Intersection Graphs and Geometric Objects in the Plane

vorgelegt von
M. Sc. Mathematik
Udo Hoffmann
geb. in Mönchengladbach

Von der Fakultät II – Mathematik und Naturwissenschaften
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
– Dr. rer. nat. –

genehmigte Dissertation

Promotionsausschuss

Vorsitzender:	Prof. Dr. Boris Springborn
Berichter:	Prof. Dr. Stefan Felsner
	Prof. Dr. Wolfgang Mulzer
	Prof. Jean Cardinal

Tag der wissenschaftlichen Aussprache: 24. März 2016

Berlin 2016

Acknowledgements

This work was mainly done in the *Diskrete Mathematik* working group. I would like to thank all current and former members, I had the pleasure of working and discussing with. We had a nice working atmosphere. Many thanks to: Stefan Felsner, Linda Kleist, Veit Wiechert, Nieke Aerts, Steven Chaplick, Piotr Micek, Irina Mustaţă, Daniel Heldt, Thomas Hixon, and Kolja Knauer.

Especially, thanks to Stefan, for always giving advice how to improve my work, and for always giving the possibility to talk to.

I want to thank Jean Cardinal for hosting me in Brussels, and giving many ideas for the second part of this thesis.

Many thanks to Linda, Veit, Steven, and Axel for proofreading parts of this work.

I want to mention that this work had not been possible without the funding of the DFG research training group *Methods for Discrete Structures*.

Finally, I want to thank Stefan, Jean, and Wolfgang Mulzer for agreeing to read this thesis.

Thank you!
Udo

Contents

What is this thesis about?

This thesis deals with geometric representations of graphs in the plane. Three main problems connected with geometric representations of graphs are:

- Which graphs admit a certain representation? (*Characterization problem*)
- Does a given graph admit a representation of a certain type? (*Recognition problem*)
- How complex are the representations?

We explain the three problems representatively for segment intersection graphs, the intersection graphs of straight line segments in the plane. Segment intersection graphs are known for the difficult structure of their representations. A representation using explicit coordinates cannot validated in polynomial time in the size of the graph due to large coordinates [KM94]. It seems unlikely, that there is a short description of a representation that allows to be checked in polynomial time. This indicates, that the recognition of segment intersection graphs possibly cannot be solved in nondeterministic polynomial time (NP). This situation changes, when the number of slopes that are used

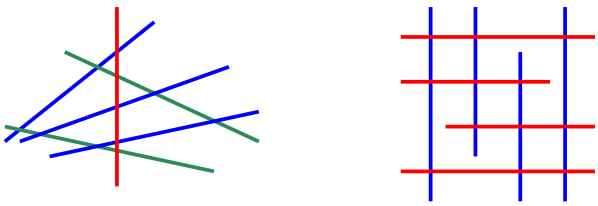

Figure 0.1.: A segment intersection representation and a grid intersection representation.

by the line segments in a representation is bounded by a constant s [KM94]: The recognition problem for segment intersection graphs that have a representation with at most s slopes is in NP, i.e., there is a segment representation of a graph can be checked in polynomial time. If the number of slopes of a segment intersection graph is bounded by two, we call the class grid intersection graphs and we can even give a "nice" combinatorial certificate, that does not (explicitly) use coordinates of a geometric representation, by a forbidden submatrix of the bipartite adjacency matrix after resorting the rows and columns.

In the first part of this thesis, we work with graph representations that have a combinatorial description, while the second part deals with the existential theory of the reals ($\exists\mathbb{R}$), a complexity class, which is connected to "complex" geometric representations.

In Chapter 1, we point out a connection of the order dimension of partial orders to grid intersection graphs that are comparability graphs the partial orders. We show, that the order dimension of grid intersection graphs is bounded by four. This observation is used to study the containment relation on many subclasses of grid intersection graphs. We argue that the order dimension is a useful tool for this purpose. We also show that the order dimension of bipartite intersection graphs of segments is not bounded, but we can give an upper bounds on the dimension that is linear in the number of slopes that is used in the representation. This leads to the study of the slope number of segment intersection graphs, the minimal number k, such that there is a segment intersection representation using k slopes.

In Chapter 2, we construct a class of intersection graphs of segments from planar graphs (with Hamiltonian path). The slope number of these segments intersection graphs is two if and only if the planar graph has a Hamiltonian cycle. We show that the Hamiltonian cycle problem in planar triangulations is NP-complete, even if a Hamiltonian path is given. This way we prove, that computing the slope number of a segment intersection graph is NP-hard. We use this connection to show that the slope number of a graph does not change "continuously": The slope number may drop from a linear number in number of vertices down to two, upon the removal of a single vertex.

Chapter 3, the beginning of the second part, deals with the realizability of circular sequences. Mnëv [Mnë88] has shown that for each semialgebraic set S, there is an order type (combinatorial descriptions of point sets), that has a realization space with the "the same geometric structure" as S. A consequence of this result is, that deciding the realizability of an order type is as hard as deciding the emptiness of a semialgebraic set. This problem is complete in the existential theory of the reals. We give a modified proof for this complexity result. Our modified proof allows to show, that deciding the realizability of a circular sequence of an order type is complete in $\exists\mathbb{R}$, even if the order type is realizable and a realization is given.

In Chapter 4, we use the proof we presented in Chapter 3 to show, that determining the slope number of a segment intersection graph is hard in $\exists\mathbb{R}$.

In Chapter 5, we use the hardness result for circular sequences from Chapter 3 to show that the recognition problem of point visibility graphs is hard in the existential theory of the reals. We also show that there are point visibility graphs that have at least one irrational coordinate in each representation. We show that deciding the existence of a representation with rational coordinates is possibly undecidable, depending on the (un)decidability of the "existential theory of the rationals".

Chapter 6 deals with the planar slope number of planar graphs. The planar slope number of a planar graph G is defined as the minimum number k, such that G has a straight line drawing with k slopes used for drawing the edges. We show that this problem is also hard in the existential theory of the reals, and point out consequences for drawings that minimize the planar slope number.

We conclude by pointing out some open questions that appeared in this thesis in Chapter 7.

0. Notations and definitions

0.1. Graphs

Throughout this thesis we use the following notations.

By $G = (V; E)$ we denote a graph G on the vertex set V with edges $E \subseteq \binom{V}{2}$. Usually we write vw for an edge instead of $\{vw\}$. By $G[V']$ we denote the induced subgraph of a graph G with the vertex set $V' \subseteq V$. The *open neighborhood* of a vertex v is the set $N(v) := \{w \in V | vw \in E\}$, the *closed neighborhood* by $N[v] := N(v) \cup v$.

In the thesis we consider geometrical representations of graphs. We represent a graph by representing vertices as geometric objects. We assign a set S_v to each vertex v. An edge between two vertices v and w exists if and only if a relation between the objects S_v and S_w hold. Examples of such relations are *intersections*, where the edge vw exists if and only if S_v and S_w intersect as shown in Figure 0.1, or *visibilities*, where the edge vw exists if S_v and S_w can *see* each other.

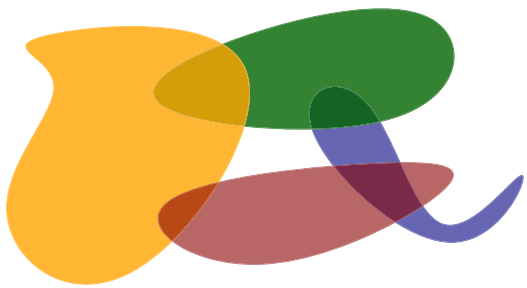

Figure 0.1.: An intersection representation of C_4, a cycle of length four, using connected sets in the plane.

With a slight abuse of notation, we will denote the geometric objects by the name of the vertex corresponding to it, for example we write "v intersects w" instead of "S_v intersects S_w".

By a representation R we denote the set geometric objects representing the vertices. Given a geometric representation R of a graph G we define the subrepresentation $R[V'] := \{S_v | v \in V'\}$, $V' \subset V$. For intersection graphs, a subrepresentation $R[V']$ gives an intersection representation of $G[V']$. This property does not hold in general for visibility graphs: The visibility graph represented by $R[V']$ can be a supergraph of

$G[V']$, because the objects of $R[V \setminus V']$ may have *blocked* visibilities between objects of $R[V']$ in the original representation $R[V]$. An induced subgraph of a visibility graph is not necessarily a visibility graph again. For example an independent set is not a visibility graph (with the usual definitions of visibility), since a non-edge requires a blocked visibility, which requires another vertex which is *seen* instead.

The two main graph classes we study in this thesis are *segment intersection graphs* and *point visibility graphs* in the plane. In the rest of this section we give an overview on representations of graphs that are used in this thesis.

0.1.1. Graph drawings

We discuss "classical" drawings of graphs: The vertices of a graph are represented as points and the edges as curves between the incident vertices. Such a drawing is a *straight-line drawing* if the curves representing the edges are line segments.

We call a drawing *planar* if it has a drawing with non-crossing edges. A planar drawing is *outerplanar* if all vertices are incident to the unbounded face.

Outerplanar graphs are exactly the graphs that have a *1-page book embedding*, a planar drawing where all points lie in one line ℓ and the edges lie in one half-space that is bounded by ℓ. In general, a *k-page book embedding* of a graph G is an ordering of the vertices along with the partition of the edges into k sets, such that each of the k sets has a 1-page book embedding with the given vertex ordering. This can be seen as a drawing of a graph with the vertices on the *spine* of a book, where the edges of one of the k blocks are drawn on only one page of the book.

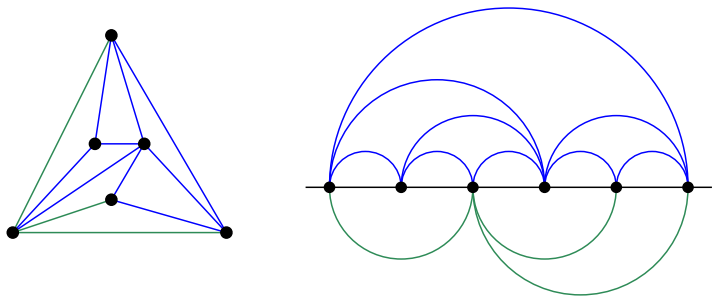

Figure 0.2.: A planar straight line drawing and a 2-page book embedding of a planar graph. The blue edges form an outerplanar graph.

Yannakakis [Yan89] has shown, that each planar graph has a 4-page book embedding. On the other hand, graphs with a 2-page book embedding are planar. The following lemma characterizes graphs with 2-page book embeddings.

Lemma 0.1 ([BK79]). *A graph G has a 2-page book embedding if and only if G is a subgraph of a planar graph with Hamiltonian cycle.*

0.1.2. Intersection graphs in the plane

While every graph can be represented as intersection graph of a collection of general sets we focus on subclasses of intersection graphs of connected sets in the plane, which often have a rich structure from connections to planarity. The most general class here is the class of *string graphs*, the intersection graphs of curves in the plane. This is equivalent to the intersection graph of connected regions. Not every graph is a string graph. A class of non-string graphs is given by *full subdivisions* (every edge is subdivided) of non-planar graphs [EET76].

Lemma 0.2 ([Sin66, HNZ91]). *The full subdivision P_G of a graph G is a string graph if and only if G is planar. If G is planar then P_G is even a* segment contact graph *of horizontal and vertical segments.*

A graph G that has a string representation, such that each pair of curves intersects at most once, then G is called *pseudosegment intersection graph.*

A *contact graph* is a special kind of intersection graphs where the sets are interiorly disjoint. The existence of a segment contact representation of P_G for planar G follows from the following more general theorem.

Lemma 0.3 ([HNZ91]). *A planar bipartite graph is a segment contact graph of vertical and horizontal segments.*

Such a contact representation of P_G for a planar graph G can also be considered as *weak bar visibility representation*, where the vertices are represented by vertical segments and the edges by horizontal segments (sight lines). If the lower endpoint of each vertical segment lies on a common horizontal line we call the representation a *weak semi-bar visibility representation.*

Lemma 0.4. *A graph G has a weak semi-bar visibility representation if and only if G is outerplanar.*

The semi-bar visibility representations of a graph G have a close connection to 1-page book embeddings: The possible orders of the semi-bars coincide with the order of vertices in a 1-page book embedding. Those orders are the possible (cyclic) orders of the outer-face cycle in an outerplanar drawing.

A large part of this thesis deals with *segment intersection graphs* (SEG) the intersection graphs of line segments in the plane. We consider only SEG representations without intersections of parallel segments. Those representations are often called *pure* representations. Every SEG graph has a pure representation. We denote the class of graphs with a pure representation using at most k directions/slopes is denoted by k-dir.

The class of graphs that can be realized using a prescribed set of slopes $\alpha_1, \ldots, \alpha_k$ is denoted by k-dir$(\alpha_1, \ldots, \alpha_k)$. The class of graphs that can be represented by the slopes $\alpha_1, \ldots, \alpha_k$ and $\beta_1, \ldots, \beta_k$ is the same if and only if the directions can be transformed into each other by a linear transformation [ČKNP02]. This is always possible for $k \leq 3$. For larger k, we can pick three directions, say $\alpha_1, \alpha_2, \alpha_3$, and map them on arbitrary other directions $\beta_1, \beta_2, \beta_3$ by a linear transformation. The image of all other directions is determined by the choice of this maps.

Pure 2-dir representations (*grid intersection representations*) have a simple combinatorial description using its bipartite adjacency matrix.

Lemma 0.5 ([HNZ91]). *A bipartite graph G is a grid intersection graph if and only if its bipartite adjacency matrix has a* cross-free *ordering, i.e., an ordering of the rows and columns, such that the matrix does not contain a submatrix of the following kind:*

$$\begin{pmatrix} & 1 & \\ 1 & 0 & 1 \\ & 1 & \end{pmatrix}.$$

For general SEG representations no simple combinatorial characterization is known.

Many properties of segment intersection graphs are easier to show, when we can fix a segment representation. A tool that allows us to construct a graph from a segment representation, such that each representation of the graph contains a copy of the old representation, is given by the following lemma.

Lemma 0.6 (order forcing lemma, [KM94]). *Let R be a simple segment representation of a graph G. There exists a segment intersection graph G_R on $O(|V(G)|^2)$ vertices, such that any segment representation of G_R contains a copy of the segment representation R. Moreover, G_R has a segment representation using s slopes ($s \geq 2$) if and only if G can be realized using at most s slopes. The graph G_R can be constructed from R in polynomial time.*

0.1.3. Visibility graphs

Next to bar- and semi-bar visibility representations, we consider *point visibility graphs*. Given a set of objects S in the plane, we say two objects p and q in the plane *see* each other if there is a lines segment that intersects exactly p and q. The edges of the visibility graph of S are the pairs of objects in S that can see each other. A *point visibility graph* is a graph that is the visibility graph of some point set. For an overview on different visibility graphs considered in literature we refer to Chapter 5.

0.2. Partial orders

A *partially ordered set* (*poset*) $P = (X, \leq_P)$ is a binary relation that is

- reflexive: $a \leq_P a$ for all $a \in X$,
- antisymmetric: $a \leq_P b$ and $b \leq_P a$ implies $a = b$ for all $a, b \in X$,
- transitive: $a \leq_P b$ and $b \leq_P c$ implies $a \leq_P c$ for all $a, b, c \in X$.

Two elements a and b are *comparable* if $a \leq_P b$ or $b \leq_P a$. The *incomparable pairs* $\mathrm{inc}(P)$ of P are all pairs $(a, b) \in X \times X$ that are not comparable. The set of *maxima* $\mathrm{Max}(P)$ is the set of elements x with no y such that $x <_P y$. Similarly, the set of *minima* $\mathrm{Min}(P)$ is defined.

A linear order $L = (X, \leq_L)$ on X is a *linear extension* of P when $x \leq_P y$ implies $x \leq_L y$. A family $\mathcal{R}$ of linear extensions of P is a *realizer* of P if $P = \bigcap_{i \in \mathcal{R}} L_i$, i.e., $x \leq_P y$ if and only if $x \leq_L y$ for every $L \in \mathcal{R}$. The *dimension* of P, denoted $\dim(P)$, is the minimum size of a realizer of P. This notion of dimension for partial orders was defined by Dushnik and Miller [DM41]. The dimension of P can also be defined as the minimum t, such that P admits an order preserving embedding into the dominance order on $\mathbb{R}^t$, i.e., we can associate a vector $x = (x_1, \ldots, x_t) \in \mathbb{R}^t$ with each element $x \in X$, such that $X \leq_P y$ if and only if $x_i \leq y_i$ for $i = 1, \ldots, t$. We denote this order on vectors as the *dominance order* and denote it by $x \leq_{\mathrm{dom}} y$. We can construct t linear extensions from an embedding of the order in $\mathbb{R}^t$ by the orthogonal projection of the points onto the t coordinate axes.

A poset can be considered as a directed graph. The undirected graph we obtain by dropping the orientation of the edges is called the *comparability graph*. In this thesis we consider mainly *bipartite posets* or posets of *height 2*, which are the posets where each element is minimum or maximum. Equivalently, bipartite posets are the posets with bipartite comparability graph. For a bipartite graph $G = (A, B; E)$, we define the poset Q_G where $A = \mathrm{Max}(Q_G)$ and $B = \mathrm{Min}(Q_G)$ and $a > b$ if $a \in A, b \in B$ and $ab \in E$.

An *interval order* is a partial order $P = (X, \leq_P)$ admitting an interval representation, i.e., a mapping $x \to [a_x, b_x]$ from the elements of P to intervals in $\mathbb{R}$ such that $x \leq_P y$ iff $b_x \leq a_y$. The *interval dimension* of P, denoted $\mathrm{idim}(P)$, is the minimum number t such that there exist t interval orders I_i with $P = \bigcap_{i=1}^{t} I_i$. Equivalently, the interval dimension can be defined as the smallest t, such that P is represented as dominance order of boxes (the Cartesian product of the different intervals of one element) in $\mathbb{R}^t$.

Since every linear order is an interval order $\mathrm{idim}(P) \leq \dim(P)$ for all P. If P is bipartite, then dimension and interval dimension differ by at most one, i.e., $\dim(P) \leq \mathrm{idim}(P) + 1$, [Tro92, p.47].

For the sake of brevity we define the *dimension of a bipartite graph* G to be equal to the dimension of Q_G. The freedom that we have in defining Q_G, i.e., the choice of the color classes, does not affect the dimension. This is an easy instance of the fact that dimension is a comparability invariant (see [TMS76]).

The prime example for (bipartite) posets of large dimension is the *standard example* S_n of dimension n, the poset Q_G, where G is the complete bipartite $K_{n,n}$ with one perfect matching removed [Tro92, p.12].

0.3. Complexity theory

In this section we remind the reader of some widely known concepts in complexity theory and give a short introduction on some infrequently used complexity classes. For an overview we refer to [AB09].

By P we denote the class of problems that can be solved by algorithms in polynomial time in the input size by a touring machine. Problems that lie in the complexity class NP are those that can be solved in nondeterministic polynomial time, i.e., a given solution (a *certificate*) of the problem can be verified by an algorithm in polynomial time.

Next to these classic complexity classes, we consideration is the *existential theory of the reals*, which we abbreviate by $\exists\mathbb{R}$. The existential theory of the reals is characterized by the following problem: Given a quantifier-free formula $F(x_1, \ldots, x_n)$ using the numbers 0 and 1, addition and multiplication operations, strict and non-strict comparison operators, Boolean operators, and the variables $x_1, \ldots, x_n$ and we are asked whether there exists an assignment of real values to $x_1, \ldots, x_n$, such that F is satisfied. This amounts to deciding, whether a system of polynomial inequalities admits a solution over the reals.

It is known from the Tarski-Seidenberg Theorem that $\exists\mathbb{R}$ (and even a more general class, the *first-order theory over a real closed field*) is decidable. Only much more recently, a polynomial space (PSPACE) algorithm for problems in $\exists\mathbb{R}$ has been proposed by Canny [Can88].

We call a problem complete in $\exists\mathbb{R}$if it is polynomially equivalent to a $\exists\mathbb{R}$-complete problem.

Some geometric problems have been shown to be complete in $\exists\mathbb{R}$, many based on the fundamental work by Mnëv [Mnë88]. The notation $\exists\mathbb{R}$ has been proposed more recently by Schaefer [Sch09].

Some geometric problems $\exists\mathbb{R}$-complete (resp. hard for optimization problems) include.

- *abstract order type realizability/stretchability of pseudoline arrangements* [Mnë88],

- recognition of segment intersection graphs [KM94],
- computing the *rectilinear* (straight line) crossing number of a graph [Bie91],
- algorithmic *Steinitz problem* [RGZ95],
- recognition of *unit distance graphs* and realizability of *planar linkages* [Sch12],
- simultaneous *geometric* (i.e., straight line) graph embedding [Kyn11],
- recognition of d-dimensional *Delauney triangulations* [APT14].

The following "non-geometrical" decision problems are also complete in $\exists\mathbb{R}$.

- solvability of a strict polynomial inequality system,
- solvability of a polynomial equation in several variables.

For some of the geometric problems above, the question for a representation on the grid (i.e., with integer coordinates) arises. For polytopes and order types the complexity of this is a longstanding open problem [Stu87]: Asking for integer solutions to the formula which certify a problem to be hard in $\exists\mathbb{R}$ do not capture that a rational solution leads to a representation on the integer grid by scaling. Thus, a representation on the integer grid exists if we find a rational solution. Similar to $\exists\mathbb{R}$ we denote the *existential theory of the rationals* by $\exists\mathbb{Q}$. The decidability of $\exists\mathbb{Q}$ is a longstanding open problem. Efforts have been made to define the integers via rational logic. For example Poonen [Poo09] characterized the integers using a formula with only two universal quantifiers. Eliminating those quantifiers would imply that the computational complexity of the existential theory of the integers and the rationals is polynomially equivalent.

Furthermore, we use the notion of *decidablility*. Roughly speaking, a decision problem is *decidable* if there exists an algorithm that solves the problem and terminates after a finite number of steps. We call an algorithm that terminates after a finite number of steps and always gives the correct answer to a decision problem an *effective* algorithm. If there is no effective algorithm for a problem it is called *undecidable*. Some of the most prominent examples of undecidable problems are

- the *halting problem*, which asks whether a program will terminate or run forever on a given input,
- *Hilbert's tenth problem*, which asks whether a *Diophantine equation*, i.e., a polynomial equation in several variables, has a solution over the integers.

Both problems above are *positive semi-decidable*: If "yes" instances of the problems are given there exists an algorithm gives the correct answer after a finite number of steps. For example, a given polynomial equation can be checked for roots in the integers by testing combinations of integers with an absolute value that is bounded by a constant M. Increasing M after testing all combinations leads to an algorithm that finds a solution after a finite number of steps. However, we cannot give an upper bound on the constant M and thus the algorithm cannot correctly answer "no".

0.4. Projective geometry

For many geometric problems it is useful to use *projective geometry* instead of Euclidean geometry. We refer to [RG11] for further reading. A *projective plane* satisfies the following axioms:

1. Any two distinct points are contained in exactly one line.
2. Any two distinct lines intersect in exactly one point.
3. There are four points such that no line is incident to more than two points.

We call a map f a *projective transformation* if it preserves point-line incidences, i.e., each line ℓ, which is spanned by two points p and q, is mapped onto the line spanned by $f(p)$ and $f(q)$.

We will use two models for the *real projective plane*, the *extended Euclidean plane* and *homogeneous coordinates*.

For the *extended Euclidean plane*, we add a new *point at infinity* for each class of parallel lines. Parallel lines intersect in this point. The new points lie on the *line at infinity*.

A way to give coordinates to the extended Euclidean plane are *homogeneous coordinates*: We consider points as 1-dimensional subspaces of the vector space $\mathbb{R}^3$. A line through two distinct points is represented by the 2-dimensional subspace containing the two points/subspaces. In this model, projective transformations correspond exactly to linear transformations of $\mathbb{R}^3$.

To obtain "a plane" from homogeneous coordinates we consider an affine plane A that does not intersect the origin. We consider the intersection of the points and lines with A. The 1-dimensional subspaces (points), that are not parallel to A, intersect A in a point. The 2-dimensional subspaces (lines), that are not parallel to A, intersect A in a line. The 2-dimensional subspace L, that is parallel to A, represents the line at infinity. The 2-dimensional subspaces that correspond to parallel lines on A intersect in a 1-dimensional subspace that is contained in L, i.e., on a point on the line at infinity.

This way, the extended Euclidean plane is contained in the model using homogeneous coordinates.

We usually choose A to be given by $z = 1$. Thus, we can assume that points (1-dimensional subspaces), that do not lie on the line at infinity, are given by $p = \lambda(a, b, 1), \lambda \in \mathbb{R}$, and its coordinate in the extended Euclidean plane by $p = (a, b)$.

