C$ if and only if there exist k spatial locations $p_1, \dots, p_k$ such that

- (1) for $i = 1, \dots, k$ we have $p_i \in R_i$, and
- (2) the sample containing the points $(p_1, t_1), \dots, (p_k, t_k)$, as well as the ones in S , is v -consistent.

We remark that we consider condition (2) from the lemma to be false when some space-time point among $(p_1, t_1), \dots, (p_k, t_k)$ shares its temporal component with a point in S , while their spatial component differs.

Proof. The “only if”-direction is obvious. We prove the “if”-direction. Assume there are locations $p_1, \dots, p_k$ satisfying (1) and (2). Now take the trajectory γ to be the linear interpolation trajectory of the sample containing the points $(p_1, t_1), \dots, (p_k, t_k)$, as well as the ones in S . It is clear that γ matches S and assumption (2) implies it is a v -bounded trajectory. Finally, assumption (1) implies that γ realises C . ◀

► **Definition 19.** We say two $(\mathcal{R}, \mathcal{T})$ -events A and B are equivalent if for every sample S and speed bound v , we have $(S, v) \models A$ if and only if $(S, v) \models B$.

The following proposition follows directly from the fact that, for $p \in \mathbb{R}^2$, we have $p \notin R(\varphi)$ if and only if $p \in R(\neg\varphi)$.

► **Proposition 20.** A $(\text{SemiAlg}, \text{Moment})$ -event of the form $\neg\text{visits}(R(\varphi), t)$ is equivalent to the event $\text{visits}(R(\neg\varphi), t)$.

Given an arbitrary $(\text{SemiAlg}, \text{Moment})$ -event, we can convert it into negation normal form and use Proposition 20 to remove all negations. The result of this process is a positive $(\text{SemiAlg}, \text{Moment})$ -event that is equivalent to the original. Because this process can be performed in linear time (with respect to the length of the event), we can assume that any given $(\text{SemiAlg}, \text{Moment})$ -event is positive, without loss of generality.

► **Theorem 21.** In terms of data complexity, the $(\text{SemiAlg}, \text{Moment})$ -realisability problem is decidable in linear time.

Proof. We describe an algorithm to decide the realisability problem of a $(\text{SemiAlg}, \text{Moment})$ -event e for input sample $S = \langle (p_1, t_1), \dots, (p_n, t_n) \rangle$, where $p_i = (x_i, y_i)$, and speed bound v . Noting the remark made above, we assume that e is positive. The first step is to convert e into its disjunctive normal form $\bar{e}$. Since we are dealing with data complexity, the possibly increased size of $\bar{e}$, compared to e , has no impact on the running time of our method. In fact, because the number of disjuncts of $\bar{e}$ is constant, it is sufficient to show that the realisability problem for a single disjunct can be decided in linear time.

Consider a disjunct C of $\bar{e}$. Because $\bar{e}$ is positive and in DNF, the event C must be a conjunction of atomic events. This means we can apply Lemma 17, and we can determine whether $(S, v) \models C$ by testing whether each of the following conditions are met:

- (1) $(\langle (p_1, t_1) \rangle, v) \models C_{(-\infty, t_1]}$,
- (2) for $i = 1, \dots, n-1$, we have $(\langle (p_i, t_i), (p_{i+1}, t_{i+1}) \rangle, v) \models C_{[t_i, t_{i+1}]}$, and
- (3) $(\langle (p_n, t_n) \rangle, v) \models C_{[t_n, +\infty)}$.

We focus our attention to condition (2). For every value of i , we want to decide whether $C_{[t_i, t_{i+1}]}$ is realisable for sample $\langle (p_i, t_i), (p_{i+1}, t_{i+1}) \rangle$ and speed bound v . Let us write the conjunction $C_{[t_i, t_{i+1}]}$ as $\text{visits}(R(\varphi_1), t'_1) \wedge \dots \wedge \text{visits}(R(\varphi_k), t'_k)$, with $t'_1 \leq \dots \leq t'_k$. We remark that this implies $t_i \leq t'_1 \leq \dots \leq t'_k \leq t_{i+1}$. From Lemma 18, we know that $(\langle (p_i, t_i), (p_{i+1}, t_{i+1}) \rangle, v) \models C_{[t_i, t_{i+1}]}$ if and only if there exist locations $q_1, \dots, q_k \in \mathbb{R}^2$, with $q_i = (x'_i, y'_i)$, such that

- (a) for $j = 1, \dots, k$ we have $q_j \in R(\varphi_j)$, and
- (b) the sample $\langle (p_i, t_i), (q_1, t'_1), \dots, (q_k, t'_k), (p_{i+1}, t_{i+1}) \rangle$ is v -consistent.

