Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 25, no. 1, pp. 195–218 (2021)
DOI: 10.7155/jgaa.00557

On the Complexity of Some Geometric Problems With Fixed Parameters

Marcus Schaefer[1]

[1]School of Computing
DePaul University, Chicago, Illinois 60604, USA

Submitted: July 2020	Reviewed: December 2020	Revised: January 2021
Accepted: January 2021	Final: January 2021	Published: January 2021
Article type: Regular paper		Communicated by: Martin Nöllenburg

Abstract. The following graph-drawing problems are known to be complete for the existential theory of the reals ($\exists\mathbb{R}$-complete) as long as the parameter k is unbounded. Do they remain $\exists\mathbb{R}$-complete for a fixed value k?

- Do k graphs on a shared vertex set have a simultaneous geometric embedding?
- Is G a segment intersection graph, where G has maximum degree at most k?
- Given a graph G with a rotation system and maximum degree at most k, does G have a straight-line drawing which realizes the rotation system?

We show that these, and some related, problems remain $\exists\mathbb{R}$-complete for constant k, where k is in the double or triple digits. To obtain these results we establish a new variant of Mnëv's universality theorem, in which the gadgets are placed so as to interact minimally; this variant leads to fixed values for k, where the traditional variants of the universality theorem require unbounded values of k.

Keywords: Simultaneous geometric embedding, rotation system, intersection graph, existential theory of the reals, Mnëv, universality theorem.

AMS Subject Classification (2020): 68Q17, 68R10, 05C10, 05C62, 05B35

1 Introduction

The complexity class $\exists\mathbb{R}$ captures the complexity of deciding truth in the existentially quantified theory of the reals. $\exists\mathbb{R}$ has turned out to be a successful model for describing the complexity of many problems in graph drawing, and computational geometry, which require placement of

E-mail address: mschaefer@cdm.depaul.edu (Marcus Schaefer)

geometric objects in the Euclidean plane (or some higher-dimensional space). A growing list of problems complete for $\exists\mathbb{R}$ can be found at [34], also see [4] for a survey.

As in the case of **NP**, just at a smaller scale, these complete problems can be used to establish $\exists\mathbb{R}$-hardness of new problems.[1] And just as in the case of **NP**, it is sometimes necessary to go back to the original problems, to see how they can be tweaked, to obtain new and stronger hardness results. For **NP**, this has led to a rich list of **NP**-complete variants of the satisfiability problem, as well as a few other core problems. For $\exists\mathbb{R}$, two problems have been fundamental in establishing hardness results in graph drawing and computational geometry: Stretchability of (arrangements of) pseudolines and the dual problem of realizing abstract order types (oriented matroids). These problems were first shown $\exists\mathbb{R}$-complete by Mnëv [28], though it was arguably Shor [33] who emphasized the computational implications of Mnëv's universality theorem. Figure 1 shows a small example of a stretchability problem. On the left we have an *arrangement of pseudolines*, x-monotone curves from $-\infty$ to ∞ so that each pair crosses once; such an arrangement is called *stretchable* if it is isomorphic to an arrangement of straight lines (the isomorphism being an isomorphism of the plane).

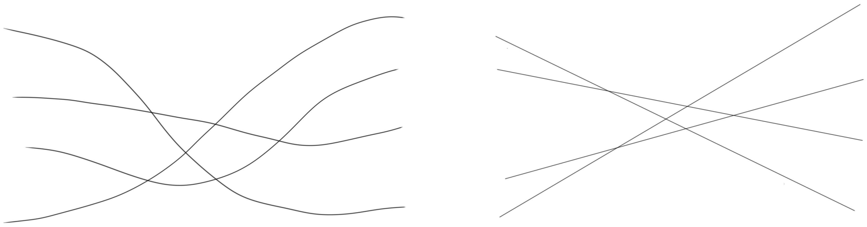

Figure 1: The stretchability problem. A pseudoline arrangement *(left)*, and its straight-line realization, an isomorphic line arrangement *(right)*. Both arrangements are *simple* in that at most two pseudolines/lines pass through each point.

Let us illustrate the issue we are addressing in this paper through an example. A graph is a *segment intersection graph* if each vertex can be assigned a line-segment in the plane so that there is an edge in the graph if and only if the corresponding line-segments cross. Kratochvíl and Matoušek [23] showed that SEG, the problem of deciding whether a graph is a segment intersection graph, is $\exists\mathbb{R}$-complete. They did so, by reducing the stretchability problem to SEG. In a first step, each pseudoline is replaced by a pseudo-segment (a subarc of the pseudoline), containing all crossings, as shown in the left illustration of Figure 2; then additional pseudo-segments are added to ensure the topology of the original pseudoline arrangement is maintained (shown in the right of the figure).

And here lies the issue: Each of the pseudo-segments has at least as many crossings as the pseudoline it is based on. Since every two pseudolines cross, and the number of pseudolines is unbounded, this implies that each pseudo-segments has an unbounded number of crossings, and the corresponding segment intersection graph unbounded degree. So if we want to sfhow that SEG remains $\exists\mathbb{R}$-complete for bounded-degree graphs, we need to push beyond stretchability to Mnëv's result.

Here we face the real issue: The constructions underlying proofs of Mnëv's theorem rely on placing gadgets along a line. Mnëv [28], Shor [33], and Richter-Gebert [29] do so in slightly

[1] A problem is $\mathcal{C}$-*hard* for a complexity class $\mathcal{C}$ if any problem in $\mathcal{C}$ reduces to it; the problem is $\mathcal{C}$-*complete* if it is $\mathcal{C}$-hard, and belongs to $\mathcal{C}$; membership in $\mathcal{C}$ is typically easy, so the emphasis tends to be on hardness results.

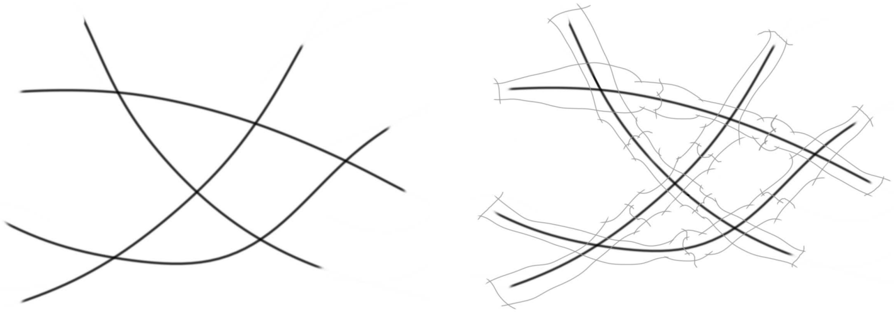

Figure 2: Pseudo-segments, based on the pseudoline arrangement in Figure 1 *(left)*, and the same with additional pseudo-segments (in gray) encoding the topology of the pseudoline arrangement as an intersection graph of pseudo-segments (curves) *(right)*.

different ways, but all constructions are linear.[2] To move information between the gadgets then forces unbounded interaction between the various gadgets, which, so far, has precluded results along the lines of the current paper.[3]

Overview of Contributions

Our main result, Theorem 1 in Section 2, shows that a partial variant of the abstract order type problem (dual to the stretchability problem) remains ∃ℝ-complete (details to be explained later). In this variant, the various, partial abstract order types, only interact in a fixed, bounded number of points. This allows us to apply Theorem 1 to establish ∃ℝ-completeness of various graph-drawing results for fixed parameters such as degree or number of crossings. We keep the, somewhat lengthy, proof of Theorem 1 separate, it can be found in Section 6; we strongly recommend the reader be familiar with the standard ∃ℝ-completeness proof for stretchability before tackling that section (the necessary background can be found in [27, 29], for example).

As a nearly direct consequence of Theorem 1, we can strengthen the result by Kratochvíl and Matoušek and show that testing whether a bounded-degree graph is a segment intersection graph is ∃ℝ-complete, see Section 5.

For the remaining results, it is more convenient to work with radial systems, a notion introduced recently in [2]. In Section 3 we state Corollary 4, the equivalent of Theorem 1 for radial systems. Using Corollary 4, we can settle the complexity of various other problems, most directly the problem of testing whether a bounded-degree graph with a given rotation system (a cyclic permutation of ends of edges at each vertex) can be realized in a straight-line drawing. The problem is ∃ℝ-complete even for graphs of degree at most 131, see Corollary 5.

Finally, we consider the *k-SGE (simultaneous geometric embedding)* problem: Given k graphs on the same vertex set, can the vertices be placed in the plane so that the straight-line drawing of each graph, by itself, on the vertex set is planar? The problem is known to be ∃ℝ-complete for

[2] The construction is never fully illustrated, because of scaling issues. Matoušek [27, page 26] shows what it looks like to combine two gadgets.

[3] We should point out that it is not impossible that the standard construction can be adapted; this would be the case, for example, if any polynomial can be computed using only a bounded amount of space, that is, storage of intermediate results. While this is unlikely to be true for all polynomials, it may just be true for the class of polynomials achieving universality, but we leave this as an open question.

an unbounded number of graphs [24, 11]. We show in Theorem 6 in Section 4 that the 240-SGE problem is $\exists\mathbb{R}$-complete, and the problem remains $\exists\mathbb{R}$-complete for a fixed number of paths, see Corollary 8.

A Brief on The Existential Theory of the Reals

Let ETR be the set of true sentences in the existential theory of the reals, that is, sentence of the form $(\exists x_1)\cdots(\exists x_n)\ \varphi(x_1,\ldots,x_n)$, where φ is a Boolean formula with atomic conditions of the form $p(x)\circ q(x)$, where $\circ \in \{<,\leq,=,>,\geq\}$, and $p(x)$ and $q(x)$ are polynomials with integer coefficients (in binary) in $x=(x_1,\ldots x_n)$. We define $\exists\mathbb{R}$ as the downward closure of ETR under polynomial-time many-one reductions, rather like the complexity class **NP** is defined from SAT, the satisfiability problem.[4] It is easy to see that $\mathbf{NP} \subseteq \exists\mathbb{R}$, since we can turn a real quantifier into a Boolean quantifier by requiring that $x^2 = x$, and truth of a Boolean formula can be encoded as a polynomial. It is also known that $\exists\mathbb{R} \subseteq \mathbf{PSPACE}$, a highly non-trivial result due to Canny [10].