An important invariant of a 4-tuple of points under projective transformation is the *cross-ratio*. The cross-ratio $(a, b; c, d)$ of four points $a, b, c, d \in \mathbb{R}^2$ is defined as

$$(a, b; c, d) := \frac{|a, c| \cdot |b, d|}{|a, d| \cdot |b, c|},$$

where $|x, y|$ is the determinant of the matrix obtained by writing the two vectors as columns. For four points a, b, c, d on a line, the cross-ratio is given by $\frac{\overrightarrow{ac} \cdot \overrightarrow{bd}}{\overrightarrow{ad} \cdot \overrightarrow{bc}}$, where $\overrightarrow{xy}$ denotes the oriented distance between x and y.

0.5. (Abstract) order types and (pseudo)line arrangements.

A *projective pseudoline arrangement* $\mathcal{L}$ is collection of closed curves in the projective plane, such that each pair of curves intersect exactly once. We call a pseudoline arrangement *simple* if no three curves intersect in one point. If there is a collection of projective lines[1] with the same combinatorial properties (the same order of intersections on each line as the collections of curves) as $\mathcal{L}$, we call $\mathcal{L}$ *stretchable* and the arrangement has a representation as *line arrangement*. Pseudoline arrangements are often described as *wiring diagrams* as shown in Figure 0.3: The pseudolines are drawn as x-monotone curves on n levels. The intersection between two pseudolines appears when the two curves *switch* their consecutive level.

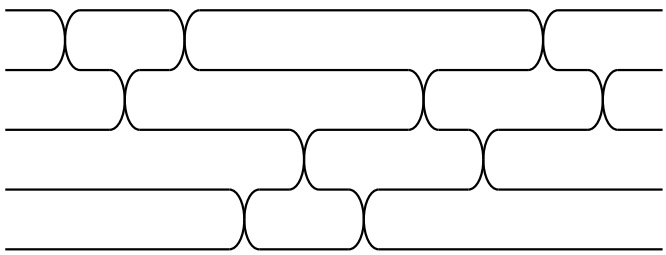

Figure 0.3.: A pseudoline arrangement represented by a wiring diagram.

The *realization space* of a line arrangement is the set possible choices for coordinates of the lines, e.g. by giving a normal vector of the planes in $\mathbb{R}^3$ that determine the lines in the projective plane. (Thus we have coordinates in $\mathbb{R}^{3|\mathcal{L}|}$.) By definition, a pseudoline

[1] Lines in the projective plane are closed curves.

arrangement is stretchable if and only if its realization space is non-empty. Usually the realization space is considered modulo projective transformations.

The model of representing the lines by the normal vector of the planes allows for a simple definition of a *dual map* $\mathcal{D}_{proj}$: A projective line (plane in $\mathbb{R}^3$) is mapped onto the 1-dimensional subspace that is spanned by a normal vector of its plane and vice versa. This map is an involution (self-inverse). Due to nicer mapping we usually use the duality on a parabola $\mathcal{D}$. The map $\mathcal{D}$ maps the line given by $y = ax - b$ onto the point (a, b). It can be obtained by applying a projective transformation before applying the projective duality, thus it also preserves incidences and the cross-ratio. A point p that lies on the intersection of the two lines ℓ_1 and ℓ_2 is mapped onto the line $\mathcal{D}(p)$, that is spanned by the points $\mathcal{D}(\ell_1)$ and $\mathcal{D}(\ell_2)$. It also preserves the *orientation* of each triple of intersecting lines: As shown in Figure 0.4, three lines that do not intersect in one point in the extended Euclidean plane are mapped by D onto three points in *general position*, i.e., the points are not collinear. The triangle, that is bounded by the

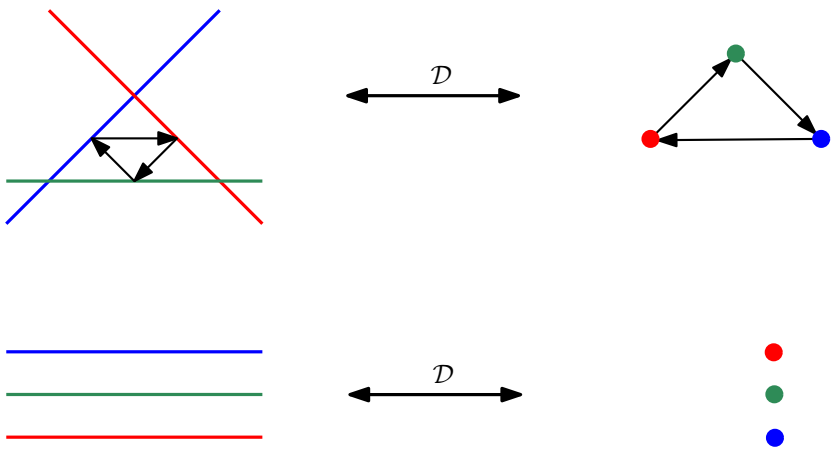

Figure 0.4.: Duality preserves the orientation of triples and the order on lines.

three lines, gives an orientation of the lines by considering the clockwise order of the lines on the boundary (ℓ_1, ℓ_2, ℓ_3). This is the same orientation of points of the dual triangle spanned by the dual points $(\mathcal{D}(\ell_1), \mathcal{D}(\ell_2), \mathcal{D}(\ell_3))$. In the case of three lines that intersect in one point (or are parallel) the dual points points are collinear. The order of the lines around their intersection point coincides with the order of the points on their common dual line.

The information of the orientation of each triple and the order of points on a line fully determines the combinatorial information of the pseudoline arrangement. On the other hand, the orientation of each triangle encodes the full the pseudoline arrangement.

Thus, a pseudoline arrangement can be dualized by giving the orientation of each triple of points, such that the triples satisfy certain axioms, see for example [BLVS$^+$99]. The

axioms come from dualizing the restrictions on curves to be a pseudoline arrangement. The orientation of each triple of points is called an *abstract order type*. An abstract order type is *realizable* if there is a point set with this orientation. This is the case if and only if its dual pseudoline arrangement is stretchable. An abstract order type is called *simple* if and only if the dual pseudoline arrangement is simple. This is the case if and only if no three points are collinear.

We often drop the word "abstract" and only use the term "order type" when we are mainly interested in the realizability.

Abstract order types are also known as *rank* 3 *oriented matroids*. The orientation of a triple of points (p_1, p_2, p_3) in homogeneous coordinates can also be determined by the map

$$\begin{aligned} \chi : \mathbb{R}^{3\times 3} &\to \{-, 0, +\} \\ (p_1, p_2, p_3) &\mapsto \operatorname{sign}\left(\det\left((p_1, p_2, p_3)\right)\right). \end{aligned}$$

If the points are given with positive z-coordinate, then the triple is oriented counterclockwise if the determinant is positive, clockwise if the determinant is negative, and they are collinear if the determinant is zero. The map χ is called the *chirotope*.

For an order type we can consider the *circular sequence*. Given a point set that realizes an order type, we define the circular sequence as the sequence of orders of points that appear as the orthogonal projection of the point set onto a rotating, directed line. This leads to a sequence of $\binom{n}{2} + 1$ permutations (assuming general position) of the points while rotating by an angle of π. The sequence of permutations is called *allowable* or *circular sequence*. An allowable sequence describes a unique abstract order type arrangement [GP80], whereas an order type usually has several allowable sequences. A circular sequence can be uniquely described by the order of *switches*, the sequence of transpositions that appear between consecutive permutations in the allowable sequence [GP80]. The sequence of the switches is the order of the slopes of the lines that are spanned by the pairs of points of the order type. In the dual line arrangement the order of switches describes the order of intersection points of lines in x-direction (if the rotating line starts and ends at vertical position). A circular sequence describes a unique order type [GP80]. On the other hand, one abstract order type can be realized by many circular sequences.

0.6. Geometry and topology

While describing realization spaces of order types, line arrangements or other geometric objects we need some basic definitions from algebraic geometry and topology.

A *semialgebraic set* is a set that can be described as the solution set of a polynomial inequality system consisting of strict and non-strict inequalities and equations. A semialgebraic set is called *primary* if it has a description only by gstrict polynomial inequalities and equations. Note that a primary semialgebraic set is an open set.

Two sets in $\mathbb{R}^n$ are *homotopy equivalent* if they can be continuously deformed into each other (they have the same topological structure).

Two semialgebraic sets V and W are *stably equivalent* if there is a homeomorphism $f : V \times \mathbb{R}^s \to W \times \mathbb{R}^t$, where both maps f and f^{-1} are polynomial maps. Stable equivalence of two sets implies homotopy equivalence.

Part I.

Combinatorial properties of intersection graphs

1. The dimension of bipartite intersection graphs

In this chapter we study bipartite intersection graphs in the plane from the perspective of order dimension. We start with the observation that partial orders of height two whose comparability graph is a grid intersection graph have order dimension at most four. Starting from this observation we provide a comprehensive study of classes of graphs between grid intersection graphs and bipartite permutation graphs and the containment relation on these classes. Order dimension plays a role in many arguments.

This chapter is based on [CFHW15].

1.1. Background of subclasses of grid intersection graphs.

Subclasses of GIGs appear in several technical applications. For example in *nano PLA-design* [STS11] and in biology [HATI11].

Other restrictions on the geometry of the representation are used to study algorithmic problems. For example, *stabbability* has been used to study hitting sets and independent sets in families of rectangles [CF13]. Additionally, computing the *jump number* of a poset, which is NP-hard in general, has been shown solvable in polynomial time for bipartite posets with interval dimension two using their restricted GIG representation [TS11].

Beyond these graph classes that have been motivated by applications and algorithmic considerations, we also study several other natural intermediate graph classes. All these graph classes and properties are defined in Subsection 1.1.1.

The main contribution of this work is to establish the strict containment and incomparability relations depicted in Figure 1.1. We additionally relate these classes to incidence posets of planar and outerplanar graphs.

In Section 1.2 we use the geometric representations to establish the containment relations between the graph classes as shown in Figure 1.1. The maximal dimension of graphs in these classes is the topic of Section 1.3. In Section 1.4 we use vertex-edge incidence posets of planar graphs to separate some of these classes from each other. Specifically, we show that the vertex-edge incidence posets of planar graphs are a subclass of *stabbable* GIG (StabGIG), and that vertex-edge incidence posets of outerplanar

graphs are a subclass of *stick intersection graphs* (Stick) and *unit* GIG (UGIG). The remaining classes are separated in Section 1.5. The separating examples are listed in Table 1.1. As part of this, we show that the vertex-face incidence posets of outerplanar graphs are *segment-ray intersection graphs* (SegRay). As a corollary we obtain that they have interval dimension at most 3.

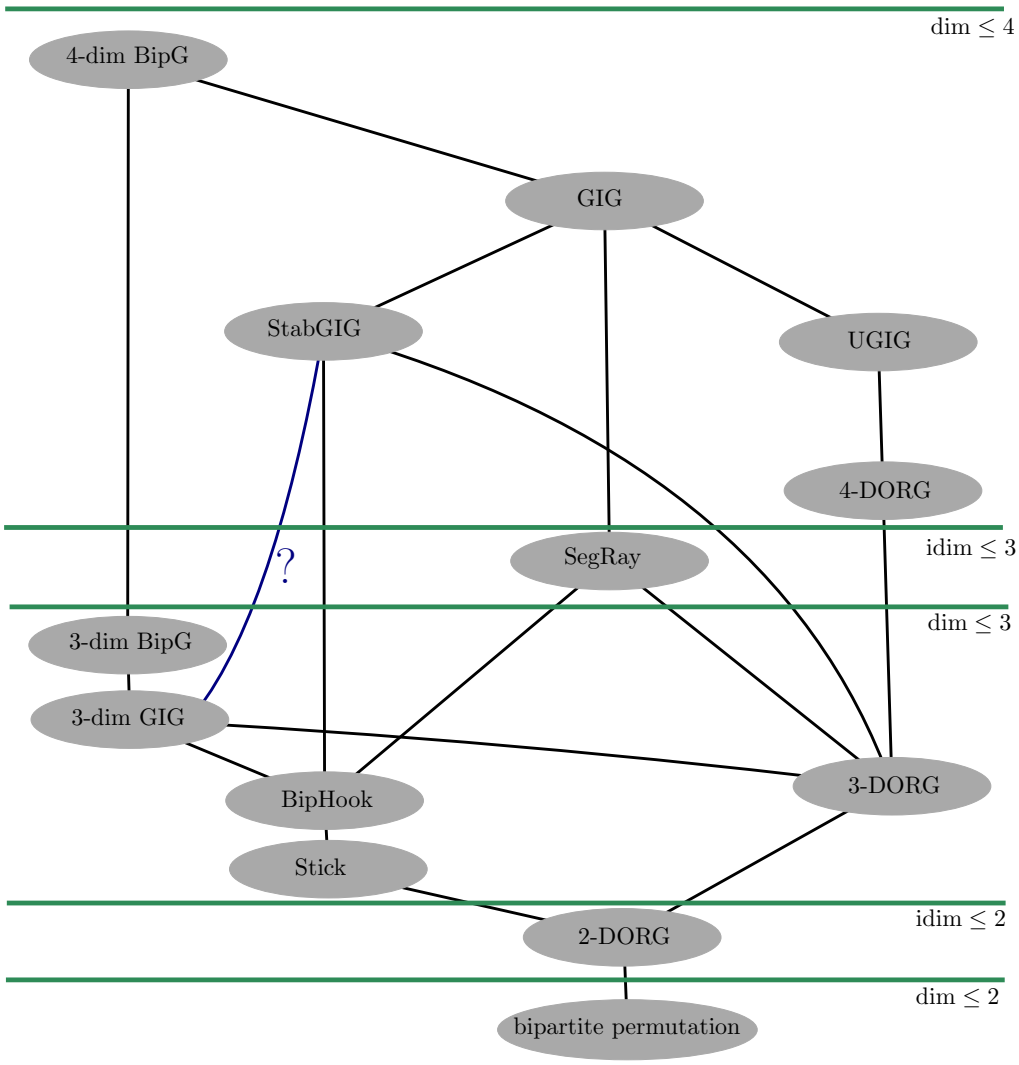

Figure 1.1.: The inclusion order of graph classes studied in this chapter.

1.1.1. Definitions of subclasses of grid intersection graphs

We introduce the graph classes from Figure 1.1. A typical drawing of a representation is shown in Figure 1.2. We denote the class of bipartite graphs by BipG. A *grid intersection*

Class I	$\not\subseteq$	Class II	Example
GIG		3-dim BipG	S_4
3-dim BipG		GIG/3-dim GIG	Proposition 1.22
3-dim GIG		SegRay	P_{K_4}, Proposition 1.15
		3-dim BipG	Proposition 1.22
StabGIG		3-dim GIG	S_4
		SegRay	S_4
SegRay		3-dim GIG	Proposition 1.20
		StabGIG	Proposition 1.30
UGIG		3-dim GIG	S_4
		StabGIG	Proposition 1.31
		4-DORG	C_{14}, see [STU10]
		SegRay	S_4, Proposition 1.9
BipHook		3-DORG	Trees
		Stick	Proposition 1.26
Stick		UGIG	Proposition 1.23
4-DORG		3-dim GIG	S_4
		StabGIG	Proposition 1.31
		SegRay	S_4
		3-DORG	S_4
3-DORG		BipHook	Proposition 1.25
2-DORG		2-dim BipG	S_3

Table 1.1.: Examples separating graph classes in Figure 1.1

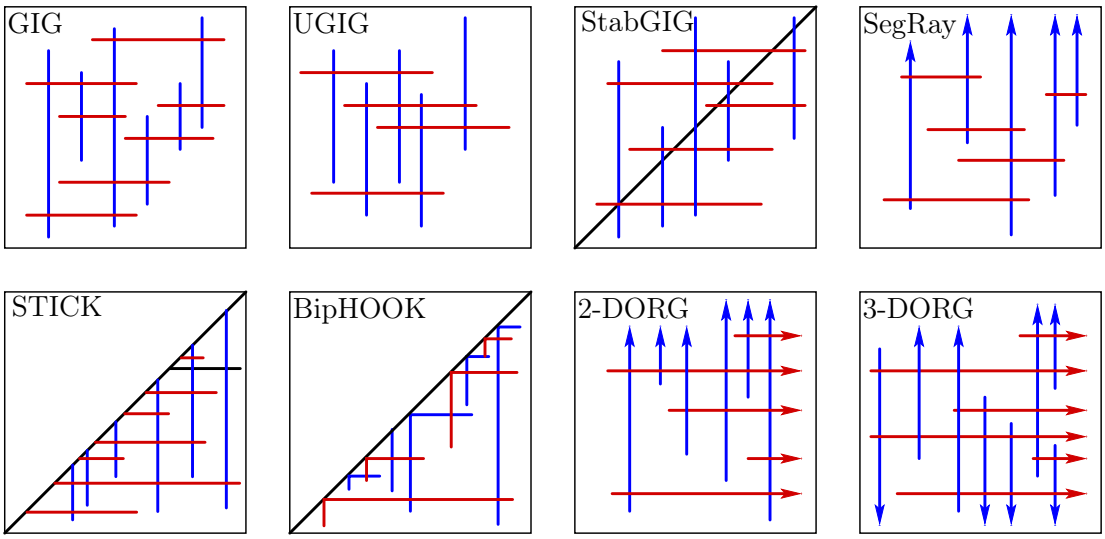

Figure 1.2.: Typical intersection representations of graphs in the classes studied in this chapter.

graph (GIG) is an intersection graph of horizontal and vertical segments in the plane where parallel segments do not intersect. If G admits a grid intersection representation such that all segments have the same length, then G is a *unit grid intersection graph* (UGIG).

A segment s in the plane is *stabbed* by a line ℓ if s and ℓ intersect. A graph G is a *stabbable grid intersection graph* (StabGIG) if it admits a grid intersection representation such that there exists a line that stabs all the segments of the representation. Stabbable representations are generally useful in algorithmic settings as they provide a linear ordering on the objects involved, see [CF13, CFS14].

A *hook* is the union of a horizontal and a vertical segment that share the left respectively top endpoint. The common endpoint, i.e., the bend of the hook, is called the *center* of the hook. A graph G is a *hook graph* if it is the intersection graph of a family of hooks whose centers are all on a line ℓ with positive slope (usually ℓ is assumed to be the line $x = y$). Hook graphs have been introduced and studied in [CCF+15, HATI11], [Hix13], and [ST15]. The graphs are called *max point-tolerance graphs* in [CCF+15] and *loss of heterozygosity graphs* in [HATI11]. Typically these graphs are not bipartite. We study the subclass of bipartite hook graphs (BipHook).

A hook graph admitting a representation where every hook is degenerate, i.e., it is a line segment, is a *stick intersection graph* (Stick). In other words, Stick graphs are the intersection graphs of horizontal or vertical segments that have their left respectively top endpoint on a line ℓ with positive slope.

Intersection graphs of rays (or half-lines) in the plane have been previously studied

in the context of their chromatic number [KN98] and the clique problem [CCL13]. We consider some natural bipartite subclasses of this class. Consider a set of axis-aligned rays in the plane. If the rays are restricted into two orthogonal directions, e.g. up and right, their intersection graph is called a *two directional orthogonal ray graph* (2-DORG). This class has been studied in [STU10] and [TS11]. Analogously, if three or four directions are allowed for the rays, we talk about 3-DORGs or 4-DORGs. The class of 4-DORGs was introduced in connection with defect tolerance schemes for nano-programmable logic arrays [STS11].

Finally, segment-ray graphs (SegRay) are the intersection graphs of horizontal segments and vertical rays directed in the same direction. SegRay graphs (and closely related graph classes) have been previously discussed in the context of covering and hitting set problems (see e.g., [KMN05, CG14, CCM13]).

In the representations defining graphs in all these classes we can assume the x and the y-coordinate of endpoints of any two different segments are distinct. This property can be established by appropriate perturbations of the segments.

For the sake of brevity we define the *dimension of a bipartite graph G* to be equal to the dimension of its comparability graph. The freedom that we have in defining the partial order on G, i.e., the choice of the color classes for minima and maxima, does not affect the dimension. This is an easy instance of the fact that dimension is a comparability invariant, i.e., independent of the choice of partial order on the comparability graph, as shown in [TMS76].

1.2. Containment relations between the classes

The diagram shown in Figure 1.1 has 19 non-transitive inclusions represented by the edges. In this section we show the inclusion between the respective classes of graphs. The inclusion 2-dim BipG $\subseteq$ 2-DORG was already noted as a consequence of Proposition 1.6. The next 8 inclusions follow directly from the definition of the classes:

$$\begin{array}{cc} \text{UGIG} \subseteq \text{GIG} & \text{StabGIG} \subseteq \text{GIG} \\ \text{2-DORG} \subseteq \text{3-DORG} & \text{3-DORG} \subseteq \text{4-DORG} \\ \text{3-dim GIG} \subseteq \text{3-dim BipG} & \text{3-dim BipG} \subseteq \text{4-dim BipG} \\ \text{Stick} \subseteq \text{BipHook} & \text{3-dim GIG} \subseteq \text{GIG.} \end{array}$$

The following less trivial inclusions follow from geometric modifications of the representation. The proofs are given in the following two propositions.

$$\text{BipHook} \subseteq \text{StabGIG} \qquad \text{2-DORG} \subseteq \text{Stick.}$$

Proposition 1.1. *Each bipartite hook graph is a stabbable GIG.*

Proof. Let $G = (A, B; E)$ be a bipartite hook graph and fix a hook representation of G in which vertices of A and B are represented by blue and red hooks, respectively. We reflect the horizontal part of each blue hook (dotted in Figure 1.3) and the vertical part of each red hook (red dotted) at the diagonal. We claim that this results in a StabGIG representation of the same graph. The edges are preserved by the operation, since each intersection is witnessed by a vertical and a horizontal segment, and either both segments are reflected or none of them. On the other hand, the transformation is an invertible linear transformation on a subset of the segments from the region below the line to the one above, hence no new intersection is introduced. The stabbability of the GIG representation comes for free. □

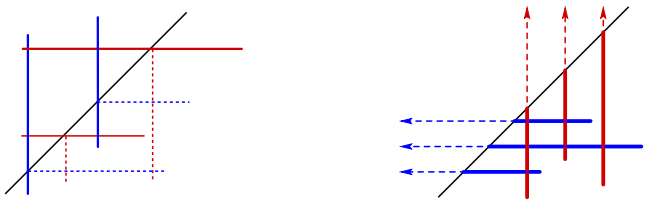

Figure 1.3.: From a BipHook to a StabGIG and from a 2-DORG to a Stick representation.

Proposition 1.2. *Each 2-DORG is a Stick graph.*

Proof. Given a 2-DORG representation of G with upward and leftward rays, we let ℓ be a line with slope 1 that is placed above all intersection points and endpoints of rays. Removing the parts of the rays that lie in the halfplane above ℓ leaves a Stick representation of G, see Figure 1.3.

Pruning of rays also yields the following three inclusions:

$$\text{3-DORG} \subseteq \text{SegRay} \qquad \text{SegRay} \subseteq \text{GIG} \qquad \text{4-DORG} \subseteq \text{UGIG}.$$

For the last one, consider a 4-DORG representation and a square of size D that contains all intersections and endpoints of the rays. Cutting each ray to a segment of length D leads to a UGIG representation of the same graph. This was already observed in [STU10].

Conversely, extending the vertical segments of a Stick representation to vertical upward rays yields:

$$\text{Stick} \subseteq \text{SegRay}.$$

To show that every 3-DORG is a StabGIG, we use a simple geometric argument as depicted in Figure 1.4 and formalized in the following proposition.

$$\text{3-DORG} \subseteq \text{StabGIG}.$$

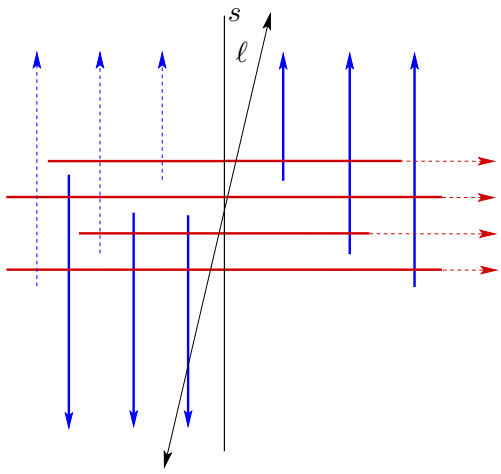

Figure 1.4.: From a 3-DORG to a StabGIG representation

Proposition 1.3. *Each 3-DORG is a stabbable GIG.*

Proof. Consider a 3-DORG representation of a graph G. We assume that vertical rays point up or down while horizontal rays point right. Let s be a vertical line to the right of all the intersections. We prune the horizontal rays at s to make them segments and then reflect the segments at s, this doubles the length of the segments (see Figure 1.4). Now take all upward rays and move them to the right via a reflection at s. This results in an intersection representation with vertical rays in both directions and horizontal segments such that all rays pointing down are left of s and all rays pointing up are to the right of s. Due to this property we find a line ℓ of positive slope that stabs all the rays and segments of the representation. Pruning the rays above, respectively below their intersection with ℓ yields a StabGIG representation of G. □

A non-geometric modification of a representation gives the 16th of the 19 non-transitive inclusions from Figure 1.1:

$$\text{BipHook} \subseteq \text{SegRay}.$$

Proposition 1.4. *Each bipartite hook graph is a SegRay graph.*

Proof. Consider a BipHook representation of $G = (A, B; E)$. We construct a SegRay representation where A is represented by vertical rays and B by horizontal segments. Let $a_1, \ldots, a_{|A|}$ be the order of the vertices of A that we get by the centers of the hooks on the diagonal, read from bottom-left to top-right. The y-coordinates of the horizontal segments and the endpoints of the rays in our SegRay representation of G will be given in the following way.