12:10 On the Complexity of the Realisability Problem for Visit Events

In other words, we have $(\langle(p_i, t_i), (p_{i+1}, t_{i+1})\rangle, v) \models C_{[t_i, t_{i+1}]}$ if and only if the formula $\psi = \exists x'_1 \exists y'_1 \exists \dots \exists x'_k \exists y'_k (\psi_a \wedge \psi_b)$ is true, where ψ_a is $\varphi_1(x'_1, y'_1) \wedge \dots \wedge \varphi_k(x'_k, y'_k)$, expressing condition (a), and to express condition (b), we take ψ_b to be the conjunction of $k+1$ distance inequalities.

Because ψ is a first-order logic sentence over the ordered field of real numbers, its truth can be determined by a decision procedure for the theory of real closed fields (first described by Tarski [19], we refer to Basu et al. [2] for a modern exposition). We note that the size of ψ is independent of n , and thus has constant size in the data complexity setting (k is bounded by the length of C). The time needed to determine the truth of ψ is thus also constant. To test for condition (2), we have to perform the above steps for $n-1$ values of i . Conditions (1) and (3) can both be tested in constant time, in a manner similar to the above, the only difference is that the constructed sentence requires one less inequality for expressing v -consistency. We have thus shown that the realisability problem for a single disjunct of $\bar{e}$ can be answered in $O(n)$ time. This concludes the proof. $\blacktriangleleft$

Our next result concerns the data complexity of the (SemiAlg, Interval)-realisability problem for positive events.

► **Lemma 22.** *If e is a positive $(\mathcal{R}, \text{Interval})$ -event containing k distinct atoms $A_1, \dots, A_k$, where $A_i = \text{visits}(R_i, T_i)$, then $(S, v) \models e$ if and only if there exist k space-time points $(p_1, t_1), \dots, (p_k, t_k)$ such that*

- (1) *the sample containing the points $(p_1, t_1), \dots, (p_k, t_k)$, as well as the ones in S , is v -consistent, and*
- (2) *the Boolean expression obtained by replacing A_i in e by true if $(p_i, t_i) \in R_i \times T_i$, and by false otherwise, evaluates to true.*

Proof. We first prove the “if”-direction. Assume that there exist $(p_1, t_1), \dots, (p_k, t_k)$ satisfying (1) and (2). Let S' be the sample containing the points $(p_1, t_1), \dots, (p_k, t_k)$, as well as the ones in S . Assumption (1) says S' is v -consistent, which means $\text{LIT}(S')$ is v -bounded. Because S' is an extension of S , and γ matches S' , it must also match S . The trajectory $\text{LIT}(S')$ realises all the atoms A_i for which $(p_i, t_i) \in R_i \times T_i$, and potentially others. Because of assumption (2) and the fact that e is positive, $\text{LIT}(S')$ realises e , and thus $(S, v) \models e$.

For the “only if”-direction, we assume that $(S, v) \models e$. That means that there exists a v -bounded γ realising e and matching S . We choose $(p_1, t_1), \dots, (p_k, t_k)$ as follows. For every atom A_i which γ realises, we know that γ visits some point in $R_i \times T_i$, and take (p_i, t_i) to be such point. For an atom A_i not realised by γ , we take (quite arbitrarily) (p_i, t_i) to be the first anchor point of S . Then, $(p_i, t_i) \notin R_i \times T_i$, since otherwise γ would realise A_i . All points of $(p_1, t_1), \dots, (p_k, t_k)$ and S are visited by γ , so the sample containing all those points must be v -consistent, satisfying condition (1). Because γ realises A_i if and only if $(p_i, t_i) \in R_i \times T_i$ and γ realises e , condition (2) is also satisfied. $\blacktriangleleft$

The following proposition follows directly from the definition of v -consistency and the fact that the distance function d obeys the triangle inequality.

