Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 27, no. 9, pp. 803–841 (2023)
DOI: 10.7155/jgaa.00646

RAC-Drawability is $\exists\mathbb{R}$-complete and Related Results

Marcus Schaefer

School of Computing, Chicago, Illinois 60604, USA

Submitted: July 2022	Reviewed: April 2023	Revised: July 2023
Accepted: November 2023	Final: November 2023	Published: December 2023
Article type: Regular Paper	Communicated by: A. Symvonis	

Expert: Alright, let's leave aside the color for the moment. You had something there also relating to perpendicularity?
Client: Seven lines, all strictly perpendicular.
Expert: To what?
Client: Um, to everything. Among themselves. I assumed you know what perpendicular lines are like!

The Expert [8]

Abstract. A RAC-*drawing* of a graph is a straight-line drawing in which every crossing occurs at a right angle. We show that deciding whether a graph has a RAC-drawing is as hard as the existential theory of the reals, even if we know that every edge is involved in at most eleven crossings and even if the drawing is specified up to isomorphism. The problem remains hard if the crossing angles are only required to be very close (doubly-exponentially so) to being right angles.

We also show that if a graph has a RAC-drawing in which every edge has at most one bend, then such a drawing can be placed on an integer grid of double-exponential area. This is in contrast to RAC-drawability on the grid which turns out to be as hard as the existential theory of the rationals.

1 Introduction

If we cannot avoid crossings in drawings, then we prefer crossings at which the (straight-line) edges cross at large angles. The *crossing angle* at a crossing is the minimum of the two angles formed by the edges at the crossing. Experiments have shown that large crossing angles simplify reading a drawing, as edges become easier to follow individually [29, 28]. In a 2009 paper, Didimo, Eades and Liotta [19] formalized this idea by introducing RAC-drawings of graphs, in which only the largest possible crossing angle, the right angle, is allowed. A straight-line drawing of a graph is a RAC-*drawing* if all crossing angles are right angles, and a graph is RAC-*drawable* if it has a RAC-drawing. Figure 1 shows that the Petersen graph is RAC-drawable.

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

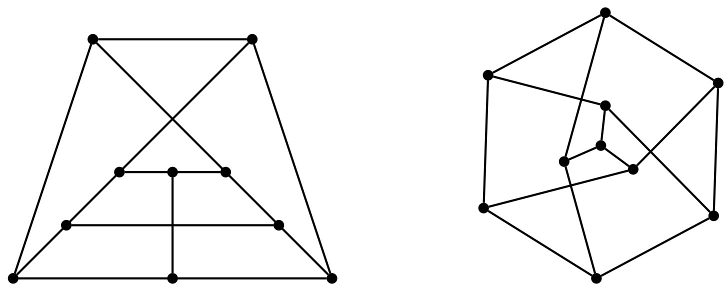

Figure 1: Two RAC-drawings of the Petersen graph, the left one is crossing-minimal.

RAC-drawings have become a popular subject in the graph drawing literature, see, for example, a recent survey by Didimo [18] summarizing our knowledge. We are specifically interested in the computational complexity of the recognition problem, that is, given a graph, how hard is it to tell whether the graph has a RAC-drawing?

Argyriou, Bekos, and Symvonis [6] showed early on that it is **NP**-hard to recognize whether a graph has a RAC-drawing. Why **NP**-hard, and not **NP**-complete? The issue is that a realization of a RAC-drawing may require real coordinates, and, a priori, we do not have any bounds on the precision required for those coordinates and we do not even know whether the graph can be realized on a grid. Bieker [11] showed that the problem lies in $\exists\mathbb{R}$, the complexity class associated with deciding the truth of the existential theory of the reals; we will introduce this class in Section 2. The exact complexity remained open (as mentioned, for example, in [21, p. 4:11/12]).

Also open, not even known to be **NP**-hard, was the complexity of the *fixed embedding* variant of RAC-drawability, in which we are given a drawing of the graph and have to decide whether the graph has a RAC-drawing isomorphic to the given drawing. The following result settles the computational complexity of all of these variants.

Theorem 1 *Testing whether a graph (with or without fixed embedding) has a* RAC-*drawing is* $\exists\mathbb{R}$-*complete, even if each edge has at most eleven crossings.*

$\exists\mathbb{R}$-hardness implies **NP**-hardness [41], so the fixed embedding variant is **NP**-hard as well.

What does $\exists\mathbb{R}$-hardness add that **NP**-hardness does not already give us? Perhaps nothing, since it is possible (but considered unlikely) that $\mathbf{NP} = \exists\mathbb{R}$. Nevertheless, $\exists\mathbb{R}$-hardness reductions are typically very geometric and can be used to obtain stronger conclusions. For example, we show in Section 6 that Theorem 1 remains true for crossings angles sufficiently close (asymptotically) to right angles. Another common consequence of $\exists\mathbb{R}$-hardness results is that problem instances may require algebraically complex solutions, and RAC-drawings are no exception:

(*i*) realizing a RAC-drawing may require double-exponential area (assuming all points have at least unit distance from each other), see Corollary 10;

(*ii*) not every RAC-drawing can be realized on an integer grid, see Corollary 11; and

(*iii*) testing whether a graph has a RAC-drawing on an integer grid is as hard as $\exists\mathbb{Q}$, the existential theory of the rationals, see Corollary 14.

Item (*iii*) implies that there are (at this point at least) no algorithms for testing RAC-realizability on a grid, since $\exists\mathbb{Q}$ is not known to be decidable. These are the types of results that do not follow from **NP**-hardness.

While these results are purely negative, we can also use the theory to establish positive consequences of $\exists\mathbb{R}$-membership. For example, Corollary 12 shows that a RAC-drawing, if it is realizable, can be realized in double-exponential area, showing that item (*i*) above is tight. The double-exponential area upper bound also applies to RAC$_k$-drawings, in which every edge is allowed to have up to k bends. For $k = 1$ we sharpen the area upper bound in Theorem 15 which shows that if a graph has a RAC$_1$-drawing, then it has a RAC$_1$-drawing on an integer grid of double-exponential size. In view of the results on RAC-drawings this is a bit surprising and it requires some delicate perturbation arguments combined with non-trivial bounds from real algebraic geometry.

1.1 Area Requirements and Bends

As far as we know Corollary 10, the double-exponential lower bound on the area of a RAC-drawing, is a new result, but there have been (single) exponential lower bounds in constrained settings, e.g. for upward RAC-drawings [5], RAC-drawings in which a given horizontal order of the vertices must be realized, and for 1-plane RAC-drawings [13], drawings in the plane with at most one crossing per edge.

Allowing bends along edges changes the situation dramatically. A RAC$_k$-drawing of a graph is a RAC-drawing in which every edge has at most k bends. RAC-drawings are just the RAC$_0$-drawings.

Every graph has a RAC$_3$-drawing [19], but not necessarily a RAC$_2$-drawing: any graph with a RAC$_2$-drawing has at most linearly many edges [7]. The complexity of recognizing graphs with RAC$_1$- and RAC$_2$-drawings remains intriguingly open [21, Problem 6]. Neither of these problems is even known to be **NP**-hard.

There are polynomial upper bounds on the area of RAC$_k$-drawings for $k \geq 3$ [24], but no bounds seem to be known for $k = 1, 2$. Standard arguments give double-exponential upper bounds in these cases, see Corollary 12. $\exists\mathbb{R}$-hardness of these cases would likely imply matching double-exponential lower bounds as it does for $k = 0$.

In Theorem 15 we strengthen Corollary 12 in the case $k = 1$ to show that RAC$_1$-drawings if they are realizable, can always be realized on a grid of double-exponential area.

1.2 Overview of the Paper

Our proof that the RAC-drawability problem is $\exists\mathbb{R}$-complete consists of a sequence of two reductions. There is no convenient graph drawing problem to reduce from, so in Section 2 we present an $\exists\mathbb{R}$-complete algebraic problem which we will use as the starting point.

For the intermediate drawing problem we introduce *junctions*, vertices with a specific type of angle constraints on incident edges. The proof of Theorem 1 then breaks into two major parts. The first part shows that drawing a graph with junction constraints is an $\exists\mathbb{R}$-complete problem (even for crossing-free drawings), since it is powerful enough to encode the algebraic problem, see Section 3. The second part shows how to simulate junctions within the RAC-model; this is done in Section 4.

In Section 5 we discuss area requirements and grid-drawings, in RAC-drawings with and without bends. Finally, Section 6 discusses a relaxation of RAC-drawings in which all crossings angles of a drawing have to be above a certain bound. We close the paper with a short list of open questions.

2 The Existential Theory of the Reals

The *existential theory of the reals*, ETR, is the set of existentially quantified, true statements over the real numbers, allowing arithmetical operations $+, -, \cdot$, constants 0 and 1, comparisons $<, \leq, =$ and the Boolean connectives. For example,

$$\exists x : x \cdot x = 1 + 1$$

and

$$\exists x_1, \ldots, x_n : x_1 = 1 + 1 \wedge x_2 = x_1 \cdot x_1 \wedge x_3 = x_2 \cdot x_2 \wedge \ldots \wedge x_n = x_{n-1} \cdot x_{n-1}$$

are both sentences in ETR, while $\exists x : x \cdot x + 1 = 0$ is not (since it is false over $\mathbb{R}$). The first sentence expresses the existence of the square root of 2; the second sentence uses repeated squaring to compute the double-exponential value $2^{2^{n-1}}$, giving a glimpse of the expressive power of ETR.

We can define a complexity class $\exists\mathbb{R}$ from ETR as the set of languages that polynomial-time many-one reduce to ETR. This is analogous to defining **NP** from Boolean formula satisfiability, though there also is a machine model for $\exists\mathbb{R}$ [23]. A problem is $\exists\mathbb{R}$-*hard* if every problem in $\exists\mathbb{R}$ reduces to it; it is $\exists\mathbb{R}$-*complete*, if it is $\exists\mathbb{R}$-hard and lies in $\exists\mathbb{R}$.

We start with a specific $\exists\mathbb{R}$-completeness result which occurs in several early papers [12, 14], for a recent proof see [40, Corollary 4.2].

Theorem 2 *Testing whether a polynomial $f : \mathbb{R}^n \to \mathbb{R}$ with integer coefficients has a real zero is $\exists\mathbb{R}$-complete.*

The problem lies in $\exists\mathbb{R}$, since it can be written as $\exists x \in \mathbb{R}^n : f(x) = 0$, the tricky part is reducing ETR to it.

$\exists\mathbb{R}$ captures the complexity of many natural problems in graph drawing and computational geometry, since it can express conditions on real coordinates. For example, Lemma 4 will show the details of how to express RAC-drawability in the language of $\exists\mathbb{R}$.

Some recent problems shown $\exists\mathbb{R}$-complete relevant to graph drawing and computational geometry include[1]: the art gallery problem [2], recognizing geometric hypergraphs [10], matroid realizability [31], packing problems [33], and covering polygons by triangles [1].

As we mentioned, **NP** $\subseteq \exists\mathbb{R}$ [41], in particular, $\exists\mathbb{R}$-hard problems are also **NP**-hard. On the other hand, $\exists\mathbb{R} \subseteq$ **PSPACE** [15], so $\exists\mathbb{R}$-complete problems are solvable in polynomial space (and, therefore, exponential time).[2]

2.1 An $\exists\mathbb{R}$-Complete Problem

For our initial reduction we will be working with an $\exists\mathbb{R}$-complete problem, which, as is often the case, is tailor-made for the situation we find ourselves in.

[1] These are some of the results which have been published since 2022. Bieker's thesis [11] surveys many of the relevant graph drawing results. The Wikipedia page mentioned earlier [42] is host to a growing list of complete problems, including areas other than graph drawing and computational geometry.

[2] For a thorough introduction to the existential theory of the reals, check out [32]. For a quick intro, the Wikipedia page [42] will serve.

Theorem 3 *The following problem is $\exists\mathbb{R}$-complete: Given equations of the form $x_i = 2$, $x_i = x_j$, $x_i = x_j + x_k$, and $x_i = x_j \cdot x_k$ for variables $x_1, \ldots, x_n$, decide whether the equations have a solution with $x_i > 1$ for all $i \in [n]$.*

The construction in the proof combines ideas from Mnëv [34], Shor [41], and Richter-Gebert [36]; a worked example follows the proof.

Proof: By Theorem 2, testing whether a given polynomial f with integer coefficients has a zero is $\exists\mathbb{R}$-complete. Multiplying the polynomial by 2 we can ensure that all coefficients of f are even numbers. We then replace every variable x of f with the difference of two new variables $x' - x''$. Then f has a zero if and only if the new polynomial has a zero in which all variables are greater than 1 (since any x can be written as the difference of two numbers greater than 1). We move negative terms to the other side of the equation to obtain two polynomials g, h with positive, even coefficients so that f has a zero if and only if $g = h$ has a solution with all variables being greater than 1.

We now calculate the values of g and h from the original variables step by step using (at most polynomially many) new intermediary variables and only using equalities of the type $x_i = 2$, $x_i = x_j + x_k$ and $x_i = x_j \cdot x_k$. This includes the coefficients, which, since they are even, can be built from 2 in polynomially many steps by following the binary encoding of the coefficient. All the new intermediate variables are greater than 1 (as products and sums of values greater than 1). Finally, to test whether $g = h$ we add one more equation $x_i = x_j$, where x_i is the new variable that computes g and x_j the variable that computes h. This completes the proof. □

As an example of the construction described in the proof consider $f(x_1, x_2) = 3x_1^2 - x_2$. We first replace f with $2f$, giving us $6x_1^2 - 2x_2$ with all even coefficients. We then replace each of the variables with a difference of two new variables; in this example (reusing variable names) we get $6(x_1 - x_2)^2 - 2(x_3 - x_4)$. Collecting positive terms gives us $6x_1^2 + 6x_2^2 + 2x_4 = 12x_1x_2 + 2x_3$, so $g(x_1, x_2, x_3, x_4) = 6x_1^2 + 6x_2^2 + 2x_4$ and $h(x_1, x_2, x_3, x_4) = 12x_1x_2 + 2x_3$ in this example. We then compute g and h term by term, using additional variables. E.g., let us show how to compute the term $6x_1^2$: we add equations $x_5 = x_1 \cdot x_1$, $x_6 = 2$, $x_7 = x_6 \cdot x_6$, $x_8 = x_6 + x_7$, $x_9 = x_8 \cdot x_5$; then x_9 computes $6x_1^2$. Similarly, we can compute all other terms into new variables, and add them up, one at a time, to get g and h. Finally, we need one more equality, $x_i = x_j$, to compare the resulting values.

2.2 Existential Theories of the Rationals

Analogously to the existential theory of the reals, one can define the existential theory of an arbitrary field, and that was first done by Buss, Frandsen, and Shallit [14]. For the purposes of this paper, we need ETQ, the *existential theory of the rationals*. Over $\mathbb{Q}$, the first of the two sample sentences we saw earlier,

$$\exists x : x \cdot x = 1 + 1$$

is false, so it does not belong to ETQ. The second one,

$$\exists x_1, \ldots, x_n : x_1 = 1 + 1 \wedge x_2 = x_1 \cdot x_1 \wedge x_3 = x_2 \cdot x_2 \wedge \ldots \wedge x_n = x_{n-1} \cdot x_{n-1}$$

is still true over the rationals.

The complexity class $\exists\mathbb{Q}$ is then defined as the set of languages that polynomial-time many-one reduce to ETQ, and $\exists\mathbb{Q}$-*completeness* and $\exists\mathbb{Q}$-*hardness* are defined as usual. In analogy to Theorem 2 we have the following result, which traces back to [12, 14].

Theorem 4 *Testing whether a polynomial* $f : \mathbb{R}^n \to \mathbb{R}$ *with integer coefficients has a rational zero is* $\exists\mathbb{Q}$*-complete.*

We do not know much about where $\exists\mathbb{Q}$ lies with respect to traditional complexity classes, we do not even know whether it is decidable. The best result in that respect is due to Poonen [35] who showed that the $\forall\exists$ theory of the rationals is undecidable (sharpening a result by Julia Robinson). The best known, trivial, upper bound on $\exists\mathbb{Q}$ is the halting problem.

We do know that $\exists\mathbb{R} \subseteq \exists\mathbb{Q}$, since deciding whether a system of strict inequalities has a solution over the reals is equivalent to deciding whether there is a rational solution, and the strict inequality problem is $\exists\mathbb{R}$-complete [40, Theorem 4.1]. It follows that $\exists\mathbb{Q}$-hardness implies **NP**-hardness. $\exists\mathbb{Q}$-completeness naturally arises in graph drawing in the context of grid realizations as first observed by Cardinal and Hoffmann [16, 27], and our new example, Corollary 14, will be of the same type.