The complexity class $\exists\mathbb{R}$ had been implicit in several papers, including Shor [33], Kratochvíl and Matoušek [23], Bienstock [5], and Buss, Frandsen, and Shallit [9]; it was more formally introduced in [30, 32]. For a very readable and rich introduction to the existential theory of the reals, we recommend Matoušek [27].

2 Partial Order Type Realizability

If p, q, and r are three points in the plane, then either r lies on the line from p to q, or it lies to the left or right of that line. We call this the *order type (orientation)*, $\chi(p,q,r)$, of (p,q,r) and denote it by 1 for left, 0 for collinear, and -1 for right. The *order type* χ of a larger point-set $P \subset \mathbb{R}^2$ is the collection of order-types of triples of P, that is, $\chi : P^3 \to \{-1,0,1\}$. Order types satisfy some rules, for example, the alternation axiom: $\chi(p,q,r)$ determines χ for any permutation of p, q, and r, by multiplying with the parity of the permutation.

For an (abstract) universe of points $U=\{u_1,\ldots,u_n\}$, we can define an *abstract order type* as any function $\tau : U^3 \to \{-1,0,1\}$. If the points in U can be assigned locations in $\mathbb{R}^2$ so that the order type of the points agrees with the abstract order type τ of the (abstract) points, we say that the abstract order type is *realizable.*[5] An abstract order type is *uniform (simple, in general position)* if no three distinct points are collinear.

We want to introduce the notion of a partial abstract order type, where only some triples are assigned an orientation (order type). We do not do so in full generality, but restrict ourselves to defining partial abstract order types for set-covers of U.

Recall that a *set-cover* $(U_i)_{i\in I}$ *of* U has to satisfy $U = \bigcup_{i\in I} U_i$. We can associate a set cover $(U_i)_{i\in I}$ of U with a family $\tau = (\tau_i)_{i\in I}$ of abstract order types, where $\tau_i : U_i^3 \to \{-1,0,1\}$. In this case, we say that a set of points $P \subset \mathbb{R}^2$ realizes τ, if P_i, the points in P corresponding to U_i, realize τ_i or its reverse, $-\tau_i$. If P_i realizes τ_i for all i, we call this a *strict realization.* We call τ *uniform* if every τ_i, $i \in I$, is uniform. The family τ itself, we call a *partial abstract order type (with respect to set-cover* $(U_i)_{i\in I}$ *of* U*).*

Our main result will be that realizing partial abstract order types remains $\exists\mathbb{R}$-complete, even while we control the interaction between elements of the set-cover. To make this precise, we say

[4] $\exists\mathbb{R}$, just like **NP**, has a machine model, a real RAM machine with certain restrictions [8, 15].

[5] One would typically require that an abstract order type satisfy some basic axioms, such as the alternation axiom mentioned above. Since we are only interested in whether an abstract order type can be realized, at which point all axioms are satisfied automatically, we ignore that issue.

that the set cover $(U_i)_{i \in I}$ of U is *thin* if it satisfies the following conditions:

(i) $|U_i| \in \{28, 36, 40\}$, for every $i \in I$,

(ii) $|U_i \cap U_j| = 12$ if U_i and U_j intersect, for every $i \neq j \in I$,

(iii) for every U_i, $i \in I$, there are at most five other U_j that intersect U_i; at most one of the six sets contains 36 or 40 points; there are at most 92 points in sets intersecting U_i which do not belong to U_i.

With this definition we can state the main result.

Theorem 1 *Testing the realizability of a partial abstract order type $\tau = (\tau_i)_{i \in I}$ associated with a set cover $(U_i)_{i \in I}$ of U is $\exists\mathbb{R}$-complete, even under the following restrictions:*

(i) *τ is uniform;*

(ii) *$(U_i)_{i \in I}$ is thin;*

(iii) *the convex hull of every U_i has at least 4 extreme points;*

(iv) *if τ is realizable, then there is a strict realization in which the convex hull of the points in U_i intersect the convex hull of at most one other U_j disjoint from U_i for every $i \in I$, and we know which U_j has this property;*

(v) *we are given an abstract order type σ so that σ restricted to U_i is τ_i for all $i \in I$, and σ is realizable if τ is.*

Item (iii) refers to the convex hull of U_i; this is well-defined by the abstract order type τ_i on U_i: pq is on the boundary of the convex hull of U_i, if $\tau_i(p, q, r)$ is either at least 0 or at most 0 for all points $r \in U_i$.

In $\exists\mathbb{R}$-completeness results so far, order types are rarely used directly; many examples start with the dual problem, pseudoline stretchability, which we saw in Figure 1. Here we pursue some different routes: in Section 5 we exhibit a reduction which is based on abstract order types via pseudo-segments without using duality; in Sections 3 and 4, we explore radial systems, which are an entirely different way to harness the power of abstract order types.

Following Directions

To round off this section, we include a problem, maybe somewhat artificial, that illustrates a more or less direct application of Theorem 1.[6] Suppose we are given a directed walk, with a direction, (L)eft or (R)ight attached to each inner vertex of the walk (a vertex can appear arbitrarily often along the walk, so, strictly speaking, a direction is assigned to the position of the vertex in the walk, not just the vertex itself). We say a straight-line drawing of the walk *realizes* the directions if for every e, u, f along the walk, f lies to the left or right of the line through e as determined by the direction at u in the walk. We introduce, as far as we know, the *directional walk problem* which asks whether the vertices of the walk can be placed so that the walk realizes the given directions.

Corollary 2 *The directional walk problem is $\exists\mathbb{R}$-complete, even if each edge of the graph is used at most 336 times.*

[6] A somewhat similar problem was recently studied in [14].

Proof: Let σ, τ, and $(U_i)_{i \in I}$, U be as in the statement of Theorem 1. For every 3-element set $\{u, v, w\} \subset U_i$, we create a walk uv, v, vw (choosing v arbitrary), and assign to v, at this point in the walk, the direction $\tau_i((u, v, w))$. We can now combine two walks uv, v, vw and xy, y, yz so that w and x do not belong to the same U_i for any i (most edges are like that). For the combined walk $uv, v, vw, w, wx, x, xy, y, yz$ we define the directions of w and x in the walk by using σ, to which we have access, by property (v). Each edge occurs at most $2 * 38 + 2 * 26 * 5 = 336$ many times. □

What is the complexity of the directional walk problem if each edge occurs at most once? If each vertex occurs only once, the problem becomes trivial.

3 Radial Orderings and Radial Systems

For a set of points $p_1, \ldots, p_n$ in general position in the plane, we can determine the *clockwise radial ordering at* p_1, that is, the (cyclic) permutation of $p_2, \ldots, p_n$ we obtain by rotating a ray at p_1 clockwise through a full turn, while recording which of the p_i the ray intersects. Radial orderings can also be defined for abstract order types, as shown in [2], the paper that introduced radial orderings: Suppose χ is a uniform abstract order type on U. We can then define the *clockwise radial ordering* at any point $u \in U$ as follows: Pick an arbitrary point $v \in U - \{u\}$. Points w with $\chi(v, u, w) = 1$ come before points w with $\chi(v, u, w) = -1$ in the radial ordering at u. Within each of these two groups, a point w then occurs before another point w' in the clockwise radial rotation, if and only if $\chi(w, u, w') = 1$. A *radial system* for U is a collection of radial orderings for each point in a set U. We call two radial systems *equivalent* if they are the same up to reversing radial orderings of a subset $U' \subseteq U$ of points.

Given a set cover $(U_i)_{i \in I}$ of U we can associate with it a family $R = (R_i)_{i \in I}$, where R_i is a radial system for U_i, $i \in I$. We call R a *partial radial system (with respect to set-cover $(U_i)_{i \in I}$ of U).* We say a set of points $P \subset \mathbb{R}^2$ *realizes* R, if P_i, the points in P associated with U_i, have radial system R_i up to equivalence. We say that P *strictly realizes* R, if each P_i has radial system R_i (not up to equivalence).

While the uniform abstract order type uniquely determines its radial system, the reverse is not, in general true (see Figure 3 in [2]). The following, however, is true [2, Theorem 1]:

Theorem 3 (Aichholzer, Cardinal, Kusters, Langerman, Valtr [2]) *Let R be a radial system on a set U equivalent to the radial system of some uniform abstract order type τ. If $|U| \geq 5$, we have the following:*

(i) The size of the convex hull of any realization of R is uniquely determined; it equals the size of the convex hull of τ, and it can be computed efficiently.

(ii) If the convex hull has at least 4 extreme points, then the only uniform abstract order types with radial system R are τ and its reverse, $-\tau$.

Cardinal and Kusters [11] showed that testing whether a radial system is strictly realizable is $\exists\mathbb{R}$-complete. Theorem 3 allows us to restate Theorem 1 for radial systems to obtain a refined version of their result.