We initialize a list $R = [a_1, \ldots, a_{|A|}]$ and a set $S = B$ of *active* vertices, and an empty list Y. We apply one of the following steps repeatedly:

1. If there is an active $a \in R$ such that $N(a) \cap S = \emptyset$, then remove a from R and append it to Y.

2. If there is an active $b \in S$ such that vertices of $N(b)$ appear consecutively in R, then remove b from S and append it to Y. □

Suppose that R and S are empty after the iteration. Then we can construct a SegRay representation of G. The endpoint of the ray representing a_i receives i as the x-coordinate and the position of a_i in Y as the y-coordinate. The segment representing $b \in B$ also obtains the y-coordinate according to its position in Y. Its x-coordinates are determined by its neighbourhood. Now it is straight-forward to verify that this defines a SegRay representation of G. It remains to show that one of the steps can always

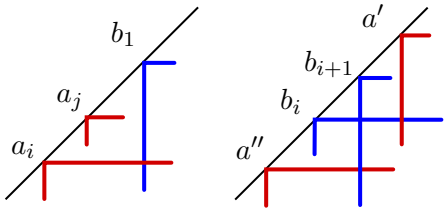

Figure 1.5.: Left: b_1 is not involved in a forward pair. Right: A forbidden configuration for a bipartite hook graph.

be applied if R and S are nonempty. Suppose that none of the steps can be applied. Then, for each $b \in S$ there are active vertices $a_i, a_k \in R \cap N(b)$ and $a_j \in R \setminus N(b)$ with $i < j < k$. We call (a_j, b) an *interesting pair*. If the center of the hook a_j lies before the center of b then we call the interesting pair *forward*, and *backward* otherwise. Let $b_1, \ldots, b_{|S|}$ be the order of centers of active vertices of S. If b_1 is involved in a forward interesting pair, then each certifying a_j is locked in the triangle between b_1 and a_i and thus has no neighbour in S (see Figure 1.5), and so step 1 could be applied. Hence every interesting pair involving b_1 is a backward pair. Symmetrically, every interesting pair involving $b_{|s|}$ is a forward pair. We conclude that there are active vertices b_i, b_{i+1}, such that b_i is involved in a forward interesting pair, and b_{i+1} in a backward one.

Let $a' \in N(b_i)$ be the hook corresponding to an active vertex that encloses $a \notin N(b_i)$, i.e., $b_i < a < a'$ on the diagonal. Since a is active its hook intersects some b_j with $i + 1 \leq j$. Therefore, $b_i < b_{i+1} < a'$ on the diagonal. By symmetric reasoning we also find $a'' \in N(b_{i+1})$ such that on the diagonal $a'' < b_i < b_{i+1} < a'$ and $a''b_{i+1} \in E$. The order on the diagonal and the existence of edges $a''b_{i+1}$ and $b_i a'$ implies that the hooks of b_i and b_{i+1} intersect (see Figure 1.5). This contradicts that b_i and b_{i+1} belong to the same color class of the bipartite graph.

1.3. Dimension

From the 19 inclusion relations between classes that have been mentioned at the beginning of the previous section we have shown 16. The remaining three inclusions will be shown by using order dimension in this section. Specifically, we bound the maximal dimension of the graphs in the relevant classes. First, we will show that the dimension of GIGs is bounded. It has previously been observed that $\mathrm{idim}(G) \leq 4$ when G is a GIG [CHO$^+$14]. As already shown in [Fel14] this can be strengthened to $\dim(G) \leq 4$.

We define four linear extensions of G as depicted in Figure 1.6. In each of the directions left, right, top and bottom we consider the orthogonal projection of the segments onto a directed horizontal or vertical line. In each such projection every segment corresponds to one interval (or point) per line. We choose a point from each interval on the line by the following rule. For minimal elements we take the minimal point in the interval in the direction of the line, for maximal elements we choose the maximal one. We denote those total orders according to the direction of their oriented line by $L_\leftarrow, L_\rightarrow, L_\uparrow, L_\downarrow$.

Proposition 1.5. *For every GIG G, $\{L_\leftarrow, L_\rightarrow, L_\uparrow, L_\downarrow\}$ is a realizer of G, and hence* $\dim(G) \leq 4$.

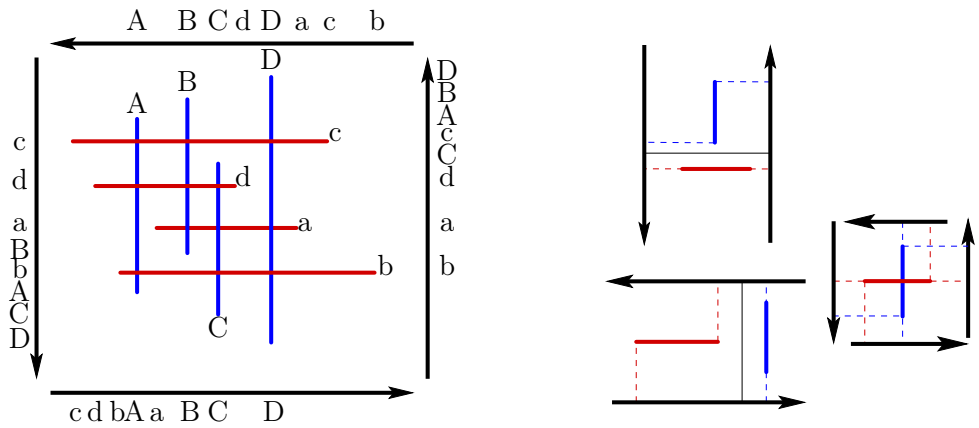

Figure 1.6.: A realizer of a GIG and an illustration of the correctness.

Proof. For two intersecting segments the minimum always lies before the maximum, see Figure 1.6. It remains to check that every incomparable pair (s_1, s_2) is reversed in the realizer. Every disjoint pair of segments is separated by a horizontal or vertical line. The separated vertices appear in different order in the two directions orthogonal to this line, see Figure 1.6.

It is known that a bipartite graph is a bipartite permutation graph if and only if the dimension of the poset is at most 2. Thus, by Proposition 1.5, the maximal dimension of the graphs in the other classes that we consider must be 3 or 4.

The class of 2-DORGs is characterized by the interval dimension of the graph.

Proposition 1.6. *2-DORGs are exactly the bipartite graphs of interval dimension 2.*

This has been shown in [STU10] using a characterization of 2-DORGs as the complement of co-bipartite circular arc graphs. Below we give a simple direct proof.

To explain it we recall the geometric version of interval dimension: Vectors $a, b \in \mathbb{R}^d$ with $a \leq_{\text{dom}} b$ define a *standard box* $[a, b] = \{v : a \leq_{\text{dom}} v \leq_{\text{dom}} b\}$ in $\mathbb{R}^d$. Let $P = (X, \leq_P)$ be a poset. A family of standard boxes $\{[a_x, b_x] \subseteq \mathbb{R}^d : x \in X\}$ is a *box representation* of P in $\mathbb{R}^d$ if it holds that $x <_P y$ if and only if $b_x \leq_{\text{dom}} a_y$. Then the interval dimension of P is the minimum d for which there is a box representation of P in $\mathbb{R}^d$. Note that if P has height 2 with $A = \text{Min}(P)$ and $B = \text{Max}(P)$, then in a box representation the lower corner a_x for each $x \in A$ and the upper corner b_y for each $y \in B$ are irrelevant, in the sense that they can uniformly be chosen as $(-c, \ldots, -c)$ respectively $(c, \ldots, c)$ for a large enough constant c.

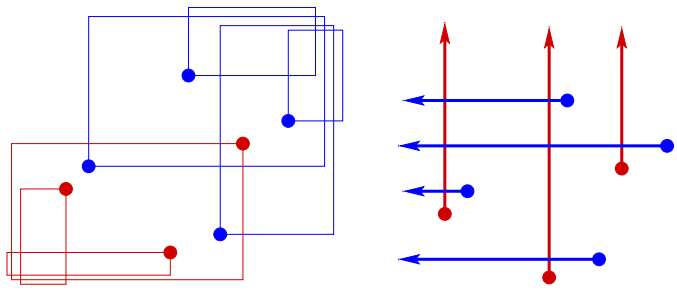

Figure 1.7.: A representation with boxes showing that $\text{idim}(G) = 2$ and the corresponding 2-DORG representation.

Proof of Proposition 1.6. Let $G = (A, B; E)$ be a bipartite graph and suppose $\text{idim}(G) = 2$. Consider a box representation $\{[a_x, b_x]\}$ of G in $\mathbb{R}^2$. Clearly, for each $x \in A$, $y \in B$ we have $xy \in E$ if and only if $b_x \leq_{\text{dom}} a_y$. To obtain a 2-DORG representation of G, draw upward rays starting from upper corners of boxes representing A, and leftward rays starting from lower corners of boxes representing B (see Figure 1.7). Then for each $x \in A$, $y \in B$ we have $b_x \leq_{\text{dom}} a_y$ if and only if the rays of x and y intersect.

Now, the converse direction is immediate together with the observation from the previous paragraph.

Since the 2-DORGs are exactly the bipartite graphs of interval dimension 2, and interval dimension is bounded by dimension, we obtain

$$\text{2-dim BipG} \subseteq \text{2-DORG}.$$

In the following we show that BipHooks and 3-DORGs have dimension at most 3. For these results we note that graphs with a special SegRay representation have dimension at most 3 and that the interval dimension of SegRay graphs is bounded by 3. This latter result was shown previously by a different argument in [CHO$^+$14].

Lemma 1.7. *For a SegRay graph G,* $\dim(G) \leq 3$ *when G has a SegRay representation satisfying the following: whenever two horizontal segments are such that the x-projection of one is included in the other one, then the smaller segment lies below the bigger one.*

Proof. Consider such a SegRay representation of G with horizontal segments as maximal and downward rays as minimal elements of Q_G. The linear extensions $L_\rightarrow$, $L_\leftarrow$, and $L_\downarrow$ defined for Proposition 1.5 form a realizer of Q_G. □

Corollary 1.8. *For every 3-DORG G,* $\dim(G) \leq 3$.

Proof. Consider a 3-DORG representation of G where the horizontal rays use two directions. We cut the horizontal rays so that they have the same length D. When D is large enough, this yields a SegRay representation of the same graph. Note that such a representation has no nested segments. Thus, Lemma 1 implies $dim(Q_G) \leq 3$. □

Proposition 1.9. *For every SegRay graph G,* $\mathrm{idim}(G) \leq 3$.

Proof. Suppose that the rays correspond to minimal elements of Q_G. By Lemma 1.7 the linear extensions $L_\rightarrow$, $L_\leftarrow$ and $L_\downarrow$ reverse all incomparable pairs except some that consist of two maximal elements. We convert these linear extensions to interval orders and extend the intervals (originally points) of maximal elements in $L_\rightarrow$ far to the right to make them intersect. In this way we obtain three interval orders whose intersection gives rise to Q_G. □

Proposition 1.10. *For every bipartite hook graph G,* $\dim(G) \leq 3$.

Proof. Let A and B be the color classes of G. We construct the graph G' by adding private neighbours to vertices of B. Then G' is also a BipHook graph as we can easily add hooks intersecting a single hook in a representation of G. By Proposition 1.4 we know that G' has a SegRay representation R with downward rays representing A. By construction, each horizontal segment in R must have its private ray intersecting it. Thus, R satisfies the property of Lemma 1.7 and $\dim(Q_{G'}) \leq 3$. Since Q_G is an induced subposet of $Q_{G'}$ we conclude $\dim(Q_G) \leq 3$.

Since Stick $\subset$ BipHook we know that $\dim(Q_G) \leq 3$ if G is a Stick graph. However, a nicer realizer for a Stick graph is obtained by Proposition 1.5, since $L_{\leftarrow}$ and $L_{\downarrow}$ coincide in a Stick representation.

In Section 1.5 we will show that these bounds are tight.

1.4. Vertex-edge incidence posets

We proceed by investigating the relations between the classes of GIGs and incidence posets of graphs.

For a graph G, P_G denotes the vertex-edge incidence poset of G, and the comparability graph of P_G is the graph obtained by subdividing each edge of G once. The vertex-edge incidence posets of dimension 3 are characterised by Schnyder's Theorem.

Theorem 1.11 ([Sch89]). *A graph G is planar if and only if* $\dim(P_G) \leq 3$.

Even though some GIGs have poset dimension 4, we will see that the vertex-edge incidence posets with a GIG representation are precisely the vertex-edge incidence posets of planar graphs. A weak bar visibility representation of G gives a GIG representation of P_G. On the other hand, a GIG representation of P_G can be transformed into a weak bar visibility representation of G. In particular, since the segments representing edges of G intersect two segments representing incident vertices, they can be shortened until their intersections become contacts. Hence P_G is a GIG if and only if G is planar. We next show the stronger result, that there is a StabGIG representation of P_G for every planar graph G.

Proposition 1.12. *A graph G is planar if and only if P_G is a stabbable GIG.*

We use the following definitions. A *generic floorplan* is a partition of a rectangle into a finite set of interiorly disjoint rectangles that have no point where four rectangles meet. Two floorplans are *weakly equivalent* if there exist bijections Φ_H between the horizontal segments and Φ_V between the vertical segments, such that a segment s has an endpoint on t in F if and only if $\Phi(s)$ has an endpoint on $\Phi(t)$. A floorplan F *covers* a set of points P if and only if every segment contains exactly one point of P and no point is contained in two segments. The following theorem has been conjectured by Ackerman, Barequet and Pinter [ABP06], who have also shown it for the special case of *separable* permutations. It has been shown by Felsner [Fel13] for general permutations.

Theorem 1.13 ([Fel13]). *Let P be a set of n points in the plane, such that no two points have the same x- or y-coordinate and F a generic floorplan with n segments. Then there exists a floorplan F', such that F and F' are weakly equivalent and F' covers P.*

Proof of Proposition 1.12. Consider a weak bar-visibility representation of G. The lowest and highest horizontal segments h_b and h_t can be extended, such that their left as well as their right endpoints can be connected by new vertical segments v_l and v_r. The segments h_b, h_t, v_l and v_r are the boundary of a rectangle. Extending every horizontal segment until its left and right endpoints touch vertical segments leads to a floorplan F. By Theorem 1.13 there exists an equivalent floorplan F' that covers a pointset P consisting of n points on the diagonal of the the big rectangle with positive slope. Shortening the horizontal segments and extending the vertical segments of F' by $\epsilon > 0$ on each end leads to a GIG representation of P_G that can be stabbed by the line through the diagonal.

On the other hand, every GIG representation of P_G leads to a weak bar-visibility representation, and hence G is planar.

We will now show that P_G is in the classes of Stick and bipartite hook graphs if and only if G is outerplanar.

Proposition 1.14. *P_G is a Stick graph if and only if G is outerplanar.*

Proof. Outerplanar graphs have been characterized by linear orderings of their vertices by Felsner and Trotter [FT05]: A graph $G = (V, E)$ is outerplanar if and only if there exist linear orders L_1, L_2, L_3 of the vertices with $L_2 = \overleftarrow{L_1}$, i.e., L_2 is the reverse of L_1, such that for each edge $vw \in E$ and each vertex $u \notin \{v, w\}$ there is $i \in \{1, 2, 3\}$, such that $u > v$ and $u > w$ in L_i.

Consider a Stick representation of P_G where the elements of V correspond to vertical sticks. Restricting the linear extensions $L_1 = L_{\leftarrow}$, $L_2 = L_{\rightarrow}$, and $L_3 = L_{\uparrow}$ (cf. the proof of Proposition 1.5) obtained from a Stick representation of P_G to the elements of V yields linear orders satisfying the property above. Thus G is outerplanar.

For the backward direction let G be an outerplanar graph. In [CCF+15] it is shown that the class of *hook contact* graphs (each intersection of hooks is also an endpoint of a hook) is exactly the class of outerplanar graphs. Given a hook contact representation of G we construct a Stick representation of P_G. To this end we consider each hook as two sticks, a vertical one for the vertices and a horizontal one as a placeholder for the edges. For each contact of the horizontal part of a hook v we place an additional horizontal stick slightly below the center of v. The k-th contact of a hook with the horizontal part is realized by the k-th highest edge that is added in the placeholder as shown in Figure 1.8.

The construction used in the previous proposition directly produces a weak semi-bar visibility representation of an outerplanar graph. Just extend all vertical segments upwards until they hit a common horizontal line ℓ and reflect the plane at ℓ, now ℓ can play the role of the x-axis for the weak semi-bar visibility representation.

Figure 1.8.: A hook contact representation of G transformed into a Stick representation of P_G.

Proposition 1.15. *A graph G is outerplanar if and only if the graph P_G has a SegRay representation where the vertices of G are represented as rays.*

Proof. Cutting the rays of a SegRay representation with rays pointing downwards somewhere below all horizontal segments leads to a weak semi-bar visibility representation of G and vice versa. Thus, Lemma 0.4 gives the result.

Proposition 1.16. *A graph G is outerplanar if and only if the graph P_G has a hook representation.*

Proof. If G is outerplanar then P_G has a hook representation by Proposition 1.14. On the other hand, assume that P_G has a hook representation for a graph G. According to Proposition 1.4 we construct a SegRay representation with vertices as rays and edges as segments. This representation shows that G is outerplanar by Proposition 1.15. □

Proposition 1.17. *If G is outerplanar, then the graph P_G has a SegRay representation where the vertices of G are represented as segments.*

Proof. Consider a hook representation R' of P_G. According to the proof Proposition 1.4 we can transform P_G into a SegRay representation with a free choice of the colorclass that is represented by rays. Choosing the subdivision vertices as rays leads to the required representation. □

In contrast to Proposition 1.15, the backward direction of Proposition 1.17 does not hold: Figure 1.9 shows a SegRay representation of $P_{K_{2,3}}$ with vertices being represented as horizontal segments, but $K_{2,3}$ is not outerplanar. Together with Proposition 1.15 this also shows that the class of SegRay graphs is not symmetric in its color classes.

In the following we construct a UGIG representation of P_G for an outerplanar graph G.

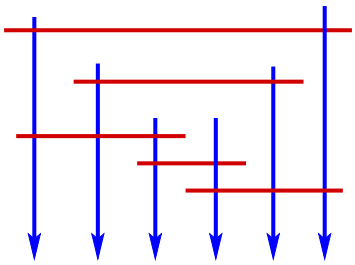

Figure 1.9.: A SegRay representation of $P_{K_{2,3}}$.

Proposition 1.18. *If G is outerplanar then P_G is a UGIG.*

Proof. We construct a UGIG representation of P_G for a maximal outerplanar graph $G = (V, E)$ with outer-face cycle $v_0, \ldots, v_n$. The vertices of V are drawn as vertical segments. Starting from v_0 we iteratively draw the vertices of breadth-first-search layers (BFS-layers). Each BFS-layer has a natural order inherited from the order on the outer-face, i.e., the increasing order of indices. When the i-th layer L_i has been drawn the following invariants hold:

1. Segments for all vertices and edges of $G[L_0, \ldots, L_{i-1}]$, all vertices of L_i, and all edges connecting vertices of L_{i-1} to vertices of L_i have been placed.

2. The upper endpoints of the segments representing vertices in L_i lie on a strict monotonically decreasing curve C_i. Their order on C_i agrees with the order of the corresponding vertices in L_i. Their x-coordinates differ by at most one.

3. No segment intersects the region above C_i.

We start the construction with the vertical segment corresponding to v_0. The curve C_0 is chosen as a line with negative slope that intersects the upper endpoint of v_0.

We start the $(i+1)$-th step by adding segments for the edges within vertices of layer L_i. Afterwards we add the segments for edges between vertices in layer L_i and L_{i+1} and the segments for the vertices of layer L_{i+1}. The construction is indicated in Figure 1.10.

First we draw unit segments for the edges within layer L_i. Since the graph is outerplanar such edges only occur between consecutive vertices of the layer. For a vertex v_k of L_i which is not the first vertex of L_i we define a horizontal ray r_k whose start is on the segment of the predecessor of v_k on this layer such that the only additional intersection of r_k is with the segment of v_k. The initial unit segment of ray r_k can be used for the edge between v_k and its predecessor.

All segments that will represent edges between layer L_i and L_{i+1} are placed as horizontal segments that intersect the segment of the incident vertex $v_k \in L_i$ above the

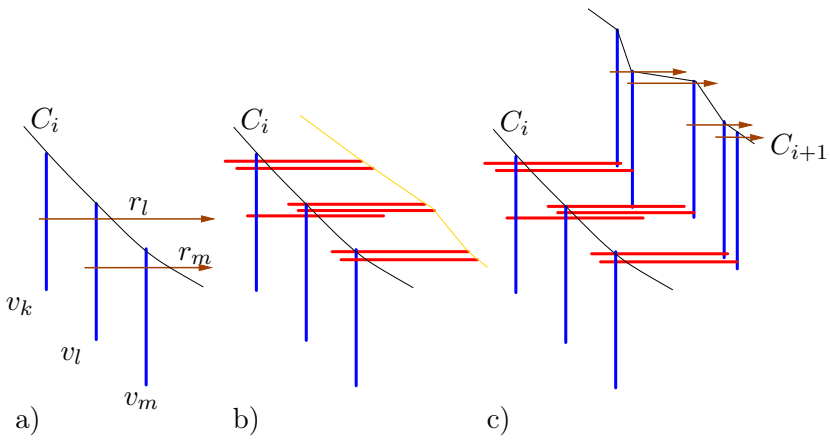

Figure 1.10.: One step in the construction of a UGIG representation of P_G: a)The situation before the step. b) The edges between layer L_i and L_{i+1} and within layer L_i are added. c) The vertices of layer L_{i+1} are added.

ray r_k. We draw these edge-segments such that the endpoints lie on a monotonically decreasing curve C and the order of these endpoints on C corresponds to the order of their incident vertices in L_{i+1}.

Now the right endpoints of the edges between the two layers lie on the monotone curve C and no segment intersects the region above this curve. Due to properties of the BFS for outerplanar graphs, each vertex of layer L_{i+1} is incident to one or two edges whose segments end on C and if there are two then they are consecutive on C. We place the unit segments of vertices of L_{i+1}, such that their lower endpoint is on the lower segment of an incident edge with the x-coordinate such that they realize the required intersections.

With this construction the invariants are satisfied. □

There are graphs G where P_G is a UGIG and G is not outerplanar, for example $G = K_{2,3}$ as shown in Figure 1.11. On the other hand there exist planar graphs G, such that P_G is not a UGIG as the following proposition shows.

Proposition 1.19. *P_{K_4} is not a UGIG.*

Proof. Suppose to the contrary that P_{K_4} has a UGIG representation with vertices as vertical segments. By contracting vertical segments to points one can obtain a planar embedding of K_4 from such a representation. As K_4 is not outerplanar, there is a vertex v that is not incident to the outer face in this embedding. For the initial UGIG representation this means that v is represented by a vertical segment which is enclosed by segments representing vertices and edges of $K_4 - \{v\}$. Notice that these segments represent a 6-cycle of P_{K_4}. However, the largest vertical distance between any pair of

horizontal segments in this cycle is less than 1. Thus, there is not enough space for the vertical segment of v, contradiction. □

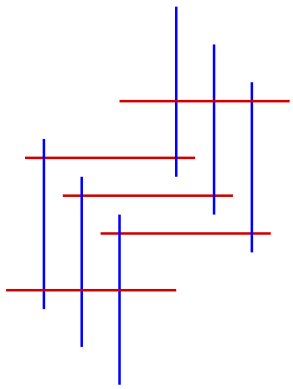

Figure 1.11.: A UGIG representation of $P_{K_{2,3}}$.

1.5. Separating examples

In this section we will give examples of graphs that separate the graph classes in Figure 1.1. For this purpose we will show that the classes we have observed to be at most 4-dimensional indeed contain 4-dimensional graphs. This is done in Subsection 1.5.1 using standard examples and vertex-face incidence posets of outerplanar graphs. The remaining graph classes will be separated using explicit constructions in Subsection 1.5.2 and Subsection 1.5.3.