3 Drawings with Junctions

We introduce two types of special vertices that come with angle and rotation constraints. Recall that the rotation at a vertex in a drawing is the clockwise permutation of edges incident to the vertex.

- a $\top$-*junction* is a vertex v which is incident to three special edges e_1, e_2, e_3. A straight-line drawing *respects* the $\top$-junction if there are right angles between e_1 and e_2 and e_2 and e_3 at v; additional edges at v can occur, at any angle, between e_1 and e_3 (opposite of e_2),
- a $\times$-*junction* is a vertex v which is incident to four special edges e_1, e_2, e_3, e_4. A straight-line drawing *respects* the $\times$-junction if the rotation of the special edges at v is $e_1e_2e_3e_4$, or the reverse, and there are right angles between e_i and e_{i+1}, for $1 \leq i \leq 3$; additional edges may occur, at any angle, inside one of the quadrants, e.g. between e_3 and e_4.

Figure 2 shows both junction types, and how we symbolize them in drawings. Lemma 3 implies that $\times$-junctions can be simulated by $\top$-junctions, so they are not, strictly speaking, necessary, but they do simplify the constructions.

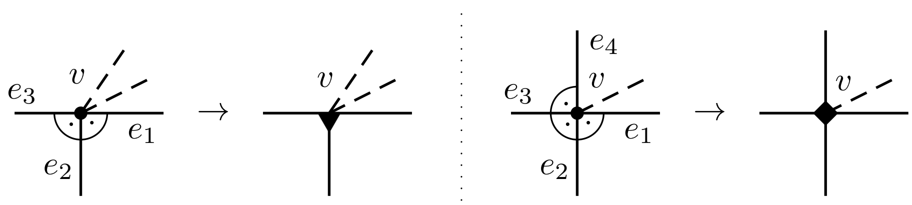

Figure 2: $\top$- and $\times$-junctions, and how we visualize them in graphs using a ▼ and a ⧫. Dashed edges are additional edges at the junction.

If special edge e_2 in a $\top$-junction ends in a leaf, we can think of the junction as a straight-line, e_3e_1, with a vertex on it. Viewing e_1e_3 and e_2e_4 as parts of a line, we will sometimes say that the lines "cross" at the $\times$-junction v, but strictly speaking v is a vertex, and not a crossing in the drawing with junctions.

3.1 The Complexity of Junction Drawings

Two drawings of a graph (with or without junctions) are *isomorphic* if there is a homeomorphism of the plane (which may be orientation-reversing) that maps the graphs to each other.

Theorem 5 *Given a graph G with $\top$- and $\times$-junctions and a plane drawing D of G, it is $\exists\mathbb{R}$-complete to decide whether G has a drawing respecting all junction constraints. The problem remains $\exists\mathbb{R}$-complete even with the following restrictions:*

- (i) *the only non-junction vertices in G have degree 1, all $\times$-junctions have at most one additional edge, and all $\top$-junctions have at most two additional edges, and*
- (ii) *if G does have a straight-line drawing respecting all junction constraints, it has such a drawing D_+ which is isomorphic to D, and*
  - (ii-$\times$) *if a $\times$-junction has an additional edge, it forms an angle of $\pi/4$ in D_+ with the two edges it neighbors in the rotation, and*
  - (ii-$\top$) *if a $\top$-junction has two additional edges, they form a right angle in D_+.*

A drawing of G respecting all junction constraints may potentially have arbitrary (non-RAC) crossings. The theorem implies that this does not have to be the case: if G has a drawing respecting all junction constraints, then G has a drawing respecting all junction constraints without any crossings by (ii), since D_+ is isomorphic to the planar drawing D. It follows that testing whether a graph with $\top$- and $\times$-junctions has a crossing-free drawing is also $\exists\mathbb{R}$-complete, but we need the sharper statement of Theorem 5 for the proof of Theorem 1.

The proof of Theorem 5 can be found in Section 3.3; we prepare for the proof by constructing various gadgets to simulate arithmetic in Section 3.2.

Remark 6 *The inquisitive reader may wonder about conditions (ii-$\times$) and (ii-$\top$) which are very particular, and not used in this paper. They are used in a sequel paper [39] to show that testing the angular resolution of a graph is $\exists\mathbb{R}$-complete.*

3.2 Gadgets

For a reduction from the $\exists\mathbb{R}$-complete problem from Theorem 3 we need to encode equations $x_i = 2$, $x_i = x_j + x_k$, $x_i = x_j \cdot x_k$ and $x_i = x_j$. This requires several gadgets which we will build in this section.

How does a drawing of a graph encode a real number? It is tempting to encode a number x as the absolute distance between two points in a drawing of a gadget, but it does not seem to be possible to maintain absolute distances in junction- or RAC-drawings. Instead we work with relative distances. Given three collinear points (these will be vertices of the gadget) labeled 0, 1 and x, we say x *represents* $(-1)^s d(0, x)/d(0, 1)$, where d is the Euclidean distance of two points, and s is 0 if x lies on the same side of 0 as does 1 (on the common line), and 1 otherwise. We will refer to $d(0, 1)$ as the *scale* of the encoding.

Given collinear points $p_1, \ldots, p_\ell$, where $\ell \geq 3$ and two of the points are labeled 0 and 1, each point represents a number, and we can talk about another set of collinear points $p'_1, \ldots, p'_\ell$ representing the same numbers, if the ratios of the distances are the same for both point sets.

Data will be shared between gadgets using parallel lines (via their relative distances), and we typically visualize gadgets with input points along a horizontal line at the bottom and output points along another horizontal line at the top, e.g. see Figure 4. Data enters and leaves such a gadget as vertical lines. There are a few exceptions: some of the turning gadgets have outputs along a vertical line, e.g. see Figure 5, so data leaves as horizontal lines; some gadgets do not have inputs, like the variable gadget in Figure 3 and some do not have outputs, like the equality gadget in Figure 9.

3.2.1 Gadgets for Creating and Moving Data

Variable Gadget

Figure 3 shows the gadget we use to encode a variable x_i for which we know that $x_i > 1$. The *variable gadget* uses six $\times$-junctions arranged (and connected) in a 2×3 grid; the top row is labeled 0, 1, and x_i in this order (with the e_3-edge of a junction identified with the e_1-edge of the next junction) and the bottom row a, b, c. The e_4-edges of the top row can then be used to send data to another gadget. Any unused edges leaving the gadget are capped with a degree-1 vertex. (We do this for all subsequent gadgets as well, without explicitly mentioning it every time. In the drawings, these unused edges are generally short.)

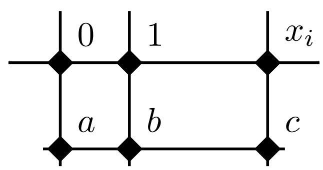

Figure 3: A gadget for variable x_i, with $x_i > 1$.

In a drawing of the x_i-gadget that respects junctions, all e_4-edges must lie on the same side of the line through 0, 1, and x_i, since a, b, and c also lie on a line, and that line is parallel to the first line. In this way, the x_i-gadget can be used to represent any number $x_i > 1$.

By relabeling the junctions in the top row 0, x_i, 1, or x_i, 0, 1, we obtain gadgets for variables between 0 and 1, and variables less than 0.

Copy Gadget and Copying Information

Our next gadget allows us to duplicate information by creating two copies of a point set representing the same numbers. The two copies will even have the same scale as the original set.

The *copy gadget* consists of two $\times$-junctions a and b on a common line (so the e_1 edge of a is the same as the e_3-edge of b). On the e_2-edge of a we have $\top$-junctions $c, p_1, \dots, p_\ell, d$, in this order, and the e_2-edge of b has $\top$-junctions $p'_1, \dots, p'_\ell, d'', c'', p''_1, \dots, p'_\ell$, in this order. We connect c to c'', d to d'', and p_i to both p'_i and p''_i, for each $i \in [\ell]$, and replace each crossing with a $\times$-junction, as shown in Figure 4.

Let us argue that the copy gadget works correctly. The two e_2-edges incident to the $\times$-junctions a and b are parallel, since they are both orthogonal to ab. They also leave ab in the same direction: if they left in opposite directions, then c and c' (for example) are on opposite sides of the line through ab, so the lines through cc' and ab would have to be orthogonal, but then cc' would overlap with ca, which we do not allow. Because of the relative order of the points on the top and bottom line, line cc'' crosses dd' as well as all lines $p_ip'_i$ at a $\times$-junction, so all these lines are

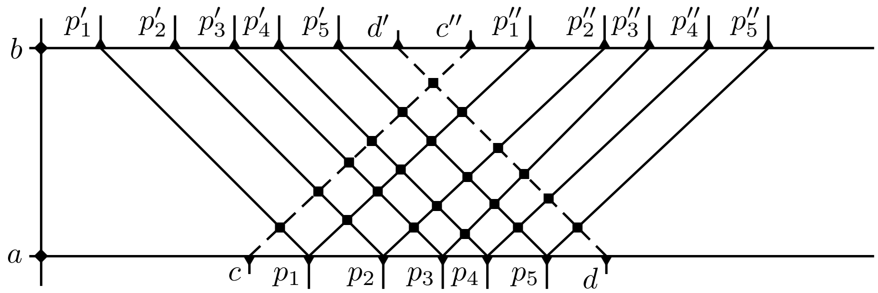

Figure 4: The copy gadget. The dashed lines are orthogonal because they intersect in a $\times$-junction; this forces all the $p_ip'_i$ as well as all the $p_ip''_i$ lines to be parallel as shown.

parallel. And since every line $p_jp''_j$ is crossed by the line dd', these lines are also all parallel. It follows that $d(p_i, p_j) = d(p'_i, p'_j) = d(p''_i, p''_j)$ for all i, j. In particular $p'_1, \dots, p'_\ell$ and $p''_1, \dots, p''_\ell$ both represent the same numbers as $p_1, \dots, p_\ell$. By making $d(a, b)$ large relative to $d(c, d)$ we can move the two copies far apart if necessary.

By chaining copy gadgets, that is, connecting the parallel lines leaving one of the output point-sets, say $p'_1, \dots, p'_\ell$ into the inputs of another copy gadget, we can make an arbitrary number of copies of a collinear set of points.

Turning Gadgets

Consider the *right-turn gadget* shown in Figure 5. It consists of a $\times$-junction v for which the line through the e_1-edge has $\top$-junctions labeled $a, p_1, \dots, p_\ell$, in that order towards v, and the line through the e_4-edge has $\top$-junctions labeled $b, p'_1, \dots, p'_\ell$, again as read towards v. We connect a to b, and p_i to p'_i, for each $i \in [\ell]$. Finally, we add a line intersecting each of ab, $p_1p'_1, \dots, p_\ell p'_\ell$ in a $\times$-junction, in that order. Let c be the $\times$-junction on ab.

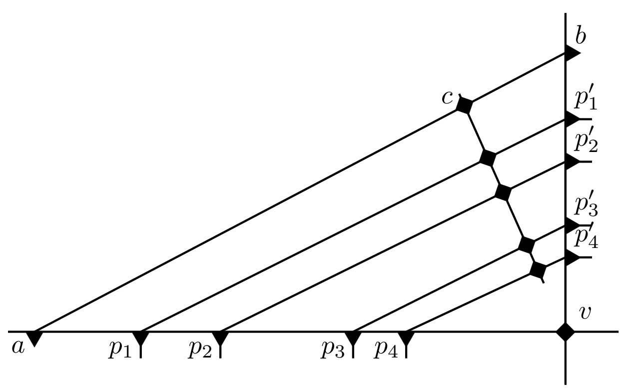

Figure 5: Right-turn gadget making a (scaled) copy $p'_1, \dots, p'_\ell$ of $p_1, \dots, p_\ell$ at a right angle (for $\ell = 4$).

Let us first argue that the right-turn gadget works, in the sense that $p_1, \dots, p_\ell$ and $p'_1, \dots, p'_\ell$ represent the same numbers. The vertices p_i and p'_i are $\top$-junctions, as are a and b, and all the junctions on vc are $\times$-junctions. Since a and b must lie on opposite sides of c, that point lies between lines va and vb. The lines $p_ip'_i$ are parallel (being orthogonal to the line through c), so the p'_i have the same relative distances from each other as do the p_i, and so they represent the same numbers.

The right-turn gadget will play two roles for us. First of all, we can use the right-turn gadget to change the direction in which parallel lines carry data. A single right-turn gadget makes a right-angle turn. By mirroring the right-turn gadget, we obtain a *left-turn gadget*, allowing us to make a left turn. By chaining two right-angle gadgets, that is, by connecting the p'_i of one right-turn gadget to the p_i of a second right-turn gadget, we can achieve a turning angle of π, see Figure 6, and by adding a third right-turn gadget, $3\pi/2$.

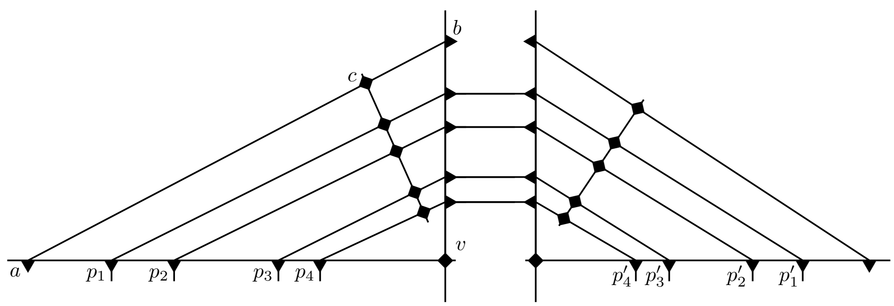

Figure 6: Gadget for a turning angle of π.

The second role the right- and left-turn gadgets play for us, is that they allow us to change the scale of an encoding: The distances $d(a, v)$ and $d(b, v)$ are independent of each other, so we can choose the scale on each axis as needed. In other words, we can arbitrarily scale information (while turning it as well).

3.2.2 Arithmetical Gadgets

Constant Gadget

Let us start with the number 2. The *number* 2 *gadget* is shown in Figure 7; we obtain it by creating a 4×5 grid of $\times$-junctions of which we remove the rightmost junctions in the lower two lines. We label the resulting grid cells like a chessboard, so $A1$ is the lower-left cell and $D3$ is the top-right cell. We add diagonals inside the cells $A1, C1, A3$ and $D3$; we make the "intersection" of each pair of diagonals a $\times$-junction.

Let us consider a drawing of the number 2-gadget respecting the junctions. The $\times$-junctions forming the corners of each grid cell ensure that the shape they bound is a rectangle. The diagonals inside a rectangle with diagonals "intersect" at a right angle, since the intersection is a $\times$-junction, so the sides of the rectangle have the same length, and the cell is bounded by a square.

The four squares with diagonals then force the distance between 0 and 1 to be the same as the distance between 1 and 2: the distance between 1 and 2 is the side-length of square $D3$ which

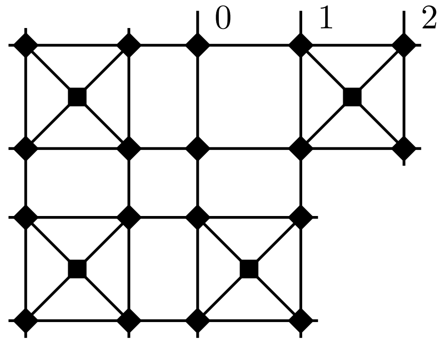

Figure 7: The number 2 gadget.

equals the side-length of square $A3$ which in turn has the same side-length as square $A1$ which finally has the same side-length as $C1$ which equals the distance between 0 and 1. We conclude that the point labeled 2 accurately represents the number 2.

(Why do we not place two squares with diagonals right next to each other? The reason is that our $\times$-junctions only allow additional edges in one quadrant.)

Negation Gadget

We next build a special negation gadget that, for given input points 0, 1, and x creates four output points, $-x$, $-x+1$, 0, and 1 that correctly represent their labels. To build the *negation gadget*, shown in Figure 8, we start with a 3×4-grid of $\times$-junctions, removing the rightmost junction of the bottom line. We add diagonals to cells $A2$ and $C2$ replacing the "intersections" with $\times$-junctions, as we did in the number 2 gadget. We label the three input junctions on the bottom line 0, 1 and x, and the four output junctions on the top line $-x$, $-x+1$, 0 and 1.