Corollary 4 *Testing the realizability of a partial radial system $R = (R_i)_{i \in I}$ associated with set cover $(U_i)_{i \in I}$ of U is $\exists\mathbb{R}$-complete, even under the following restrictions:*

(i) $(U_i)_{i \in I}$ is thin;

(*ii*) *if R is realizable, then there is a strict realization in which the convex hull of the points in U_i intersect the convex hull of at most one other U_j disjoint from U_i for every $i \in I$, and we know which U_j has this property;*

(*iii*) *we are given a radial system S which determines the radial ordering of all points in U with respect to all of U, and so that S induces R on the set cover $(U_i)_{i \in I}$. If R is realizable, then there is a strict realization of S that also satisfies condition (ii): all points are in general position, and the convex hull of the points in U_i intersect the convex hull of at most one other U_j disjoint from U_i for every $i \in I$, and we know which U_j has this property.*

Proof: Let $\tau = (\tau_i)_{i \in I}$, U and $(U_i)_{i \in I}$ of U be as in Theorem 1. As we saw, τ determines a radial system R_i on each U_i. Let $R = (R_i)_{i \in I}$ be the family of these systems. We also have an abstract order type σ, that induces τ on $(U_i)_{i \in I}$. Let S be the radial system on U for all of U determined by σ.

We need to show that R is realizable if and only if τ is, and that R satisfies the restrictions of the corollary. The fact that the U_i are thin follows directly from Theorem 1(*ii*).

If τ is realizable, we know that σ, and therefore, S is realizable, by Theorem 1(*v*). In particular, R is realized, and we know which U_i have intersecting convex hulls. This shows (*iii*).

Let P be a set of points in the plane strictly realizing τ. Since R is defined based on τ, the points in P strictly realize the radial system R, and property (*ii*) of R follows from Theorem 1(*iv*).

On the other hand, if R is realizable, let $P \subset \mathbb{R}^2$ be a set of points realizing R. Let P_i be the points realizing U_i. Since $|P_i| = |U_i| \geq 5$, and the convex hull of each U_i contains at least 4 extreme points, we can invoke Theorem 3 to argue that P_i realizes τ_i or $-\tau_i$, the reverse of τ_i, on U_i. This is how we defined P realizing τ.[7] □

The proof of Corollary 4 has an interesting consequence for a fundamental graph-drawing problem. For a given graph, a *rotation* of a vertex is a cyclic permutation of the edges incident to the vertex. A *rotation system* is the family of rotations for each vertex. A drawing of a graph *realizes* a given rotation system, if the rotation at each vertex corresponds to the clockwise order of the ends of edges at the vertex.

Corollary 5 *Testing straight-line realizability of a graph with rotation system and maximum degree at most* 131 *is* $\exists\mathbb{R}$*-complete.*

Kynčl showed that the straight-line realizability of a complete graph with a given rotation system is $\exists\mathbb{R}$-complete [24]; in comparison, the topological version can be decided in polynomial time for complete graphs [24, 25]. There are results for non-complete graphs (same references), but they are phrased for AT-graphs which, if the graph is not complete, is not the same as specifying the rotation system.

Proof: Let $R = (R_i)_{i \in I}$, $(U_i)_{i \in I}$ and S be as in the statement of Corollary 4. We want to define a graph with rotation system which enforces all the radial systems R_i. Define $G = (U, E)$, where $E = \bigcup_{i \in I} \binom{U_i}{2}$. For each u in U_i we need to define its rotation with respect to all its neighbors; since u can belong to multiple U_i, we work with S restricted to u and its neighbors, to determine the rotation at u. How many neighbors can u have? It has at most $40 - 1 = 39$ neighbors in the set U_i it belongs to; by property (*iii*) of being thin, there are at most 92 other neighbors in U_j which intersect U_i (which they have to, if they also contain u). Therefore, u has at most $39 + 92 = 131$ neighbors. □

[7] Note that we do not have to show that P *strictly* realizes τ, since we are reducing from the realizability problem.

The bound can be lowered to 103 by looking at the actual construction in the proof of Theorem 1 (rather than just working with the definition of thin). It is tempting to speculate that even the case of cubic graphs with rotation system is $\exists\mathbb{R}$-complete, but that would probably require a different approach. Our proof allows rotations at vertices to arbitrarily reverse, and the problem remains $\exists\mathbb{R}$-complete. For degree-3 vertices there are only two rotations, and they are reverses of each other. This stops being true for degree 4.

4 Simultaneous Geometric Embeddings

As a second application of Corollary 4, we consider simultaneous geometric embeddings.

Theorem 6 *The* 240*-SGE problem is* $\exists\mathbb{R}$*-complete.*

The 1-SGE problem is equivalent to planarity testing, which is in **P**. For two graphs, the problem is known to be **NP**-hard [16]. There are some results that place 2-SGE in **P**, under certain additional constraints [18], so it is not impossible to image that 2-SGE lies in **NP**, but it may be $\exists\mathbb{R}$-complete already.

We saw earlier that the unbounded version of the k-SGE problem was first shown to be $\exists\mathbb{R}$-complete by Kynčl [24]. Our proof is closer to the one given by Cardinal and Kusters [11] in that it also works with radial systems. Cardinal and Kusters asked whether the k-SGE problem was **NP**-complete for $k = O(\log n)$. Theorem 6 shows that this is not the case, unless $\exists\mathbb{R} = \textbf{NP}$, which seems unlikely.

Proof of Theorem 6: Let $R = (R_i)_{i\in I}$ and $(U_i)_{i\in I}$ be as in Corollary 4. We define a graph H with vertex set I and edges ij if the convex hulls of U_i and U_j intersect (by property (ii) we can build this graph, even if we do not know whether R is realizable); by (ii), graph H has degree at most 5, so it is 6-colorable; let $c(i)$ be the color of U_i in this coloring. Also, assume the vertices in each U_i are ordered (arbitrarily) as $u_{i,1}, \ldots, u_{i,n_i}$, where $n_i \leq 40$. We color each $u_{i,j}$ with color $(c(i), j)$. Note that the same vertex may belong to different U_i, and may therefore have more than one color.

For every $u_{i,j}$ we create a modified wheel $W_{i,j}$ centered at $u_{i,j}$ with the outer cycle being all the neighbors of $u_{i,j}$ in U_i in the order determined by the radial system R_i, and each outer edge subdivided once.

We now define graphs $G_{c,j}$, $1 \leq c \leq 6$, $1 \leq j \leq 40$, as follows: $G_{c,j} = \bigcup_{i:c(i)=c} W_{i,j}$. This is a family of at most 240 graphs. Each wheel $W_{i,j}$ enforces that $u_{i,j}$ has radial system R_i or its reverse. Each wheel can be realized (by itself), whatever the angles are, because we subdivided the outer edges of the wheel. Two wheels belonging to the same $G_{c,j}$ cannot interfere: Since j is the same, their centers must belong to two different U_i, say U_{i_1} and U_{i_2}, which have the same color: $c(i_1) = c(i_2)$. It follows that the convex hulls of U_i and U_j do not intersect, so the wheels (and their convex hulls) do not intersect. □

The bound of 240 can probably be reduced quite a bit if we take into account the structure of the intersection graph of the U_i, where the large U_i, of size 40, are far from each other.

Remark 7 (Sunflowers and Geometric Thickness) Each edge in the construction belongs to at most two graphs in the family of 240 graphs. Does the problem remain $\exists\mathbb{R}$-complete, if every edge belongs to exactly one graph? In the terminology of simultaneous graph drawing, we would phrase this variant as saying that there are no public edges (edges belonging to more than one graph), all edges are private, they belong to a specific graph. Having no public edges is an extreme case of the sunflower variant, where the set of public edges is the same for all graphs. Can the sunflower variant, with or without public edges be shown $\exists\mathbb{R}$-complete? Why would that be interesting? Such a result could be a step towards showing that the geometric thickness problem is $\exists\mathbb{R}$-complete. The *geometric thickness* of a graph is the smallest k for which the graph has a straight-line drawing in which the edges can be colored with k colors so that there are no crossings between edges of the same color. This is similar to the k-SGE problem, except that we do not have control over the coloring, and every edge belongs to at most one color. It was only shown recently that testing for geometric thickness at most 2 is **NP**-hard [13]. It appears harder to control edges for larger thickness. Is it possible that geometric thickness at most 2 is $\exists\mathbb{R}$-hard? Or does it lie in **NP**? ♦

We can extend our construction slightly to turn all graphs into paths; this is interesting, as one of the first non-trivial results in the area was that any two paths *have* a simultaneous geometric embedding [7]; the same paper also showed that it is not always possible to embed five paths simultaneously (in a geometric embedding).

Corollary 8 *Testing whether* 9360 *paths have a simultaneous geometric embedding is* $\exists\mathbb{R}$*-complete.*

In the proof we make use of the following result: If H is a subgraph of G and $\mathcal{H}$ is a straight-line embedding of H so that G has a plane embedding extending $\mathcal{H}$, then G has a plane embedding in which each edge in $G - H$ is a polygonal chain consisting of at most $O(|V(H)|)$ straight-line segments; this is Theorem 1 from [12].

Proof: Consider the 240 graphs $G_{c,i}$, $1 \leq c \leq 6$, $1 \leq i \leq 40$ constructed in the proof of Theorem 6. Each of these graphs is a disjoint union of (subdivided) wheels on at most 40 vertices (counting the center vertex). Each such wheel can be written as the union of 39 paths which start at the center, use one of the 39 spokes, and then continue clockwise around the perimeter of the wheel as far as possible (all but one edge of the perimeter belong to the path).

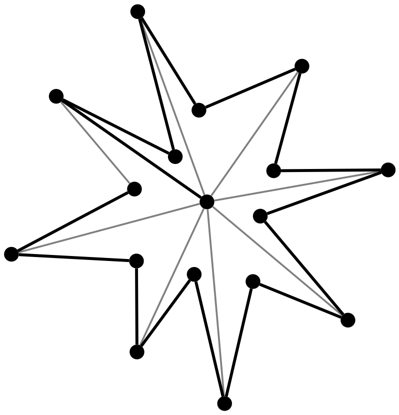

Figure 3: A (subdivided) wheel on eight spokes (gray). One of the eight subpaths that together cover all pairs of independent edges in the wheel is shown in black.