Using the observations of Section 1.4 about vertex-edge incidence posets we can immediately separate the following graph classes.

$$\text{StabGIG} \not\subset \text{BipHook} \qquad \text{StabGIG} \not\subset \text{3-DORG}$$
$$\text{SegRay} \not\subset \text{4-DORG} \qquad \text{Stick} \not\subset \text{2-DORG}.$$

In [STU10] it is shown that the graph C_{14} (cycle on 14 vertices) is not a 4-DORG, and in particular is not a 3- or 2-DORG. In other words, P_{C_7} is not a 4-DORG. Since C_7 is outerplanar, by the propositions of the previous section we know that P_{C_7} is a SegRay, a StabGIG and a Stick graph. This shows the three seperations involving DORGs. For the first one let G be a planar graph that is not outerplanar. Then P_G is a StabGIG (Proposition 1.12) but not a BipHook graph (Proposition 1.16).

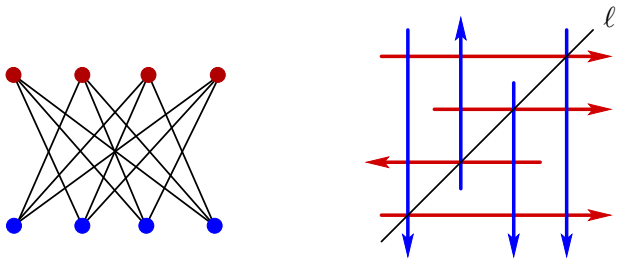

Figure 1.12.: The poset S_4 and a stabbable 4-DORG representation of it.

1.5.1. 4-dimensional graphs

First of all, some graph classes are already separated by their maximal dimension. The *standard example* S_n of an n-dimensional poset, cf. [Tro92], is the poset on n minimal elements $a_1, \ldots, a_n$ and n maximal elements $b_1, \ldots, b_n$, such that $a_i < b_j$ in S_n if and only if $i \neq j$. To separate most of the 4-dimensional classes from the 3-dimensional ones, the standard example S_4 is sufficient. As shown in Figure 1.12 it has as a stabbable 4-DORG representation. From this it follows that:

$$\text{StabGIG} \not\subset \text{BipHook} \qquad \text{StabGIG} \not\subset \text{3-DORG}$$
$$\text{4-DORG} \not\subset \text{3-DORG} \qquad \text{StabGIG} \not\subset \text{3-dim GIG}$$

Since the interval dimension of S_n is n we get the following relations from Proposition 1.9.

$$\text{StabGIG} \not\subset \text{SegRay} \qquad \text{4-DORG} \not\subset \text{SegRay}$$

We will now show that the vertex-face incidence poset of an outerplanar graph has a SegRay representation. In [FN11] it has been shown that there are outerplanar maps with a vertex-face incidence poset of dimension 4. Together with Proposition 1.20 below this shows that there are SegRay graphs of dimension 4. We obtain

$$\text{SegRay} \not\subset \text{3-dim GIG}.$$

Proposition 1.20. *If G is an outerplanar map then the vertex-face incidence poset of G is a SegRay graph.*

Proof. Let G be a graph with a fixed outerplanar embedding. First we argue that we may assume that G is 2-connected. If G is not connected then we can add a single edge between two components without changing the vertex-face poset. Now consider adding an edge between two neighbours of a cut vertex on the outer face cycle, i.e., two vertices of distance 2 on this cycle. This adds a new face to the vertex-face-poset, but keeps

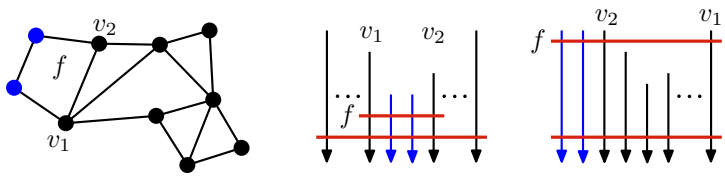

Figure 1.13.: Illustration for the induction step in Proposition 1.20

the old vertex-face-poset as an induced subposet. Therefore, we may assume that G is 2-connected.

By induction on the number of bounded faces we show that G has a SegRay representation in which the cyclic order of the vertices on the outer face agrees with the left-right order (cyclically) of rays representing these vertices. If G has one bounded face then the claim is straight-forward. If G has more bounded faces then consider the dual graph of G without the outer face, which is a tree. Let f be a face that corresponds to a leaf of that tree. Define G' to be the plane graph obtained by removing f and incident degree-2 vertices from G. Then exactly two vertices v_1, v_2 of f are still in G', and they are adjacent via an edge at the outer face of G'. Note that G' is 2-connected. Applying induction on G' we obtain a SegRay representation in which the two rays representing v_1 and v_2 are either consecutive, or left- and rightmost ray.

In the first case we insert rays for the removed vertices between v_1 and v_2 with endpoints being below all other horizontal segments. Then a segment representing f can easily be added to obtain a SegRay representation with the required properties of G, see the middle of Figure 1.13.

If the rays of v_1 and v_2 are the left- and rightmost ones, then observe that the endpoints of both rays can be extended upwards to be above all other endpoints. We can insert the new rays to the left of all the other rays and the segment for f as indicated in Figure 1.13 on the right. This concludes the proof.

Propositions 1.20 and 1.9 also give the following interesting result about vertex-face incidence posets of outerplanar maps which complements the fact that they can have dimension 4 [FN11].

Corollary 1.21. *The interval dimension of a vertex-face incidence poset of an outerplanar map is bounded by* 3.

We have separated all the graph classes which involve dimension except for the two classes of 3-dimensional GIGs and stabbable GIGs. As indicated in Figure 1.1 it remains open whether 3-dim GIG is a subclass of StabGIG or not. More comments on this can be found at the end of Subsection 1.5.3.

1.5.2. Constructions

In this subsection we give explicit constructions for the remaining separations of classes not involving StabGIG.

In the introduction we mentioned that every 2-dimensional order of height 2, i.e., every bipartite permutation graph, is a GIG. We show now that this does not hold for 3-dimensional orders of height 2.

Proposition 1.22. *There is a 3-dimensional bipartite graph that is not a GIG.*

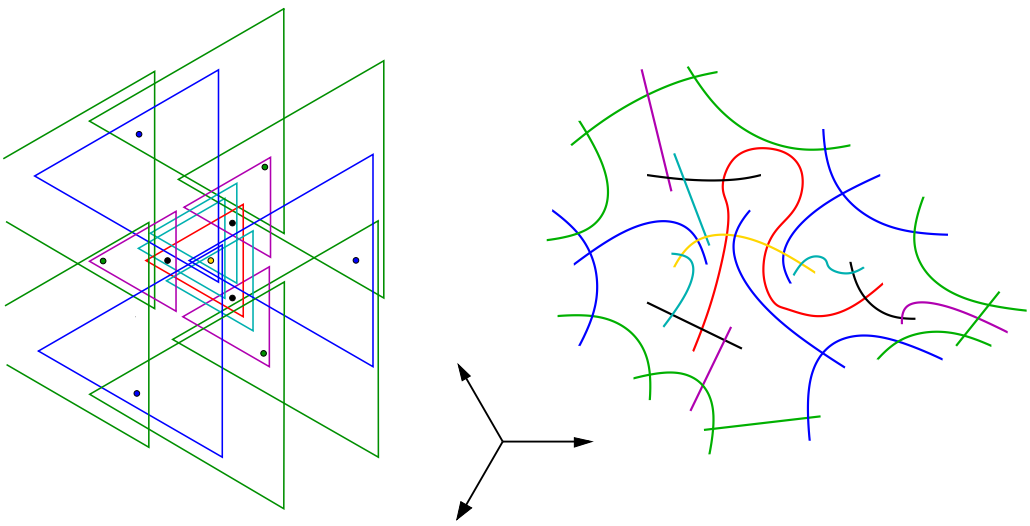

Figure 1.14.: The drawing on the left defines an inclusion order of homothetic triangles. This height-2 order does not have a pseudosegment representation.

Proof. The left drawing in Figure 1.14 defines a poset P by ordering the homothetic triangles by inclusion. Some of the triangles are so small that we refer to them as points from now on. Each inclusion in P is witnessed by a point and a triangle, and hence P has height 2. To see that it is 3-dimensional we use the drawing and the three directions depicted in Figure 1.14. By applying the same method as we did for Proposition 1.5 we obtain three linear extensions forming a realizer of P.

We claim that P is not a pseudosegment intersection graph , and hence not a GIG. Suppose to the contrary that it has a pseudosegment representation. The six green triangles together with the three green and the three blue points form a cycle of length 12 in G. Hence, the union of the corresponding pseudosegments in the representation contains a closed curve in $\mathbb{R}^2$. Without loss of generality assume that the pseudosegment representing the yellow point lies inside this closed curve (we may change the outer face using a stereographic projection). The pseudosegments of the three large blue triangles

intersect the yellow pseudosegment and one blue pseudosegment (corresponding to a blue point) each. The yellow and the blue pseudosegments divide the interior of the closed curve into three regions. We show that each of these regions contains one of the pseudosegments representing black points.

Each purple pseudosegment intersects the cycle in a point that is incident to one of the three bounded regions. Now, each black pseudosegment intersects a purple one. If such an intersection lies in the unbounded region, then the whole black pseudosegment is contained in this region. This is not possible as for each of the black pseudosegments there is a blue pseudosegment representing a small blue triangle that connects it to the enclosed yellow pseudosegment without intersecting the cycle. Thus, the three intersections of purple and black pseudosegments have to occur in the bounded regions, and in each of them one. It follows that each of the three bounded regions contains one black pseudosegment.

Now, the red pseudosegment intersects each of the three black pseudosegments. Since they lie in three different regions whose boundary it may only traverse through the yellow pseudosegment, it has to intersect the yellow pseudosegment twice. This contradicts the existence of a pseudosegment representation. □

In the following we give constructions to show that

Stick $\not\subset$ UGIG	UGIG $\not\subset$ Stick
BipHook $\not\subset$ 3-DORG	BipHook $\not\subset$ Stick

Proposition 1.23. *The Stick graph shown in Figure 1.15 is not a UGIG.*

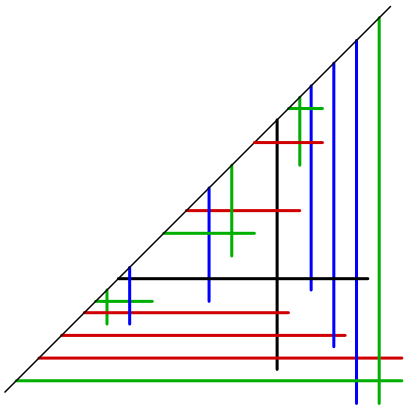

Figure 1.15.: A stick representation of a graph that is not a UGIG.

Proof. Let G be the graph represented in Figure 1.15. Let v and h be the two adjacent vertices of G that are drawn as black sticks in the figure. There are five pairs of intersecting blue vertical and red horizontal segments $v_1, h_1, \ldots, v_5, h_5$. Each v_i intersects h and each h_i intersects v. Four of the pairs v_i, h_i form a 4-cycle with a pair of green segments q_i, r_i.

Suppose that G has a UGIG representation. We claim that in any such representation the intersection points p_i of v_i and h_i form a chain in $<_{\text{dom}}$ after a suitable rotation of the representation. Note that one quadrant formed by the segments v and h (without loss of generality the upper right one) contains at least two of the p_i's by the pigeonhole principle. Assume without loss of generality that p_1 and p_2 lie in this quadrant. If p_1 and p_2 are incomparable in $<_{\text{dom}}$, then the horizontal segment h_1 of the lower intersection point has a forbidden intersection with the vertical segment v_2 of the higher one, see Figure 1.16 left.

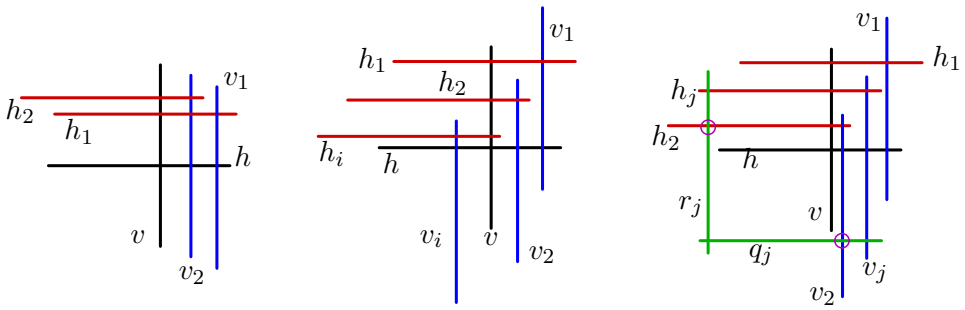

Figure 1.16.: Left: The intersection points p_1, p_2 in the upper right quadrant form a chain in $<_{\text{dom}}$. Middle: p_i does not dominate p_2 in $<_{\text{dom}}$. Right: The green segments q_j, r_j for the middle pair of segments h_j, v_j cannot be added.

So p_1 and p_2 are comparable in $<_{\text{dom}}$. We may assume that v_2, h_2 is the pair of segments whose intersection point is dominated in $<_{\text{dom}}$ by all other intersection points in the upper right quadrant. We observe that the lower endpoint of v_2 lies below the lower endpoint of v, and the left endpoint of h_2 lies to the left of the left endpoint of h as shown in the middle of Figure 1.16. It follows that if an intersection point p_i does not dominate p_2, then p_i lies below h_2 and to the left of v_2, but not in the upper right quadrant by our choice of p_2 (see Figure 1.16 for an example). It is easy to see that the remaining two intersection points p_j ($j \notin \{1, 2, i\}$) then have to dominate p_2 in $<_{\text{dom}}$, as otherwise we would see forbidden intersections among the blue and red segments.

We conclude that, in each case, four of the points $p_1, \ldots, p_5$ lie in the upper right quadrant and that they form a chain with respect to $<_{\text{dom}}$. Thus at least one pair of segments v_j, h_j with p_j being in the middle of the chain has neighbours q_j, r_j. However, as indicated in the right of Figure 1.16, q_j and r_j cannot be added without introducing

forbidden intersections. Hence G does not have a UGIG representation. □

We now show that there is a 3-DORG that is not a BipHook graph. We will use the following lemma for the argument.

Lemma 1.24. *Let G be a bipartite graph and G' be the graph obtained by adding a twin to each vertex of G (i.e., a vertex with the same neighbourhood). Then G' is a hook graph if and only if G is a Stick graph.*

Proof. Suppose that G' has a hook representation. Consider twins $v, v' \in V(G)$ and the position of their neighbours in a hook representation. Suppose that there are vertices $u, w \in N(v)$, such that the order on the diagonal is u, v, v', w. One can see that this order of centres together with edges uw and $v'u$ would force the hooks of v and v' to intersect, which contradicts their non-adjacency. Thus either no neighbour of v occurs before v or no neighbour of v' occurs after v' on the diagonal. This shows that the hook of v or v' can be drawn as a stick, and it follows that G has a Stick representation.

Conversely, in a stick representation of G twins can easily be added to obtain a stick representation of G'. □

Proposition 1.25. *The* 3-DORG *in Figure 1.17 is not a Stick graph.*

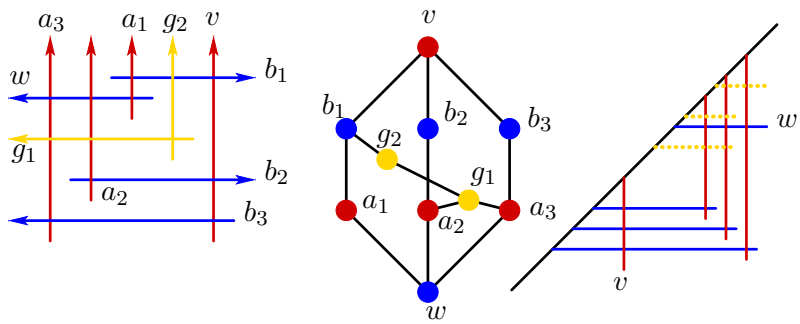

Figure 1.17.: A 3-DORG that is not a Stick graph.

Proof. Suppose to the contrary that a Stick representation of the graph exists. We may assume that v is a vertical and w a horizontal stick. Observe that w has to lie above v on the diagonal: Otherwise, two of the a_i's have to lie either before v or after v, however, for the outer one of such a pair of a_i's it is impossible to place a stick for b_i that also intersects v. Hence, the Stick representation of v, w and the a_i's and b_i's have to look as in Figure 1.17. By checking all possible positions of g_1, i.e., permutations of $\{a_1, a_2, a_3\}$ and the correspondingly forced permutation of $\{b_1, b_2, b_3\}$ in the representation, it can be verified that the representation cannot be extended to a representation of the whole graph. The cases are indicated in Figure 1.17. □

As a consequence, there is a 3-DORG that is not a bipartite hook graph. Indeed, if we add a twin to each vertex of the graph shown in Figure 1.17 then the obtained graph is still a 3-DORG. It can not be a BipHook graph as otherwise by Lemma 1.24 we would conclude that the graph in Figure 1.17 is a Stick graph.

We next show a construction of a bipartite hook graph that is not a Stick graph. A related construction was also presented in [Hix13].

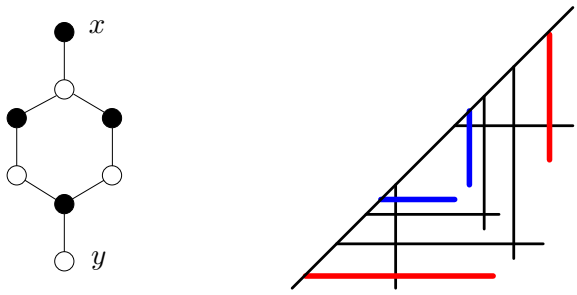

Figure 1.18.: The graph Φ and the two possible positions of x and y in a Stick representation of G.

Proposition 1.26. *There is a bipartite hook graph that is not a Stick graph.*

Proof. The proof is based on the graph Φ shown in Figure 1.18. The vertices x and y are the *connectors* of Φ. Let G be a graph that contains an induced Φ and a path p_{xy} from x to y such that there is no adjacency between inner vertices of p_{xy} and the 6-cycle of Φ. Observe that the Stick representation of the 6-cycle is essentially unique. Now it is easy to check that in a Stick representation of G the sticks for the connectors have to be placed like the two blue sticks or like the two red sticks in Figure 1.18, otherwise the sticks of x and y would be separated by the 6-cycle, whence one of the sticks representing inner vertices of p_{xy} and a stick of the 6-cycle would intersect. Depending on the placement the connectors are of type *inner* (blue) or *outer* (red).

Consider the graph Φ^4 depicted in Figure 1.19 together with a hook representation of it. Suppose for contradiction that Φ^4 has a Stick representation. It contains four copies $\Phi_1, \ldots, \Phi_4$ of the graph Φ with connectors $x_1, \ldots, x_5$. By our observation above, the connectors of each Φ_i are either of inner or outer type. We claim that for each $i \in \{1, 2, 3\}$, connecters of Φ_i and Φ_{i+1} are of different type. If the type of both connectors of Φ_i and Φ_{i+1} is inner, then such a placement would force extra edges, specifically an edge between the two 6-cycles of Φ_i and Φ_{i+1}. And if both are outer then such a placement would separate x_i and x_{i+2}, see Figure 1.20 on the left.

It follows that the connector type of the Φ_i's is alternating. In particular, there is $i \in \{1, 2\}$ such that the connectors of $\Phi_i, \Phi_{i+1}, \Phi_{i+2}$ are of type inner–outer–inner in

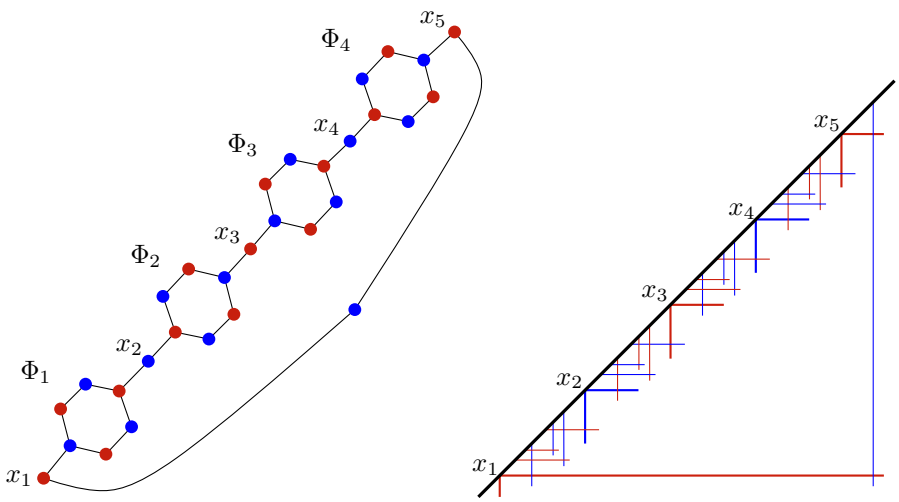

Figure 1.19.: A bipartite hook graph (with hook representation) that is not a Stick graph.

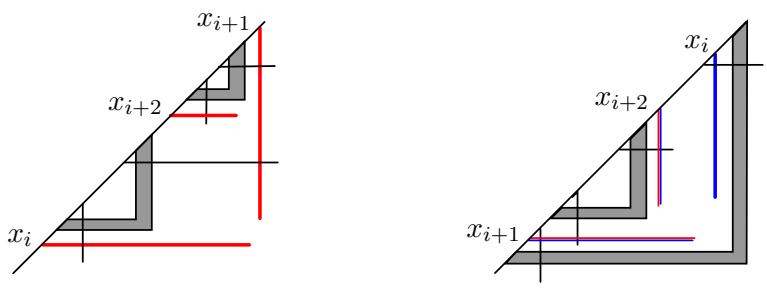

Figure 1.20.: Stick representations of Φ_i and Φ_{i+1} with connecters of type inner–inner (left) and inner–outer (right).

this order. The right-hand side of Figure 1.20 illustrates how Φ_i and Φ_{i+1} have to be drawn in a Stick representation. Since x_{i+2} is one of the inner type connectors of Φ_{i+2}, there is no chance of adding the sticks for Φ_{i+2} to the drawing without intersecting sticks representing Φ_i and Φ_{i+1}. This is a contradiction and hence Φ^4 is not a Stick graph.

1.5.3. Stabbability

We proceed to show that

$$\text{SegRay} \not\subset \text{StabGIG} \qquad\qquad \text{4-DORG} \not\subset \text{StabGIG}.$$

As an intermediate step we prove that there are GIGs that are not stabbable. Techniques used in the proof will be helpful to show the two seperations.

Proposition 1.27. *There exists a GIG that is not a StabGIG.*

Proof. Consider a GIG representation of a complete bipartite graph $K_{n,n}$. The GIG representation forms a grid in the plane. Now we add segments such that for every pair of cells in the same row or in the same column there is a segment that has endpoints in both of the cells. Furthermore, those segments can be drawn in such way that a horizontal and a vertical segment intersect if and only if both intersect a common cell completely, that is, they do not have an endpoint in this cell. Denote the resulting GIG representation by R_n and the corresponding GIG by G_n.

Suppose for contradiction that G_n has a stabbable GIG representation R'_n for all $n \in \mathbb{N}$. By the Erdős-Szekeres theorem for monotone subsequences, for every $k \in \mathbb{N}$ there exists $n \in \mathbb{N}$ such that in R_n there are subsets H and V consisting of k horizontal and k vertical segments that represent vertices of the $K_{n,n}$ in G_n, such that they appear in the same order (up to reflection) as segments in R_n representing the same set of vertices. In R_n those segments induce a subgrid to which we added the blue segments depicted in Figure 1.21. That is, for each cell c in the subgrid we have a horizontal segment h_c and a vertical one v_c such that h_c and v_c intersect only each other and the segments building the boundary of c.

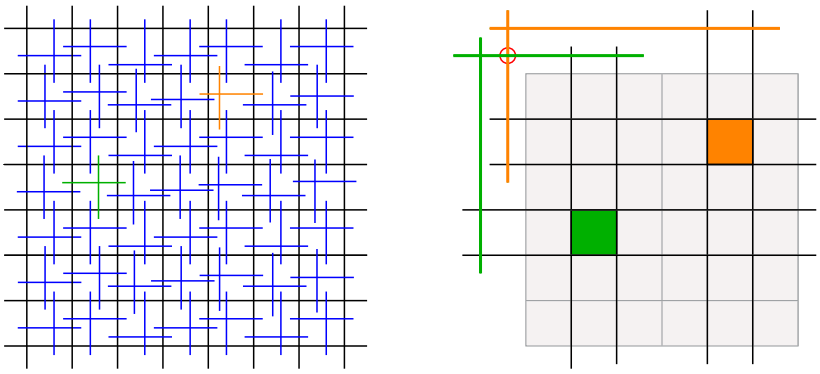

Figure 1.21.: Partial representation of R_n on the left. Replaced cell segments in R'_n on the right.