In a drawing respecting the junctions, all cells are rectangles, so the distance between inputs 0, 1 and x is the same as the distance between outputs $-x$, $-x+1$, and 0. As we argued in the case of the number 2 gadget, the cells containing diagonals are squares, so the distance between outputs 0 and 1 and outputs $-x$ and $-x+1$ is the same, which also implies that the input and output 0 and 1s have the same distance. It follows that the output points accurately represent their labels.

We use this gadget only for x with $x > 1$, so the sides of the squares do not interfere on the top lines.

Equality Gadget

We can use a modified copy gadget to enforce $x_i = x_j$ for two variables x_i and x_j which we know to be larger than 1.

We assume that for each variable $x \in \{x_i, x_j\}$ we have three incoming lines labeled 0, 1 and x, with all six lines parallel to each other (and oriented the same direction). Figure 9 shows the *equality gadget*; it is made up of an upside-down copy gadget with three points $p_1, \ldots, p_3$ labeled 0, 1 and $x_i = x_j$, and p'_1, p'_2, p'_3 labeled 0, 1, x_i and p''_1, p''_2, p''_3 labeled 0, 1, x_j.

By the properties of the copy gadget, the equality gadget is only realizable (respecting junctions), if x_i and x_j represent the same value larger than 1, and the scales of the two variables

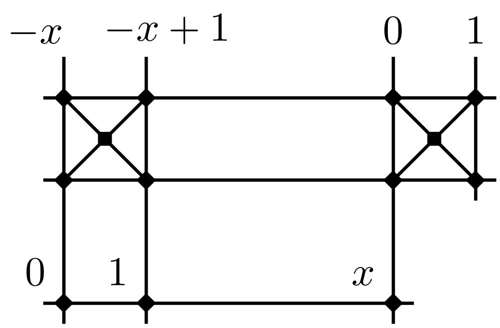

Figure 8: Negation gadget for x representing $-x$ and $-x+1$.

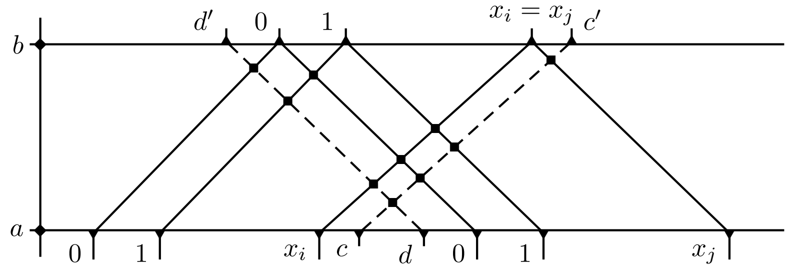

Figure 9: Testing equality $x_i = x_j$ using the copy gadget.

are the same. Since we use turning gadgets to connect gadgets, the second condition is not a restriction, since the turning gadgets can rescale the data.

Addition Gadget

Computing $x_i = x_j + x_k$ appears easy since we can use a modified copy gadget to geometrically add x_j to x_k. A direct implementation of this idea runs into the problem that we do not know which one of x_j or x_k is greater (or if they are the same). This leads to a problem in constructing the gadget, since we need to place the points on a line, which forces a particular order. To avoid this problem we work with the negation gadget to create $-x_j$ and $-x_j + 1$ which we can then geometrically add to x_k. (The idea of working with negation to avoid the ordering problem is due to Richter-Gebert [36], also see Matoušek [32].)

The *addition gadget* is, like the equality gadget, an upside-down copy gadget with the outputs on the top line labeled 0, 1, and $x_j + x_k$ together with some unlabeled points, and the inputs along the bottom line labeled $-x_j$, $-x_x + 1$, 0, 1 in the left half, and 0, 1, x_k in the right half, again with some unlabeled points, as shown in Figure 10. One of the lines incident to $x_j + x_k$ has been dropped, since there is no need for a point corresponding to $x_j + x_k$ in the left input to the gadget.

When using the addition gadget, the x_j-inputs come from a negation gadget for x_j. Assuming that, and assuming that we have a drawing of the addition gadget respecting the junctions, we can argue that the outputs correctly represent their labels. Since the addition gadget is simply

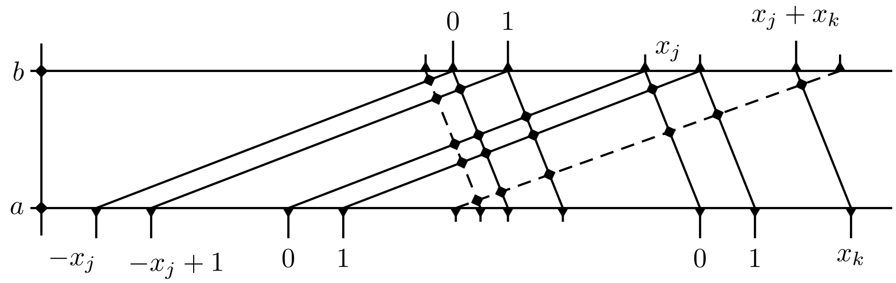

Figure 10: Addition gadget computing $x_i = x_j + x_k$.

an upside-down copy gadget, we argue as before that the two dashed edges are orthogonal, which forces the other lines to be parallel. If we ignore the two dashed edges and their endpoints, the remaining points on top represent $-x_j$, $-x_j + 1$, 0, 1, x_k. If we relabel the first and second points as 0 and 1, then the fifth point represents $x_j + x_k$, and the addition gadget works as needed. Like the equality gadget, it requires that the scales of the two inputs are the same.

Multiplication Gadget

To compute $x_i = x_j \cdot x_k$ we face the same issue we encountered when encoding addition: we do not know which of x_j and x_k is greater, so we cannot place them along the same line. Instead of working with negation, we work with the reciprocal of x_j, which lies (strictly) between 0 and 1 since, as usual, we assume that $x_i > 1$. Given three collinear points labeled 0, 1 and x, simply relabeling them as 0, $1/x$ and 1 (in the same order), gives us the reciprocal of x, see Figure 11.

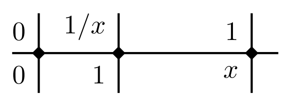

Figure 11: Computing the reciprocal $1/x$ of a variable $x > 1$.

This is cheating a bit, since for most other gadgets (except the turning gadgets) the distance between 0 and 1 (the scale) does not change, and most gadgets assume the distance between 0 and 1 is the same when processing inputs. Since we connect gadgets using turning gadgets, which can rescale arbitrarily, an output 0, 1, x from some other gadget, can be rescaled to 0, $1/x$, 1 to be connected to the multiplication gadget, so we do not even need an explicit gadget computing the reciprocal, we simply relabel the lines.

The product $x_i = x_j \cdot x_k$ can then be calculated using the *multiplication gadget* shown in Figure 12. It is a copy gadget upside down allowing us to merge $1/x_j$ and x_k into a common scale. (We dropped two lines, one incident to output 1 and one incident to output $x_j \cdot x_k$ since there is no need for corresponding input points.)

The points along the top line (ignoring the endpoints of dashed lines which enforce orthogonality) represent 0, $1/x_j$, 1 and x_k, which we can relabel as 0, 1, x_j and $x_j \cdot x_k$, at which point we can drop the x_j to obtain a gadget computing $x_j \cdot x_k$. The multiplication gadget changes the

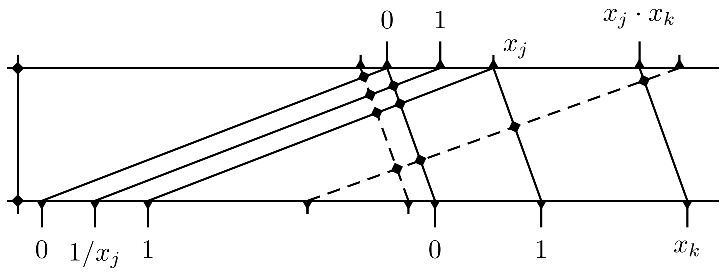

Figure 12: Multiplication gadget computing the product $x_i = x_j \cdot x_k$.

scale which is fine, since when we connect its output to the input of another gadget, the turning gadgets can be used to rescale 0 and 1 to the standard distance.

3.3 Proof of Theorem 5

We can assume that we are given a system of equations over variables x_i, $i \in [n]$ as described in Theorem 3. Our goal is to (efficiently) construct a graph G with junctions, so that the system of equations is solvable, if and only if the graph G has a drawing respecting the junctions. We will also have to show that conditions (i) and (ii) of the theorem are satisfied.

We build G and a plane drawing D of G in several stages. We first create gadgets for the main operations.

- For every variable x_i occurring in the system of equations create a variable gadget.
- For every occurrence of a variable x_i in an equation create a new copy gadget. For every occurrence of the number 2 in an equation, create a number 2 gadget.
- For every equation of the type $x_i = x_j + x_k$ create a new negation- and an addition-gadget and for each equation of the type $x_i = x_j \cdot x_k$ create a new multiplication gadget.
- For every equation, of any of the four types: $x_i = 2$, $x_i = x_j$, $x_i = x_j + x_k$, and $x_i = x_j \cdot x_k$, create an equality gadget.

Let us say this gives us m gadgets in total. Place all m gadgets so that their output points lie along the same vertical line. Incoming and outgoing (information-carrying) edges are then vertical. A (somewhat simplified example) is shown in Figure 13.

We need the following connections between gadgets:

- the output of the variable x_i-gadget needs to be connected as an input to one of the copy gadgets representing it, and all the copy gadgets representing the variable x_i need to be connected to each other; this will leave one free set of output points representing x_i in each of the copy gadgets for x_i,
- if x_i occurs in an equation, we need to connect an unused set of output points representing x_i to one of the inputs of the equality gadget associated with the equation,
- if 2 occurs in an equation, we need to connect the output of an unused number 2 gadget to one of the input sets of the equality gadget representing the equation,

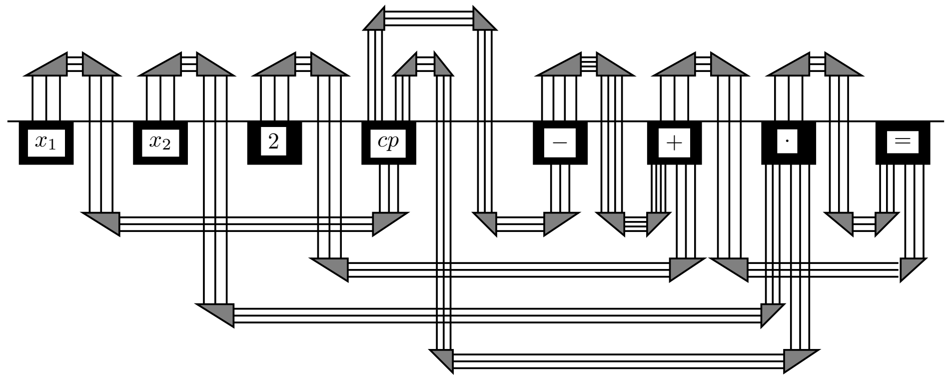

Figure 13: Translating $x_1 + 2 = x_1 \cdot x_2$ into a drawing D with junctions; the black boxes represent gadgets corresponding to their labels, and the gray triangles are right- and left-turn gadgets. Intersections of lines in the drawings are realized as $\times$-junctions (not explicitly shown).

- for every equation of the type $x_i = x_j + x_k$ or $x_i = x_j \cdot x_k$ we connect the unused output of a x_i-gadget and the output of the addition (or multiplication gadget) for the term $x_j + x_k$ ($x_j \cdot x_k$, rsp) to the inputs of an unused equality gadget.

We still need to explain how to connect a set of outputs from gadget α to be an input to gadget β, where $0 \leq \alpha, \beta < m$. This can be done with four turning gadgets: We leave α upwards, turn right twice, and move vertically downwards to the right, and close to, gadget α. We move downwards to a unique level reserved for connecting α and β, say at depth $\alpha + \beta m$ below the common line. We then turn left, move below the inputs of gadget β, and make another left-turn, connecting to β vertically from below. In this way, each connection between two gadgets requires four turning gadgets (two right- and two left-turns).

Connecting the gadgets introduces (orthogonal) crossings between some of the lines connecting the turning gadgets, but we can determine exactly which of these lines cross (independently of whether the gadgets are realizable or not). We replace each such crossing with a $\times$-junction. We obtain a graph G as well as a plane embedding D of G (which may not satisfy the junction-constraints, of course).

If the system of equations is solvable, we can use a solution to create a drawing of each gadget respecting the junctions; moreover, we can assume that in each gadget, the output scale, that is the distance between the outputs labeled 0 and 1 is 1. Place the gadgets along the common line as described by D, moving them sufficiently far apart so they do not interfere. We can then draw the connections between the gadgets as described in D: using the turning gadgets, we can ensure that the connecting lines have the right scale when leaving and entering a gadget, and, in between those two points, are sufficiently close so as not to interfere with any other gadget. Since line intersections were replaced with $\times$-junctions, we have built a drawing D_+ isomorphic to D which respects all junctions.

On the other hand, if there is a drawing of G respecting junctions, all the gadgets work as described, and each occurrence of a variable represents the same value, so there is a solution to the system of equations. This proves (ii), without (ii-$\times$) and (ii-$\top$).

By construction, all vertices of G are $\top$-junctions, $\times$-junctions, or have degree 1. All uses of junctions are illustrated: inspecting Figures 3–12 shows that the ×-junctions used in gadgets have at most one additional edge, resulting from adding diagonals to a rectangle; those edges always form an angle of $\pi/4$ with their neighbors; the only other place we use ×-junctions occurs when connecting gadgets, to remove crossings, as in Figure 13; these ×-junctions do not have additional edges. Similarly, Figures 4–6, 10 and 12 show all the occurrences of ⊤-junctions in gadgets. Inspecting those illustrations shows that all ⊤-junctions in gadgets have at most two additional edges, and in the cases where they do, namely in the copy gadget, Figure 4 and the gadgets based on it, Figures 9, 10 and 12, the two additional edges form a right angle. This proves (i), (ii-×) and (ii-⊤), completing the proof.

Example 7 *Figure 13 illustrates our construction for the simple equational system*

$$x_3 = 2, x_4 = x_1 + x_3, x_5 = x_1 \cdot x_2, \text{and } x_5 = x_6.$$

Since even a simple example like this would require a lot of gadgets, we eliminate some of the variables and reduce the system to the equivalent

$$x_1 + 2 = x_1 \cdot x_2,$$

and we create a junction drawing based on this equation following the ideas of the proof above (rather than all the details). We create variable gadgets for x_1 *and* x_2 *and a gadget for* 2*. We also need a copy gadget, since* x_1 *occurs twice, an addition gadget (with a negation gadget), a multiplication gadget, and one equality gadget. We line them all up along a line, and use the turning gadgets to connect them as shown in Figure 13. For example, the addition gadgets receives as inputs* $-x_1$, $-x_1 + 1$, 0, 1 *(from the negation gadget), as well as* 0, 1, *and* 2, *and the equality gadgets compares the results of* $x_1 + 2$ *to* $x_1 \cdot x_2$*. In the illustration, we did not use a unique level for each connection so as to keep the drawing small.*

3.4 Forcing Empty Faces

When replacing ⊤- and ×-junctions with normal vertices in RAC-drawings, we make use of a restricted drawing model that simplifies the construction. Given a graph G with k pairwise disjoint sets of vertices $V_i \subseteq V(G)$, $i \in [k]$, we are interested in RAC-drawings of G in which the vertices of each V_i lie on the boundary of an empty face, for all $i \in [k]$.

The following theorem shows that this drawing constraint can be removed in RAC-drawings.

Theorem 8 *Let* G *be a graph, and let* $(V_i)_{i\in[k]}$ *be* k *pairwise disjoint sets of vertices of* G*. We can construct, in polynomial time, a graph* G' *so that* G *has a* RAC*-drawing in which the vertices of each* V_i *lie on the boundary of an empty face, if and only if* G' *has a* RAC*-drawing.*

For the proof, we will use that a RAC-drawable graph can have at most $4n - 10$ edges [19].

Proof: Let $v_1, \ldots, v_k$ be k new vertices not in G. Connect v_i to each vertex in V_i by a path of length n, where $n = |V(G)|$, for $1 \leq i \leq k$. Finally, replace each edge of the newly added paths by $\ell = 12n^2$ paths of length 2 (that is, a $K_{2,\ell}$). See Figure 14 for an illustration. Call the resulting graph G'. Suppose G' has a RAC-drawing.