Note that any two independent edges of a wheel belong to one of these at most 39 paths. We can then define $G_{c,i,j}$, $1 \leq j \leq 39$ as follows: replace each wheel in $G_{c,i}$ with the path-device starting with the j-th spoke, and connect these paths into a single path. Such a path is plane, and can be drawn so that is extends the path-device for each wheel; by the discussion before the proof, we can assume that the path is drawn as a polygonal chain if we subdivide the edges used to connect the individual paths sufficiently often (on the order of $O(|V(G)|)$ times). This gives us a collection of $39 * 240 = 9360$ paths. □

5 Stretchability and Intersection Graphs

A family of curves in the plane is called an *arrangement of pseudo-segments*, if every pair of curves crosses at most once. The arrangement is *stretchable* if it is isomorphic to a drawing in which all curves are straight-line segments (an *arrangement of straight-line segments*). Isomorphism here refers to an isomorphism of the plane, so it maintains the topological structure of the arrangement, e.g. in which order the curves cross each other, not just which curves cross each other.

If the curves are x-monotone, from $-\infty$ to ∞, then we speak of an *arrangement of pseudolines*, if every pair of curves crosses exactly once. Such an arrangement is *stretchable* if it is isomorphic to an arrangement of straight lines.

An arrangement is *simple* if at most two pseudo-segments, pseudolines, segments, or lines pass through each point.

Theorem 9 *Testing the realizability of a simple pseudo-segment arrangement in which every pseudo-segment is involved in at most* 72 *crossings is* $\exists\mathbb{R}$*-complete.*

The standard approach to the proof would be to work with the pseudoline arrangement dual to the abstract order type σ from Theorem 1, and then restrict the pseudolines to pseudo-segments. With our approach it seems hard to control the intersections along the resulting pseudo-segments to just the local gadgets. We instead take a more simple-minded approach, that would not work for pseudolines, but works fine for pseudo-segments. The proof modifies the proof of Theorem 1.

Proof: Let $\tau = (\tau_i)_{i\in I}$, σ, U and $(U_i)_{i\in I}$ be as in the proof of Theorem 1 *before* we make the abstract order type uniform. So the illustrations in Figures 4 to 7 accurately reflect what each gadget looks like. Instead of viewing these figures as definitions of abstract order types, we reinterpret them as illustrations of pseudo-segments, one for each line. This already gives us a pseudo-segment arrangement whose stretchability implies that the original gadgets work correctly, and thereby decide realizability of τ. We note that every point has at most seven lines passing through it; the ∞ point of a von Staudt gadget has the most such lines: three from the gadget, and one each from the incoming scale, two incoming variables, and one outgoing variable. We can now use a trick that Shor attributes to Mnëv [28] and a preprint version of [21], but it's just the dual construction introduced by Las Vergnas to make point configurations uniform (see Lemma 3). See Figures 12 and 13 in Shor [33], as well as his Lemma 4. The pseudo-segments in each gadget are constructible (in the sense of constructibility for pseudolines), and chaining gadgets together does not change that. Each pseudo-segments gets replaced with four new, but very close, pseudo-segments. Each such replacement adds at most four crossings along existing pseudo-segments. As we look at the varying gadgets, the line ℓ in the negated addition, and inverted gadgets are involved in the most crossings: 6 within the gadget, and 3 each for the four gadgets transporting information into or away from the gadget, leading to 18 pseudo-segments crossings overall. Since each pseudo-segments gets replaced with at most 4 new pseudo-segments as we uniformize the arrangement, each pseudo-segment has at most 72 crossings with other pseudo-segments. □

The pseudo-segment arrangement problem can be turned into a segment intersection graph problem. The original reduction, due to Kratochvíl and Matoušek [23] was quite complicated; we build on the simplified reduction from [30], which is also described in [27]. The reduction is essentially as shown in Figure 2 except that it is missing the framing device, which can be adapted to not interfere with the conclusion that each pseudo-segment after the reduction has at most three times as many crossings as the original pseudo-segment, or if it is one of the new pseudo-segments, at most three crossings.

Corollary 10 *Realizability of segment intersection graphs of degree at most* 216 *is* $\exists\mathbb{R}$*-complete.*

Essentially the same reduction also establishes the hardness of recognizing intersection graphs of convex sets [30], so we can draw the same conclusion in this case. The $\exists\mathbb{R}$-completeness of the unbounded version was first shown in [30]; Kratochvíl and Matoušek [23] had shown the problem **NP**-hard.

Corollary 11 *Realizability of convex set intersection graphs of degree at most* 216 *is* $\exists\mathbb{R}$*-complete.*

This last result makes it tempting to conjecture that disk or unit disk intersection graphs of bounded degree are $\exists\mathbb{R}$-complete. For graphs of unbounded degree, unit disk intersection graphs are known to be $\exists\mathbb{R}$-complete to recognize [22], while for disk intersection graphs only **NP**-hardness is known [19].

6 Proof of Theorem 1

The $\exists\mathbb{R}$-hardness proof follows the standard outline, with one major deviation, which is the placement of the gadgets. Traditionally, gadgets are placed along a line (two intersecting lines, really), but we will separate them and place them in a grid. We prepare the ground for that placement in Section 6.1. The remaining parts are then relatively standard: We briefly review how to turn polynomials into a normal form for evaluation in Section 6.2, and how the von Staudt gadgets can be used to implement the steps of the evaluation of a polynomial, and the comparison of values in Section 6.3; we also discuss constructibility, a notion we need to make the order type uniform. Finally, in Section 6.4, we show how to combine these elements for the proof.

6.1 Drawing in General Position

To place the gadgets safely in a grid, where safely means that we can control the (abstract) order type of points we place using the grid, we prove a graph-drawing result. In graph-drawing language it states that we can assume that every graph has a subdivision with a 1-plane drawing in general position on a polynomial-size grid. In a 1-*plane* drawing of a graph every edge is involved in at most one crossing.

Lemma 1 *Every graph G on n vertices has a subdivision H so that the vertices of H can be placed on the points of an $O(n^5) \times O(n^5)$ square grid so that the vertices of H are in general position (no line through two vertices of H contains a third vertex of H and no two vertices of H are on the same grid-line), and so that every edge of H has at most one crossing. Each edge of G is subdivided an odd number of times. The placement can be done efficiently.*

Since the placement is efficient, the abstract order type of the vertices of H can be computed in polynomial time.

We combine two results from the literature to prove Lemma 1; the first is due to Biedl and Kaufmann [3]. In an *orthogonal* drawing, edges follow the grid, with bends at grid-points only. In a 1-*bend drawing* every edge has exactly one bend (so it looks like an L). Since grid-points have at most four incoming edges, we relax the representation of a vertex, and allow it to be drawn as an axis-parallel rectangle in the grid (the corners being grid-points). Edges attach to the outside of the rectangle, and there are no vertices or edges inside the rectangle.

Theorem 12 (Biedl, Kaufmann [3]) *Every connected graph on n vertices and m edges has a* 1-*bend orthogonal drawing on an $\frac{n+m}{2} \times \frac{n+m}{2}$ grid. Each vertex is represented by a rectangle of perimeter at most twice its degree.*

The second result we need is proved—in passing—in Brass, *et al.* [7, p.128-129]. The *edge resolution* of a drawing is the smallest distance between a vertex and a non-incident edge in the drawing. The result requires *unit edge resolution*, where a unit is the distance between two neighboring grid-points.

Theorem 13 (Brass, *et al.* [7]) *Suppose H has a planar straight-line drawing in an $O(k) \times O(k)$ grid which has unit edge resolution. Then there is a straight-line drawing of H in an $O(k^2) \times O(k^2)$ grid in which no three vertices of H are collinear and no two vertices of H lie on the same grid-line. The new drawing of H can be found in polynomial time, and does not depend on the edges in H, just the position of its vertices.*

Proof of Lemma 1: We can assume that G is connected. Theorem 12 gives us a 1-bend drawing of G in an $O(n^2) \times O(n^2)$ grid. Crossings between two edges occur at grid-points. We subdivide the four segments coming together in each crossing close to the crossing. Tripling the number of gridlines is sufficient to make sure the subdivision vertices are grid-points; we also replace the bend in each edge by a vertex.

We assume that every rectangle has height and width at least 3. If that is not the case, we can extend the rectangle slightly and add a new grid-line (or -lines) to ensure the rectangle has sufficient width and height. We can then choose a grid-point in the interior of each rectangle as the location of the vertex represented by that rectangle. We connect it by straight-line segments to the ends of edges along the rectangle's boundary, creating a new vertex at each such point.

This results in a grid drawing of a subdivision H of G with all vertices placed on grid-points, and so that every crossing occurs between a horizontal and a vertical edge, and no edge is involved in more than one crossing. Also, each edge of G was subdivided an odd number of times, once for the initial 1-bend drawing, and twice for each crossing removed, and once for each of the two endpoints.

To apply Theorem 13 we need to ensure unit edge resolution of the drawing. Outside the rectangles, we have unit edge resolution, since the drawing is orthogonal. Inside the rectangles we may have violations though: two edges from the center vertex to two boundary vertices of the rectangle may get too close to each others end-vertices. To deal with these vertices, we refine the grid by a factor of n (separating any two gridlines by n new gridlines). This achieves unit edge resolution of the drawing.