It is easy to see that this partial representation in R_n is not stabbable if k is large. Since the segments of the subgrid appear in the same order (up to reflection) in R'_n, we only have to consider the placement of cell segments h_c and v_c. We restrict our attention to cells not lying on the boundary of the grid and fix a stabbing line ℓ for R'_n. There are two possibilities for the placement of h_c and v_c in R'_n. One case is that the intersection point p_c of h_c and v_c lies in c or in one of the eight cells surrounding c. Then the segments h_c and v_c can only be stabbed by ℓ if at least one of those eight cells around c is intersected by ℓ. The cells intersected by ℓ in R'_n are only $O(k)$ many, so

their neighbouring cells are only $O(k)$ many as well. This shows that $\Omega(k^2)$ intersection points p_c have to lie outside of the grid in R'_n (as depicted in Figure 1.21 on the right). However, we show that this is possible for only $O(k)$ of them.

If an intersection point lies outside of the grid it is assigned to one quadrant, i.e., p_c lies above or below and left or right of the interior of the grid. Every quadrant contains at most $O(k)$ points p_c: We index each cell by its row and column in the grid so that the bottom- and leftmost cell is $c_{1,1}$. If the intersection points corresponding to cells $c_{u,v}$ and $c_{x,y}$ lie in the upper left quadrant, then $u < x$ implies $y \leq v$. This is illustrated in Figure 1.21, where it is shown that otherwise the cell segments of the colored cells produce a forbidden intersection. It follows that at most $O(k)$ intersection points of cell segments can lie in one quadrant, and hence $O(k)$ of them lie outside of the grid. We conclude that G_n has no stabbable GIG representation for a sufficiently large n. □

For SegRay graphs we give a similar construction that shows that there are SegRay graphs which do not belong to StabGIG. First we will construct a graph that cannot be stabbed in any SegRay representation.

Lemma 1.28. *Let R be a SegRay representation of a cycle C with $2n$ vertices. For the vertices in C being represented as rays it holds that their order in C is up to reflection and cyclic permutation equal to the order of the rays representing them in R .*

Proof. Let L be a horizontal line below all horizontal segments in R. Contracting each ray to its intersection point with L yields a planar drawing of a cycle C' with n vertices such that the vertices lie on L and edges are drawn above L. This is also known as a *1-page embedding* of C'. It is easy to see that edges in C' have to connect consecutive vertices on L or the two extremal ones. Now the conclusion of the lemma is straightforward. □

Proposition 1.29. *There exists a SegRay graph that has no stabbable SegRay representation.*

Proof. Consider the graph defined by the SegRay representation R in Figure 1.22. Let R' be an arbitrary SegRay representation of this graph. The order of the rays in R' is up to reflection and cyclic permutation equal to the one in R by Lemma 1.28. Hence without loss of generality the rays of the left half in R appear consecutively in R'. Now observe that the two yellow segments below the red segment in the left half of R also have to be below the red segment in R'. Similarly, the two yellow segments above the red segment in R must lie above the red segment in R'. Furthermore, these two yellow segments have to lie below the top of the red vertical ray in R'. It follows that, as in R, the red segment and the red ray seperate the plane into four quadrants in R' such that

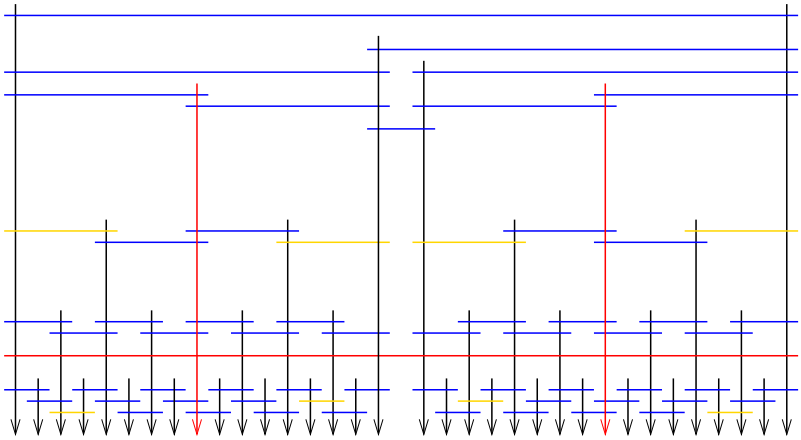

Figure 1.22.: A SegRay graph with no stabbable SegRay representation.

each quadrant contains exactly one of the considered yellow segments. Any line in the plane can intersect at most three of the quadrants and thus will miss a yellow segment in R'. Therefore, R' is not stabbable and the conlusion follows. □

We add a vertex h to the graph in Figure 1.22 that is adjacent to all rays. This graph is still a SegRay graph. We call this graph a *bundle* and the set of horizontal segments its *head*. The bundle is not stabbable in any SegRay representation by the proposition above. This means, in any StabGIG representation of a bundle there is one horizontal segment above the segment representing h and one below. Indeed, otherwise the representation of the bundle can be modified by extending vertical segments to rays in one direction to obtain a stabbed SegRay representation. Using this property of a bundle we can show the following.

Proposition 1.30. *There exists a SegRay graph that is not a stabbable GIG.*

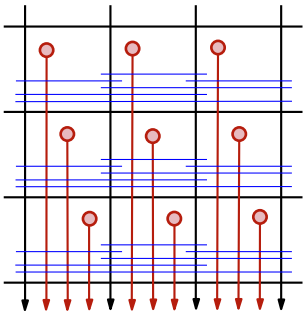

Figure 1.23.: Illustration of a SegRay graph that is not a StabGIG

Proof. Similar to the construction in Proposition 1.27, consider a SegRay representation of a complete bipartite graph $K_{n,n}$. In this representation we see a grid with cells. We place in each of the cells the head of a bundle as indicated in Figure 1.23. Now, for each pair of cells in the same row of the grid, add a spanning horizontal segment with endpoints in the given cells. We do it in such a way that the rays of a bundle are intersected by the segment if the head of the bundle lies in a cell between the two given cells.

Denote by R_n the resulting SegRay representation and let H_n be the SegRay graph defined by R_n. Suppose that H_n has a stabbable GIG representation R'_n. As in the proof of Proposition 1.27, given an integer $k \geq 1$ it follows by the Erdős-Szekeres theorem that for sufficiently large n there is a subgrid of size k in R'_n, where the order of the horizontal and vertical segments is either preserved or reflected with respect to R_n. Assume that it is preserved. Now we restrict our attention to the relevant bundles and horizontal segments of R_n according to the cells of the subgrid. In R_n again this looks like in Figure 1.23, but this time with respect to the fixed subgrid.

Let us now consider the placement of the bundles and blue segments in R'_n. Given a cell c in the subgrid, let y_c be the horizontal grid segment bounding c from below. By Proposition 1.29 and its consequences, the bundle lying in cell c contains a horizontal segment that lies above y_c in R'_n. We denote this segment by h_c and let x_c be an arbitrary ray of the bundle intersecting h_c. Consider now the left side of Figure 1.24 showing a 3×3 box and a ray x of the subgrid that lies strictly to the left of the box in the representation R_n. Let c_1, c_2, c_3 be the three shaded cells. Then we claim that at least one of $h_{c_1}, h_{c_2}, h_{c_3}$ is placed to the right of x in R'_n.

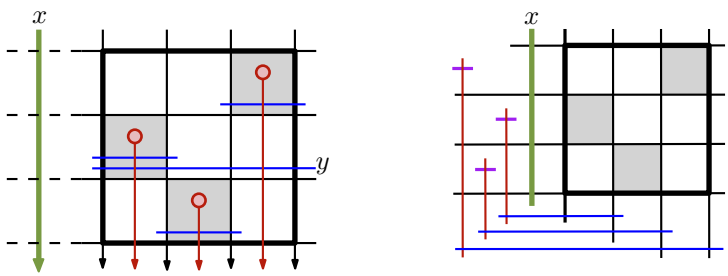

Figure 1.24.: For at least one gray cell c, the purple segment h_c lies to the right of x in R'_n.

Suppose to the contrary that all lie to the left. If we use the fact that h_{c_i} is above x_{c_i} in R'_n for each $i \in \{1, 2, 3\}$, then it is straightforward to see that $h_{c_1}, h_{c_2}, h_{c_3}, x_{c_1}, x_{c_2}, x_{c_3}$ and the three short blue horizontal segments depicted on the left of Figure 1.24 have to be placed in R'_n as shown on the right of Figure 1.24 (segments h_{c_i} are colored purple). But then the segment y, which is the long blue one on the left, cannot be added to the partial representation without creating forbidden crossings. This shows our subclaim.

In the next step we consider the green box of the fixed subgrid shown on the left of Figure 1.25. Using our subclaim we have that each of the three shaded 3×3 boxes contains a cell c such that h_c is placed to the right of x in R'_n. Now we apply the symmetric version of this claim to deduce that one of these three segments also lies to the left of x' in R'_n. We conclude that there is a segment that is strictly contained in the green box in R'_n.

In the final step we consider four copies of the green box that are placed in the fixed subgrid of R_n as shown on the right of Figure 1.25. Since each copy strictly contains a segment in R'_n, each line in the plane will miss at least one of the four segments. This shows that R'_n is not stabbable for sufficiently large n and completes the proof. □

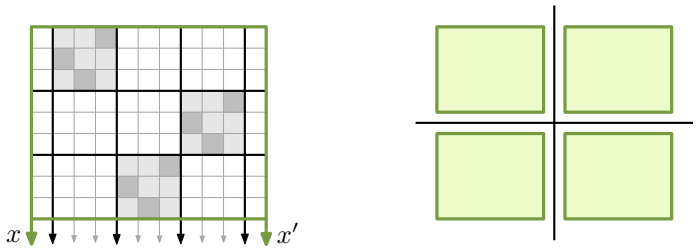

Figure 1.25.: There is a cell c in the green box such that h_c is drawn inside the box in R'_n.

Proposition 1.31. *There exists a 4-DORG that is not a stabbable in any GIG representation.*

Proof. Since the ideas here are similar to those used for Propositions 1.29 and 1.30, we only provide a sketch of this proof. Consider the following construction. Take a 4-DORG representation of a complete bipartite graph $K_{n,n}$. Similarly to previous constructions this yields a grid with cells. For each cell we add four rays starting in this cell, one in each direction and such that a vertical and a horizontal ray intersect if and only if they entirely intersect a common cell. Call this representation R_n and the corresponding intersection graph G_n. We claim that for sufficiently large n there is no stabbable GIG representation of G_n.

Suppose to the contrary that there exists a StabGIG representation R'_n of G_n. Again by applying the Erdős-Szekeres theorem we may assume that there is a large subgrid of size k in R'_n, such that the order of the grid segments in R'_n agrees with the order in R_n (up to reflection).

Given the representation R'_n, we want to partition the cells of the subgrid according to the placement of the four segments representing the rays that start in a given cell of our construction. Note that these four segments intersect in such a way that they enclose a rectangle in R'_n. Therefore, we can distinguish the following cases: the rectangle (1) is

contained in a grid cell, (2) it does not intersect a grid cell, (3) it contains some but not all grid cells, and (4) it contains all of the grid cells (see Figure 1.26 for the cases from left to right). Each of these cases again can be split into at most four natural subcases. For instance, if the rectangle contains some of the grid cells, then it also has to contain a corner of the grid, which gives rise to four subcases.

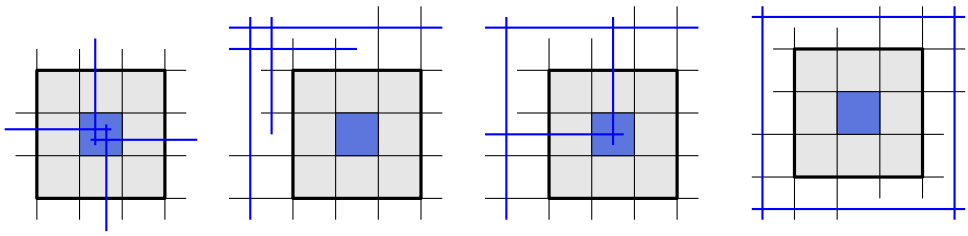

Figure 1.26.: Four different situations for the blue cell and its segments in R'_n

Using similar arguments as in previous proofs of the section and the assumption that R'_n is stabbed by a line, one can show that each partition class contains at most $O(k)$ cells. Thus, for large enough n and k we get a contradiction since our subgrid has k^2 cells. This observation completes the proof. □

It remains open whether there exists a 3-dimensional GIG that is not stabbable. It is tempting to look for an example that produces again a large grid in every representation (to get non-stabbability), but it turned out that all these examples seem to have dimension 4. We also tried with SegRay graphs satisfying the properties of Lemma 1.7 since they have dimension 3. However, we did not succeed with finding such a SegRay that is not stabbable.

1.6. The dimension of bipartite segment intersection graphs

In this section we show that the dimension of a k-dir graph, a segment intersection graph that can be represented using at most k slopes, is bounded by a linear function in k. Furthermore, we show that there are k-dir graphs that have a dimension linear in k, by showing that we can represent the standard example S_{k+2} using k slopes.

Lemma 1.32. *The dimension of a bipartite k-dir graph is at most $2k$.*

Proof. Given a k-dir representation of a bipartite graph G, we give a realizer of the partial order by using the sweeping technique from Proposition 1.5 in the k directions. Therefore, we show that two non-intersecting segments in a k-dir representation are separated by a line in one of the k directions. We consider two non-intersecting segments a and b in direction d_a and d_b. We show that a and b can be separated by a line of

direction d_a or d_b. We apply a linear transformation T such that $T(d_a)$ and $T(d_b)$ are both vertical or horizontal segments. The vertical or horizontal segments are separated by a vertical or a horizontal line ℓ as shown in the proof of Proposition 1.5. Inverting the linear transformation leads to a line $T^{-1}(\ell)$ of direction d_a or d_b, which separates the segments a and b.

This implies that the dimension of a bipartite k-dir graph is at most $2k$. □

After showing that the dimension of bipartite k-dir graphs is bounded linearly in k, we show that this bound is up to a constant factor tight.

Theorem 1.33. *The standard example S_{k+2} for even k has a segment representation using k different slopes.*

Proof. We construct a segment representation of S_{k+2}, for even n, iteratively. We start with the representation of S_4 that is shown in Figure 1.27. The construction will be symmetric to the horizontal line ℓ. We keep the following invariant during the

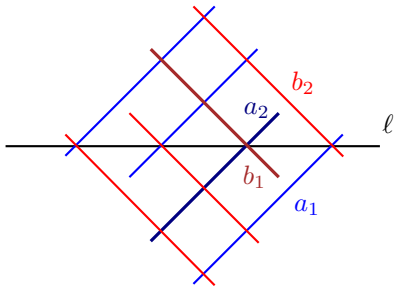

Figure 1.27.: A symmetric representation of S_4.

construction: Removing the segment b_2 leads to segment representation, such that the minima (blue) can be extended to rays into the upper half-plane without creating new intersections. Symmetrically, removing the segment a_1 from the representation, such that the maxima (red) can be extended to rays into the lower half-plane without creating new intersections. In addition, the segments a_1 and b_2 share the leftmost or rightmost intersection point on ℓ.

We use the invariant to construct a segment representation of S_k for even k, with $k-2$ slopes from a representation of S_{k-2} using $k-4$ slopes. The construction will be symmetric with respect to a horizontal line ℓ. Note that the representation of S_4 shown in Figure 1.27 satisfies all those invariants. We describe the construction in the case that b_2 and a_1 share the rightmost intersection point on ℓ.

First, we add two parallel lines with a positive slope that is smaller than all slopes of the segments in the representation of S_{k-2}. The lines intersect ℓ to the left of the representation of S_{k-2} and do not intersect the representation. We extend the blue

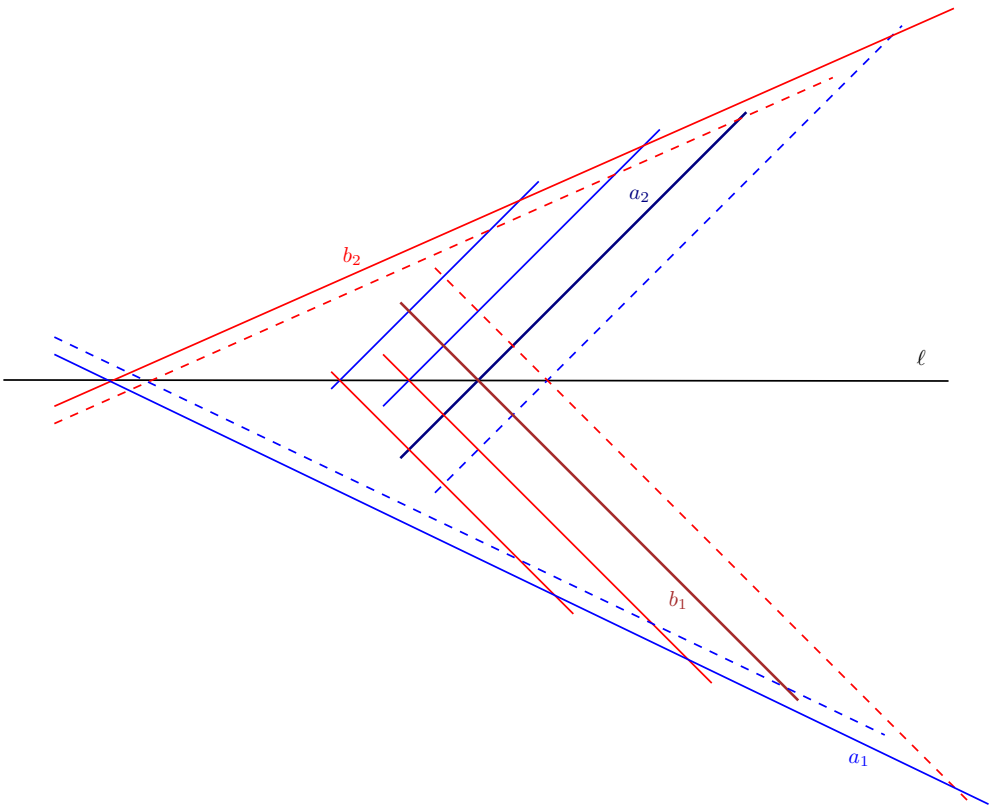

Figure 1.28.: A segment representation of S_6.

segments, such that all segments except for a_2 intersect the upper of the two new lines, which is possible since we assumed the slope of the new segments is small. The segment a_2 ends after intersecting the lower of the new lines. The upper of the new lines is the supporting line for the new segment b_2. We fix a position of the new segment b_2, such that it contains all intersection points of segments and the new line. For now, we consider b_2 as a ray pointing in the lower left direction of the line. The lower of the new lines supports a segment that contains all but the rightmost intersection points with the blue segments and is also considered as a ray to the left direction of its supporting line. The representation in the half-plane below ℓ is symmetric to the representation above ℓ.

We show that this construction gives a segment representation of S_k, i.e., we have to show that segments of one color do not intersect and that each segment intersects $k-1$ other segments.

Note, that the segments of one color do not intersect by construction and the invariant: The invariant states that we can extend the blue segments in the upper half-plane and the red segments in the lower half-plane without creating new intersections.

The two new red segments intersect all but one blue segment by construction. The rightmost blue that does not intersect the new segment on the lower line lies on the supporting line of the former segment a_1, the segment which could not be extended without creating a new intersection. Thus this segment intersects $k-1$ segments, all but the red segment on the lower line. The segment a_2 intersects all except the new segment b_2. The modifications of all other blue segments cause two new intersections with the segments on the new lines. Thus, the number of intersections is $k-1$ for the red segments on the new supporting lines and the blue segments on the old supporting lines. This implies the correct number of intersections for each segment by symmetry. We added two new directions which gives a $k-2$-dir representation of S_k inductively.$\square$

2. The slope number of segment intersection graphs

In this chapter, we investigate the *slope number of segment intersection graphs*. We define the slope number of a segment intersection graph G to be the minimal k, such that G has a k-dir representation.

The main tool is a connection between Hamiltonian cycles in maximal planar graphs G and grid intersection representations of a graph P_G^* associated with G. A similar connection exists between Hamiltonian cycles in G and segment intersection representations of P_G^*. This is presented in Section 2.2. In Section 2.3 we consider the behaviour of the slope number under removal of vertices. We use the methods from Section 2.2 to show that there are graphs with a slope number that is linear in the number of vertices (and edges), and the removal of one special vertex leads to a grid intersection representation, i.e., the slope number drops down to two. Another consequence of Section 2.2 is that the recognition of grid and of segment intersection graphs is NP-hard. This follows from known NP-hardness results for the Hamiltonian cycle and the Hamiltonian path problem in maximal planar graphs [Wid82, AG11]. In Section 2.4 we extend this by showing that the Hamiltonian cycle problem in maximal planar graphs with given Hamiltonian path is NP-complete. With our tools this corresponds to the problem of finding a grid representation of a graph with a given segment representation. This implies that the problem of determining the slope number of a graph is NP-hard. The input graphs in the reduction have at most slope number four and thus it is NP-hard to determine if a 4-dir graph has a 2-dir (GIG) representation.

The bipartite k-dir graphs we consider in the first sections always use an even number of directions k. In Section 2.5 we raise the question whether there is a bipartite SEG with an odd slope number. We answer this question affimatively and show that the class of bipartite 3-dir graphs that do not have a 2-dir representation has no *monochromatic* direction, that is, all segments of one slope have the same color in a proper two coloring.

Parts of this chapter appeared in [Hof15].

2.1. Background

Kratochvíl and Matoušek [KM94] considered SEG graphs. They show that for each fixed k the recognition of k-dir graphs is in NP, while the recognition problem for segment intersection graphs is hard in the existential theory of the reals, and thus is not known to be in NP. Kratochvíl [KM89] has also shown that the recognition problem of k-dir graphs is NP-hard for each fixed $k \geq 2$.

Kratochvíl and Matoušek also consider segment intersection graphs that can be represented using a fixed sets of directions. They prove that for $k \geq 4$ the set of graphs that have a k-dir representation using a fixed set of directions depends on the directions. In fact, the set of graphs that can be represented as k-dir graph using the directions $\{\alpha_1 \ldots, \alpha_k\} =: A$ differs from the ones that can be represented using the directions $\{\beta_1, \ldots, \beta_k\} =: B$, unless there is a linear transformation that maps A onto B [ČKNP02].

The computational complexity of slope minimization of a segment intersection graphs has not been considered explicitly, but an anonymous referee pointed out that the graphs constructed in the proof of NP-completeness of GIGs by Kratochvíl [Kra94] have a realization as SEG.

Usually the *slope number* of a graph G is the number of slopes that is necessary to represent the edges in a non-degenerate straight-line drawing of G. We consider this parameter for planar drawings, the *planar slope number*, in Chapter 6. So we refer to this chapter for the background about different slope numbers known in literature.

2.2. Segment intersection graphs and Hamiltonian paths

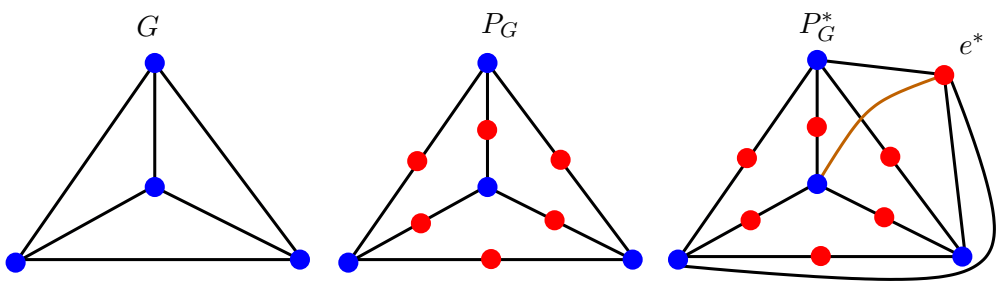

Figure 2.1.: The graphs K_4, P_{K_4} and $P^*_{K_4}$.

We start with the definition of the graph P_G^*, which is the main tool for our observations on the slope number of segment intersection graphs. First recall that P_G denotes the full subdivision of a graph $G = (V, E)$ (or its vertex-edge incidence poset). Let

$$P_G^* := (V \cup E \cup \{e^*\}, E(P_G) \cup \{ve^* | v \in V\})$$

be the graph obtained by the full subdivision of G, where one vertex e^* is added that is adjacent to every original vertex of G. We are interested in segment and GIG representations of P_G^*. By Lemma 0.2 the graph P_G admits a GIG representation if and only if G is planar. The following lemma characterizes the situation when P_G^* is a grid intersection graph.