Since we added at most n paths of length n to G, we have at most n^2 of the $K_{2,\ell}$-graphs in G'. We consider them one at a time. Let $E_0 = E(G)$. An edge can cross at most one edge of a $K_{1,\ell}$ at right angles (otherwise, the edges of $K_{1,\ell}$ would overlap). Hence, an edge can cross at

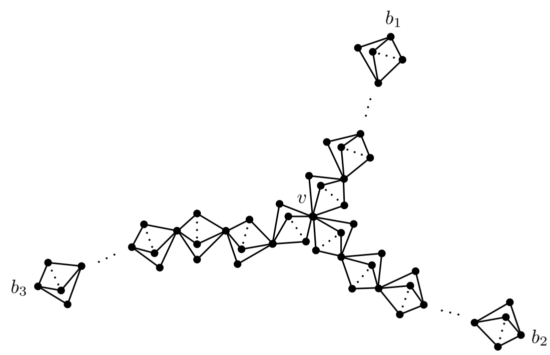

Figure 14: Forcing an empty face; in this example $v = v_i$ and $V_i = \{b_1, b_2, b_3\}$.

most two edges of a $K_{2,\ell}$. Since $\ell > 8n > 2|E_0|$, the first $K_{2,\ell}$ contains a path P_1 of length 2 that crosses none of the edges in E_0. Let $E_1 = E_0 \cup E(P_1)$. Inductively, we obtain $E_i = E_{i-1} \cup E(P_i)$ with $|E_i| < 4n + 2i$. The $(i+1)$-st $K_{2,\ell}$ must contain a path P_{i+1} of length 2 which crosses none of the edges in E_i, since $2|E_i| < 2(4n + 2i) \leq \ell$, for all $1 \leq i \leq n^2$. With P_{i+1}, we can let $E_{i+1} = E_i \cup E(P_{i+1})$.

We conclude that the RAC-drawing of G' contains a RAC-drawing of G together with the vertices v_i and paths from each v_i to every vertex in V_i so that none of the paths are involved in any crossings. In other words, for each i there is a crossing-free (subdivided) wheel with center v_i and a perimeter containing V_i. Removing all edges not belonging to G gives us a RAC-drawing of G in which all vertices of V_i lie on the boundary of the same face (the one that contains v_i).

For the other direction, suppose G has a RAC-drawing in which all vertices of each V_i lie on the boundary of the same face F_i. For each i, we create a new vertex v_i. We place v_i inside and close to the boundary of F_i so that we can connect v_i by a path of length n to each vertex in V_i by closely following the boundary of F_i. We can then replace each edge of the newly added paths ℓ parallel paths of length 2 to obtain a RAC-drawing of G'. □

4 Proof of Theorem 1

$\exists\mathbb{R}$-membership of RAC-drawability was first shown by Bieker [11, Section 6.2]; it also follows from Lemma 4, as discussed after that lemma. To prove $\exists\mathbb{R}$-hardness, we are missing one more ingredient, a way to simulate junctions in RAC-drawings.

Theorem 9 *Let D be a planar drawing of a graph G with some vertices identified as $\top$- and $\times$-junctions. We can efficiently construct a graph G' without junctions, vertex sets $(V_i)_{i\in[k]}$, and a drawing D' of G' in which all vertices of each V_i lie on the boundary of an empty face, so that:*

- *(i) if D is isomorphic to a drawing of G respecting the junctions, then D' is isomorphic to a* RAC-*drawing of G', and all edges are involved in at most eleven crossings,*
- *(ii) if G does not have a drawing respecting the junctions, then G' does not have a* RAC-*drawing in which all the vertices of each V_i lie on the boundary of an empty face.*

The proof of Theorem 9 can be found in Section 4.1. Let us see how this theorem completes the proof of Theorem 1; we first consider the fixed embedding case. By Theorem 5 it is $\exists\mathbb{R}$-hard to test whether a graph G with junctions has a drawing respecting the junctions, even if we know that the graph either has no drawing respecting the junctions, or that it has a crossing-free drawing isomorphic to a given planar drawing D. Using Theorem 9, we construct a graph G' and a drawing D' so that G has a drawing respecting junctions if and only if D' is isomorphic to a RAC-drawing of G'. For the forward direction, we use (i), and for the backward direction (ii) and the fact that D' forces all vertices of each V_i to lie on the boundary of an empty face.

To show that the problem remains $\exists\mathbb{R}$-complete without fixing the embedding, we need to take one more step by enforcing that all the vertices of each V_i lie on the boundary of an empty face without relying on the embedding D'. For this, we apply Theorem 8 to G' and the vertex sets $(V_i)_{i\in[k]}$ to obtain a graph G'', without a fixed embedding. Then G has a drawing respecting junctions if and only if G'' has a RAC-drawing.

4.1 From Junctions to RAC-Drawings

The double cap. We need gadgets to simulate the $\top$- and $\times$-junction restrictions in RAC-drawings. We will base these gadgets on the graph we call the *double-cap* shown in Figure 15. The double-cap consists of two cycles, the *a-cap*, bounded by $a_0, \dots, a_{10}$ and the *b-cap*, bounded by $b_0, \dots, b_{10}$, two center vertices, c_0, c_1, and three *vertices of attachment*, α, β, and γ which will be identified with vertices from other gadgets. Curves a_0a_{10} and b_0b_{10} (shown as dashed in the illustration) are subdivided sufficiently often, so they can be realized as polygonal curves.

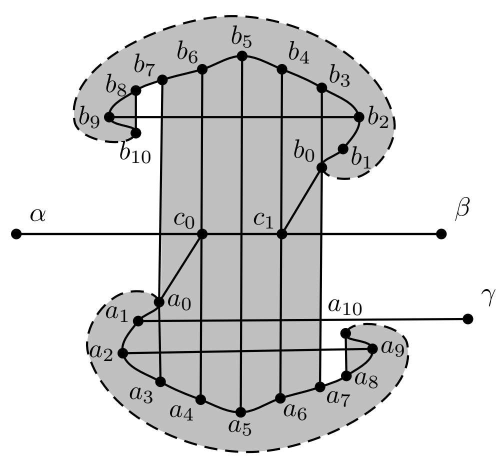

Figure 15: The double-cap with attachments $c_0\alpha$, $c_1\beta$, and $a_1\gamma$. Each of the vertex sets $\{a_0, a_1, \dots, a_{10}\}$ and $\{b_0, b_1, \dots, b_{10}\}$ lies on the boundary of a face. Edges a_0a_{10} and b_0b_{10} (dashed) are subdivided so they can be realized by a polygonal curve. The inside of the double-cap is shaded.

We will argue that the double-cap simulates a line: αc_0 and $c_1\beta$ will be collinear, and $a_1\gamma$ will be parallel to that line; the parallel line will be essential in building the $\top$-gadget below, which simulates a $\top$-junctions.

Assuming the double-cap (without attachments) is drawn as shown in the illustration, we can define the *inside* of the double-cap as the region bounded by $a_0b_7b_8b_{10}b_0a_7a_8a_{10}a_0$; this region is shaded in the illustration. We can then say that a point lies *outside the double-cap* if it does not lie inside the closure of inside of the double-cap.

Lemma 1 *Suppose we have a* RAC*-drawing of the double-cap, in which both* $\{a_0, a_1, \ldots, a_{10}\}$ *and* $\{b_0, b_1, \ldots, b_{10}\}$ *bound an empty face.*

(i) *Without the attachments, the drawing is isomorphic to the drawing shown in Figure 15 (without the attachments).*

(ii) *If* α*,* β*, and* γ *lie outside the double-cap, then the drawing of the double-cap with attachments is isomorphic to the one shown in Figure 15; in particular,* $c_0\alpha$ *and* $c_1\beta$ *are collinear and* $a_1\gamma$ *and* $\alpha\beta$ *are parallel.*

Proof: We start with (i) and ignore the attachments to α, β, and γ. Since $\{a_0, a_1, \ldots, a_{10}\}$ lie on the boundary of an empty face, and the ends of a_2a_9 alternate with the ends of both a_0a_3 and a_8a_{10}, a_2a_9 must cross both of those edges and do so orthogonally. Then a_0c_0 cannot cross a_2a_9, since otherwise it would overlap with a_0a_3. It follows that c_0a_4 must cross a_2a_9 (since c_0a_4 cannot cross the empty face bounded by the a-cap). This implies that c_0c_1 cannot cross a_2a_9 (edges would overlap), so c_1a_6 must cross a_2a_9. Repeating the same argument, c_1b_0 cannot cross a_2a_9, so b_0a_7 must cross a_2a_9. Similarly, c_0b_6 and c_1b_4 cannot cross a_2a_9, and since b_4, b_5, b_6 lie on the boundary of the same face, a_5b_5 must cross a_2a_9. At this point, we know that a_0a_3, c_0a_4, b_5a_5, c_1a_6, and b_0a_7 are all orthogonal to a_2a_9 and therefore all parallel to each other.

By symmetry, we can conclude analogous facts about the upper cap. It then follows that a_5b_5 must cross c_0c_1, so c_0c_1 is parallel to a_2a_9 and b_2b_9. And edges a_0a_3 and a_0b_7 are collinear, as are c_0a_4 and c_0b_6, as well as c_1a_6 and c_1b_4, and b_0a_7 and b_0b_3.

This, in turn implies that $a_3a_0b_7$, $a_4c_0b_6$, a_5b_5, and $a_6c_1b_4$ lie on parallel lines and must cross edge b_2b_9 (since they connect vertices on opposite caps, which lie on empty face boundaries).

In summary, without attachments, the drawing of the double-cap is (up to a homeomorphism) as shown in Figure 15.

This implies that it is meaningful to assume, for part (ii), that α, β and γ lie outside the double-cap (as defined earlier).

Since γ does not lie inside the region bounded by $\{a_0, a_1, a_2, a_3\}$ and $a_1\gamma$ cannot pass through the empty face bounded by $\{a_0, a_1, \ldots, a_{10}\}$, $a_1\gamma$ must cross a_0a_3, entering the rectangle formed by $b_7a_0a_3$, a_2a_9, $a_7b_0b_3$, and b_2b_9. Since it does not lie inside, it must continue crossing c_0a_4, b_5a_5, c_1a_6 and b_0a_7 (note that a_0c_0 cannot cross $a_1\gamma$, so c_0c_1 lies between $a_1\gamma$ and a_2a_9 both of which it is parallel to). Hence, $a_1\gamma$ is as shown in the drawing.

Next, consider α. Since α does not lie inside the rectangle formed by $b_7a_0a_3$, a_2a_9, $a_7b_0b_3$, and b_2b_9, but c_0 does, it follows that αc_0 crosses one of the four sides of that rectangle, but three of the directions, c_0b_6, c_0c_1, and c_0a_4 are already taken (since there can be no overlap). So αc_0 must cross either a_0a_3 or a_0b_7. If it crosses a_0a_3 it would also have to cross the boundary of the a-cap (since α does not lie inside the double-cap), which is not allowed, since the face bounded by $\{a_0, a_1, \ldots, a_{10}\}$ is empty. It follows that αc_0 must cross a_0b_7.

A symmetric argument for β (rotate the double-cap upside-down) shows that $c_1\beta$ crosses b_0a_7. Hence, the drawing with attachments is as shown in the figure.

The chain of crossings: $c_0\alpha$ with a_0b_7, a_0b_7 with b_2b_9, b_2b_9 with a_5b_5, a_5b_5 with a_2a_9, a_2a_9 with a_7b_0 and finally a_7b_0 with $c_1\beta$ shows that $c_0\alpha$ and $c_1\beta$ are collinear. And since a_5b_5 crosses $a_1\gamma$, $a_1\gamma$ and $\alpha\beta$ are parallel. □

The $\top$-gadget. By combining two double-caps we can now build a $\top$-gadget, as shown in Figure 16; the $\top$-gadget contains four caps, one a- and one b-cap for each of the double-caps. The $\top$-gadget has three attachments: $c_0\alpha$, $c_0'\gamma$, and $v\gamma$. We label the non-attachment vertices in each double-cap as before, and add a prime symbol ($'$) to distinguish the vertices in the right double-cap.

We define the *outside region* of a $\top$-gadget to be everything that is outside both the double-caps (as defined earlier) as well as outside the region bounded by the boundary c_0,a_0,a_1,a_1',c_0',c_0. The inside region is shaded in the drawing.

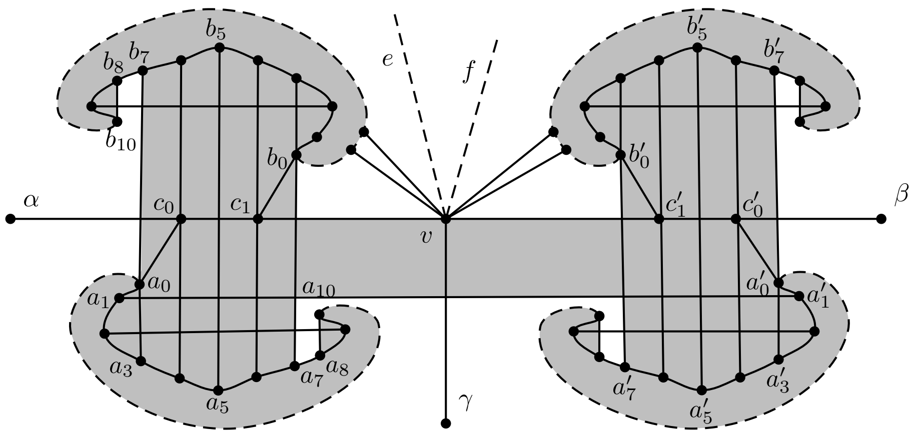

Figure 16: Simulating a $\top$-junction with two double-caps. The inside region of the $\top$-gadget is shaded.

Lemma 2 *Suppose we have a* RAC-*drawing of the* $\top$*-gadget in which all four caps bound empty faces. Then the drawing without attachments is isomorphic to the drawing shown in Figure 16 (without the attachments). If* α*,* β *and* γ *lie outside the* $\top$*-junction and* $v\gamma$ *starts inside the* $\top$*-gadget, then the drawing is isomorphic to the one shown in the figure.*

The two additional edges e and f in Figure 16 could leave v in other directions as well, the drawing shows the intended direction. We note that the caps can be made arbitrarily small, and close to v, so that e and f can be realized anywhere in the rotation at v between α and β.

Proof: By Lemma 1, we can assume that both double-caps, without attachments are drawn as shown in the figure, except each one of them could be reversed (we will exclude that possibility later).

We attached v to two new vertices on the boundary of the b-cap of the left double-cap, call the connecting edges g and h. We claim that v must lie outside the left double-cap. If v lies inside the double-cap, edges g and h would have to cross b_8b_{10}, a_8a_{10}, b_0a_7 or a_0b_7 to get inside (all other boundary edges have no crossings, since they are on the boundary of an empty face). Two incident edges cannot both cross the same edge orthogonally, as they would overlap, so g and h must cross two different edges. This is only possible if v lies inside the region bounded by the double-cap and g and h cross a_0b_7 and b_0a_7. Since a_0b_7 and b_0a_7 are parallel this forces g and h to be collinear, creating a crossing through the double-cap with endpoints outside. This is not possible, because of the two diagonal edges c_0a_0 and c_1b_0. We conclude that v lies outside of the left double-cap, and, with the same argument, outside the other double-cap as well. Moreover, the two double-caps are outside each other, since otherwise vc_1 would overlap with vc'_1. In particular, the two double-caps are on opposite sides of v, and therefore oriented as shown in the figure.

Consider the edge $a_1a'_1$. Since a'_1 is outside the left double-cap, $a_1a'_1$ crosses b_0a_7, b_5a_5 and a_0a_3 orthogonally. Similarly, because a_1 is outside the right double-cap, $a_1a'_1$ crosses $b'_0a'_7$, $a'_5b'_5$, and $a'_0a'_3$ orthogonally. It follows that the six edges, b_0a_7, b_5a_5, a_0a_3, $b'_0a'_7$, $a'_5b'_5$, and $a'_0a'_3$ are parallel to each other, which, in turn implies that αc_0, c_0c_1, c_1v, vc'_1, $c'_1c'_0$ and $c'_0\beta$ are parallel (since each of them is orthogonal to one of the six edges), and, since they form a path, must be collinear. In particular, v lies on the line through α and β.

If γ lies outside the $\top$-gadget, and leaves v by starting inside the gadget, it also has to cross $a_1a'_1$ forcing $v\gamma$ to be orthogonal to $\alpha\beta$. □

Replacing ×-junctions. Finally, as mentioned earlier, we can replace each ×-junction with four $\top$-junctions, see Figure 17.