We still cannot apply Theorem 13 to H directly, since H is not planar, however, we can apply the theorem to the empty graph on the vertices of H. Since the redrawing performed by Theorem 13 depends only on the vertices of H and their location, not the edges which are present, any planar subgraph of H remains planar in the redrawing. Therefore, the redrawing does not introduce new crossings into the drawing. We conclude that if we straight-line draw H in the new grid, this drawing of H still has at most one crossing per edge, and the vertices of H are in general position. □

6.2 Evaluating Polynomials

To prove $\exists\mathbb{R}$-hardness, we will work with the problem STRICT INEQ, which consists of strict inequalities between polynomials. To encode this problem we will need a way to express evaluating a polynomial and comparing the values of two polynomials using abstract order types. This can be done using the von Staudt gadgets, as we will see in Section 6.3, but to apply these gadgets, we need to perform the polynomial evaluation carefully. Suppose, for example, we have to evaluate $X + Y$ from X and Y. If we do not know which of X and Y is larger, we cannot build the abstract order type for the corresponding von Staudt gadget. Here is what looks to be an even worse problem: suppose we have to decide whether there are X, Y, Z so that $X^2 + (Z - Y)Y - XY^2 > 0$. Even if we could calculate the value of the expression $X^2 + (Z - Y)Y - XY^2$, how could we compare it to 0, if, to build the gadget we already need to know whether it is smaller or larger than 0? Which also depends on the values of X, Y, and Z. There have been various, slightly different solutions for this (Mnëv [28], Shor [33]); we follow the approach taken by Richter-Gebert [29], also used by Matoušek [27].

A program in *Richter-Gebert normal form (RG-NF)* for a family of strict inequalities in variables $X_1, \ldots, X_n$ consists of a sequence of statements of the form

- $V_1 = 1$; and
- $V_i = X_j$ for some j, where $X_j > 1$; or
- $V_i = -V_j$, for $j < i$, where $V_j > 1$ or $j = 1$; or
- $V_i = (-V_j) + V_k$, for some $j, k < i$, where $V_j < 0$ and $V_k > 1$ or $k = 1$; or
- $V_i = 1/V_j$, for $j < i$, where $V_j > 1$; or
- $V_i = (1/V_j) * V_k$, for some $j, k < i$, where $0 < V_j < 1$ and $V_k > 1$;

together with a set of conditions $V_i < V_j$ for some pairs (i, j) with $1 \leq i, j \leq m$ over the computed variables V_i. We say a program in RG-NF is *solvable* if there is an assignment to all the variables X_i, $1 \leq i \leq n$, and V_i, $1 \leq i \leq m$ which satisfies all statements and restrictions, and all conditions.

Lemma 2 (Richter-Gebert [29]) *Let $(f_i)_{i \in I}$ be a family of (multivariate) polynomials with integer coefficients. We can efficiently construct a program in RG-NF so that the program is solvable if and only if there is a solution $x \in \mathbb{R}^\ell$ to $f_i(x) > 0$ for all $i \in I$.*

Even more is true: any semialgebraic set is "essentially" the solution set of an RG-NF program, and that result is shown in both [29, 27]. For our purposes, we only need equivalence of solvability, which we illustrate with a simple example. Suppose we are given the single strict inequality $XY^2 - 5X > 0$. We replace each variable by a difference of two new variables $X = X' - X''$ and $Y = Y' - Y''$. This allows us to assume that all variables $X', X'', Y', Y'' > 1$. We then order terms so there is no subtraction. The expression $(X' - X'')(Y' - Y'')^2 - 5(X' - X'') > 0$ turns into $X'Y'^2 + X'Y''^2 + 2X''Y'Y'' + 5X'' > X''Y'^2 + X''Y''^2 + 2X'Y'Y'' + 5X'$. To calculate the coefficient 5, we work with the binary presentation: $5 = 1 * 2^2 + 1$. The corresponding RG-NF program would be $V_1 = 1$, $V_2 = -V_1$, $V_3 = (-V_2) + V_1$, which is 2, $V_4 = 1/V_3$, $V_5 = (1/V_4) * V_3$, which is 4, and $V_6 = (-V_2) + V_5$. Similarly, we can evaluate the monomials, and add them.

The main point of the RG-NF is that for every statement we exactly know the ordering of the variables involved, both with respect to each other, and with respect to $-\infty$, 0, and 1. For example, in $V_i = (1/V_j) * V_k$ we assume that $0 < V_j < 1$ and $V_k > 1$, so $V_i > V_k$, and the order is $-\infty < 0 < V_j < 1 < V_k < V_i$; analogous claims are true for all the other statements. This is what makes the construction of the von Staudt gadgets possible, as we will see in the next section.

6.3 Von Staudt Gadgets

Figure 4 shows the two main von Staudt gadgets as adapted by Richter-Gebert [29].

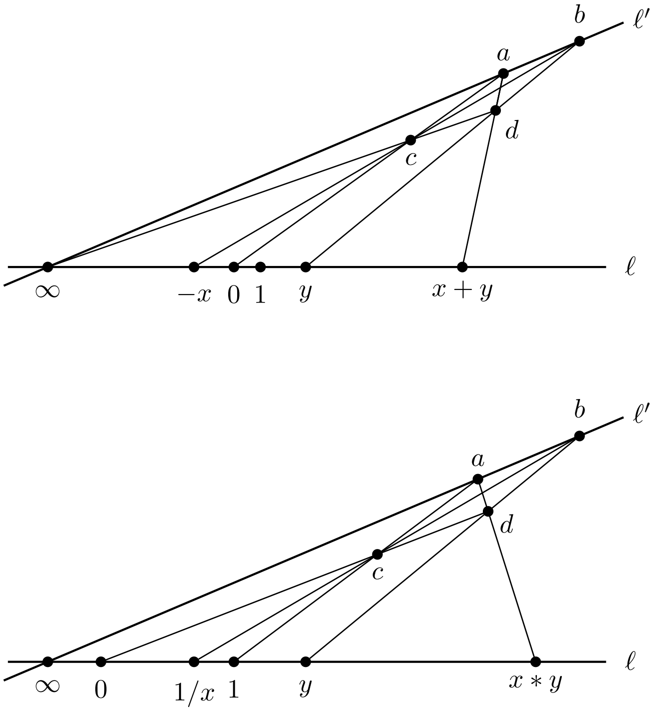

Figure 4: Von Staudt gadgets for negated addition (top) and inverted multiplication (bottom). Based on illustration by Richter-Gebert [29].

The workings of the gadgets are based on the notion of cross-ratios from projective geometry. The *cross-ratio* of four points p, q, r, and s lying on a common line ℓ is defined as

$$(p, q; r, s) := \frac{d(p, r) \cdot d(q, s)}{d(p, s) \cdot d(q, r)},$$

where $d(x, y)$ is the distance between points x and y. The cross-ratio is invariant under projective transformations, and can also be defined if one of the points lies at infinity. Consider the gadget for negated addition. Then $-(-x, 1, 0, \infty) + (y, 1, 0, \infty) = (x + y, 1, 0, \infty)$; in that sense, the point labeled $x + y$ represents the sum of x and y in the projective scale determined by 1, 0, and ∞. Similarly, $1/(1/x, 1, 0, \infty) * (y, 1, 0, \infty) = (x * y, 1, 0, \infty)$, for inverted multiplication.

Gadgets for $V_1 = 1$, and $V_i = X_j$ could be done using three or four points ordered on a line, but we simply consider these special cases of the negated addition gadget. Similarly, $V_i = -V_j$ is a special case of the negated addition gadget, and $V_i = 1/V_j$ a special case of the inverted multiplication gadget. This allows us to treat all these cases with just the two von Staudt gadgets. For $V_i = (-V_j) + V_k$, where $k = 1$ we identify the points y and 1; in that case, the addition gadget has 9 points.

We also need a gadget to compare two values $x, y > 1$, given as $-x$ and y. For that, as Richter-Gebert [29] showed, we can use a slightly modified addition gadget: set y in the gadget to 0, and think of $x + y$ as y, see Figure 5. Let c be the intersection of the line segment between $-x$ and b and a and 0. As we traverse move from 0 to a along the connecting line segment, we encounter c before we encounter the intersection with ∞-to-d if and only if $x < y$.

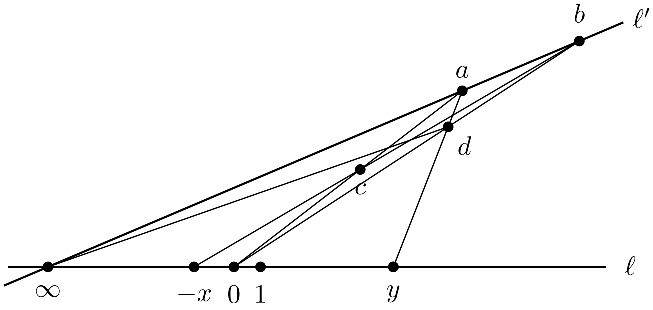

Figure 5: Von Staudt gadgets for ensuring $x < y$. Based on illustration by Richter-Gebert [29].

All von Staudt gadgets can be made arbitrarily flat by moving a and b out along ℓ' and reducing the angle between ℓ' and ℓ. A bit more formally: given a disk D of radius R and $\varepsilon > 0$, there is a projective transformation which places either gadget at an arbitrary point inside D so that all the points of the gadget lie in a disk of radius ε, and any line through two points of the gadget has distance at most ε from ℓ inside D.

Finally, we need a way to transport ratios between point distances from one line (belonging to one gadget) to another line (belonging to another gadget). We do this using the simple inversion gadget shown in Figure 6.

The mappings from the points on ℓ_- to ℓ is projective, and therefore maintain ratios; it does invert the order of the points, but that is not an issue. By moving a arbitrarily close to ℓ, we can make r, s, and t arbitrarily close to each other. To move information farther, we can chain the inversion gadgets, and that can be done in two different ways: with two inversion gadgets having their a-points close to the same line ℓ, as in the top of Figure 7, or with the next inversion gadget having its a-point close to the next line, as in the bottom of Figure 7.