Lemma 2.1. *The graph P_G^* is a grid intersection graph if and only if G is a subgraph of a planar graph with Hamiltonian cycle.*

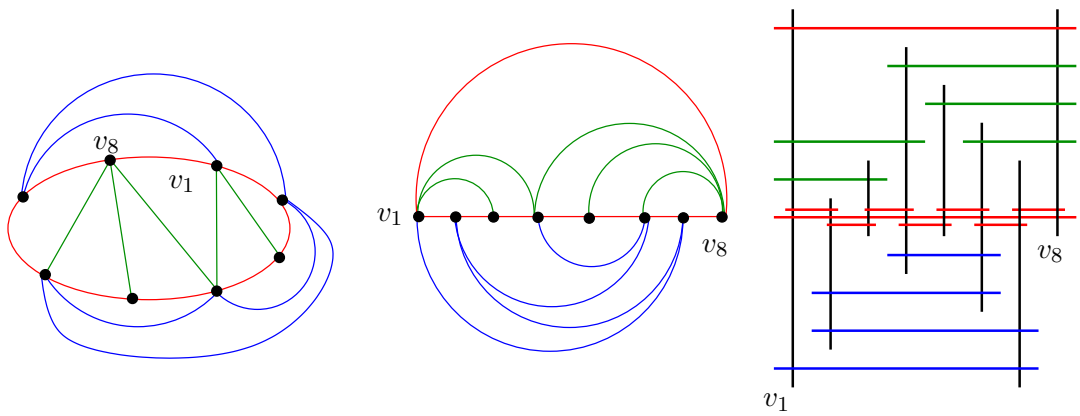

Figure 2.2.: A graph with Hamiltonian cycle, its 2-page book embedding and GIG representation of P_G^*.

Proof. The right-hand side of this lemma, the existence of a planar supergraph with Hamiltonian cycle, is the known characterization for graphs with 2-page book embedding according to Lemma 0.1.

A 2-page book embedding of G and a grid intersection representation of P_G^* coincide as shown in Figure 2.2: Consider a GIG representation of P_G^*, such that e^* is represented by a horizontal segment. The segment e^* partitions the segments of E into two parts, the segments above e^* and the ones below. Considering only the part above e^* we have a representation that can easily be modified to a semi-bar visibility representation, which corresponds to an outerplanar graph by Lemma 0.4. The lower part gives analogously a semi-bar visibility representation. The order in which e^* intersects the segments of the original vertices V is the order of the semi-bars in both representations. Thus, we have a partition of the edges in two outerplanar graphs that can be extended to outerplanar graphs with the same outer face cycle. We obtain two 1-page book embedding with the same order of vertices, a 2-page book embedding. □

As a short remark we give a short idea of the characterization by Bernhart and Kainen [BK79]: A Hamiltonian cycle in a planar graph partitions the edges that are not use by the cycle

into parts. One part consists of the edges inside of the cycle, the other one of the edges outside. Each of the blocks forms, together with the edges of the cycle, an outerplanar graph with the same outer face cycle. The 1-page book embeddings of these outerplanar graphs can be combined to a 2-page book embedding. For the other implication we notice that two consecutive vertices as well as the first and last vertex in the order of the Hamiltonian cycle can be connected by an edge without violating properties of the 2-page book embedding, which gives a Hamiltonian cycle.

The main tool of this chapter is a characterization of segment representations of P_G^*.

Theorem 2.2. *The graph P_G^* is a segment intersection graph if and only if G is a subgraph of a planar graph with Hamiltonian path. Moreover, a* SEG *representation of P_G^* can be computed from a Hamiltonian path in polynomial time.*

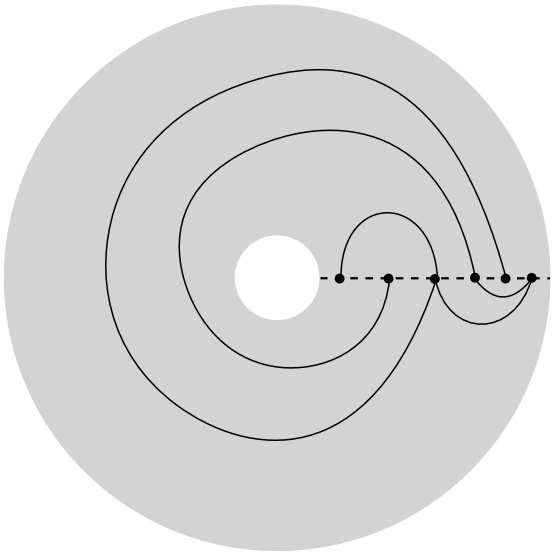

Figure 2.3.: A linear cylindric drawing (on an unwrapped cylinder).

The class of graphs with a planar supergraph with Hamiltonian path has also been characterized as the graphs with a *linear cyclindric drawing* [ABB+11, AG11]: A linear cylindric drawing of a graph is essentially a 2-page book embedding on a cylinder. The difference here is, that edges may be connected to two different sides of the spine by encircling the cylinder. These edges correspond exactly to the bad edges in a (pseudo)segment representation of P_G^*.

Before we delve into the construction of a segment representation of P_G^* using a Hamiltonian path, we show a version of the theorem for pseudosegment intersection graphs.

Theorem 2.3. *The graph P_G^* is a pseudosegment intersection graph if and only if G is a subgraph of a planar graph with Hamiltonian path.*

Proof. We first show, that we can add edges to G such that we obtain a planar graph with Hamiltonian path if P_G^* has a pseudosegment representation: The pseudosegment of e^* gives a linear order of the vertices $v_1, \ldots, v_n$. We add a pseudosegment parallel to e^*, that intersects exactly v_i and v_{i+1} in the representation for each $i \in \{1, \ldots, n-1\}$. The modified representation is a pseudosegment representation $P_{G'}^*$ for a graph G' that has the Hamiltonian path $(v_1, \ldots, v_n)$. The planarity of G' follows from Lemma 0.2.

To show that P_G^* has a pseudosegment representation if G has a Hamiltonian path, we start with a planar straight line drawing of G, see Figure 2.4. We can interpret this drawing as contact representation of P_G, where the edges are represented by line segments ending on circles which correspond to the vertices. We can obtain a pseudosegment contact representation of P_G by removing a small part from the circles such that the contacts are preserved. We decide which part we remove after drawing e^* in the following way: Let $v_1, \ldots, v_n$ be the order of the vertices along the Hamiltonian path. Then, we draw a pseudosegment for e^* starting inside of the cycle of v_1. Iteratively, we leave the cycle of v_i into a face that is incident to v_{i+1} and enter circle v_{i+1} without intersecting any other segment or cycle. A face with this property exists because $v_i v_{i+1}$ is an edge of G. We finish the construction by removing a segment of each circle, such that the pseudosegment representing e^* intersects each circle exactly once. □

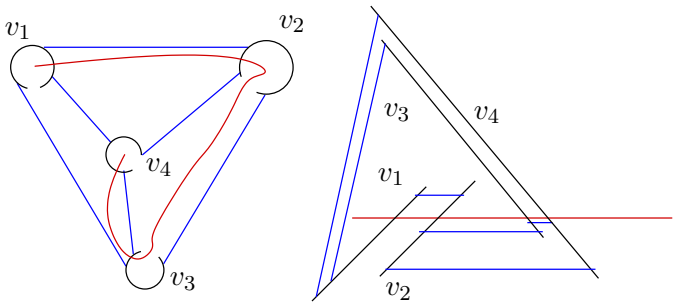

Figure 2.4.: A pseudosegment and a segment representation of $P_{K_4}^*$

We continue by considering the differences between a GIG representation and a SEG representation of P_G^*. Let R be a SEG representation of P_G^* with horizontal segment e^*. We call the part of a segment that lies above e^* the *upper part*, the one below e^* the *lower part*. In contrast to a GIG representation, a segment representation may have edge segments that intersect the lower part of one vertex segment and the upper part of another vertex segment as shown in Figure 2.4 right. We call those edges the *bad edges*. Recall, that a segment representation of P_G^* gives an embedding of G. The bad edges

vw are exactly those, such the unique closed curve C_{vw} on the segments v, w, vw and e^* separates the first vertex of the Hamiltonian path v_1 from the last one v_n. The curve C_{vw} corresponds to the cycle consisting of the Hamiltonian path and the bad edge in the graph G. This implies that removing all bad edges leads to a planar graph in which v_1 and v_n lie in the same face.

Observation 2.4. *Removing the bad edges from G leads to graph that is a subgraph of a planar graph with Hamiltonian cycle.*

Note, that fixing a Hamiltonian path in a planar graph also fixes the bad edges. The bad edges can be partitioned into several types: We call a bad edge v_iv_j with $i < j$ an *upward edge* if its segment connects the lower part of v_i with the upper part of v_j. Otherwise it is called *downward edge*. We call a bad edge vw a *front* back edge if v_1 lies in the bounded region of C_{vw}. A vertex incident to a bad front edge is a bad front vertex, and a bad back vertex if it is incident to a bad back edge. The partition of the bad edges in upward and downward edges depends (up to interchanging the types of all edges) only on the Hamiltonian path. Whether a bad edge is a front or a back edge is determined by the choice of the outer face in the embedding of G. With these definitions we can make the following observation on the slopes of a segment representation of P_G^*. Note that we only consider *normal representations*, where $e*$ is represented by a horizontal segment and all other segments have positive slope.

Observation 2.5. *If the edge v_iv_j is a front upward edge or a back downward edge, then $s(v_i) < s(v_j)$. In the case that v_iv_j is a front downward or a back upward edge, we have $s(v_j) < s(v_i)$. The slopes of the bad edge v_iv_j lies strictly between the slopes of its incident vertices.*

Proof of Theorem 2.2. We give a construction of a segment representation of P_G^* from an embedding of G where the end of the Hamiltonian path lies on the outer face, i.e., all bad edges are front bad edges. By symmetry, we only show how to construct a representation that has only upward bad edges. We can assume without loss of generality, that G is a maximal planar graph.

We start to build the representation with the segment e^* has a horizontal ray in positive x-direction. We construct a representation such that all slopes of segments are non-negative. We first add the part of the vertices incident to bad edges and the bad edges to the representation.

We will choose the slopes, such that the following properties and the conditions from Observation 2.5 are satisfied.

a. For each pair of upward bad edges of the form v_iv_k and v_jv_l with $i \leq j < k \leq l$ we have $s(v_iv_k) \leq s(v_jv_l)$ (see Figure 2.5).

b. For each pair of upward bad edges of the form v_iv_k and v_jv_k with $i < j < k$, we have $s(i) \leq s(j) < s(k)$ and $s(v_iv_j) \leq s(v_jv_k)$.

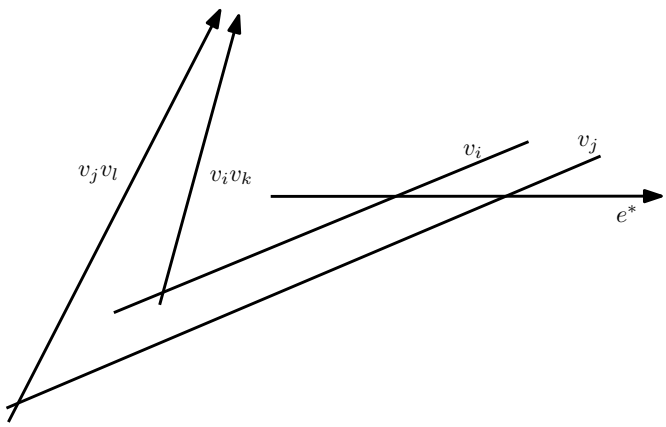

Figure 2.5.: Forbidden configuration of Property a.

With these properties, we construct a representation by adding v_j and all bad edges v_iv_j with $i < j$ to the representation in step j. We keep the following invariants for the representation R_j after step j.

1. If v_iv_k with $i \leq j < k$ is a bad upward edge, then there is an interval on the lower part of v_i, such that a ray starting from this interval in positive $s(v_iv_k)$ direction does not intersect any segment of R_j.

2. If v_iv_k with $i \leq j < k$ is a bad downward edge, then there is an interval on the upper part of v_i, such that a ray starting from this interval in negative $s(v_iv_k)$ direction does not intersect any segment of R_j.

In step j, we pick a point p_j on e^* right of the intersections with all other vertices v_i with $i < j$, such that a line ℓ_j of slope $s(v_j)$ through the point does not intersect any other segment of the representation R_{j-1} than e^*. This can be done by choosing the point far enough to the right on e^*. For each incident bad upward edge v_iv_j of v_j we extend the lower part of v_j until it sees the upper part of v_i with by a sight line of slope $s(v_iv_j)$. If for each bad upward edge v_jv_k we further extend the lower part of v_j until a line of slope $s(v_jv_k)$ starting from an interval on v_j does not intersect any other segment of the representation R_{j-1}. Symmetrically, we extend the upper part of v_j until all its neighbors v_i with a bad upward edge v_iv_j can be seen by a sight line of slope $s(v_iv_j)$, or until a line of slope $s(v_jv_k)$ does not intersect the rest of the representation through an interval on the upper part of v_j if v_jv_k is a bad downward edge. After determining the ends of v_j on ℓ_j this way we can draw the bad edges v_iv_j on the constructed sight lines. These sight lines do not cross which follows from Property b.

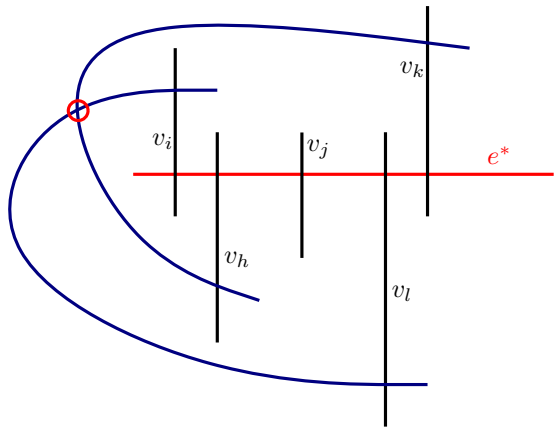

Figure 2.6.: There is no active upward and downward edge in one step.

It remains to show that the invariants remain valid. First note, that there are no upward and downward bad edges $v_h v_k$ and $v_i v_l$ with $h, i \leq j < k, l$ simultaneously in one step, because of planarity reasons as shown in Figure 2.6. We assume that each bad edge $v_i v_k$ with $i \leq j < k$ is an upward edge and show that Invariant 1 holds, i.e., the upper part of v_j does not block a ray from the lower part of some v_i with slope $s(v_i v_k)$. This follows directly from Property a. We constructed a representation of R_j that satisfies the invariant.

We proceed to describe an algorithm that assigns slopes to the bad edges and the incident vertices that satisfies Property b. We call a bad edge $v_i v_k$ with $i \leq j < k$ *active in step* j and recall that a non-empty set of active edges in one step either consists only of upward edges or downward edges. Because of symmetry reasons, we assume we only deal with upward edges. We fix a set of positive slopes $\cdots < s_1 < s_2 < s_3 < \ldots$. The slopes with odd index will be assigned to vertices and the slopes with even index to edges. We assign the slope s_1 to v_1 and keep track of the active edges while iterating through the vertices along the Hamiltonian path. We set $a_1 := 2$ to store a lower bound the active edges will obtain. Let A_j be the set of active after step j. If vertex v_{j+1} is incident to a bad edge in A_j, then we assign slope s_{a_j+1} to v_{j+1}. The bad edges $v_i v_{j+1}$ with $i < j+1$ are obtain slope a_j. If v_{j+1} is incident to a bad upward edge $v_{j+1} v_k$ then we set $a_{j+1} := a_j + 2$.

Finally, we add the remaining vertices and the good edges. The good edges will be represented by horizontal segments. We assign a slope to the vertices that are only incident to good edges, which is already used by a bad vertex. By Observation 2.4, we can construct a GIG representation R of the subgraph of P_G^*, which we obtain after removing the bad edges. We consider the order of upper and lower endpoints of the vertical segments representing the vertices of G. We add the remaining vertices, such

that the order of the y-coordinates of the endpoints agrees with the order of endpoints in R. This can be done, since we can assume the part of the vertices that is intersected by a bad edge is unbounded, since the bad vertices lie on the unbounded face in R. The horizontal segments representing the good edges can be added after adding the good vertices in a small stripe around e^* as shown in Figure 2.7.

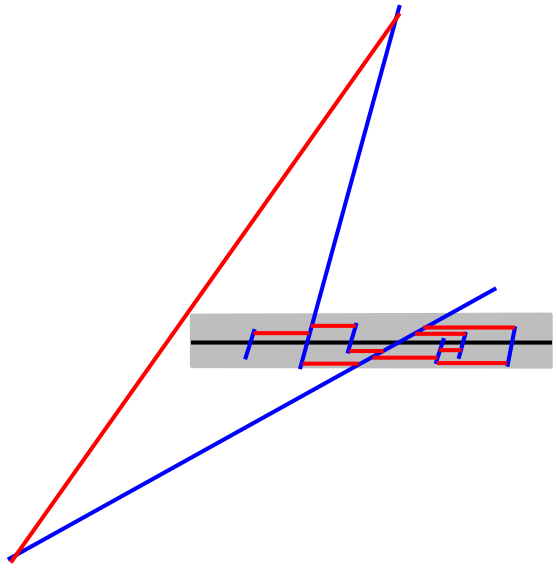

Figure 2.7.: Adding the good edges and remaining vertices to the segment representation.

It remains to show that we can compute the coordinates for the segment representation in polynomial time. Therefore, we choose the slopes s_i from the set $\{1, \ldots, n\}$. By a result from Kratochvíl and Matoušek [KM94, Theorem 1.1(ii)(a)] we can encode the realizability of a segment representation with a given slope for each segment in a linear inequality system with coefficients that depend on the slopes. Since we can choose slopes that have a small encoding, the inequality system can be solved in polynomial time in the dimension of the system. □

2.3. Behavior of the slope number

In the last section we have shown that we can draw P_G^* with as segment intersection graph, where the order of segments intersecting e^* is given by the order of the vertices on a Hamiltonian path. In this section we ask the question, how many slopes are required for a segment representation of P_G^* if the embedding of G and the Hamiltonian path p are not fixed. We show that there is a family planar graphs G_n, such that for each embedding and each Hamiltonian path a segment representation of $P_{G_n}^*$ has $\theta(|V(G_n)|)$

slopes. First we consider the number of slopes of P_G^* for a fixed Hamiltonian path under different embeddings of G.

Let $w_i w_{i+1}, w_{i+1} w_{i+2}, \ldots, w_{i+k-1} w_{i+k}$ be bad edges of the same type (upwards/downwards). We call the vertices a *chain of length* k. If all the edges of the chain are of the same front or back type the chain is called *monotone*.

Lemma 2.6. *Let p be a Hamiltonian path in G and C be a chain of bad edges. Any segment representation of P_G^*, whose vertex order along e^* is given by p, requires at least $2(\lceil \frac{|C|}{2} \rceil + 1)$ slopes.*

Proof. Consider a segment representation of P_G^*. Without loss of generality, let C' be a chain of front upward bad edges. For a bad front edge we know that the segment with the upper part has a larger slope then the segment with the lower part. By Observation 2.5, the bad edges connecting vertices in the chain have slopes that lie between the slopes of their incident vertices. Thus the segment representation uses at least $2(|C'|+1)$ slopes; $|C'|+1$ slopes for the vertices incident to the bad edges, $|C'|$ for the bad edges, and one for e^*.

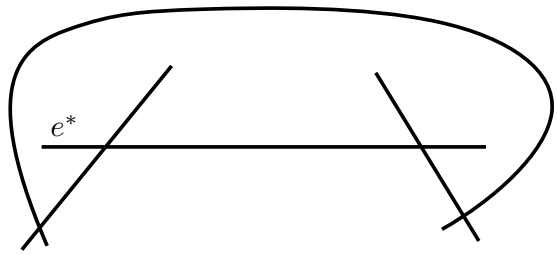

Figure 2.8.: No good edge separates the ends of e^* from the unbounded face.

Therefore, we only have to show that a chain of bad edges of length $|C|$ leads to a monotone chain of bad edges C' of length $\lceil |C|/2 \rceil$. We observe that the unbounded face of $P_G^* - B$, where B is the set of bad edges, is the face that is incident to the two ends of the Hamiltonian path: Otherwise we have good edge that separates both ends of e^* from the unbounded face, which is not possible as shown in Figure 2.8.

On the other hand, we can obtain a pseudosegment representation of each such embedding by flipping one front bad edge over to a back bad as shown in Figure 2.9. This flipping splits the chain into two chains, one of which has the claimed length. □

We show that there is a family of planar triangulations that have only Hamiltonian paths including long chains of bad edges. This way, we prove that there are graphs G_n, such that P_G^* requires a large number of slopes in any segment representation. Therefore, we need the following tools.

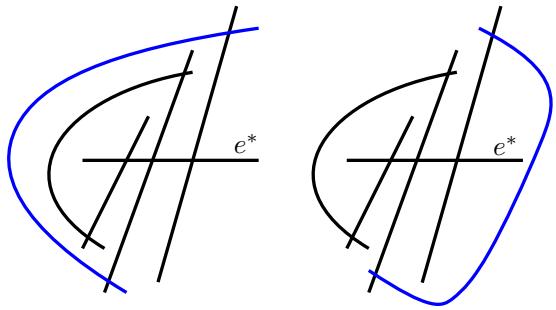

Figure 2.9.: Flipping a front bad edge to the back.

Proposition 2.7. *By P and R we denote the graphs from Figure 2.10.*

1. *Let G be a planar triangulation that contains P as proper subgraph, such that the vertices of the outer triangle are the only vertices of P that are adjacent to vertices of $G - P$. Then each Hamiltonian path of G traverses all vertices of P consecutively, or has an endpoint in P.*

2. *Let G be a planar triangulation that contains R as proper subgraph, such that the three outer vertices of R are a separating triangle. Then each Hamiltonian path of G has one endpoint inside R. In addition, the vertices of the connected components of the inner vertices are traversed consecutively, unless both end vertices of the Hamiltonian path lie inside of R.*

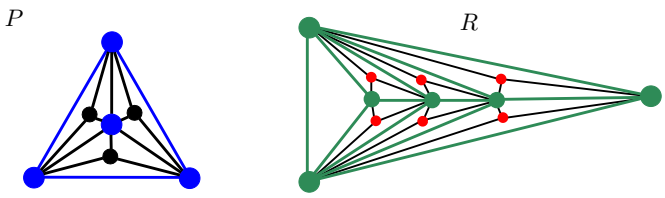

Figure 2.10.: Gadgets for fixing a Hamiltonian path according to Proposition 2.7.

Proof. The proof of this proposition is based on the observation that removing a separator S from G results in a subgraph with at most $|S| + 1$ connected components if G has a Hamiltonian path p (and at most $|S|$ components if G has a Hamiltonian cycle [Chv73]). If the removal of S leaves exactly $|S| + 1$ connected components, then the two ends of h lie in two different components of $G - S$. These observations follow from the fact that a Hamiltonian path p traverses at least one vertex of S whenever it goes from one connected component $G - S$ to another. If the number of connected components of $G - S$ is exactly $|S| + 1$, this implies that all vertices of one component of $G - S$ are traversed consecutively by p.

To show part 1 we assume that G is a planar graph with proper subgraph P, such that the three outer vertices are the only vertices of P connected to $G - P$. We observe that the three outer vertices along with the central blue vertex of P in Figure 2.10 are a separator. Removing these four vertices leads to a graph with at least four connected components, three inside of the triangle and a fourth component outside, because we assumed that P is a proper subgraph of G. Assume G has a Hamiltonian path p with both endpoints in $G - P$, then H traverses the three connected components inside of P consecutively: Since one vertex of the separator is traversed between the connected components by the Hamiltonian path p between two vertices of different connected components we have shown part 1.

To show part 2 we consider a planar graph G with R as a proper subgraph, such that the outer vertices of R are a separator of G. The six green vertices of R are a separator of G. The removal of these vertices leads to (at least) seven connected components. Thus G has no Hamiltonian path with the ends in the same component. This implies that also one of the components of the red vertices of R contains one end of the Hamiltonian path. If one end vertex of the Hamiltonian path lies in the connected component that is separated from the inner vertices of R by the its outer triangle, then the vertices of R are traversed consecutively by the path. □

We use this proposition to construct triangulations with almost fixed Hamiltonian path.

Lemma 2.8. *Let G be a planar graph with Hamiltonian path $p = v_1, \ldots, v_n$. There exists a planar graph G' with $\Theta(|V(G)|)$ vertices, such that G is an induced subgraph of G' and each Hamiltonian path in each triangulation of G' visits all vertices of G in the same order (up to the reverse) as p.*

Proof. Given a planar graph G with a Hamiltonian path $p = v_1, \ldots, v_n$, we fix the order of the vertices of path p in any Hamiltonian path of a graph G' as shown in Figure 2.11. We add a copy of R from Proposition 2.7 to v_1 and one copy to v_n by identifying v_1 and v_n with one of the points of the outer triangle. For each edge $v_i v_{i+1}$ of p we glue a copy of P from Proposition 2.7 onto the edge by identifying v_i and v_{i+1} with two vertices from the outer triangle of P. We call this new graph G'.