► **Proposition 23.** *If M is an (unordered) finite set of space-time points, then the sample containing all points in M is v -consistent if and only if for every pair of points (p_1, t_1) and (p_2, t_2) from M , we have $d(p_1, p_2) \leq v \cdot |t_1 - t_2|$.*

It will be of interest later that the condition $d(p_1, p_2) \leq v \cdot |t_1 - t_2|$ from the above proposition is equivalent to $d(p_1, p_2)^2 \leq v^2 \cdot (t_1 - t_2)^2$, which, if $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$, is a polynomial inequality with variables $x_1, y_1, t_1, x_2, y_2, t_2, v$.

► **Theorem 24.** *In terms of data complexity, the (SemiAlg, Interval)-realisability problem for positive events can be decided in polynomial time.*

Proof. We describe an algorithm to decide the realisability of a positive (SemiAlg, Interval)-event e for input sample S , of length n , and speed bound v . Lemma 22 gives a condition equivalent to $(S, v) \models e$. We can express this condition using a first-order logic sentence over the ordered field of real numbers $\psi = \exists x_1 \exists y_1 \exists t_1 \dots \exists x_k \exists y_k \exists t_k (\psi_1 \wedge \psi_2)$, where ψ_1 expresses part (1) of Lemma 22, and ψ_2 expresses part (2). To express part (1), stating that the sample containing $(x_1, y_1, t_1), \dots, (x_k, y_k, t_k)$ as well as the points from S is v -consistent, we can take ψ_1 to be a conjunction of $(n+k)^2$ distance inequalities, as per Proposition 23. To express the second part, we construct ψ_2 by taking e and replacing every atom $\text{visits}(R(\varphi_i), [t_i^-, t_i^+])$ by the formula $\varphi_i(x_i, y_i) \wedge t_i^- \leq t_i \wedge t_i \leq t_i^+$.

We have now constructed a formula ψ which is true if and only if $(S, v) \models e$. Thus, if there is a method to determine the truth ψ , we can decide the realisability problem for positive (SemiAlg, Interval)-events. Because ψ is an existential sentence, we can apply known decision procedures for the existential theory of the reals. Of course, the time complexity required to decide the realisability of positive (SemiAlg, Interval)-events in the described manner, depends on the time complexity of the existential theory of the reals. In [2] (see theorem 13.14), an upper bound of $s^{m+1}d^{O(m)}$ is given, where s is the number of polynomials occurring in the formula, m the number of variables, and d the maximum degree of the polynomials. Because we are considering data complexity, the number of polynomials in the φ_i 's, as well as their maximum degree, is constant, and so is k . This implies that ψ contains $O(n^2)$ polynomials of constant degree, and $3k$ variables. Thus, by Theorem 13.14 from [2], the truth of ψ can be decided in $O((n^2)^{3k+1}) = O(n^{6k+2})$ time, which is polynomial in n . It is also clear that ψ can be constructed in polynomial time. ◀

4.3 A class of events for which the realisability problem has polynomial time combined complexity

Until now, we have only seen hardness results in the query (and thus, combined) complexity setting. In this section, we give an example of a class of events for which the realisability problem has polynomial-time combined complexity.

An $(\mathcal{R}, \mathcal{T})$ -literal is either an atomic $(\mathcal{R}, \mathcal{T})$ -event, or the negation of an atomic $(\mathcal{R}, \mathcal{T})$ -event. Conjunctions of (Point, Moment)-literals provide a class of events for which the realisability problem has an efficient solution in terms of combined complexity. Before giving our result, we first prove a lemma.

► **Lemma 25.** *If S_1 and S_2 are trajectory samples, then there exists a v -bounded trajectory matching S_1 and not visiting any point in S_2 if and only if*

- (1) S_1 does not contain a point in S_2 ,
- (2) S_1 is v -consistent, and
- (3) if $\text{LIT}(S_1)$ visits a point from S_2 , on the line segment between points (p_i, t_i) and (p_{i+1}, t_{i+1}) from S_1 , then $d(p_i, p_{i+1}) < v \cdot (t_{i+1} - t_i)$.