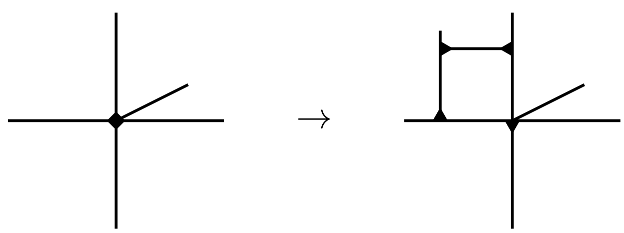

Figure 17: How to replace a ×-junction with four $\top$-junctions.

Lemma 3 *Given a graph G with ×- and $\top$-junctions, we can build a graph G' with $\top$-junctions only so that G has a drawing respecting junctions if and only if G' does, and if the drawing of G is crossing-free, then so is the drawing of G'. Given a drawing D isomorphic to a drawing of G respecting junctions we can efficiently find a drawing D' isomorphic to a drawing of G' respecting junctions.*

Proof: We replace each ×-junction with four $\top$-junctions as shown in Figure 17. This does not affect drawability respecting junctions, and constructs D' efficiently from D. If the original drawing is crossing-free, then the $\top$-junctions can be placed arbitrarily close to the vertex of the original ×-junction so that the new drawing remains crossing-free. □

4.2 Proof of Theorem 9

We are given a graph G with $\top$- and ×-junctions, together with a crossing-free drawing D of G. We have to construct a graph G', a drawing D' of G' and vertex sets $(V_i)_{i\in[k]}$ satisfying the conditions of Theorem 9.

Because of Lemma 3, we can assume that G contains only $\top$-junctions. By Theorem 5 we know that a drawing of G respecting junctions, if it exists, will be isomorphic to D. We start by replacing each $\top$-junction with a $\top$-gadget.

Consider an edge uv in G. If it connects two non-junction vertices, we do nothing. Suppose uv connects $\top$-junction v to a vertex u in G. If u is α or β in the $\top$-gadget, we connect u with two edges to one of the neighboring two caps belonging to the $\top$-gadget belonging to v. See the vertex u on the left of Figure 18. If u is γ in the $\top$-gadget, we connect it with two edges to the lower caps of both double-caps belonging to v (this situation is not shown in the illustration).

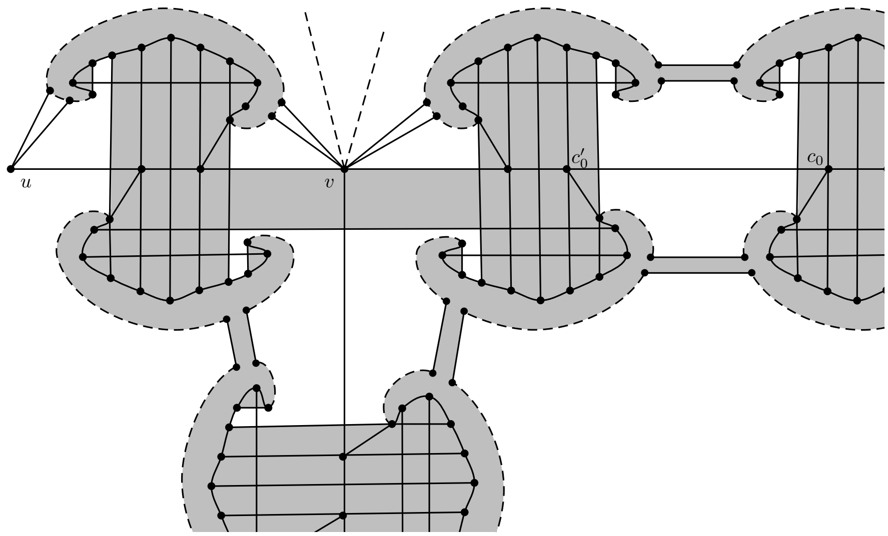

Figure 18: Connecting a $\top$-gadget v to a neighboring vertex u on the left and a neighboring $\top$-gadget w on the right (only the leftmost double-cap of the gadget for w is visible). Vertex c_0' in the $\top$-gadget for v is identified with α in the $\top$-gadget for w, and vertex c_0 in the $\top$-gadget for w is identified with β in the $\top$-gadget for v.

Similarly, if an edge vw connects two $\top$-junctions in G, we connect the boundaries of all caps (of $\top$-gadgets for u and v) that lie in the same region and *merge* them, by making them the boundary of a common empty face: we connect pairs of consecutive vertices on each of the two boundaries with each other, creating a narrow tunnel between two caps. See Figure 18 which shows the full $\top$-gadget for v, and two partial $\top$-gadgets, the one on the right representing w (the vertex w itself lies outside the frame). The $c_0'\beta$-edge of the $\top$-gadget for v was identified with the αc_0-edge of the $\top$-gadget for w (on the right), and the corresponding a- and b-caps were merged. This forces the c_0-vertex belonging to the $\top$-gadget for w to lie outside the $\top$-gadget for v and the c_0'-vertex belonging to the $\top$-gadget for v to lie outside the $\top$-gadget for w. In this fashion we ensure that for each $\top$-gadget, its α-, β-, and γ-vertices lie outside the gadget. For γ-vertices merging the caps also enforces that $v\gamma$ leaves the $\top$-gadget in the right direction: below the $\top$-gadget for u in Figure 18 there is a (side-ways) double-cap belonging to another $\top$-gadget. Its c_0' vertex is identified with γ from the $\top$-gadget for v. The merged caps force $v\gamma$ to leave v starting inside the gadget. By Lemma 2 this implies that the $\top$-gadget correctly represents a $\top$-junction.

From the crossing-free drawing D of G we have constructed a graph G' (without $\top$- or $\times$-junctions) and a drawing D' of G'. Let $(V_i)_{i\in[k]}$ be the collection of boundaries of (merged) caps.

If D is isomorphic to a drawing of G respecting junctions, then we can draw the junction gadgets as intended (that is, as shown in the illustrations), and D' is isomorphic to a RAC-drawing of G'. Inspecting the gadgets, we find that no edge in any gadget is involved in more than 11 crossings (this number is achieved for edges of type a_1a_1' connecting two double-caps in a $\top$-gadget). Hence, we can assume that the RAC-drawing is 11-planar. This proves Property (i) of the theorem.

For (ii) suppose that G' has a drawing in which the vertices of each V_i lie on the boundary of an empty face. This is sufficient, as we argued, for all the junction gadgets to work correctly, and we can conclude that G has a RAC-drawing. This completes the proof.

5 Area Bounds and Grid Drawings

The *area* of a RAC-drawing is the area of a smallest bounding box containing the drawing in which all points (vertices, crossings, bend-points, if allowed) have at least distance 1 from each other. Using the reduction from Theorem 1 we can build a graph G representing equations

$$x_1 = 2, x_2 = x_1 \cdot x_1, x_3 = x_2 \cdot x_2, \ldots, x_n = x_{n-1} \cdot x_{n-1}.$$

Then any RAC-drawing of G contains three points representing 0, 1, and $2^{2^{n-1}}$. So G is a graph of polynomial size which requires double-exponential area. In Section 5.1 we will see a corresponding upper bound.

Corollary 10 *There are graphs which are* RAC-*drawable but require double-exponential area.*

In the graph G constructed for the corollary, the points in the gadgets can even be placed so as to lie on a grid. This is not always possible, since we can also build a graph G representing equations

$$x_1 = 2, x_1 = x_2 \cdot x_2.$$

A RAC-drawing of G will contain triples of points representing 0, 1, $\sqrt{2}$; since 1 and $\sqrt{2}$ are linearly independent over $\mathbb{Q}$, it is not possible to place the three points on a grid, so G can have an RAC-drawing which is not realizable on a grid.

Corollary 11 *There are graphs which are* RAC-*drawable but not realizable on an integer grid.*

We have a closer look at grid drawings in Section 5.2 on the computational complexity of RAC-drawability on the grid, and Section 5.3 on the geometric complexity of RAC_1-drawings on the grid.

5.1 Area Upper Bound

Recall that RAC_k-drawings are RAC-drawings in which each edge may have up to k bends.

Corollary 12 *Every graph that has a* RAC_k*-drawing, has a* RAC_k*-drawing with at most double-exponential area, for every* $k \geq 0$.

Since we have polynomial area upper bounds for $k = 3$, this result is only interesting for $k < 3$, where it gives the currently best-known upper bounds. For $k = 1$ we will strengthen the result in Theorem 15.

This type of exponential upper bounds on the geometric complexity of realizations was first obtained by Goodman, Pollack and Sturmfels [25] in a paper on the spread of point configurations. Our situation differs in that the realization space of RAC_k-drawings of a graph is not open in the topological sense. This forces us to go back to one of the original tools, a result due to Grigor'ev and Vorobjov [26]. This will be the main technical ingredient in the proof of Corollary 12.

Theorem 13 (Grigor'ev and Vorobjov [26, Lemma 10]) *Suppose f is a polynomial in n variables of (total) degree less than d and with integer coefficients of bitlength less than M. If $f = 0$ has a solution, then it has a solution of distance at most $R = 2^{Md^{cn}}$ from the origin, where $c > 0$ is a fixed constant, independent of f.*

Bieker showed that RAC-drawability can be expressed in the existential theory of the reals. We need a slightly more precise form of that statement as the second main ingredient in the proof of Corollary 12, so we reprove the result.

Lemma 4 (Bieker [11, Section 6.2]) *Given a graph $G = (V, E)$ we can efficiently compute a polynomial f_G with variables (v_x, v_y) for $v \in V$, (c_x, c_y) for $c \in \binom{E}{2}$, and additional variables λ_i, $i \in I$ so that the following two statements are equivalent:*

- *G has a* RAC-*drawing in which every vertex v is placed at location (v_x, v_y) and for every pair of edges $c \in \binom{E}{2}$ that crosses, the crossing is at (c_x, c_y),*
- *$f_G((v_x, v_y)_{v\in V}, (c_x, c_y)_{c\in\binom{E}{2}}, (\lambda_i)_{i\in I}) = 0$ for some choice of the λ_i, $i \in I$.*

The polynomial f_G has total degree at most 12, the bit-length of f_G is $O(\log n)$, and $|I| = O(n^2)$, where $n = |V|$, so the total number of variables of f_G is $O(n^2)$.

In particular f_G has a zero if and only if G has a RAC-drawing, this is all we need for membership of RAC-drawability in $\exists\mathbb{R}$ (and the proof of Theorem 1).

Proof: We show how to translate certain conditions, such as two line segments being parallel, into nonnegative polynomials so that the condition is true if and only if the polynomial has a zero. We then combine these conditions into the single polynomial f_G.

Range Constraints. The condition $\lambda \geq 0$ is equivalent to there being a $\lambda_\#$ for which $\lambda = \lambda_\#^2$. So $\lambda \geq 0$ is true if and only if the (nonnegative) polynomial $(\lambda - \lambda_\#^2)^2$ is zero for some $\lambda_\# \in \mathbb{R}$. We can also express $\lambda \neq 0$ as there being a $\lambda_\#$ so that $\lambda\lambda_\# = 1$, or, equivalently, $(\lambda\lambda_\# - 1)^2$ being zero for some $\lambda_\# \in \mathbb{R}$.

A value λ lies in the range $(0, 1)$, that is, $0 < \lambda < 1$ if there is a $\lambda_\# \in \mathbb{R}$ for which $\lambda(1+\lambda_\#^2) = 1$; defining the polynomial

$$f_{\in(0,1)}(\lambda, \lambda_\#) := (\lambda(1 + \lambda_\#^2) - 1)^2$$

we have that $f_{\in(0,1)}(\lambda, \lambda_\#)$ is zero for some $\lambda_\# \in \mathbb{R}$ if and only if $0 < \lambda < 1$.

Similarly, we can express that λ does not lie in the range $(0, 1)$, so either $\lambda \leq 0$ or $\lambda \geq 1$. This is equivalent to $\lambda^2 - \lambda \geq 0$ which is equivalent $\exists\lambda_\# : \lambda^2 - \lambda - \lambda_\#^2 = 0$. So we can define the polynomial

$$f_{\not\in(0,1)}(\lambda, \lambda_\#) := (\lambda^2 - \lambda - \lambda_\#^2)^2$$

which is zero for some $\lambda_\# \in \mathbb{R}$ if and only of λ does not lie in the range $(0, 1)$.

Geometric Constraints. Two points $p = (p_x, p_y)$ and $q = (q_x, q_y)$ are distinct, if $p_x \neq q_x$ or $p_y \neq q_y$. We define $f_{\neq}(p, q, \lambda_1, \lambda_2) = f_{\neq}(p_x, p_y, q_x, q_y, \lambda_1, \lambda_2)$ as

$$((p_x - q_x)\lambda_1 - 1)^2((p_y - q_y)\lambda_2 - 1)^2.$$

Then p and q are distinct if and only if $f_{\neq}(p, q, \lambda_1, \lambda_2)$ is zero for some $(\lambda_1, \lambda_2) \in \mathbb{R}^2$.

Given three points $p = (p_x, p_y), q = (q_x, q_y), r = (r_x, r_y) \in \mathbb{R}^2$ we can express that r lies on the line through p and q (assuming they are distinct) as follows:

$$\exists \lambda : \lambda(q_x - p_x) + p_x = r_x \wedge \lambda(q_y - p_y) + p_y = r_y.$$

If we define $f_|(r, p, q, \lambda) = f_|(r_x, r_y, p_x, p_y, q_x, q_y, \lambda)$ as

$$(\lambda(q_x - p_x) + p_x - r_x)^2 + (\lambda(q_y - p_y) + p_y - r_y)^2,$$

then r lies on the line through pq if and only if $f_|(r, p, q, \lambda) = 0$ for some $\lambda \in \mathbb{R}$, and, in that case, $r = q + \lambda(q - p)$. Moreover, the point r lies in the interior of the line segment pq if and only if $0 < \lambda < 1$, which we can express using the polynomial $f_{\in(0,1)}(\lambda, \lambda_\#)$. We can also express that r does not lie in the interior of the line segment pq using the polynomial $f_{\notin(0,1)}(\lambda, \lambda_\#)$, but this assumes that r does lie on the line through pq.

The point r does not lie on the line through pq if and only if

$$\exists \lambda : \lambda(q_x - p_x) + p_x = r_x \wedge \lambda(q_y - p_y) + p_y \neq r_y,$$

so if we define $f_{\cdot|}(r, p, q, \lambda, \lambda_\#) = f_{\cdot|}(r_x, r_y, p_x, p_y, q_x, q_y, \lambda, \lambda_\#)$ as

$$(\lambda(q_x - p_x) + p_x - r_x)^2 + (\lambda_\#(\lambda(q_y - p_y) + p_y - r_y) - 1)^2,$$

then $f_{\cdot|}(r, p, q, \lambda, \lambda_\#)$ is zero for some $(\lambda, \lambda_\#) \in \mathbb{R}^2$ if and only if r does not lie on the line through pq. (The second term of $f_{\cdot|}$ encodes that $\lambda(q_y - p_y) + p_y - r_y \neq 0$ using $\lambda_\#$.)

Given four pairwise distinct points $p = (p_x, p_y), q = (q_x, q_y), s = (s_x, s_y), t = (t_x, t_y) \in \mathbb{R}^2$ the two line-segments pq and st are parallel (including the possibility that they are collinear) if and only if $(p_x - q_x)(s_y - t_y) = (p_y - q_y)(s_x - t_x)$. So if we define $f_{\|}(p, q, s, t) = f_{\|}(p_x, p_y, q_x, q_y, s_x, s_y, t_x, t_y)$ as

$$((p_x - q_x)(s_y - t_y) - (p_y - q_y)(s_x - t_x))^2,$$

then pq and st are parallel if and only if $f_{\|}(p, q, s, t)$ is zero (assuming the four points are pairwise distinct, conditions we already know how to enforce).

If pq and st are not parallel, then there is a common point c belonging to the lines through pq and st. We can express this as the polynomial

$$f_\dagger(p, q, s, t, c, \lambda_1, \lambda_2) := f_|(p, q, c, \lambda_1)^2 + f_|(s, t, c, \lambda_2)^2$$

being zero for some $(\lambda_1, \lambda_2) \in \mathbb{R}^2$.

The common point c on lines pq and st belongs to both line segments if $0 < \lambda_1 < 1$ and $0 < \lambda_2 < 1$ in $f_\dagger(p, q, s, t, c, \lambda_1, \lambda_2)$, something we already know how to express using $f_{\in(0,1)}$.