Note that when chaining the gadget, the next gadget does not have to lie on the same side of ℓ as the previous gadget, it may lie on the opposite side, namely when ℓ_- and ℓ_+ are on opposite

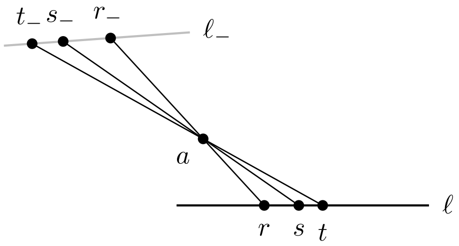

Figure 6: Moving ratios between lines. We have $d(r,s)/d(s,t) = d(r_-,s_-)/d(s_-,t_-)$.

sides of ℓ (this will be well-defined). If we have control of the points on ℓ_-, we can then make sure that a, b, r, s, and t lie within a disk of given radius ε and any line through two of these points has distance at most ε from ℓ, wherever we place the gadget in the disk D we use for the construction. The same is true for the points r_+, s_+, t_+ or any points on a later chained gadget.

We need one final ingredient to complete the proof. The gadgets we are using work with collinear points, which we need to avoid. This cannot always be done, but it is possible for our construction. The main result we need we take from Matoušek [27], but it can also be found in Mnëv [28] and Shor [33] (Richter-Gebert [29] does not need constructibility for his purposes).

An abstract order type τ over U is *constructible*, if the elements of U can be arranged in a linear order $u_1, \ldots, u_n$ so that (i) no three points in $\{u_1, \ldots, u_4\}$ are collinear according to τ, and (ii) every u_i, $i > 4$, lies on at most two lines through points in $\{u_1, \ldots, u_{i-1}\}$, according to τ.

Las Vergnas [26] described a method to turn a constructible order type into an equivalent (in terms of realizability) uniform one. A formal proof that the method works, seems to have first been given in [21]; that proof can also be found in [6, Proposition 8.6.3]; finally, Matoušek [27] also sketches a proof.

Lemma 3 (Las Vergnas [26]) *For every constructible abstract order type τ over U there is a uniform abstract order type τ' over U' so that τ is realizable if and only τ' is. Moreover, $|U'| \leq 4|U| - 3$.*

In the construction all but the first three points get replaced by four new points which lie "close-by". The von Staudt gadgets are well-known to be constructible, but since we are chaining them differently from the standard construction, let us include some more detail.

Lemma 4 *The von Staudt and inversion gadgets are constructible.*

Proof: For the negated addition gadget, we can order points as ∞, $-x$, 0, 1, y, b, d, c, a, $x+y$. Similarly, for the inverted multiplication gadget, we use ∞, 0, $1/x$, 1, y, b, d, c, a, $x * y$. For the comparison gadget, we use ∞, $-x$, 0, 1, y, b, d, a, c.

For the inversion gadget, we have t_-, s_-, r_-, r, s, a, t, or any permutation of the r_-, s_-, t_- and any permutation of r, s, and t. □

6.4 Putting Pieces Together

We have all the ingredients required to prove Theorem 1. Clearly, the realizability problem described in the theorem lies in $\exists\mathbb{R}$, so we are left with the proof of $\exists\mathbb{R}$-hardness.

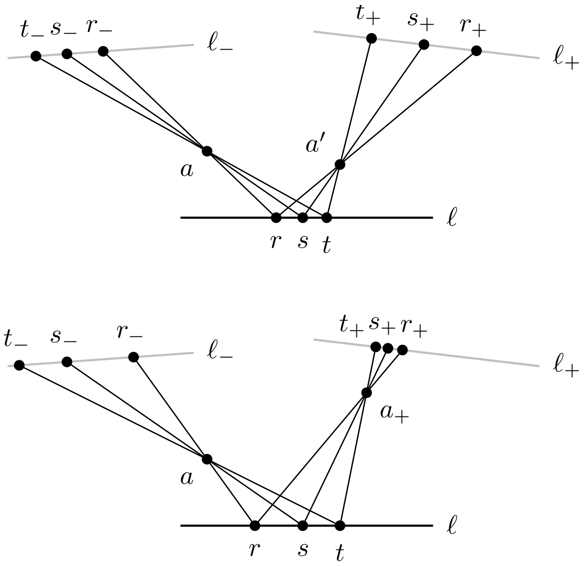

Figure 7: Chaining two gadgets. In opposite direction (above), and the same direction (below). We have $d(r,s)/d(s,t) = d(r_-,s_-)/d(s_-,t_-) = d(r_+,s_+)/d(s_+,t_+)$.

We reduce from the problem STRICT INEQ, which is well-known to be $\exists\mathbb{R}$-hard (see, for example [32]). In the STRICT INEQ problem we are given a family $(f_i)_{i\in I}$ of multivariate polynomials and ask whether there is an $x \in \mathbb{R}^n$ so that $f_i(x) > 0$ for all $i \in I$.

We saw in Lemma 2 that STRICT INEQ reduces to deciding whether the corresponding RG-NF program is solvable. So we may as well suppose we have a program in RG-NF with underlying variables $X_1, \dots, X_n$, computed variables $V_1, \dots, V_m$, and conditions $V_i < V_j$ for some $1 \le i, j \le m$. From the program we create a directed graph G as follows: We start with a special vertex s, which will encode the scale for ∞, 0, and 1. We then create a new vertex in G for each underlying variable X_i, for each statement computing a variable V_i, and for each condition $V_i < V_j$ in the program. We need to connect s to all vertices corresponding to computed variables and conditions (not to vertices corresponding to underlying variables); these are the vertices that contain $-\infty$, 0, and 1. We do so by making these vertices leaves of a binary tree rooted in s (we need to add the new vertices of the tree to G). For any two vertices u and v so that u corresponds to a statement computing a variable which is used in v, we add an edge uv. Then v either corresponds to a computation of another variable that uses the variable computed in u, or it is a comparison involving the variable computed in u. By the definition of RG-NF, G is a directed acyclic graph. See Figure 8 for an example.

By Lemma 1 there is a subdivision H of G so that the vertices of H can be placed on the points of a polynomial-size grid with no two points on the same grid-line and no three points collinear. Moreover, every edge in G corresponds to a path of even length in H. We can assume that each path has length at least 4 (in the proof of Lemma 1, each path starts out as having length 2, but each edge is then subdivided n times, giving length at least $2(n+1) \ge 4$).

Our goal now is to construct a partial abstract order type τ, as described in the statement of the theorem. It is sufficient to construct σ and specify the set cover U, since together they induce

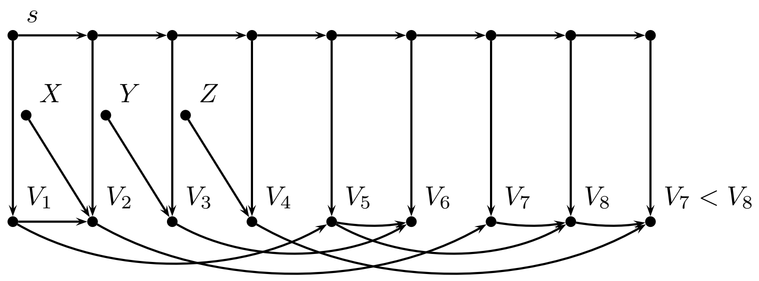

Figure 8: The inequality $X(Y+1) > Z$, where $X, Y, Z > 1$, translates into RG-NF $V_1 = 1$, $V_2 = X$, $V_3 = Y$, $V_4 = Z$, $V_5 = -V_1$, $V_6 = -V_5 + V_3$, $V_7 = 1/V_2$, $V_8 = 1/V_7 * V_5$ and $V_4 < V_8$. A directed, acyclic graph G corresponding to this program is shown.

τ. To describe σ we will start by choosing locations for each gadget in a geometric grid, and aligning them in that location. Roughly speaking, the local order type of each gadget, together with the abstract order type forced by the geometry of the grid will give us σ.

We start with a sufficiently large square grid to place the vertices of H as points in the grid, as in Lemma 1. We can assume that the whole grid lies inside a disk D of radius R. Vertices of H are assigned to grid-points. Suppose v is a vertex of H that belongs to G. To each v we assign a gadget.

If v is one of the interior vertices of the binary tree rooted at s, we create three collinear vertices labeled ∞, 0 and 1. If v corresponds to the underlying variable X_i, we create three collinear vertices labeled ∞, 1, X_i. Both these cases, we can view as part of an inversion gadget with the three vertices lying on a line ℓ.

If v corresponds to a statement or a condition, there is a corresponding von Staudt gadget for each type. We place each gadget so that its line ℓ lies on the horizontal grid-line through grid-point v, just to the right of v. We know that there is an $\varepsilon > 0$ so that all points in the gadget for v lie in an ε-neighborhood of v and any line through two points of the gadget has distance at most ε from ℓ inside D; since no other vertices will be placed on the gridline, these lines do not come close to any other vertices of H.

The remaining vertices of H are vertices subdividing an edge of G. Let P be the path in H resulting from subdividing an edge uv in G. By assumption, P has even length at least 4. We replace P with a chain of inversion gadgets, each edge resulting in one inversion gadget. Each inversion gadget starts and ends close to two grid-points containing points from other inversion gadgets, or, at the end of P, points of a von Staudt gadget. We saw that there are two ways to chain consecutive inversion gadgets. We orient all inversion gadgets towards the next vertex of P (that is, the a-point of a gadget connecting x to y is close to the next vertex, y), with the exception of the last edge of P where we place the point a close to the last P-vertex before v. That vertex then has two a-points (from consecutive inversion gadgets) close to it, just as pictured in the top of Figure 7.