By Proposition 2.7.2 the two green triangles contain the end vertices of each Hamiltonian path p' of G'. Since the vertices of those triangles are traversed consecutively by p' the vertices v_1 and v_n are the first and last vertex of p' restricted on the vertices of G. When the path p' traverses v_1 it enters the copy of P that is glued on the edge $v_1 v_2$. Since the vertices of a copy of P are traversed consecutively by Proposition 2.7.1, the vertex v_2 is the second vertex in p' restricted on the vertices of G. Since two copies

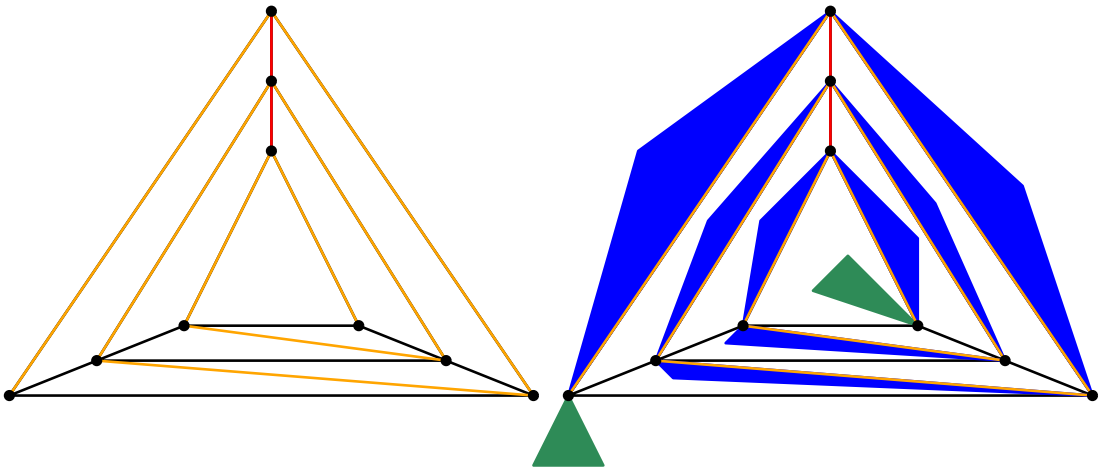

Figure 2.11.: Fixing a Hamiltonian path. The green and blue triangles represent copies of P and R from Figure 2.10. The red edges in the left figure form a chain of bad edges.

of P that are glued onto the edges $v_{i-1}v_i$ and v_iv_{i+1} intersect in v_i, we can iterate this argument to show that the order of vertices of G in p' agrees with the order in p. This concludes the proof of the lemma. □

Theorem 2.9. *There exists a family of graphs G_n with $|V(G_n)| = n$, such that each segment representation of G_n requires $\Theta(n)$ slopes. There exists a distinguished vertex v in G_n, such that $G_n - \{v\}$ is a* GIG.

Proof. Consider the graph G_n with $3n$ vertices that is the triangulation of n nested triangles with the Hamiltonian path spiraling from the outer face to the inner triangle. A subgraph of G_3 is shown in Figure 2.11 left. The graph G_n has a chain of bad edges of length $n-1$, which is indicated by the red edges in Figure 2.11. Constructing G'_n from Lemma 2.8, that fixes the Hamiltonian path, gives a graph that $P^*_{G'_n}$ has a chain of bad edges of length $n-1$ independent of the choice of the Hamiltonian path. By Lemma 2.6 each segment representation of $P^*_{G'_n}$ has a linear number of slopes. Removing the vertex e^* leads to a graph that has a segment contact representation with only two slopes. □

2.4. Computational complexity

With the tools from Section 2.2 we show that the recognition problems for grid-, segment-, and pseudosegment intersection graphs is NP-hard. We do so by applying the following results on the Hamiltonian path and cycle problem in planar triangulations together with Lemma 2.1 and Theorem 2.2.

Theorem 2.10. *The Hamiltonian cycle problem [Wid82, Chv85] and the Hamiltonian path problem [AG11] in planar triangulations is NP-complete.*

Corollary 2.11. *The recognition problem of grid intersection graphs, segment intersection graphs, and pseudosegment intersection graphs is NP-hard.*

We show that minimizing the slope number of a segment intersection graph is NP-hard.

Theorem 2.12. *Computing the slope number of a segment intersection graph is NP-hard.*

The theorem above follows from the following theorem, Lemma 2.1 and Theorem 2.2: Consider planar triangulations G that have a Hamiltonian path. Then the graph P_G^* has a 4-dir representation. It has a 2-dir representation if and only if G has a Hamiltonian cycle. To determine this is NP-complete.

Theorem 2.13. *The Hamiltonian cycle problem in planar triangulations with Hamiltonian path is NP-complete, even if the Hamiltonian path is given.*

The idea to show this theorem is the following: We follow the proof by Chvátal [Chv85] who shows that the Hamiltonian cycle problem in maximal planar graphs is NP-complete. The proof builds upon the reduction of the 3SAT problem to the Hamiltonian cycle problem in planar, cubic bipartite graphs by Akiyama, Nishizeki, and Saito [ANS80]. We apply their reduction on certain 3SAT instances. These instances guarantee to find a Hamiltonian path in the resulting maximal planar graph. The NP-hardness of the Hamiltonian cycle problem in maximal planar graphs with Hamiltonian path follows from the fact that the satisfiability problem on our special 3SAT instances is still NP-hard.

The 3SAT instances we consider are the *almost satisfiable* 3SAT instances. We call a 3SAT instance *almost satisfiable* if there exists a truth assignment, such that all but one clause is satisfied. To my knowledge, almost satisfiable instances have not been considered before.

Lemma 2.14. *The problem of deciding the satisfiability of almost satisfiable 3SAT with given almost satisfying truth assignment is NP-complete.*

Proof. We give a reduction from the 3SAT problem. Given a 3SAT instance we modify all clauses in the following way. We replace each literal in each clause with a new variable, so each variable appears only once. In addition we introduce a variable x and the clause $\overline{x}$. For each pair of variables y_1, y_2, where both variables corresponded to a

literal of the same variable, we add the clauses $x \vee y_1 \vee \overline{y_2}$ and $x \vee \overline{y_1} \vee y_2$ to make them equivalent if they are both non-negated or both negated, or $x \vee y_1 \vee y_2$ and $x \vee \overline{y_1} \vee \overline{y_2}$ if exactly one literal is negated. The constructed 3SAT is satisfiable if and only if the original is: In a satisfying truth assignment x is false, hence the introduced clauses give the equivalence between y_1 and y_2, or their negation respectively. Thus a satisfying instance of the almost satisfiable 3SAT results in a satisfying truth assignment for the original 3SAT. On the other hand, a truth assignment satisfying all but one clause is given by setting x and all other variables true. The size of the new 3SAT is polynomial in the size of the original one. This concludes the proof that almost satisfiable 3SAT is NP-complete. □

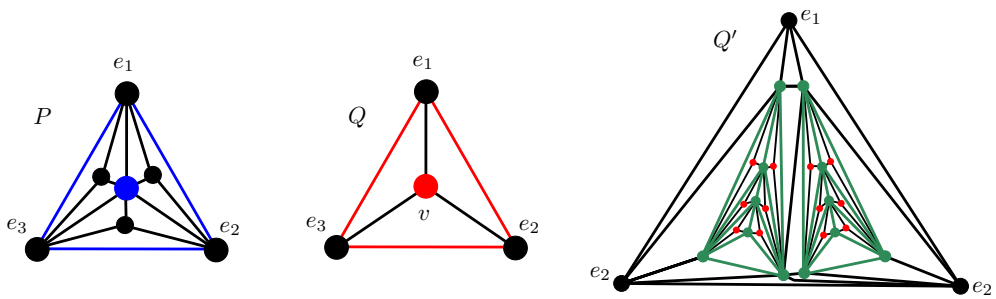

Figure 2.12.: Gadgets used in Theorem 2.13

Proof of Theorem 2.13. In order to show Theorem 2.13 we have to delve into the proof of the NP-hardness of the Hamiltonian cycle problem in planar triangulations. Therefore, we first explain a construction that reduces the Hamiltonian path problem in planar, cubic, bipartite graphs to the Hamiltonian path problem in maximal planar graphs by Chvátal [Chv85]. Wigderson [Wid82] gave a similar but more involved construction to reduce the Hamiltonian cycle problem in planar, cubic (and not necessarily bipartite) graphs to the Hamiltonian cycle problem in planar triangulations. In a given planar, cubic, bipartite graph $G = (A, B; E)$ we subdivide each edge. We glue a copy P_a of the graph P shown in Figure 2.12 on each vertex $a \in A$. Here the middle vertex represents a and the three vertices e_1, e_2, e_3 of P are identified with the subdivision vertices of the three edges incident to a. The vertices of B are replaced in the same way with a copy of the graph Q that is shown in Figure 2.12. The graph we constructed is denoted by G'.

Claim 2.15. *For each triangulation G'' of G' the following holds: The graph G'' has a Hamiltonian cycle if and only if G has a Hamiltonian cycle.*

Proof. We first observe that a Hamiltonian cycle in G'' does not traverse two vertices of B without traversing a vertex of A in between. This follows from Proposition 2.7.1:

Consider two vertices b_1 and b_2 of B in a Hamiltonian cycle in G''. Consider one of the two paths between b_1 and b_2 that are given by the Hamiltonian cycle. At the vertex of the path leaves the copy Q_{b_1} of Q it also enters the vertex of a copy P_a of P. By Proposition 2.7 the path has to enter this copy and thus traverses $a \in A$ before b_2.

The similar property holds for vertices of A. Assume two vertices a_1 and a_2 of A are traversed by the Hamiltonian path without a vertex of B in between. Because the graph G is regular we know that $|A| = |B|$. If a_1 and a_2 are traversed without a vertex of B in between then there are two vertices of B traversed without a vertex of A in between. This is a contradiction as shown before.

With those observations we can show that the edges that were added to triangulate G' are not used in a Hamiltonian path: Assume such an edge is used. Both incident vertices of this edge belong to a copy of P. This implies that the corresponding vertices of A are visited consecutively, a contradiction. □

Note that replacing one of the copies Q in G' by a copy of Q' leads to a graph that has a Hamiltonian path if and only if G' has a Hamiltonian cycle: The graph Q' contains two copies of the graph R from Proposition 2.7. Each of these copies will contain one end of each Hamiltonian path. The continuation of the path outside of Q' will agree with a Hamiltonian cycle in G'.

With the claim above we can proceed with the proof of the NP-hardness of the Hamiltonian cycle problem in planar, cubic, bipartite graphs by Akiyama, Nishizeki, and Saito [ANS80]. Their reduction is a modification of a NP-hardness proof of the Hamiltonian cycle problem in planar, cubic, 3-connected graphs by Garey, Johnson and Tarjan [GJT76]. It involves some gadgets which gives conditions on the edges a Hamiltonian

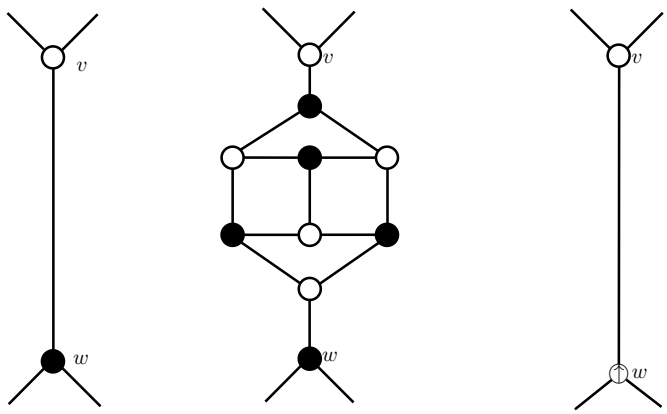

Figure 2.13.: The edge vw is modified to be traversed by each Hamiltonian cycle. The abbreviation of this gadget using the ⇧-symbol is shown on the right.

path in the graph uses. For example the construction in Figure 2.13: Adding the graph

on the edge forces the any Hamiltonian cycle to use this edge (,just as subdividing the edge would without the requirement that the graph has to be cubic and bipartite). This gadget is symbolized by the ⊕-symbol on one neighbouring vertex as shown in the left part of Figure 2.13. The graph in Figure 2.14 is a gadget that forces each Hamiltonian

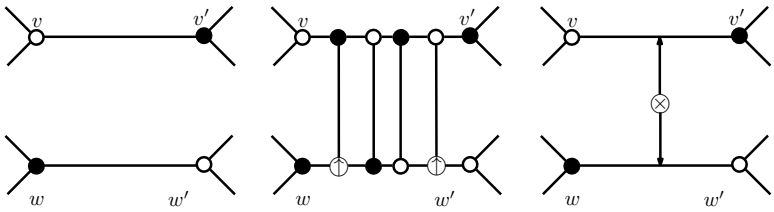

Figure 2.14.: The edge vw is modified to be traversed by each Hamiltonian cycle. The abbreviation of this gadget using the $\otimes$-symbol is shown on the right.

cycle to traverse exactly one of the edges vv' or ww' ('exclusive or'). It is abbreviated by the $\otimes$-symbol as shown in the left of the figure. We spare to explain the 'or'-gadget which forces each Hamiltonian path to use at least one of the connected edges. This gadget is symbolized by the ⊕-symbol.

With those gadgets we can explain the reduction of the 3SAT problem to the Hamiltonian cycle problem in cubic bipartite graphs from [ANS80]. The general idea of the

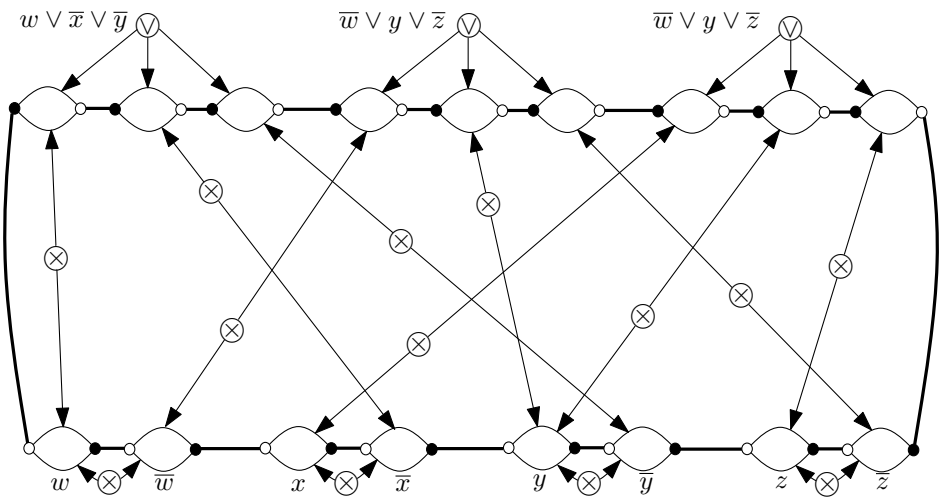

Figure 2.15.: A graph which has a Hamiltonian cycle iff the 3-SAT $(w \vee \overline{x} \vee \overline{y}) \wedge (\overline{w} \vee y \vee \overline{z}) \wedge (\overline{w} \vee y \vee \overline{z})$ is satisfiable.

construction is the following. Given a 3-SAT instance Φ we construct the following planar graph G_Φ. In the lower part of the graph in Figure 2.15 a path of length $4n$, where n is the number of variables in the 3SAT. Every second edge in this path is a double edge. The encoding of the truth assignment of the variables is represented by

the following choice of the Hamiltonian path: The double edges are paired along the path. Each pair corresponds to a variable. If the Hamiltonian cycle uses the upper edge on the first double edge then the corresponding variable is set true, if the upper edge of the second double edge is chosen then the negation is true. An $\otimes$-gadget connecting the lower two of the double edges guarantees that at most one of the upper edges is traversed by a Hamiltonian cycle. The upper part of the Figure 2.15 is used to construct the clauses. The literals of the clauses are represented by double edges. One literal is true if the upper edge is used by the Hamiltonian cycle. The literals of one clause appear consecutively on the path. This way the upper edges can be connected by a $\bigvee$-gadget. If a literal appears positively the lower of the double edges of a gadget is connected to the upper double edge of the positive copy of the variable via a $\otimes$-gadgets. The crossings of the $\otimes$-gadgets can be "planarized" by an idea from [GJT76], which is depicted in Figure 2.16. In the normal $\otimes$-gadget between vv' and ww' the four vertical

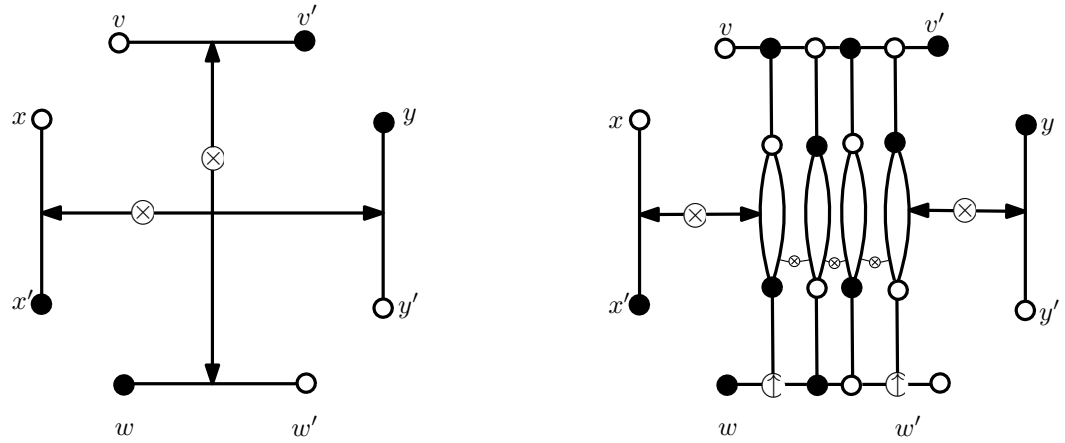

Figure 2.16.: Planarizing crossings of $\otimes$-gadgets.

edges are subdivided twice. The middle edge becomes a double edge. Which of the subdivided edge is taken is copied via an $\otimes$-gadget from xx' and goes through the whole gadget onto yy'. This way we can delete one crossing. Several crossing of the $\otimes$-gadget between vv' and ww' can be resolved via more subdivisions.

The two paths for the variables in the lower part and for the clauses in the upper part are connected to a cycle.

Given a 3-SAT instance Φ, the graph constructed for this instance described above has a Hamiltonian cycle if and only if Φ has a satisfying truth assignment: A Hamiltonian cycle uses exactly one of the lower double edges that correspond to a variable. If the lower part of the negation is used the variable is set true, and false otherwise. The information whether the Hamiltonian cycle uses the lower or upper arc is copied onto the literal double edges in the upper part. The $\bigvee$-gadget can only be traversed by a

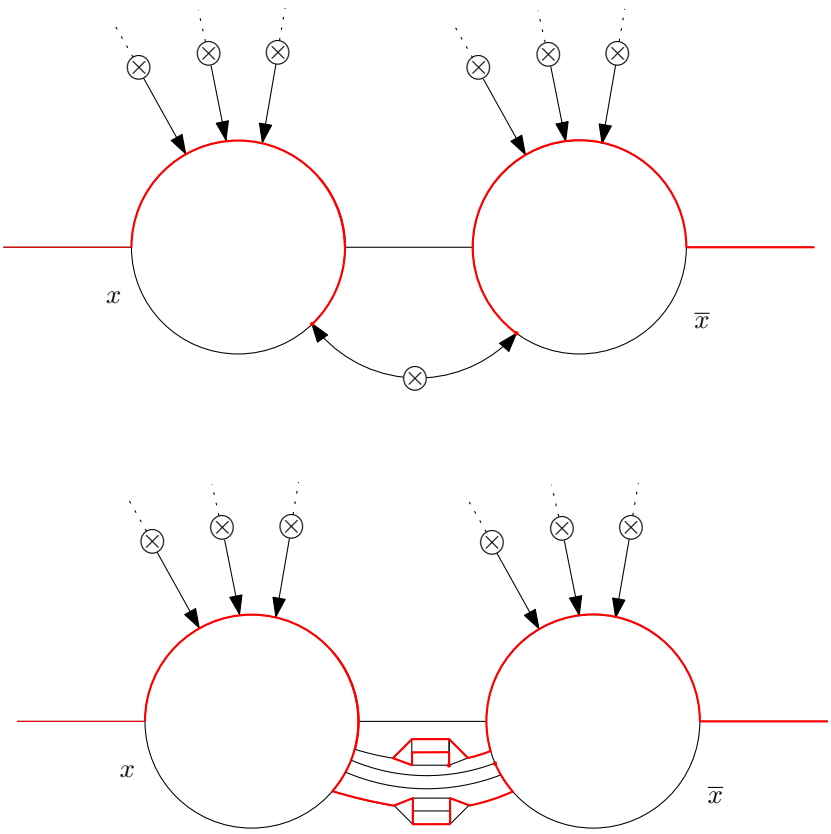

Figure 2.17.: A Hamiltonian path in G_Φ for an almost satisfiable 3-SAT: The variable x is in "superposition".

Hamiltonian cycle if one of the connected literals is true. On the other hand, a truth assignment of the 3-SAT can be translated back to the choice of the upper or lower arc. A satisfying truth assignment respects all the logical connections introduced by the gadgets. This leads to a Hamiltonian cycle of the constructed graph.

With this construction we have all tools to finish the proof of Theorem 2.13. We consider the graphs G_Φ for almost satisfiable 3-SAT instances Φ. Since the satisfiability problem on those 3-SAT instances is NP-complete the Hamiltonian cycle problem on the corresponding graphs is NP-complete. In the following paragraph we show that a graph G_Φ constructed from an almost satisfiable 3-SAT constructed with the method of Lemma 2.14 has a Hamiltonian path.

The way we construct the Hamiltonian path in G_Φ is the following. We take an almost satisfying truth assignment of the 3-SAT. Let x be a variable, such that putting this variable x in "superposition", i.e., x and $\overline{x}$ are true, satisfies Φ. We construct the Hamiltonian path according to the truth assignment with 'superposition' as shown in

Figure 2.17. Now all but one of the logical conditions on the edges, which are constructed by the different gadgets, are satisfied. The condition that is not satisfied is the $\otimes$-gadget between the two lower arcs of the literals x and $\overline{x}$. Using the ends of the Hamiltonian path instead of a cycle we can overcome this problem as shown in Figure 2.17. This concludes the proof. □

2.5. Odd slope number and bipartite graphs

All our examples of bipartite segment graphs used only even slope numbers. So we ask the following natural question: Are there bipartite segment intersection graphs that need an odd number of slopes in a representation with the minimal number of slopes? We answer this question affirmatively by constructing a bipartite 3-dir graph that is not a GIG.

Observation 2.16. *There is a bipartite* 3*-dir graph which is not a* 2*-dir graph.*

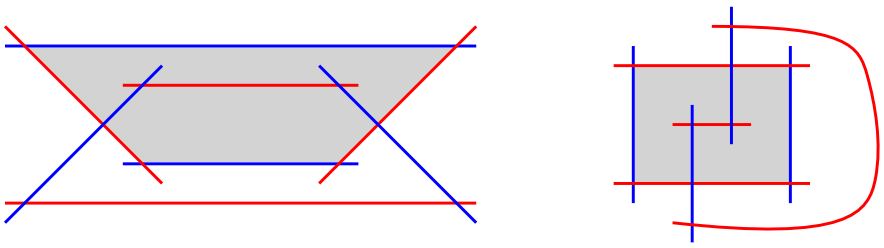

Figure 2.18.: Left: A 3-dir representation that cannot be transformed in a 2-dir representation (GIG).

Proof. We give a construction of a bipartite 3-dir graph that is not a GIG. To construct this graph we apply the order forcing lemma (Lemma 0.6) on the 3-dir graph in Figure 2.18 to obtain a graph G. Then we show that the graph G is not a 2-dir graph (GIG) by showing that the representation in Figure 2.18 left cannot be realized by segments in two directions.

Therefore, assume there is a GIG representation R of G. By the order forcing lemma the representation of the segments in Figure 2.18 is kept in R. Especially, one of the horizontal red segments lies inside the gray region and one outside. Furthermore, one of the blue segments connecting these red segments intersects the upper boundary of the region and one the lower boundary. The outer red segment intersects one blue segment above the gray region and the other one below, which leads to a contradiction as shown in Figure 2.18 right. This concludes the proof that G is not a GIG. □

In contrast to the previous examples none of the directions in the last lemma is monochromatic. We show that this is the case in general by showing that one monochromatic direction in a bipartite 3-dir representation allows us to transform the graph to a grid intersection graph.