To test whether the lines through pq and st cross at a right angle, we need to check that $(q_y - p_y)(t_y - s_y) = (q_x - p_x)(t_x - s_x)$. The polynomial $f_\times(p, q, s, t)$ defined as

$$((q_y - p_y)(t_y - s_y) - (q_x - p_x)(t_x - s_x))^2$$

is zero if and only if pq and st are orthogonal.

Building f_G. We can now express that $G = (V, E)$ has a RAC-drawing using a polynomial f_G as follows: for every vertex v in V we have variables (v_x, v_y) and for every pair of edges $c \in \binom{E}{2}$ we have variables (c_x, c_y) (representing a potential crossing). We start with $f_G = 0$. Using the polynomials we built, we express that

(i) all vertices and crossings are pairwise distinct,

(ii) no vertex lies in the interior of an edge,

(iii) every pair of edges is either parallel or crosses in c, where c is the variable representing the pair of edges, and

(iv) if c is a crossing of two edges that are not parallel, it is a right-angle crossing.

For (i) we add $f_{\neq}(p, q, \lambda_1^{p,q}, \lambda_2^{p,q})$ to f_G for every two points p and q (vertices and crossings).

For (ii) we express that if r lies on the line through pq, then it does not lie in the interior of the segment pq for any three vertices p, q, r for which pq is an edge in G. This is sufficient to guarantee (ii) since in (i) we excluded r coinciding with p or q. We implement the implication as a disjunction, and add

$$f_{\cdot|}(r, p, q, \lambda_1^{r,p,q}, \lambda_2^{r,p,q}) \cdot f_{\not\in(0,1)}(\lambda_1^{r,p,q}, \lambda_3^{r,p,q})$$

to f_G. For the product to be zero, either r does not lie on the line through pq, or it does, in which case $\lambda_1^{r,p,q}$ does not lie in $(0, 1)$, meaning r does not lie in the interior of pq.

To ensure (iii) we add

$$f_{\|}(p, q, s, t) \cdot f_{\dagger}(p, q, s, t, c^{p,q,s,t}, \lambda_1^{p,q,s,t}, \lambda_2^{p,q,s,t})$$

to f_G for all pairs of edges pq and st. Again, we use the product to encode disjunction, this time between pq and st being parallel and $c^{p,q,s,t}$ being their intersection point.

Finally, for (iv) we add

$$f_{\not\in(0,1)}(\lambda_1^{p,q,s,t}, \lambda_3^{p,q,s,t}) \cdot f_{\not\in(0,1)}(\lambda_2^{p,q,s,t}, \lambda_4^{p,q,s,t}) \cdot f_{\times}(p, q, s, t)$$

to f_G for every pair of edges pq and st in G. Here $\lambda_1^{p,q,s,t}$ and $\lambda_2^{p,q,s,t}$ are the same variables as in (iii), so the term added for (d) is zero if either $c^{p,q,s,t}$ does not lie on line segment pq or $c^{p,q,s,t}$ does not lie on line segment st or pq is orthogonal to st.

We obtain a polynomial f_G in the variables (v_x, v_y), (c_x, c_y) as well as several λ-variables, so that (v_x, v_y) encodes the point locations of a RAC-drawing with crossings (for actually crossing edges) at point locations (c_x, c_y) if and only if f_G has a zero for some choice of the λ-variables.

The total degrees of the special polynomials we defined are 4 for $f_{\not\in(0,1)}$, $f_|$, $f_\|$, 6 for $f_{\in(0,1)}$, $f_{\cdot|}$, and 8 for $f_{\neq}$ and $f_{\dagger}$. The terms created for (i) have total degree 8, the terms for (ii) degree $6 + 4 = 10$, for (iii) degree $4 + 8 = 12$, and for (iv) degree $4 + 4 + 4 = 12$, so the total degree of the polynomial is at most 12.

Our polynomial expressions only use coefficients 1 and -1. So if we multiply out all the terms (and do not collect them), the coefficients will still be -1 and 1. Let $n = |V|$; we know that $|E| = O(n)$ (if $|E| > 4n - 10$, then G does not have a RAC-drawing [19, Theorem 1], and we can let $f_G = 1$), and so there are at most $O(n^2)$ terms. Collecting terms then leads to coefficients of bit-length at most $O(\log n)$. Finally, the $O(n^2)$ conditions require at most four λ-variables each, so $|I| = O(n^2)$, and therefore f_G has at most $O(n^2)$ variables overall. □

Combining Lemma 4 with Theorem 13 yields Corollary 12 that every graph that has a RAC$_k$-drawing has such a drawing in at most double-exponential area.

Proof: [Proof of Corollary 12] Let us consider the case $k = 0$ first. For a graph G let f_G be the polynomial constructed in Lemma 4. If G is RAC-drawable with vertex v at location (v_x, v_y) and a crossing c (if it occurs) at location (c_x, c_y), then f_G is zero for some choice of the λ-variables.

We extend f_G so it enforces that any two points (vertices and crossings) have distance at least 1. This does not affect realizability, since we can scale the drawing. We have that $d(u, w) \geq 1$ if and only if $(u_x - w_x)^2 + (u_y - w_y)^2 \geq 1$, which is the case if and only if $(u_x - w_x)^2 + (u_y - w_y)^2 = 1 + \lambda_{u,w}^2$ which is equivalent to $((u_x - w_x)^2 + (u_y - w_y)^2 - 1 - \lambda_{u,w}^2)^2$ being zero for some $\lambda_{u,w} \in \mathbb{R}$. We add the term $((u_x - w_x)^2 + (u_y - w_y)^2 - 1 - \lambda_{u,w}^2)^2$ to f_G for every pair of points u and w obtaining a polynomial f. Then f is zero for some choice of the λ-variables if and only if G has a RAC-drawing in which all vertices and crossings have distance at least 1 from each other.

By Theorem 13, $f = 0$ has a solution within distance $R = 2^{2^{cn^2}}$ of the origin, for some fixed $c > 0$. In particular, all vertices and crossings lie within a disk of radius R centered at the origin, and all vertices and crossings have distance at least 1 from each other, by the conditions we added. The claim of the corollary follows for $k = 0$.

For $k > 0$ we build a graph H by subdividing each edge of G with k bend-points. Since G has a RAC_k drawing if and only if H has a RAC-drawing, the case $k = 0$ applies to H yielding the double-exponential upper bound on the area of G. □

5.2 RAC-Drawings on the Grid

What happens if we restrict our RAC-drawings to grids; that is, if the vertices of the graph need to be placed on the points of an integer grid? The proof of Theorem 1 essentially answers that question. RAC-drawability on a grid is as hard as $\exists\mathbb{Q}$, the existential theory of the rationals; see Section 2.2 for some details on this complexity class.

Corollary 14 *Determining whether a graph has a* RAC*-drawing on a grid is* $\exists\mathbb{Q}$*-complete.*

This effect, of an $\exists\mathbb{R}$-hard drawing problem turning $\exists\mathbb{Q}$-hard when restricted to a grid, has been observed before, for example, for point-visibility graphs, by Cardinal and Hoffmann [16], and for the planar slope number by Hoffmann [27].

As we mentioned in Section 2.2, little is known about the relationship between $\exists\mathbb{Q}$ and traditional complexity classes; we do know that it contains $\exists\mathbb{R}$, so the problem is at least **NP**-hard, but for all we know, it may even be undecidable. Not knowing whether $\exists\mathbb{Q}$ is decidable or not, means that we do not have any algorithms for $\exists\mathbb{Q}$-complete problems like RAC-drawability on a grid, and little hope for one.

Proof: [Proof of Corollary 14] We first argue that the problem lies in $\exists\mathbb{Q}$. If we limit the existential quantifiers for the positions of the vertices in the RAC-drawing to range over the rational numbers, then a solution corresponds to a grid solution: scale the drawing by multiplying all coordinates with the greatest common divisor of the denominators of all vertices, to obtain a RAC-drawing on an integer grid. (If required, we can even ensure that the crossings points lie on the integer grid, since a crossing point of two line segments with rational endpoints has rational coordinates.)

To show $\exists\mathbb{Q}$-hardness we work with Theorem 1. For a given polynomial f the proof of Theorem 1 constructs a graph G such that f has a (real) zero if and only if G has a RAC-drawing. We claim that f has a rational zero (an $\exists\mathbb{Q}$-complete problem by Theorem 4) if and only if G has a RAC-drawing on the grid.

If f has a rational zero, then all variables and intermediate values we compute with G are rational, and we can place all gadgets so that all vertices (and thereby crossing points) occur at rational coordinates. Then G has a RAC-drawing on a grid.

If G has a RAC-drawing on a grid, then all the variables represented by G must be rational, so the solution to f is rational. □

If the existential theory of the rationals turns out to be undecidable, then Corollary 14 implies that there are no computable upper bounds on the size of a grid on which a graph has a RAC-drawing.

5.3 Grid Drawings with Bends

We already mentioned that every graph has a (polynomial area) RAC$_3$-drawing [19], so the two remaining cases of interest are RAC$_1$- and RAC$_2$-drawings. While we cannot settle the computational complexity of these two problems, we can say something about the combinatorial complexity of RAC$_1$-drawings. The following result tells us that a RAC$_1$-drawing, different from a RAC-drawing, can always be realized on an integer grid of double-exponential size.

Theorem 15 *If a graph has a* RAC$_1$*-drawing, then it has a* RAC$_1$*-drawing in which all vertices and bend points are placed on the points of a* $2^{2^{O(n^2)}} \times 2^{2^{O(n^2)}}$*-grid.*

The theorem implies that every RAC$_1$-drawing can be realized in double-exponential area, but it is stronger than that since the points are placed on an integer grid, so the theorem improves on Corollary 12 for $k = 1$.

The proof of Theorem 15 requires some perturbation arguments. We collect them as lemmas below. Call a bend-point *proper* if the angle at the point is not π; in other words, if the bend-point is not the interior vertex of a line segment connecting its two neighbors. The *bend-angle* at a bend-point is the smaller of the two angles formed by the edges meeting at the bend-point.

Lemma 5 *If a graph has a* RAC$_1$*-drawing, then it has a* RAC$_1$*-drawing in which each edge has exactly one bend-point and that bend-point is proper.*

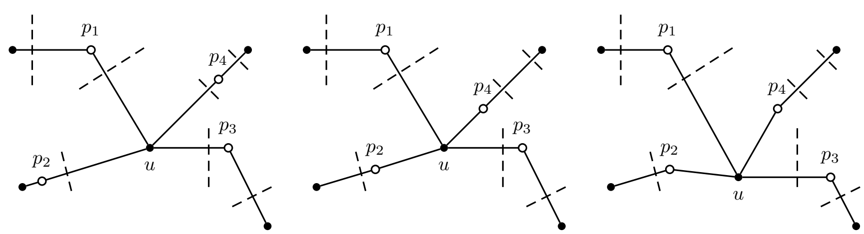

Figure 19: *(Left)* vertex u incident to four bend-point p_1, p_2, p_3, p_4, two of them proper, p_1 and p_3; *(middle)* moving the non-proper bend-points p_2 and p_4 so that up_2 and up_4 are crossing-free; *(right)* perturbing u; p_2 and p_4 do not move, up_1 and up_3 move parallel to their original drawings.

Proof: We start with a RAC$_1$-drawing; we subdivide all each without a bend-point by adding a non-proper bend-point. Suppose the drawing contains at least one bend-point which is not proper. Let u be a vertex of the graph incident on that bend-point, see the left illustration in Figure 19. We move all bend-points p incident to u which are not proper close to u so that the segments up are free of crossings (this is possible, since the bend-points, not being proper, lie on the line segment connecting the two endpoints of the edge they subdivide), see the middle illustration of Figure 19. We can now perturb u as follows: Consider an edge upv incident to u with bend-point p. If p is proper, we move the segment up parallel to the original segment as shown in the right illustration of Figure 19; this changes the location of p. If p is not proper, we keep the location of p fixed, which works since up is crossing-free. All bend-points incident to u are now proper. By making the perturbation of u sufficiently small, we can ensure that the proper bend-points incident to u (which move) do not move past a crossing edge or another vertex. □

The following two lemmas quantify the effect of perturbing points; the first lemma deals with the situation described in Lemma 5.

Lemma 6 *Suppose two lines, one through u and one through v, intersect in a single point p with a bend-angle of α. If u' has distance less than ε from u, then a line through u' parallel to up intersects the line through vp in a point p' which has distance at most $\varepsilon/\sin(\alpha)$ from p.*

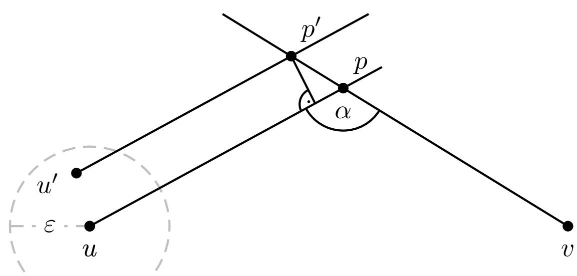

Figure 20: The effect of perturbing a vertex u by at most ε on a bend-point p when moving up in parallel.

Proof: See Figure 20. By assumption, the distance of p' from up is at most ε and $\sin(\alpha) = \sin(\pi - \alpha) = d(p', up)/d(p, p')$, from which we get that $d(p, p') < \varepsilon/\sin(\alpha)$, which is what we had to show. □

Lemma 7 *Suppose two lines up and vq cross at right angles in a point c_1, and the line vq forms a bend-angle of α with line wq. If p' has distance at most ε from p, p and u have at least distance 1, and the line through v which is orthogonal to up' crosses wq in a point q' at a bend-angle of α', then q and q' have distance at most $d(v, q)\varepsilon/\sin(\alpha')$ and $|\sin(\alpha') - \sin(\alpha)| < 2\varepsilon$ if ε is sufficiently small.*

Proof: See Figure 21. The angle between vq and vq' is the same as the angle between up and up', which is β. By assumption, $\sin(\beta) \leq \varepsilon/d(u, p) \leq \varepsilon$.

Because $\alpha' = \alpha + \beta$, we have $\sin(\alpha') = \sin(\alpha)\cos(\beta) + \sin(\beta)\cos(\alpha)$. Then

$$\begin{aligned}
|\sin(\alpha') - \sin(\alpha)| &= |\sin(\alpha)(\cos(\beta) - 1) + \sin(\beta)\cos(\alpha)| \\
&\leq |\sin(\alpha)||\cos(\beta) - 1| + |\sin(\beta)||\cos(\alpha)| \\
&< |\cos(\beta) - 1| + \varepsilon \\
&\leq |\sin(\beta)| + \varepsilon \\
&\leq 2\varepsilon,
\end{aligned}$$

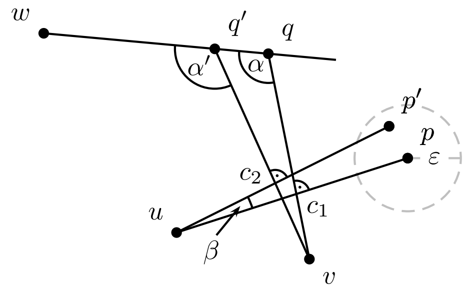

Figure 21: The effect of perturbing a bend-point p by at most ε on a bend-point q.

where we used that $\sin(\beta) + \cos(\beta) > 1$ for $0 < \beta < \pi/2$ (here we use that ε is sufficiently small). By the law of sines, we have

$$\frac{d(q, q')}{\sin(\beta)} = \frac{d(v, q)}{\sin(\pi - \alpha - \beta)}.$$

Now $\sin(\pi - \alpha - \beta) = \sin(\alpha + \beta) = \sin(\alpha')$, so we get $d(q, q') < d(v, q)\varepsilon/\sin(\alpha')$, as required. □

Finally, we need a quantitative version of Lemma 5, that is, a version that establishes a lower bound on the bend-angles and upper bounds on the other parameters.

Lemma 8 *If a graph has a* RAC_1-*drawing, then it has a* RAC_1-*drawing inside a disk of radius* $R = 2^{2^{cn^2}}$ *such that all points (vertices, crossings and bend-points) have horizontal and vertical distance at least* 2 *from each other, and* $|\sin(\alpha)| > 2/R$ *for every bend-angle* α. *The constant* c *is fixed and independent of the graph.*

Proof: Given an n-vertex graph G, let H be the result of subdividing each edge of G exactly once. Then G has a RAC_1-drawing if and only if H has a RAC-drawing. By Lemma 4 there is a polynomial f_G in variables (v_x, v_y) for $v \in V$, and (c_x, c_y) for every (unordered) pair of edges in H, and $O(n^2)$ additional variables λ_i so that H has a RAC-drawing with vertex v at location (v_x, v_y) and so that an unordered pair of edges c if it crosses, crosses in (c_x, c_y) if and only if $f_G = 0$ for some choice of the λ_i.