The path P connects two vertices u, v in G, propagating the scale of three points (α, β, γ), that is, we ensure that $d(\alpha, \beta)/d(\beta, \gamma)$ remains unchanged. If u is an interior vertex of the binary tree rooted at s, we are propagating the scale of $(\alpha, \beta, \gamma) = (\infty, 0, 1)$ from u to v; if u corresponds to an underlying variable X_i, we are propagating $(\alpha, \beta, \gamma) = (\infty, 1, X_i)$ from u to v. Otherwise,

u corresponds to a variable V_i which is computed in u and used in v. In that case, we propagate (α, β, γ), where the triple contains ∞, 0 and V_i in the right order (instead of ∞ and 0 we could choose any two of $\{\infty, 0, 1\}$). In all cases, v corresponds to a van Staudt gadget, and we identify the three of its vertices labeled α, β, and γ with the corresponding three vertices of the inversion gadget which are not close to a. Each inversion gadget in P reverses the order of the points (α, β, γ) along the line. Since P has even length, the two gadgets corresponding to vertices u and v have their points (α, β, γ) in the same order.

For each of the inversion gadgets, we can choose an $\varepsilon > 0$ so that the points of the gadget lie in an ε-neighborhood of the grid-vertex it has been assigned to, with ℓ lying along the grid-line to the right of the vertex, and a slightly above, or below the line.

Lines passing through two points of an inversion gadget either lie on the grid-line, along which ℓ is placed, or they pass through a and a point on ℓ. In the later case, the line passes through a point in another gadget, and it does not pass within a distance of ε of any other grid-point containing a vertex of H.

With this set-up, we can now define the abstract order type σ. So we are given three points and need to define the chirotope for those three points. We distinguish three cases.

The three points are close to three distinct grid-points. In this case, the geometry of the grid determines the abstract order type of the vertices.

Two of the points are close to one grid-point, the third point to a different grid-point. Look at the two points close to the same grid-point. If these two points belong to the same gadget, we promised that a line through those two points will be arbitrarily close to ℓ within the disk D, so the abstract order type of the three points is determined by the geometry of the grid-points. If the two points belong to different gadgets, they must both be a-points of chained inversion gadgets. In the later construction, we will ensure that the line through the two a-points intersects passes through the shared grid-point of the two inversion gadgets, and otherwise remains close to ℓ, which determines the order type of the two a and the third vertex (remember that no two vertices of H lie on the same horizontal grid-line).

The three points are close to the same grid-point. If all three points belong to the same gadget, then this follows, because the abstract order type of the von Staudt and inversion gadgets is fully determined. So two of the points must be a-points from an inversion gadget, and the third point is a shared r, s, t-vertex. As we said in the previous case, we will ensure that the line through the two a-points passes through the shared grid-vertex of the two gadgets, again determining the order type of the three points.

For the i-th vertex of H, we let U_i be the set of points involved in the gadget. This defines τ from σ.

We claim the following are true:

(a) If the program is solvable, then σ is realizable.

(b) If τ is realized, then the program is solvable.

Part (a). If the program is solvable, let $(X_1, \ldots, X_n) = (x_1, \ldots, x_n) \in \mathbb{R}^n$ be a solution. This determines a value $\text{val}(V_i)$ for each variable V_i of the program. Each von Staudt gadget corresponding to a statement of the program or a condition involves at most 6 points on a line together with at most 4 additional points.

We can then find an $\varepsilon > 0$ so that the following is true. We can place the points of each von Staudt gadget in a disk of diameter less than ε so that the cross-ratios corresponding to points labeled as V_i correctly represent their underlying value (that is, if p_{V_i} represents V_i in the gadget, then $(p_{V_i}, 1, 0, \infty) = \text{val}(V_i)$), and so that if the gadget is placed anywhere inside of D (the disk of radius R containing the whole grid), then any line through two points of the gadget has distance at most ε from the line through ℓ inside of D (we can do this by moving a and b out along ℓ' and reducing the angle between ℓ and ℓ'). We also require that $d(\infty, 0) = d(0, 1) = d(\infty, 1)/2$; this allows us to work with the inversion gadgets which propagate ratios, not cross-ratios.

For each grid-point that has been assigned a von Staudt gadget, we then place that gadget just to the right of the grid-point, so ℓ comes to lie on the horizontal line through the grid-point, and all points of the gadget are within distance less than ε from the grid-point. We also do this for the interior vertices of the binary tree rooted at s, which propagate the scale. For these vertices, we also ensure that $d(\infty, 0) = d(0, 1) = d(\infty, 1)/2$.

We are left with placing the inversion gadgets. We remember that each path in H between two vertices of G has even length at least 4. Consider a specific path P. Each inversion gadget is assigned to a grid-point, and the last two gadgets belonging to P are assigned to the same grid-point, y say. Suppose P starts at p and ends with xyz. We place the t-point of each gadget within an ε-neighborhood of the grid-point assigned to the gadget. Draw a polygonal chain P_t, following the t-points from the gadget corresponding to p via the inversion gadgets to z. See Figure 9.

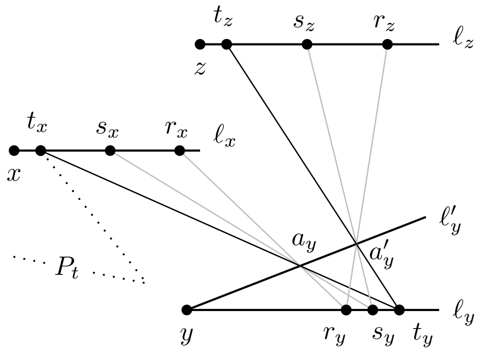

Figure 9: The end of path P_t (thin black edges) at grid-points x, y, and z, with the two inversion gadgets assigned to y. In this case, both gadgets lie on the same side of ℓ_y. *Note:* In an actual drawing, ℓ_x, ℓ_y, and ℓ_z would not overlap horizontally.

For y let us consider the case that the two gadgets sharing y are on the same side of ℓ (this is the situation depicted in Figure 9). Let ℓ' be a line through the grid-point y. This line intersects P_t in two points, these will be our a and a' control points for the two gadgets at y. As we reduce the angle between ℓ and ℓ', this defines r-, and s-points at y and x. By making the angle arbitrarily small, we can ensure that these points are within an ε-neighborhood of their grid-points, and arbitrarily close together. Moreover, we can ensure that the line through a and a' is within distance ε from ℓ inside D. We can now choose a-points for the part of P connecting p to x. We can make all of these arbitrarily close to their ℓ-lines, and we can ensure that when the gadgets meet at x, the points have exactly the same distance (whichever side is too large can be made smaller by controlling either the a-points or ℓ'). If two a-points belonging to y are on opposite sides of ℓ, we proceed essentially the same way, except we have two lines ℓ' and ℓ' we can control to make the line through a and a' arbitrarily close to ℓ.

This gives us a point configuration realizing σ, and, thereby, τ.

Part (b). This is because the von Staudt gadgets mimic the algebraic computation correctly, individually. The inversion gadgets ensure that all the von Staudt gadgets use the same scale, with $d(\infty, 0) = d(0, 1)$, which makes it possible to use the simple inversion gadgets to propagate values between arbitrary von Staudt gadgets, and do so correctly. This then follows a standard argument. Note that we do not have to assume that all of σ is realized; τ is sufficient, since then each gadget works. It follows, then, that the program is solvable, and then, from part (a) we can conclude that even σ is realizable.

By Lemma 3 all of the gadgets are constructible, but we still need to ensure that when we combine them, constructibility is maintained. For that, consider any of the paths uPv in H, where u and v are vertices of G. We will just consider the last two inversion gadgets, as shown in Figure 7 (left), since the preceding gadgets can be added similarly. If u is an interior vertex of the binary tree rooted at s, the special vertex, we are propagating the scale from u to v; this includes the case that v is a leaf of the tree, corresponding to a von Staudt gadget. The scale points get propagated to v for the first time, so we can order the points as $r_-, s_-, t_-, t, s, a, r, t_+, s_+, a', r_+$. If u is one of the remaining vertices, then the von Staudt gadget already contains two of three vertices we are propagating; without loss of generality, let those be t_+ and s_+. We can then order the points as $r_-, s_-, t_-, t, s, a, r, a', r_+$. We conclude that σ over U is constructible, so we can obtain uniform abstract order types τ' and σ' which are realizable if and only if τ and σ are, respectively. Let the domain of τ'_i be U'_i. We know by Lemma 3, that $|U'_i| \leq 4|U_i|$; duplicating points if necessary, we can assume that $|U'_i| = 4|U_i|$.

We are left with verifying properties $(ii) - (iv)$ of the Theorem, repeated here:

(ii) $(U_i)_{i \in I}$ is thin;

(iii) the convex hull of every U_i has at least 4 extreme points;

(iv) if τ is realizable, then there is a strict realization in which the convex hull of the points in U_i intersect the convex hull of at most one other U_j disjoint from U_i for every $i \in I$, and we know which U_j has this property;

By inspection, we can verify the following facts about the gadgets used in constructing τ and σ: Inversion gadgets consist of 7 points, von Staudt gadgets have 9 or 10 points. Any two gadgets that intersect do so in exactly 3 points. Von Staudt gadgets share points with at most four other gadgets, all of them inversion gadgets: one for the scale, two for in-coming variables, and one for the computed, out-going variable, so there are at most $4 * 4 = 16$ points in the inversion gadgets not belonging to the von Staudt gadget. Most inversion gadgets intersect two or three other inversion gadgets (depending on whether they are on a path or in the binary tree at the special vertex), with the exception of the inversion gadgets that intersect a von Staudt gadget (corresponding to the last interior vertex on a path P). That inversion gadget on U_i intersects the von Staudt gadget, the three other inversion gadgets intersecting the von Staudt gadget, and the inversion gadget preceding it on P, for a total of five U_j. These U_j contain at most $4 + 7 + 3 * 4 = 23$ points not belonging to U_i. When constructing τ' and σ' from τ and σ, each of the numbers goes up by a factor of 4, which implies that the partition of U' induced by τ' is thin.