Proof. It is obvious that the conditions (1), (2) and (3) are necessary for the existence of a v -bounded trajectory matching S_1 and not visiting points in S_2 . To show that they are also sufficient, we assume that all three conditions are met. Consider the trajectory $\text{LIT}(S_1)$. By condition (2), it is v -bounded. In case $\text{LIT}(S_1)$ visits one or more points from S_2 , we show that we can extend (by adding points) S_1 to a sample S^* , such that $\text{LIT}(S^*)$ is v -bounded, matches S_1 and does not visit points from S_2 . Because S^* is obtained by adding points to S_1 ,

it is obvious that $\text{LIT}(S^*)$ matches S_1 . Let us say that $\text{LIT}(S_1)$ visits a point (q, t) from S_2 on the line segment between (p_i, t_i) and (p_{i+1}, t_{i+1}) . From (1) we know that $t_i < t < t_{i+1}$, and from (3) we have $d(p_i, p_{i+1}) < v \cdot (t_{i+1} - t_i)$. Together, these observations imply there must be a small disk around q , such that for every p in this disk, the extension of S_1 with the point (p, t) remains v -consistent. Informally, this means we can deviate $\text{LIT}(S_1)$ slightly between t_i and t_{i+1} , while the trajectory remains v -bounded. Now, for every point (q', t') of S_2 with $t_i < t' < t_{i+1}$, there is at most one choice for p inside the disk that makes the linear interpolation trajectory of the extended sample visit (q', t') . Because there are infinitely many points in the disks, we can choose p such that, between t_i and t_{i+1} , none of the points from S_2 are visited. We can repeat this process for each line segment of $\text{LIT}(S_1)$ that visits a point from S_2 . The linear interpolation trajectory of the resulting sample S^* then matches S_1 , is v -bounded and does not visit points from S_2 , as desired. ◀

► **Theorem 26.** *In terms of combined complexity, the (Point, Moment)-realisability problem for a conjunction of k literals and a trajectory sample of length n is decidable in $O(n + k \log k)$ time.*

Proof. We assume that we are given a conjunction C of k (Point, Moment)-literals, a sample S of length n and a speed bound v as input. Let $P_1, \dots, P_m$ be the positive atoms occurring in C and let $N_1, \dots, N_\ell$ be the atoms occurring negated in C . First we compute the sample $S_1 = \langle (p_1, t_1), \dots, (p_{n+m}, t_{n+m}) \rangle$, containing both the points in S and the ones occurring in the atoms $P_1, \dots, P_m$. Computing this sample requires ordering the points occurring in $P_1, \dots, P_m$ by time, and merging these with those from S (which are already ordered by time). This can be done in $O(n + k \log k)$ time. We also compute a sample S_2 , containing the points from $N_1, \dots, N_\ell$, in $O(k \log k)$ time. Now we have $(S, v) \models C$ if and only if there exists a v -bounded trajectory matching S_1 and not visiting any point in S_2 . This can be determined by testing for the conditions of Lemma 25 in $O(|S_1| + |S_2|)$ time, where $|S_1| + |S_2| = n + m + \ell = n + k$. Thus, the total time required by the described procedure is $O(n + k \log k)$. ◀

► **Corollary 27.** *In terms of combined complexity, the (Point, Moment)-realisability problem for an event in disjunctive normal form, of length k , and a trajectory sample of length n is decidable in $O(kn + k \log k)$ time.*

Proof. We assume that we are given a (Point, Moment)-event e in disjunctive normal form, of length k , a sample S of length n and a speed bound v as input. Then, e is of the form $C_1 \vee \dots \vee C_\ell$, where every disjunct C_i is a conjunction of literals. Let us say that the event C_i contains k_i literals. Because e is realisable if and only if one of the C_i 's is realisable, we can use Theorem 26 to determine the realisability of e , by determining it for each of the disjuncts. This takes $\sum_{i=1}^{\ell} O(n + k_i \log k_i)$ time, which is $O(kn + k \log k)$. ◀

5 Conclusion

We have proposed a family of query languages for trajectory sample databases. Query expressions in these languages contain events, which are used to describe constraints on trajectories. When performing query evaluation, an essential problem required to be solved is that of the realisability of an event. We studied the complexity of this realisability problem in terms of data, query and combined complexity. These results are summarised in Table 1. These complexity results are given in a computational model wherein (arithmetic and comparison) operations on rational numbers take unit time. It is not clear whether our

results remain true in a model in which we measure the cost of these operations in terms of the length of the bit-representation of rational numbers. Also, we did not give much attention to the evaluation of query expressions outside of the realisability problem. We assumed that query expressions are evaluated in some standard way, but more efficient strategies might exist.