We extend f_G so it enforces two additional constraints: (i) the x- and y-coordinates of any two points (original vertices, bend-points, and crossing points) have horizontal and vertical distance at least 2, and (ii) every bend-vertex v is proper. Neither type of constraint affects realizability; for (ii) this follows from Lemma 5, and for (i) we can always rotate the drawing so no two points lie on the same horizontal or vertical line, and then scale the drawing to achieve the required minimum distance.

To encode condition (i), say for the x-coordinates u_x and v_x of two vertices u and v, we need to ensure that $|u_x - v_x| \geq 2$. Equivalently, we can require that $(u_x - v_x)^2 - 4 \geq 0$, or $(u_x - v_x)^2 - 4 = \lambda^2_{(u,v)}$ for some new variable $\lambda_{(u,v)}$. In other words, we can add the term $((u_x - v_x)^2 - 4 - \lambda^2_{(u,v)})^2$ to f_G. For condition (ii) let p be a bend-point subdividing edge uv of G, and define vectors $a = u - p$ and $b = v - p$. Suppose the bend-angle at p is $0 \leq \alpha < \pi$. The absolute value of the cosine of α is $|a \cdot b|/(|a||b|)$, where $\cdot$ denotes the dot-product of two vectors. Then $\sin(\alpha) \geq \sin^2(\alpha) = 1 - \cos^2(\alpha) = 1 - \frac{(a \cdot b)^2}{|a|^2|b|^2}$. We introduce a new variable $\lambda_{(u,p,v)}$ and add the conditions $\lambda_{(u,p,v)} \geq 0$ and $\lambda_{(u,p,v)}(1 - \frac{(a \cdot b)^2}{|a|^2|b|^2}) = 2$; these conditions can be rewritten as polynomial expressions in the underlying variables.

Let f be the result of adding these conditions to f_G. Since $f_G = 0$ has a solution, so has $f = 0$, as we argued earlier. By Theorem 13, $f = 0$ has a solution within distance $R = 2^{2^{cn^2}}$ of the origin, where $c > 0$ is a fixed (integer) constant. Consider a bend-point p subdividing edge uv, and let α be the bend-angle at p. By the conditions we added, $\lambda_{(u,p,v)} > 0$, and so $\sin(\alpha) \geq 2/\lambda_{(u,p,v)} > 2/R$. □

We are finally in a position to prove Theorem 15 that a RAC_1-drawing can be realized with all vertices and bend-points placed on a $2^{2^{O(n^2)}} \times 2^{2^{O(n^2)}}$-grid.

Proof: [Proof of Theorem 15] Suppose G has a RAC_1-drawing. By Lemma 5 we can assume that all bend-points are proper, and by Lemma 8 there is a drawing in which all points (vertices, bend-points and crossings) lie in a disk of radius $R = 2^{2^{cn^2}}$ for some fixed $c \geq 1$ independent of G, any two points have horizontal and vertical distance at least 2, and $|\sin(\alpha)| > 2/R$ for every bend-angle α.

Let $\varepsilon = \sqrt{2}/R^3$. We overlay the drawing with an infinite square grid of unit-length $1/R^3$. Any disk of radius ε contains at least one grid-point.

We will prove the theorem in three steps: In the first step we show how to "snap" vertices to the grid; snapping requires perturbing both the vertices and bend-points of the drawing to ensure that all the vertices lie on the grid and the drawing remains RAC. In the second step, we will snap each bend-point to the grid; snapping a bend-point may move other bend-points off the grid. In the third and final step we analyze the drawing to show that refining the grid is sufficient to ensure all points lie on the grid.

Step 1: Snapping vertices to the grid. To place all vertices on grid-points we perturb both vertices and bend-points of the drawing while keeping the drawing RAC. As we are doing so, the distances between vertices will change, as will the bend-angles. We will guarantee that $\sin(\alpha)$, which initially is larger than $2/R$ will remain larger than $1/R$, that all points remain in a disk of radius at most $R + 1$ around the origin, and that the pairwise distance between each pair of points (initially at least 2) remains larger than 1.

Start with a vertex u of G (not a bend-point). We can perturb u by at most ε so that it lies on the grid. We move edges incident to u in parallel, so that the drawing remains RAC, as pictured in Figure 20 but with $\alpha = \pi/2$. We say vertex u has been *snapped* to the grid. By Lemma 6, snapping u requires moving bend-points incident to u by at most $\varepsilon/\sin(\alpha)$; all bend-angles remain the same. After snapping every vertex of the graph to the grid, all vertices of G are located on the grid; every bend-point was moved at most twice (since bend-points are incident to at most two vertices), so by Lemma 6 every bend-point has distance at most $2\varepsilon/\sin(\alpha)$ from its original location. Since $|\sin(\alpha)| > 2/R$, and the bend-angles did not change, this tells us that every bend-point moved by less than $2\varepsilon/(2/R) = \sqrt{2}/R^2 < 1/R$. So every point (vertex or bend-point) moved by at most $1/R$, so all points have distance at most $R + 1/R$ from the origin, and the pairwise distance of points is at least $2 - 2/R$.

Step 2: Snapping bend-points to the grid. Consider a bend-point p. Perturbing it by at most ε we can ensure that it lies on the grid, but we are destroying the right-angle crossings involving edges incident to p. For example, if pu is an edge incident to p which crosses another edge qw, where q is a bend-point, we will move q to ensure that pu and qw remain orthogonal, as shown in

Figure 21. But qw may cross another edge xr, so other edges may be affected. We capture this with the following definition.

We say two edges e and f are RAC-*linked* if there is a sequence of edges $e = e_1, \dots, e_k = f$ so that e_i crosses e_{i+1} in the drawing, for $1 \leq i < k$. Being RAC-linked is an equivalence relation. We will show how to move an end-point of an edge, by rotating all edges RAC-linked to it by the same angle.

To *snap* bend-point p to a grid-point p' we move p to p' and modify the drawing as follows, so it remains RAC. Let L be the set of edges RAC-linked to pu, and let β be the angle (p, u, p'). We rotate all edges in L by β (at their vertex, that is, non-bend, end-vertex), just as we did in Figure 21. This keeps the drawing RAC: any two edges in L are rotated by the same angle, so they still cross at a right angle, if they cross; for any edge not in L its bend-point may move slightly (as q does for wq in Figure 21), but its angle does not change, so no other crossings are affected, unless the bend-point moves beyond a crossing; we will show below that the grid we chose is fine enough that that does not happen. If the second edge incident to bend-point p belongs to L we are done with p; in this case, p' moves to a new bend-point p'', since the second edge it is incident rotates as well. Otherwise, we repeat the same procedure with L and β defined for that second edge.

For Step 2 we snap each bend-point to the grid, in some (arbitrary) order.

Snapping p to the grid forces us to move other bend-points (potentially all of them, including points that were already on the grid, moving them off again; they will not be snapped to the grid again, they will be dealt with in Step 3). Since there are at most $6n$ bend-points (an overestimate, using the number of edges in a RAC_1-drawing [3]) and each bend-point is snapped only once, each bend-point moves at most $6n$ times. By Lemma 7, the sine of the crossing angle at a bend-point changes by at most 2ε in each of these steps, so by at most $12n\varepsilon < 1/R$ overall (we assumed $c \geq 1$). Hence every crossing-angle α at any point during the perturbation process satisfies $\sin(\alpha) > 2/R - 1/R = 1/R$. Lemma 7 also tells us that bend-point q gets moved by at most $d(v,q)\varepsilon/\sin(\alpha')$ when perturbing p, where α' is the new bend-angle at q. As we just saw, $\sin(\alpha') > 1/R$, so the distance q moves to q' is at most $d(v,q)/R^2$, in other words $d(v,q') < (1 + 1/R^2)d(v,q)$. Since q may move $6n$ times, the final distance of a bend-point from its original location could be as large as $d(v,q)(1 + 1/R^2)^{6n}$. Since $(1 + 1/R^2)^{6n} \leq \exp(1/R^2) < 1 + 1/R$, the overall moved distance is at most $1/R$. In particular, all bend-points have distance at most $R + 1$ from the origin, and their pairwise distance is at least $2 - 2/R > 1$.

By construction, at the end of Step 2 we have a RAC_1-drawing of G with all vertices still lying on grid-points (since they were not moved in Step 2). So by the end of Step 2 every point has moved by at most $1/R$ as we have argued (for vertices in Step 1, for bend-points in Step 2). This bound also applies to crossing-points, since they move less than their corresponding bend-points (see Figure 21). We conclude that the drawing remains isomorphic to the original drawing. In Step 3 we now deal with the bend-points that were moved off the grid after being snapped.

Step 3: Refining the grid. While the vertices of G all lie on grid-points at the end of Step 2, this need not be true for bend-points, since these may get moved as a side-effect of snapping a bend-point (even the bend-point itself, in case the two edges incident to the bend-point are RAC-linked). Suppose a bend-point p does not lie on the grid; then at least one of the edges incident to p, say pu, was moved as the result of a RAC-linked edge qx being moved, when the bend-point q was being snapped to the grid. Choose q as the last bend-point being snapped which moves pu, and let qx be the edge RAC-linked to pu. Then the slope of pu must be the same as qx (after snapping q) or orthogonal to it. In either case, the slope of pu corresponds to a slope occurring between two grid-points. The other edge pv incident to v either does not change slope, in which case it maintains its original slope between p and v after p was snapped to the grid, or the same analysis as for pu applies. In all cases, we conclude that the final slopes of pu and pv are slopes that occur between grid-points. Since u and v lie on the grid, we will be able to argue that sufficiently refining the grid ensures that p does as well.

If we express the slope of a line through two grid-points as the reduced ratio s/t between two integers s and t, the denominator t satisfies $1 \leq t \leq (R+1)R^3 \leq 2R^4$ (since our current grid has unit distance $1/R^3$ and its side-length is at most $R+1$). Hence it is sufficient to refine the grid by a factor of at most $(1/(2R^4))^2$ to ensure that the bend-point p, as the intersection between two lines pu and pv which pass through two points of the grid (namely u and v), lies on the new grid. Since we have at most $6n$ bend-points, this leads to a grid with unit distance at most $(2R)^{-4-8\cdot 6n}$ and side-length 1 which positions all vertices and bend-points as grid-points. Since $(2R)^{-4-8\cdot 6n} \leq 2^{-2^{c'n^2}}$, for a fixed constant $c' > c$ independent of n, an integer grid of size $2^{2^{O(n^2)}} \times 2^{2^{O(n^2)}}$ is sufficient to realize the RAC_1-drawing of G. □

6 Large-Angle Crossings

Huang, Eades, and Hong [28] studied the impact of large angles on the readability of drawings, and concluded that large angles improve the readability of drawings, but the angles do not have to be right angles. They expressed the hope that the computational problem becomes easier if the right-angle restriction is relaxed. The *crossing resolution* of a drawing is the smallest angle formed by any two crossing edges. RAC-drawings are drawings of crossing resolution $\pi/2$.

Using common tricks for the existential theory of the reals, we can show that deciding whether an n-vertex graph has a straight-line drawing with crossing resolution at least $\pi/2 - \varepsilon_n$ is $\exists\mathbb{R}$-complete, where ε_n depends on n, doubly exponentially so.

Remark 16 *The large-angle crossing problem was introduced by Dujmović, Gudmundsson, Morin, and Wolle [22] who refer to* α-angle crossing (αAC) *drawings, and independently by Di Giacomo, Didimo, Liotta, and Meijer [17] who talk about* Large Angle Crossing (LAC_α) *drawings. For more background, see the survey by Didimo and Liotta [20, Section 4.2].*

For the following theorem, the *area* of a drawing is the area of a smallest bounding box containing the drawing, assuming that every two vertices and every vertex and non-adjacent edge have at least unit distance.

Theorem 17 *An n-vertex graph G has a* RAC-*drawing if and only if it has a straight-line drawing of area at most $2^{2^{c_1 n^4}}$ and crossing resolution at least $\pi/2 - \varepsilon_n$, where $\varepsilon_n = 2^{-2^{c_2 n^4}}$, and $c_1, c_2 > 0$ are fixed constants not depending on the graph (or n). The two drawings are isomorphic.*

By the theorem, RAC-drawability cannot be distinguished from the α-crossing resolution problem if α is sufficiently (double-exponentially) close to a right angle and the drawing area is bounded (again doubly-exponentially). Why do we need to bound the area? The reason is that a graph may have straight-line drawings with crossing resolution arbitrarily close to $\pi/2$, without a crossing resolution of $\pi/2$ being realizable. As the crossing resolution approaches the limit $\pi/2$, vertices in the drawing may be forced to overlap other vertices, or lie on edges they are not incident to. If we require unit distance between vertices and vertices and edges, then the drawing area becomes unbounded.

This effect, however, does not occur with the graphs we construct in Theorem 1, since we do not create pairs of vertices, or vertices and edges which have to come arbitrarily close to each other. This allows us to drop the area requirement from the following result, which combines Theorem 1 with Theorem 17.

Corollary 18 *Testing whether an n-vertex graph has a straight-line drawing with crossing resolution at least $\pi/2 - \varepsilon_n$ is $\exists\mathbb{R}$-complete, for $\varepsilon_n = 2^{-2^{cn^4}}$ for a fixed integer $c > 0$.*

We do not know whether testing if a graph has crossing resolution at least $\pi/2 - \varepsilon$ remains $\exists\mathbb{R}$-hard for a fixed $\varepsilon > 0$. The current construction very much relies on precision to simulate the existential theory of the reals. It is not clear whether gadgets can be braced to still work if angles are only approximate. We say $\exists\mathbb{R}$-hard, rather than $\exists\mathbb{R}$-complete, since we cannot necessarily express crossing resolution at least $\pi/2 - \varepsilon$ in $\exists\mathbb{R}$, depending on ε; this issue could be resolved by parameterizing the problem by the sine of the angle rather than the angle itself.

To prove the theorem we need the following definability result, which is similar to Lemma 4 and was again first proved by Bieker [11].

Lemma 9 (Bieker [11, Section 6.3]) *Given a graph $G = (V, E)$ and a variable δ we can efficiently compute a polynomial f_G with variables (v_x, v_y) for $v \in V$, (c_x, c_y) for $c \in \binom{E}{2}$, δ and additional variables λ_i, $i \in I$ so that the following two statements are equivalent:*

- *G has a straight-line drawing in which every vertex v is placed at location (v_x, v_y), every vertex has at least unit distance from every other vertex, and every edge it is not incident to, and for every pair of edges $c \in \binom{E}{2}$ that crosses, the crossing is at (c_x, c_y) and the absolute value of the cosine of the crossing angle at (c_x, c_y) is at most δ,*
- *$f_G((v_x, v_y)_{v\in V}, (c_x, c_y)_{c\in\binom{E}{2}}, \delta, (\lambda_i)_{i\in I}) = 0$ for some choice of the λ_i, $i \in I$.*

The polynomial f_G has total degree at most 12, the bit-length of f is $O(\log n)$, and $|I| = O(n^4)$, where $n = |V|$.

Proof: We follow the proof of Lemma 4 and only sketch the differences. We need to express that the lines through pq and st cross at an angle with cosine at most δ. With vectors $a = q - p$ and $b = t - s$ the cosine of the angle formed by pq and st is $a \cdot b/(|a||b|)$, where $\cdot$ is the dot-product of two vectors. So the condition turns into $a \cdot b \leq \delta(|a||b|)$; squaring both sides gives us a polynomial condition in $p_x, p_y, q_x, q_y, \delta$. We then build a polynomial f as follows: for every vertex v in V we have variables (v_x, v_y) and for every pair of edges $c \in \binom{E}{2}$ we have variables (c_x, c_y). Using the polynomials we built, we express that

- every two vertices have at least unit distance,
- every vertex has at least unit distance from every edge it is not incident to,
- every pair of edges is either parallel or crosses in c, where c is the variable representing the pair of edges
- if c is a crossing of two edges, the cosine of their crossing angle is at most δ.

The resulting polynomial f_G has a zero for some choice of the λ-variables if and only if G has a straight-line drawing with vertices at (v_x, v_y) and crossings at (c_x, c_y) with all crossing angles having cosine at least δ.