To see (iii), we inspect each gadget; the von Staudt gadgets have three vertices on their convex hull; after making these gadgets uniform, this ends up being $3 + 3 = 9 > 4$ vertices. The inversion gadgets start with four extreme vertices on their convex hull, so they already satisfy (iii), and they will even have $3 * 4 = 12 > 4$ after making the order type uniform.

If the convex hulls of U_i and U_j intersect, but U_i and U_j are disjoint, then U_i and U_j must correspond to two inversion gadgets whose corresponding edges in H cross in the drawing of H. By the way we drew H this can happen for at most one j for each i, proving (iv).

7 Conclusion

Are there other examples where $\exists\mathbb{R}$-complete problems remain $\exists\mathbb{R}$-complete even if one of the parameters is small and fixed? Building on the work in this paper, the author shows that deciding whether a graph has a straight-line drawing with local crossing number at most 867 is $\exists\mathbb{R}$-complete [31]. A negative example is the planar slope number; as Hoffmann [20] showed, the problem is $\exists\mathbb{R}$-complete, but turns **NP**-complete if the number of slopes is fixed (based on a similar result for segment intersection graphs by Kratochvíl and Matoušek [23] and extending an earlier result for two slopes by Garg and Tammassia [17]).

Other problems which may be worth investigating include geometric thickness, see Remark 7, recently shown **NP**-hard for two layers [13], and (unit) disk intersection graphs of bounded degree. Another intriguing candidate is the recent result by Abrahamsen, Miltzow, and Seiferth [1] that shows that packing puzzles are $\exists\mathbb{R}$-complete. Does this remain true for a fixed number of (types of) pieces, maybe even a single type of piece?

Returning to the results of the current paper, one can ask, whether the values of k can be improved further? More precisely, can the parameters in Theorem 1 and Corollary 4 be lowered? There is some flexibility in the construction that could probably be exploited to get set covers of U that are thinner than the ones we defined as thin; also, as in the case of Theorem 9, one may not need a uniform abstract order type, in which case the parameters reduce by a factor of 4. Are there other ways to approach the construction, other gadgets, that lead of even smaller parameters?

References

[1] M. Abrahamsen, T. Miltzow, and N. Seiferth. Framework for $\exists\mathbb{R}$-completeness of two-dimensional packing problems. *ArXiv e-prints*, 2020. arXiv:2004.07558 (last accessed 12/18/2020).

[2] O. Aichholzer, J. Cardinal, V. Kusters, S. Langerman, and P. Valtr. Reconstructing point set order types from radial orderings. *Internat. J. Comput. Geom. Appl.*, 26(3-4):167–184, 2016. doi:10.1142/S0218195916600037.

[3] T. C. Biedl and M. Kaufmann. Area-efficient static and incremental graph drawings. In *Algorithms—ESA '97 (Graz)*, volume 1284 of *Lecture Notes in Comput. Sci.*, pages 37–52. Springer, Berlin, 1997. doi:10.1007/3-540-63397-9_4.

[4] N. Bieker. Complexity of graph drawing problems in relation to the existential theory of the reals. Bachelor's thesis, Karlsruher Institut für Technologie, Karlsruhe, 2020. https://i11www.iti.kit.edu/_media/teaching/theses/ba-bieker-20.pdf (last accessed 12/17/2020).

[5] D. Bienstock. Some provably hard crossing number problems. *Discrete Comput. Geom.*, 6(5):443–459, 1991. doi:10.1007/BF02574701.

[6] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G. M. Ziegler. *Oriented matroids*, volume 46 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, second edition, 1999. doi:10.1017/CBO9780511586507.

[7] P. Brass, E. Cenek, C. A. Duncan, A. Efrat, C. Erten, D. P. Ismailescu, S. G. Kobourov, A. Lubiw, and J. S. B. Mitchell. On simultaneous planar graph embeddings. *Comput. Geom.*, 36(2):117–130, 2007. doi:10.1016/j.comgeo.2006.05.006.

[8] P. Bürgisser and F. Cucker. Counting complexity classes for numeric computations. II. Algebraic and semialgebraic sets. *J. Complexity*, 22(2):147–191, 2006. doi:10.1016/j.jco.2005.11.001.

[9] J. F. Buss, G. S. Frandsen, and J. O. Shallit. The computational complexity of some problems of linear algebra. *J. Comput. System Sci.*, 58(3):572–596, 1999. doi:10.1006/jcss.1998.1608.

[10] J. Canny. Some algebraic and geometric computations in PSPACE. In *STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 460–469, New York, NY, USA, 1988. ACM. doi:10.1145/62212.62257.

[11] J. Cardinal and V. Kusters. The complexity of simultaneous geometric graph embedding. *J. Graph Algorithms Appl.*, 19(1):259–272, 2015. doi:10.7155/jgaa.00356.

[12] T. M. Chan, F. Frati, C. Gutwenger, A. Lubiw, P. Mutzel, and M. Schaefer. Drawing partially embedded and simultaneously planar graphs. *J. Graph Algorithms Appl.*, 19(2):681–706, 2015. doi:10.7155/jgaa.00375.

[13] S. Durocher, E. Gethner, and D. Mondal. Thickness and colorability of geometric graphs. *Comput. Geom.*, 56:1–18, 2016.

[14] A. Efrat, R. Fulek, S. Kobourov, and C. D. Tóth. Polygons with prescribed angles in 2D and 3D. In *28th International Symposium on Graph Drawing and Network Visualization, GD'20 (Vancouver, BC)*, 2020.

[15] J. Erickson, I. van der Hoog, and T. Miltzow. Smoothing the gap between NP and $\exists\mathbb{R}$. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1022–1033. IEEE, 2020. doi:10.1109/FOCS46700.2020.00099.

[16] A. Estrella-Balderrama, E. Gassner, M. Jünger, M. Percan, M. Schaefer, and M. Schulz. Simultaneous geometric graph embeddings. In *Graph drawing*, volume 4875 of *Lecture Notes in Comput. Sci.*, pages 280–290. Springer, Berlin, 2008. doi:10.1007/978-3-540-77537-9_28.

[17] A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.*, 31(2):601–625, 2001. doi:10.1137/S0097539794277123.

[18] L. Grilli, S.-H. Hong, J. Kratochvíl, and I. Rutter. Drawing simultaneously embedded graphs with few bends. In *Graph drawing*, volume 8871 of *Lecture Notes in Comput. Sci.*, pages 40–51. Springer, Heidelberg, 2014. doi:10.1007/978-3-662-45803-7_4.

[19] P. Hliněný and J. Kratochvíl. Representing graphs by disks and balls (a survey of recognition-complexity results). *Discrete Math.*, 229:101–124, 2001. doi:10.1016/S0012-365X(00)00204-1.

[20] U. Hoffmann. On the complexity of the planar slope number problem. *J. Graph Algorithms Appl.*, 21(2):183–193, 2017. doi:10.7155/jgaa.00411.

[21] B. Jaggi, P. Mani-Levitska, B. Sturmfels, and N. White. Uniform oriented matroids without the isotopy property. *Discrete Comput. Geom.*, 4(2):97–100, 1989. doi:10.1007/BF02187717.

[22] R. J. Kang and T. Müller. Sphere and dot product representations of graphs. *Discrete Comput. Geom.*, 47(3):548–568, 2012. doi:10.1007/s00454-012-9394-8.

[23] J. Kratochvíl and J. Matoušek. NP-hardness results for intersection graphs. *Comment. Math. Univ. Carolin.*, 30(4):761–773, 1989.

[24] J. Kynčl. Simple realizability of complete abstract topological graphs in P. *Discrete Comput. Geom.*, 45(3):383–399, 2011. doi:10.1007/s00454-010-9320-x.

[25] J. Kynčl. Simple realizability of complete abstract topological graphs simplified. *Discrete Comput. Geom.*, 64(1):1–27, 2020. doi:10.1007/s00454-020-00204-0.

[26] M. Las Vergnas. Order properties of lines in the plane and a conjecture of G. Ringel. *J. Combin. Theory Ser. B*, 41(2):246–249, 1986. doi:10.1016/0095-8956(86)90048-1.

[27] J. Matoušek. Intersection graphs of segments and $\exists\mathbb{R}$. *ArXiv e-prints*, 2014. arXiv:1406.2636 (last accessed 6/10/2020).

[28] N. E. Mnëv. The universality theorems on the classification problem of configuration varieties and convex polytopes varieties. In *Topology and geometry—Rohlin Seminar*, volume 1346 of *Lecture Notes in Math.*, pages 527–543. Springer, Berlin, 1988. doi:10.1007/BFb0082792.

[29] J. Richter-Gebert. Mnëv's universality theorem revisited. *Sém. Lothar. Combin.*, 34:15, 1995.

[30] M. Schaefer. Complexity of some geometric and topological problems. In *Graph drawing*, volume 5849 of *Lecture Notes in Comput. Sci.*, pages 334–344. Springer, Berlin, 2010. doi:10.1007/978-3-642-11805-0_32.

[31] M. Schaefer. Complexity of geometric k-planarity for fixed k. *J. Graph Algorithms Appl.*, 25(1):29–41, 2021. doi:10.7155/jgaa.00548.

[32] M. Schaefer and D. Štefankovič. Fixed points, Nash equilibria, and the existential theory of the reals. *Theory Comput. Syst.*, 60(2):172–193, 2017. doi:10.1007/s00224-015-9662-0.

[33] P. W. Shor. Stretchability of pseudolines is NP-hard. In *Applied geometry and discrete mathematics*, volume 4 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 531–554. Amer. Math. Soc., Providence, RI, 1991. doi:10.1090/dimacs/004/41.

[34] Wikipedia contributors. Existential theory of the reals — Wikipedia, the free encyclopedia, 2020. https://en.wikipedia.org/w/index.php?title=Existential_theory_of_the_reals (last accessed 12/17/2020).