References

- 1 Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995. URL: <http://webdam.inria.fr/Alice/>.
- 2 S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in real algebraic geometry*. Algorithms and Computation in Mathematics 10. Springer, Berlin, 2003.
- 3 Jacek Bochnak, Michel Coste, and Marie-Françoise Roy. *Real Algebraic Geometry*, volume 36 of *Ergebnisse der Mathematik und ihrer Grenzgebiete*. Springer-Verlag, 1998.
- 4 Vania Bogorny, Bart Kuijpers, and Luis Otávio Alvares. ST-DMQL: A semantic trajectory data mining query language. *Int. J. Geogr. Inf. Sci.*, 23(10):1245–1276, 2009. URL: <http://www.informaworld.com/smpp/content%7Edb=all%7Econtent=a915093101%7Efrm=abslink>.
- 5 L. Burns. *Transportation, Temporal, and Spatial Components of Accessibility*. Lexington Books, Lexington, MA, 1979.
- 6 Max J. Egenhofer. Approximation of geospatial lifelines. In Elisa Bertino and Leila De Floriani, editors, *SpadaGIS, Workshop on Spatial Data and Geographic Information Systems*. University of Genova, 2003.
- 7 Fosca Giannotti and Dino Pedreschi, editors. *Mobility, Data Mining and Privacy - Geographic Knowledge Discovery*. Springer, 2008. doi:10.1007/978-3-540-75177-9.
- 8 R. Güting and M. Schneider. *Moving Object Databases*. Morgan Kaufmann, 2005.
- 9 T. Hägerstrand. What about people in regional science? *Papers of the Regional Science Association*, 24:7–21, 1970.
- 10 Kathleen Hornsby and Max J. Egenhofer. Modeling moving objects over multiple granularities. *Ann. Math. Artif. Intell.*, 36(1-2):177–194, 2002. doi:10.1023/A:1015812206586.
- 11 Tomasz Imieliński and Witold Lipski. Incomplete information in relational databases. *J. ACM*, 31(4):761–791, September 1984. doi:10.1145/1634.1886.
- 12 Donald Janelle and Michael Goodchild. Diurnal patterns of social group distributions in a canadian city. *Economic Geography*, 59(4):403–425, 1983. URL: <http://www.jstor.org/stable/144166>.
- 13 B. Lenntorp. *Paths in Space-Time Environments: A Time-Geographic Study of the Movement Possibilities of Individuals*. Number 44 in Series B. Lund Studies in Geography, 1976.
- 14 H.J. Miller. Modeling accessibility using space-time prism concepts within geographical information systems. *International Journal of Geographical Information Systems*, 5:287–301, 1991. doi:10.1080/02693799108927856.
- 15 H.J. Miller. A measurement theory for time geography. *Geographical Analysis*, 2005. doi:10.1111/j.1538-4632.2005.00575.x.
- 16 Christos H. Papadimitriou. The Euclidean travelling salesman problem is NP-complete. *Theoretical Computer Science*, 4(3):237–244, 1977. doi:10.1016/0304-3975(77)90012-3.
- 17 Christine Parent, Stefano Spaccapietra, Chiara Renso, Gennady L. Andrienko, Natalia V. Andrienko, Vania Bogorny, Maria Luisa Damiani, Aris Gkoulalas-Divanis, José Antônio Fernandes de Macêdo, Nikos Pelekis, Yannis Theodoridis, and Zhixian Yan. Semantic trajectories modeling and analysis. *ACM Comput. Surv.*, 45(4):42:1–42:32, 2013. doi:10.1145/2501654.2501656.
- 18 Dieter Pfoser and Christian S. Jensen. Capturing the uncertainty of moving-object representations. In Ralf Hartmut Güting, Dimitris Papadias, and Frederick H. Lochovsky, editors, *Advances in Spatial Databases, 6th International Symposium, SSD'99, Proceedings*, volume 1651 of *Lecture Notes in Computer Science*, pages 111–132. Springer, 1999. doi:10.1007/3-540-48482-5_9.

12:14 On the Complexity of the Realisability Problem for Visit Events

- 19 Alfred Tarski and J. C. C. McKinsey. *A Decision Method for Elementary Algebra and Geometry*. University of California Press, 1951.
- 20 G. Trajcevski, O. Wolfson, K. Hinrichs, and S. Chamberlain. Managing uncertainty in moving objects databases. *ACM Trans. Database Syst.*, 29(3):463–507, 2004. doi:10.1145/1016028.1016030.
- 21 Zhixian Yan, Dipanjan Chakraborty, Christine Parent, Stefano Spaccapietra, and Karl Aberer. Semantic trajectories: Mobility data computation and annotation. *ACM Trans. Intell. Syst. Technol.*, 4(3):49:1–49:38, 2013. doi:10.1145/2483669.2483682.