The analysis of f_G only differs in the bound on $|I|$; since we do not know that the drawing is a RAC-drawing, we can no longer assume a linear bound on $|E|$, so $|E| = O(n^2)$, which implies that $|I| = O(n^4)$. □

To complete the proof of Theorem 17 we need a second result from algebraic geometry, often known as an effective Łojasiewicz inequality; there are many of these, we work with the following version, which is based on Jeronimo, Perruci, Tsigaridas [30, Theorem 1].

Theorem 19 (Jeronimo, Perruci, Tsigaridas [30]) *Suppose f and g are polynomials of total degree d in n variables with coefficients of bitlength at most M. If $Z = \{x : f(x) = 0\}$ is compact, and $g(x) > 0$ for all $x \in Z$, then*

$$g(x) > 2^{-Mn^2 2^n d^{n+1}}.$$

Proof: [Proof of Theorem 17] We are given a graph $G = (V, E)$ with $n = |V|$. Let $f_G(v, c, \delta, \lambda)$ be the polynomial from Lemma 9, where $v \in \mathbb{R}^{2n}$, $c \in \mathbb{R}^{\binom{n}{2}}$, $\delta \in \mathbb{R}$, and $\lambda \in \mathbb{R}^{O(n^4)}$. Then G has a straight-line drawing with the absolute values of all cosines of all crossing angles being less than δ if and only if $f_G(v, c, \delta, \lambda)$ has a zero (for the fixed value δ). If $f_G(v, c, \delta, \lambda)$ has a zero with $\delta = 0$ then G has a RAC-drawing.

Remember that f_G has total degree at most 12, and the bit-length M of f_G is $O(\log n)$, and the total number of variables is $O(n^4)$. It follows from Theorem 13 that there is a fixed integer constant $c_1' > 0$ such that f_G has a zero with $\delta = 0$ if and only if it has such a zero within distance at most $R = 2^{2^{c_1' n^4}}$ from the origin.

In particular, if G has a RAC-drawing, then f_G has a zero with $\delta = 0$, and that zero has distance at most $R = 2^{2^{c_1' n^4}}$ from the origin. Since the upper bound of R applies to all coordinate variables in v, the area of the RAC-drawing is at most $(2R)^2 \leq 2^{2^{(c_1'+2)n^4}}$, which proves the forward direction with $c_1 = c_1' + 2$.

For the other direction, assume that $G = (V, E)$ has a straight-line drawing with area $2^{2^{c_1 n^4}}$, in which all crossing angles differ by at most ε_n from $\pi/2$. Fix such a drawing. We need to show that G has a RAC-drawing.

Suppose α is a crossing angle in the drawing we fixed, that is, $\alpha = \pi/2 + \delta$ with $|\delta| \leq \varepsilon_n$. Then $\cos(\alpha) = \sin(\delta) \leq \delta$. Hence, if all crossing angles in the drawing differ by at most ε_n from $\pi/2$, then all the cosines of crossing angles are at most ε_n.

If f_G has a zero within radius $R = 2^{2^{c_1' n^4}}$ of the origin with $\delta = 0$, then G has a RAC-drawing of area at most $2^{2^{c_1 n^4}}$ and we are done.

We can therefore assume that $f_G(v, c, \delta, \lambda)$ has no zero with $\delta = 0$ within radius R. Let $B(0, R)$ denote the ball around the origin of radius R. We build a polynomial $f(v, c, \delta, \lambda, \mu)$ from f_G by adding $O(n^4)$ variables $\mu \in \mathbb{R}^{O(n^4)}$ to compute R (using repeated squaring) and adding constraints restricting all variables of f_G to have (absolute) value at most R. Then $Z = \{x = (v, c, \delta, \lambda, \mu) : f(v, c, \delta, \lambda, \mu) = 0\}$ is a compact set lying inside $B(0, R)$. Define the projection function $g : (v, c, \delta, \lambda, \mu) \mapsto \delta$. By assumption, $g(x) > 0$ for all $x \in Z$, so we can apply Theorem 19 to conclude that $g(x) > 2^{-2^{c_2 n^4}}$ for some fixed integer $c_2 > 0$ which does not depend on G.

Therefore, if we let $\varepsilon_n = 2^{-2^{c_2 n^4}}$, then if f contains a zero in Z with $g(x) < \varepsilon_n$, we must conclude that there is a zero of f_G with $\delta = 0$, which contradicts our assumptions in this case. □

7 Open Questions

We saw that testing RAC-drawability remains $\exists\mathbb{R}$-hard, even if there is a RAC-drawing with at most 11 crossings per edge. Can the number of crossings be lowered? It is known that the problem remains **NP**-hard even for 1-planar drawings (at most one crossing per edge) [9]. Does the 1-planar version of the problem belong to **NP**? In that case, we would have a situation analogous to what happens for the geometric local crossing number, $\overline{\text{lcr}}(G)$, that is, the smallest number of crossings along each edge in a straight-line drawing of G. Testing whether $\overline{\text{lcr}}(G) \leq 1$ is **NP**-complete [37], but there is a fixed k so that testing $\overline{\text{lcr}}(G) \leq k$ is $\exists\mathbb{R}$-complete [38].

Does RAC-drawability remain $\exists\mathbb{R}$-hard for bounded-degree graphs? Nearly all of our gadgets have bounded degree. The only exception is the empty-face gadget, which is based on $K_{2,n}$'s, and requires unbounded degree. Can this gadget be replaced with a bounded degree gadget or eliminated from the construction in another fashion? Since we do not use this gadget in the fixed embedding case, we can conclude that RAC-drawability does remain $\exists\mathbb{R}$-hard for bounded degree graphs with fixed embedding. Can we get a good bound on the degree in this case? It is conjectured that all cubic graphs have a RAC-drawing, for recent progress on that question, see [4].

The *right-angle crossing number* of a graph is the smallest k so that G has a RAC-drawing with at most k crossings. Our result implies that testing whether the right-angle crossing number is finite is $\exists\mathbb{R}$-complete. Does the problem remain $\exists\mathbb{R}$-complete for small, fixed values of k, like the geometric local crossing number mentioned above? Or can we test whether a graph has a RAC-drawing with one, two, three crossings in polynomial time, as we can for the rectilinear crossing number?

When showing that every RAC$_1$-drawing can be realized on a grid, we obtained a grid of size at most $2^{2^{O(n^2)}}$; can the bound of $O(n^2)$ be improved? The bottle-neck in the current proof is Lemma 4; the encoding in that lemma introduces a quadratic number of variables, which directly leads to the n^2 in the double exponent. If the encoding could be done with a linear number of variables, the resulting grid would have size $2^{2^{O(n)}}$. Whether it is possible to push below that will likely depend on whether RAC$_1$-drawability is $\exists\mathbb{R}$-hard or not, and this is one of the big open problems, of course. A first step towards that may be to see whether there are graphs requiring exponential area in a RAC$_1$-drawing.

We saw that RAC-drawings require algebraic coordinates (Chapter 5), RAC$_1$-drawings can be realized on a double-exponential grid (Theorem 15), and it is known that RAC$_3$-drawings are possible on a polynomial-size grid [19]. Where does that leave RAC$_2$-drawings? Theorem 15 does not obviously imply any upper bound on the grid complexity of RAC$_2$-drawings, though the same techniques should be useful. Is it possible that for RAC$_2$-drawings an exponential-size grid is always sufficient?

Acknowledgments

I would like to thank the anonymous referees for their careful and detailed feedback on this paper, which has led to significant improvements to the presentation.

References

[1] Abrahamsen, M.: Covering polygons is even harder. In: 2021 IEEE 62nd Annual Symposium on Foundations of Computer Science—FOCS 2021, pp. 375–386. IEEE Computer Soc., Los

Alamitos, CA ([2022] ©2022), https://doi.org/10.1109/FOCS52979.2021.00045

[2] Abrahamsen, M., Adamaszek, A., Miltzow, T.: The art gallery problem is ∃ℝ-complete. J. ACM **69**(1), Art. 4, 70 (2022), https://doi.org/10.1145/3486220

[3] Angelini, P., Bekos, M.A., Förster, H., Kaufmann, M.: On RAC drawings of graphs with one bend per edge. Theoret. Comput. Sci. **828(829)**, 42–54 (2020), https://doi.org/10.1016/j.tcs.2020.04.018

[4] Angelini, P., Bekos, M.A., Katheder, J., Kaufmann, M., Pfister, M.: RAC drawings of graphs with low degree. In: 47th International Symposium on Mathematical Foundations of Computer Science, LIPIcs. Leibniz Int. Proc. Inform., vol. 241, pp. Art. No. 11, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern (2022), https://doi.org/10.4230/lipics.mfcs.2022.11

[5] Angelini, P., Cittadini, L., Di Battista, G., Didimo, W., Frati, F., Kaufmann, M., Symvonis, A.: On the perspectives opened by right angle crossing drawings. J. Graph Algorithms Appl. **15**(1), 53–78 (2011), https://doi.org/10.7155/jgaa.00217

[6] Argyriou, E.N., Bekos, M.A., Symvonis, A.: The straight-line RAC drawing problem is NP-hard. J. Graph Algorithms Appl. **16**(2), 569–597 (2012), https://doi.org/10.7155/jgaa.00274

[7] Arikushi, K., Fulek, R., Keszegh, B., Morić, F., Tóth, C.D.: Graphs that admit right angle crossing drawings. Comput. Geom. **45**(4), 169–177 (2012), https://doi.org/10.1016/j.comgeo.2011.11.008

[8] Beinerts, L.: The expert (2014), available at https://www.youtube.com/watch?v=BKorP55Aqvg (last accessed 5/29/2021)

[9] Bekos, M.A., Didimo, W., Liotta, G., Mehrabi, S., Montecchiani, F.: On RAC drawings of 1-planar graphs. Theoret. Comput. Sci. **689**, 48–57 (2017), https://doi.org/10.1016/j.tcs.2017.05.039

[10] Bertschinger, D., El Maalouly, N., Linda, K., Miltzow, T., Weber, S.: The complexity of recognizing geometric hypergraphs. CoRR **abs/2302.13597** (2023), http://arxiv.org/abs/2302.13597

[11] Bieker, N.: Complexity of Graph Drawing Problems in Relation to the Existential Theory of the Reals. Bachelor's thesis, Karlsruhe Institute of Technology (August 2020)

[12] Blum, L., Cucker, F., Shub, M., Smale, S.: Complexity and real computation. Springer-Verlag, New York (1998), https://doi.org/10.1007/978-1-4612-0701-6

[13] Brandenburg, F.J., Didimo, W., Evans, W.S., Kindermann, P., Liotta, G., Montecchiani, F.: Recognizing and drawing IC-planar graphs. Theoret. Comput. Sci. **636**, 1–16 (2016), https://doi.org/10.1016/j.tcs.2016.04.026

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

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

[16] Cardinal, J., Hoffmann, U.: Recognition and complexity of point visibility graphs. Discrete Comput. Geom. **57**(1), 164–178 (2017), https://doi.org/10.1007/s00454-016-9831-1

[17] Di Giacomo, E., Didimo, W., Liotta, G., Meijer, H.: Area, curve complexity, and crossing resolution of non-planar graph drawings. Theory Comput. Syst. **49**(3), 565–575 (2011), https://doi.org/10.1007/s00224-010-9275-6

[18] Didimo, W.: Right angle crossing drawings of graphs. In: Hong, S., Tokuyama, T. (eds.) Beyond Planar Graphs, Communications of NII Shonan Meetings, pp. 149–169. Springer (2020), https://doi.org/10.1007/978-981-15-6533-5_9

[19] Didimo, W., Eades, P., Liotta, G.: Drawing graphs with right angle crossings. Theoret. Comput. Sci. **412**(39), 5156–5166 (2011), https://doi.org/10.1016/j.tcs.2011.05.025

[20] Didimo, W., Liotta, G.: The crossing-angle resolution in graph drawing. In: Thirty essays on geometric graph theory, pp. 167–184. Springer, New York (2013), https://doi.org/10.1007/978-1-4614-0110-0_10

[21] Didimo, W., Liotta, G., Montecchiani, F.: A survey on graph drawing beyond planarity. ACM Comput. Surv. **52**(1) (Feb 2019), https://doi.org/10.1145/3301281

[22] Dujmović, V., Gudmundsson, J., Morin, P., Wolle, T.: Notes on large angle crossing graphs. Chic. J. Theoret. Comput. Sci. pp. Article 4, 14 (2011), https://doi.org/10.4086/cjtcs.2011.004

[23] Erickson, J., van der Hoog, I., Miltzow, T.: Smoothing the gap between NP and ER. In: 61st Annual Symposium on Foundations of Computer Science—FOCS 2020, pp. 1022–1033. IEEE Computer Soc., Los Alamitos, CA (2020), https://doi.org/10.1109/FOCS46700.2020.00099

[24] Förster, H., Kaufmann, M.: On compact RAC drawings. In: Grandoni, F., Herman, G., Sanders, P. (eds.) 28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference). LIPIcs, vol. 173, pp. 53:1–53:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020), https://doi.org/10.4230/LIPIcs.ESA.2020.53

[25] Goodman, J.E., Pollack, R., Sturmfels, B.: The intrinsic spread of a configuration in $\mathbf{R}^d$. J. Amer. Math. Soc. **3**(3), 639–651 (1990), https://doi.org/10.2307/1990931

[26] Grigor'ev, D.Y., Vorobjov, Jr., N.N.: Solving systems of polynomial inequalities in subexponential time. J. Symbolic Comput. **5**(1-2), 37–64 (1988), https://doi.org/10.1016/S0747-7171(88)80005-1

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

[28] Huang, W., Eades, P., Hong, S.: Larger crossing angles make graphs easier to read. J. Vis. Lang. Comput. **25**(4), 452–465 (2014), https://doi.org/10.1016/j.jvlc.2014.03.001

[29] Huang, W., Hong, S., Eades, P.: Effects of crossing angles. In: IEEE VGTC Pacific Visualization Symposium 2008, PacificVis 2008, Kyoto, Japan, March 5-7, 2008. pp. 41–46. IEEE Computer Society (2008), https://doi.org/10.1109/PACIFICVIS.2008.4475457

[30] Jeronimo, G., Perrucci, D., Tsigaridas, E.: On the minimum of a polynomial function on a basic closed semialgebraic set and applications. SIAM J. Optim. **23**(1), 241–255 (2013), https://doi.org/10.1137/110857751

[31] Kim, E., de Mesmay, A., Miltzow, T.: Representing matroids over the reals is ∃r-complete. CoRR **abs/2301.03221** (2023), https://doi.org/10.48550/arXiv.2301.03221

[32] Matoušek, J.: Intersection graphs of segments and ∃ℝ. ArXiv e-prints (2014), arXiv:1406.2636 (last accessed 6/10/2020)

[33] Miltzow, T., Schmiermann, R.F.: On classifying continuous constraint satisfaction problems. In: 2021 IEEE 62nd Annual Symposium on Foundations of Computer Science—FOCS 2021, pp. 781–791. IEEE Computer Soc., Los Alamitos, CA ([2022] ©2022), https://doi.org/10.1109/FOCS52979.2021.00081

[34] Mnëv, N.E.: The universality theorems on the classification problem of configuration varieties and convex polytopes varieties. In: Topology and geometry—Rohlin Seminar, Lecture Notes in Math., vol. 1346, pp. 527–543. Springer, Berlin (1988)

[35] Poonen, B.: Characterizing integers among rational numbers with a universal-existential formula. Amer. J. Math. **131**(3), 675–682 (2009), https://doi.org/10.1353/ajm.0.0057

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

[37] Schaefer, M.: Picking planar edges; or, drawing a graph with a planar subgraph. In: Graph drawing, Lecture Notes in Comput. Sci., vol. 8871, pp. 13–24. Springer, Heidelberg (2014), https://doi.org/10.1007/978-3-662-45803-7_2

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

[39] Schaefer, M.: The complexity of angular resolution. J. Graph Algorithms Appl. **27**(7), 565–580 (2023), https://doi.org/10.7155/jgaa.00634

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

[41] Shor, P.W.: Stretchability of pseudolines is NP-hard. In: Applied geometry and discrete mathematics, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., vol. 4, pp. 531–554. Amer. Math. Soc., Providence, RI (1991)

[42] Wikipedia contributors: Existential theory of the reals — Wikipedia, the free encyclopedia (2021), https://en.wikipedia.org/w/index.php?title=Existential_theory_of_the_reals (last accessed 5/28/2021)