Lemma 2.17. *Let G be a bipartite graph with 3-dir representation, such that one direction is monochromatic. Then G is a GIG.*

Proof. We use the 3-dir representation with one monochromatic direction to construct a GIG representation of G. Therefore, we assume without loss of generality that the monochromatic direction is horizontal, and the other two have slopes 1 and -1. This can be assumed since a 3-dir representation R can be transformed by a linear transformation to a 3-dir representation R' that uses three prescribed slopes s_1, s_2, s_3 [ČKNP02]. Let A be the color class using three directions. First we define a linear order on the segments of A by augmenting the segments to piecewise linear, bi-infinite curves that do not intersect and use the three slopes 0, 1 and -1 as shown in Figure 2.19. We call those curves a *laminar family of curves*. The order of intersection of a vertical line with the laminar family gives an order on those curves, which will correspond to a cross-free matrix order of the vertices in A. The existence of the extension of A to a laminar

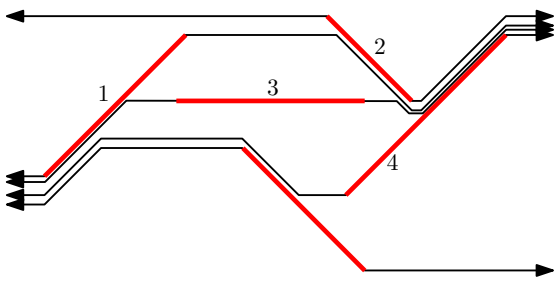

Figure 2.19.: The segments of A extended to a laminar family of curves.

family can be proven as follows. Let ε be the minimal distance between two segments of A. Consider a ball of radius $\varepsilon/2$ at the left endpoint of a segment $a \in A$. We let the ball roll by a horizontal force to the left. The other segments are considered as walls. Since the ball has radius $\varepsilon/2$ and there is no horizontal wall it does not get stuck on its way to the left. Thus the trajectory of the midpoint of the ball gives a piecewise linear curve to the left that consists of horizontal segments and segments that are parallel to the walls, i.e., they have slope 0, 1, or -1. Similarly, the trajectory of a ball rolling from the right endpoint of a to the right extends this curve to a bi-infinite curve. We extend all curves by repeating the procedure for the other segments with a ball of radius $\varepsilon/2^i$ for the i-th segment.

The order of the vertices in B is defined the following way: A segment in B intersects each of the curves in at most one interval. This stems from the fact that the slope of the segments in B is the maximal or the minimal slope of the curves. Consider the directed graph $D = (B, E)$ where $(v, w) \in E$ if and only if v and w intersect the same curve c of the laminar family and the intersection of v with c lies to the left of the intersection of w and c. We show that the graph D is acyclic and any topological ordering of D leads to a cross-free matrix order, which certifies that G is a GIG by Lemma 0.5.

We first show that D is acyclic. Assume the D is not acyclic. Then there is a shortest cycle $(b_1, \ldots, b_k)$. Since a shortest cycle has no chord there is no curve of the laminar family that intersects more than two segments corresponding to elements that lie in the cycle. This implies that $k \leq 2$, since k segments imply the existence of k curves in the laminar family, each of them intersecting exactly two segments. Since each segment intersects consecutive curves and there are only $k - 1$ consecutive pairs of curves this leads to a contradiction for $k \geq 3$. For $k = 2$ we get a contradiction as shown in Figure 2.20: Two segments b_1 and b_2 both intersect the curves c_1 and c_2. The intersection of b_1 on c_1 lies to the left of the intersection of b_2 and c_1 and vice versa on c_2. Then the segment b_1 separates the intersection points of b_2 with c_1 and c_2 in the region between c_1 and c_2. Because b_1 and b_2 do not intersect and b_2 intersects both curves in at most one interval, b_2 cannot connect both of its intersection points on c_1 and c_2. We consider a topological ordering of D. The bipartite adjacency matrix M

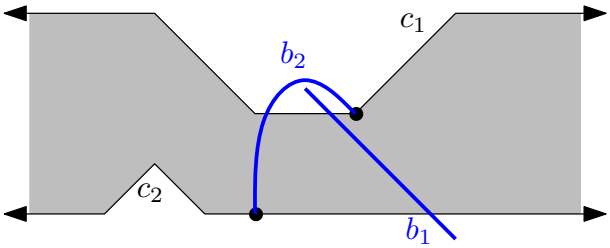

Figure 2.20.: The graph D has no 2-cycle.

ordered by this topological ordering for B and the order of the curves of the laminar family is the bipartite adjacency matrix of a GIG. To show this, assume there is a cross matrix in M. This implies that there are three curves c_1, c_2, c_3 of the laminar family in this order. Two segments $b_1, b_3 \in B$ intersect c_2 in the interval of the segment on c_2. In addition, there is a segment b_2 that intersects c_1 and c_3, and consequently also c_2. The intersection of b_2 with c_2 lies between the intersection points of b_1 and b_3 with c_2, which shows that b_2 also intersects the segment on c_2 as shown in Figure 2.21. This contradicts the non-adjacency according to M. □

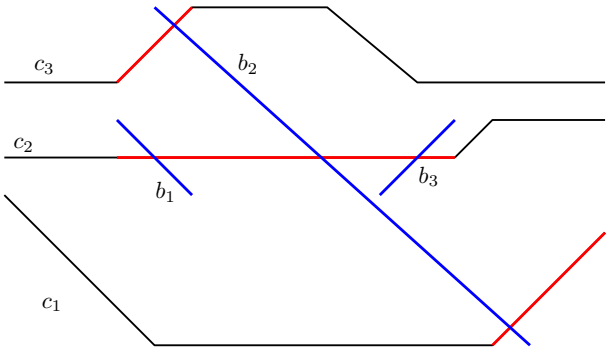

Figure 2.21.: The adjacency matrix is cross-free.

Part II.

Realizability problems and the existential theory of the reals

3. Realizability of circular sequences

In this Chapter we sketch and simplify the idea of a proof of Mnëv's universality theorem [Mnë88] for order types. We extend the theorem for circular sequences.

Theorem 3.1. *[Mnëv's universality theorem] Let S be a semialgebraic set defined by a polynomial inequality system with integer coefficients. Then there exists an order type O, such that S is stably equivalent to the realization space of O.*

The order type O of the theorem above can be constructed from the description of the semialgebraic set in form of an (in)equality system. The proof of Theorem 3.1 leads to a polynomial time algorithm to construct the order type O from the description of S given by the polynomial inequality system. This leads to the following complexity result for the problem of deciding the realizability of an order type.

Corollary 3.2. *The problem of deciding the realizability of an order type is complete in $\exists\mathbb{R}$.*

We first give an overview over the results on circular sequences in Section 3.1. Then we show how to implement and store arithmetic operations in order types in Section 3.2. Afterwards, we give a modified version of the $\exists\mathbb{R}$-reduction for the order type realizability problem based on work by Shor [Sho91], Richter-Gebert [RG95] and Matoušek [Mat14] in Section 3.3. These three papers are simplifications of the proof by Mnëv [Mnë88]. With the help of our modification we prove that the realizability of circular sequences is hard in $\exists\mathbb{R}$, even if the order type is known to be realizable.

3.1. Background

Circular sequences have been studied by Perrin [Per81] (1881), who claimed that each circular sequence defining a certain set of axioms is realizable. This claim was proven false by Goodman and Pollack [GP80] by showing that the circular sequence of a point set of five points in convex position, the *bad pentagon* shown in Figure 3.1, is not realizable. Pilz [Pil14] considered the complexity of the realizability of a circular sequence. He notices that the NP-hardness proof of Shor [Sho91] does not generalize to circular sequences. We show $\exists\mathbb{R}$-hardness, even if the order type is realizable.

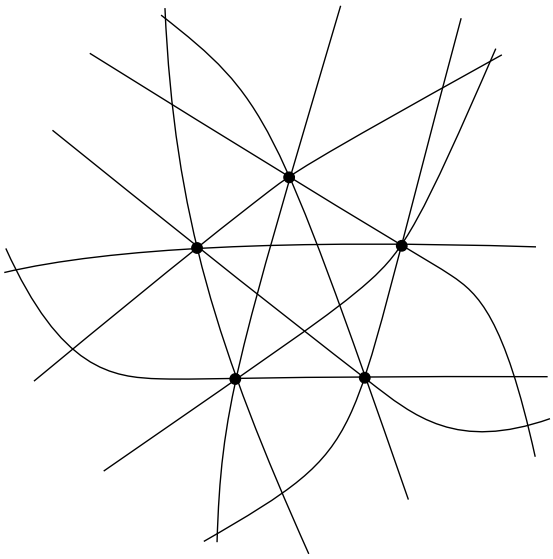

Figure 3.1.: The bad pentagon of Goodman and Pollak.

3.2. Arithmetics with order types

In order to reduce the order type realizability problem to solvability of a system of strict polynomial inequalities, we simulate arithmetic operations with order types. This uses standard constructions introduced by von Staudt in his "*algebra of throws*" [Sta47]. The constructions are well described in [RG11].

To carry out arithmetic operations using orientation predicates, we associate numbers with points on a line ℓ. We use the cross-ratio to encode the values in the following way: We fix two points on a line and associate them with 0 and 1. On the line through those points we call the point at infinity ∞. For a point x on this line the cross-ratio $(x, 1; 0, \infty)$ results in the distance between 0 and x scaled by the distance between 0 and 1. Because the cross-ratio is a projective invariant we can fix one line and use the point x for representing its value. In this way, we have established the coordinates on one line.

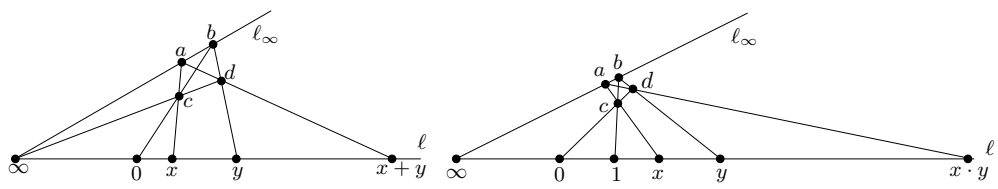

Figure 3.2.: Gadgets for addition (left) and multiplication (right) on a line.

For computing on this line we use the gadgets for addition and multiplication that are

depicted in Figure 3.2.

We first describe the case of the addition. By using a projective transformation we assume that the line ℓ_∞ is in fact the line at infinity, i.e., the lines intersecting in a are parallel and the lines intersecting in b are parallel. In addition, the line $\ell(c, d)$ is parallel to the line ℓ. With these observations we can conclude that the distance of the points c and d is the same as the distance between 0 and y, and that the distance between c and d is the same as the distance between the points x and $x+y$. Thus we can conclude that the gadget simulates the addition of the points if ℓ_∞ is the line at infinity. This is stored in the cross-ratio $(x + y, 1; 0, \infty)$, which is invariant under projective transformations. Thus the gadget simulates the addition in each realization.

Secondly, we describe the multiplication gadget. We are given the points $\infty, 0, 1, x$ and y in this order on the line ℓ We construct a point on ℓ that represents the value $x \cdot y$. Therefore, we take a second line ℓ_∞ that intersects ℓ in ∞, and two points a and b on ℓ_∞. Construct the segments $\overline{by}, \overline{b1}$ and $\overline{ax}$. Denote the intersection point of $\overline{ax}$ and $\overline{b1}$ by c. Call d the intersection point of $\overline{by}$ and $\ell(0, c)$. The intersection point of ℓ and $\ell(d, a)$ represents the point $x \cdot y =: z$ on ℓ, i.e., $(z, 1; 0, \infty) = (x, 1; 0, \infty) \cdot (y, 1; 0, \infty)$. In a projective realization of the gadget in which the line ℓ_∞ is indeed the line at infinity, the result can be obtained by applying twice the intercept theorem, in the triangles with vertices $0, d, y$ and $0, d, z$, respectively.

3.3. The reduction for circular sequences

Using the gadgets for arithmetic operations above, we can model a system of strict polynomial inequalities. However, it is not clear how we can determine the complete order type of the points on ℓ without knowing the order of the variables in solution of the system. Circumventing this obstacle was the main achievement of Mnëv [Mnë88]. We cite one of the main theorems in a version by Shor [Sho91].

Theorem 3.3. *Every primary semialgebraic set $V \subseteq \mathbb{R}^d$ is stably equivalent to a semialgebraic set $V' \subseteq \mathbb{R}^n$, with $n = \mathrm{poly}(d)$, for which all defining equations are of the form $x_i + x_j = x_k$ or $x_i \cdot x_j = x_k$ for certain $1 \leq i \leq j < k \leq n$, and all inequalities are given by $1 = x_1 < x_2 < \cdots < x_n$.*

From the computational point of view, the important property is that V is the empty set if and only V' is, and that the size of the description of V' in the theorem above is polynomial in the size of the description of V. We call the description of a semialgebraic set V' given in the theorem above the *Shor normal form*.

With Theorem 3.3 we have a tool to order the points on the line we calculate on totally. We give the idea of a slightly different method to give such a total order on

the points on this line by Richter-Gebert [RG95]. For details we refer to Matoušek's manuscript [Mat14]. We start with possible assumptions on the form of the description the semialgebraic set.

1. The values of all variables can be assumed to be larger than 1. This can be achieved by replacing each variable x_i by $x_i^+ - x_i^-$, where x_i^+ and x_i^- are new variables which can be assumed to be larger than 1.

2. All coefficients are positive. The sign of a coefficient can be changed by bringing the term on the other side of the inequality.

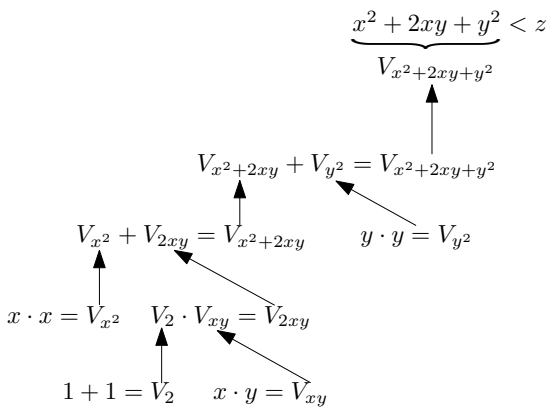

Figure 3.3.: Decomposing a formula in elementary operations.

Now we decompose both sides of the (in)equalities recursively into two subformulas as shown for the left-hand side of the equation in Figure 3.3. The result of each subformula f is stored in a new variable V_f.

The resulting decomposition tree gives an order on the variables that lie on directed paths, but not a total order on the variables. To obtain a total order we use the idea of Richter-Gebert [RG95] to introduce several *scales*. We take a topological ordering of the constructed decomposition tree. In each node of the tree we realize one calculation, an addition or a multiplication. We realize this calculation using the order types in the following way. First assume the first $i-1$ calculations of the topological ordering are already realized on the horizontal line ℓ. For calculation i, we introduce a new *scale* S_i. In this scale, we add a new point 0_i to the right of all already existing points. This point takes the role of the point 0 in scale i. We now *copy* the *input variables* of gadget i into the scale using the addition gadget: If the variable V_x is used for a computation in S_i already exists in a scale S_h with $h < i$, we add the difference between 0_h and V_x onto 0_i (using an addition gadget) to create a copy of V_x in scale S_i. This guarantees that variable x has the same value in each scale. After copying all required variables

into the scale S_i we implement the multiplication or addition using the known gadgets to construct the output variable of the gadget V_z. With this method we fix a partial order of the points on ℓ. The only point where we cannot give a total order are the two input variables in one scale, because it introduces an additional comparison of two variable. This may restrict the set of solutions of the inequality system. (The output variable is certainly larger since both input variables are assumed to be larger than 1.) We overcome this last obstacle by using the modified (relabeled) addition and multiplication gadgets shown in Figure 3.4. With these modified gadgets we can

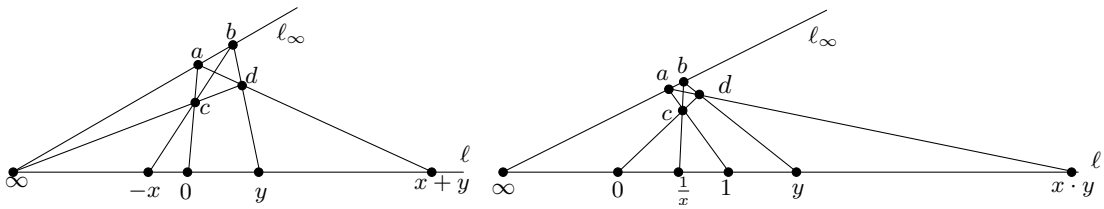

Figure 3.4.: Inverted gadgets: Computing $x + y$ from $-x$ and y (left), and $x \cdot y$ from $1x$ and y (right).

implement the computations without comparing x and y in one scale by calculating $x + y$ from $-x$ and y, and $x \cdot y$ from $1/x$ and y. The only steps that have to be added in between is to calculate $1/x$ for each first input variable of a multiplication gadget and $-x$ for the first input variable of an addition gadget. We do this in additional scales using the slight modifications of the addition and multiplication gadgets shown in Figure 3.5.

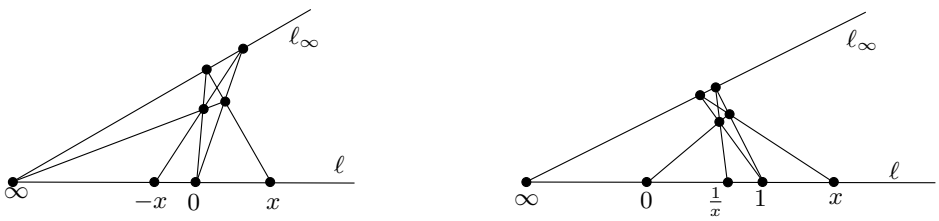

Figure 3.5.: Computing $-x$ (left) and $\frac{1}{x}$ (right).

With this construction we avoid comparisons between points on ℓ that restrict the realization space. Our new set of variables corresponds to the points on ℓ. The total order of the variables is given by the order on ℓ. Note that we did not prove Theorem 3.3, since our calculation- and copy-gadgets do not correspond exactly to standard addition and multiplication using the "original" 0. Shor [Sho91] obtains the normal form by specifying a *shift* of the value of a variables more exactly. For example, we can compute with variables V_{x+A} and V_{y+A} to obtain the result of $x \cdot y$ from $V_{xy+xA+Ay+A^2}$ if A is another variable. We refer to Shor for an exact proof of Theorem 3.3.

We obtained a *partial* order type for a semialgebraic set V that is realizable if and only if V is nonempty. The order type is not specified completely since we did not give the orientation of the triples involving points from different gadgets. A complete order type that is realizable if and only if V is nonempty can be specified using the following two observations.

Observation 3.4 ([RG95],[Mat14]). *The points a_i and b_i can be positioned arbitrarily on ℓ_∞. The position of the other points of P_i is fully determined by a_i, b_i and the input values on ℓ.*

Observation 3.5 ([RG95],[Mat14]). *All points of P_i are placed close to a_i if a_i and b_i are placed close to each other. (For each $\varepsilon > 0$ there exists a $\delta > 0$, such $|a_i - b_i| < \delta$ implies $|p - q| < \varepsilon$ for all $p, q \in P_i$.)*

The observations allow to place the points a_i and b_i of gadget g_i "close" together on ℓ_∞ and with "fast" increasing x-coordinates, such that the orientation of triples that belong to different gadgets can be fixed without restricting the position of the points on ℓ.

We do not use this construction but we use one line ℓ_∞^i for the gadget g_i as indicated in Figure 3.6. All those lines ℓ_∞^i intersect the line ℓ in ∞. Hence the gadgets still fix the same cross-ratios of the quadruples of points ℓ.

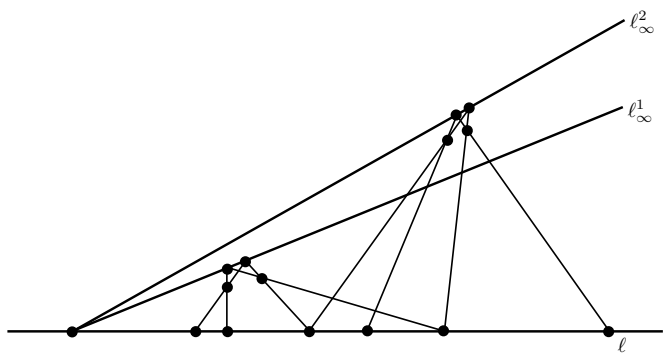

Figure 3.6.: Realizing the calculations with order types using several "lines at infinity".

We are now ready to prove the hardness of the realizability of circular sequences..

Theorem 3.6. *Let V primary semialgebraic set. There is a circular sequence C such that C is realizable if and only if V is nonempty.*

Proof. Given a primary semialgebraic set V in Shor normal form we produce a circular sequence C (that implies an order type O) that is realizable if V is nonempty.

We give an iterative construction of the circular sequence C. Starting with the points on ℓ, we construct the circular sequence C iteratively by adding the points of the

gadget g_i, that implements the arithmetic operations described in the Shor normal form.

We describe the circular sequence by the order of slopes of the lines spanned by two points of the order type.

We start the construction with the points on ℓ placed on a horizontal line, such that ∞ lies on the line at infinity. The points a_i and b_i of the gadgets g_i will be placed on horizontal lines ℓ^i_∞ with increasing y-coordinate. The slopes through points of g_i will be the smallest positive slopes spanned by points that are already placed so far.

For the exact construction, we assume that we know the lexicographic order $\prec$ of points P_{i-1} that are already placed after placing gadget g_{i-1} by decreasing y- and increasing x-coordinate. We place the points a_i and b_i on a horizontal line ℓ^i_∞ that lies above all points placed so far. We move a_i and b_i on ℓ^i_∞ to the right, such that all lines through a_i or b_i have a smaller slope than the lines with positive slope that are spanned by two points in P_{i-1}. This property implies that a_i and b_i see the points of P_{i-1} in the order defined by $\prec$. This is the case if a_i and b_i lie in the gray region in Figure 3.7. Afterwards we move the points a_i and b_i close together, such that the points c_i and d_i

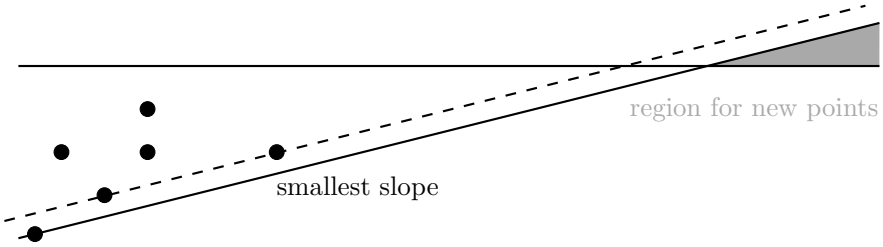

Figure 3.7.: Determining a region for the next gadget.

also lie above each point of P_{i-1}, which is possible by Observation 3.5. Furthermore, we can assume that the points of g_i lie close enough together, such that the slopes of through one fixed point $p \in P_{i-1}$ and the points of g_i appear consecutively among all slopes through a pair of points of P_i. To give the complete order of slopes we only have to give the order of the slopes of lines in $L_x := \{\ell(x, p_i) \mid p_i \in g_i\}$, the lines though one fixed point $x \in P_i$ and all the points of g_i. This order depends on the position of the point x relative to the points corresponding to the variables connected by gadget g_i. The order of slopes of the lines in L_x is indicated in Figure 3.8 and depends on the position of x relative to the lines spanned by the points of the gadget.

The order $\prec$ of P_{i-1} can be extended to an order of the points of P_i by adding the prefix $a_i \prec b_i \prec c_i \prec d_i$ if g_i is an addition- or copy-gadget and $a_i \prec b_i \prec d_i \prec c_i$ if g_i is a multiplication gadget.

By placing all the gadgets iteratively we obtain the order of switches/slopes that give an order type that realizes all the gadgets. Thus, from the cross-ratio of the points on ℓ

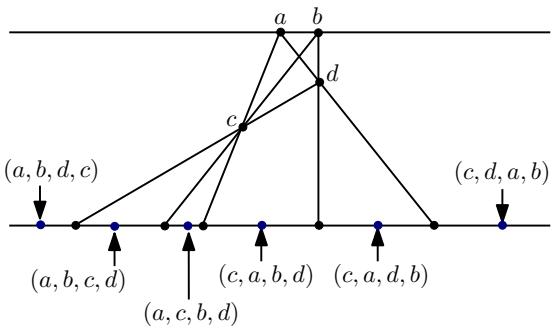

Figure 3.8.: The clockwise rotation of the lines through x (blue) and the different points of a gadget is indicated by the tuples.

in a realization of C we can compute a point in the semialgebraic set V. On the other hand, a point x in V leads to a realization of C by placing the points on ℓ such that they encode x in the cross-ratio and repeating the construction for the positioning of the gadgets above. Consequently, C is realizable if and only if V is nonempty. □

3.4. Realizable order types with non-realizable circular sequences

In this section we strengthen this result and show that deciding the realizability of a circular sequence is hard in $\exists\mathbb{R}$, even if the order type is known to be realizable.

Theorem 3.7. *Deciding the realizability of a (non-simple) circular sequence that induces a realizable order type is complete in* $\exists\mathbb{R}$.

Proof. We prove this theorem by a reduction of the solvability of a strict polynomial inequality system to the realizability of the circular sequence. The idea to prove this theorem is to make the order type constructed above realizable by removing the point ∞, the intersection point of ℓ and ℓ_∞. The orientations of the triples including ∞ will be forced by the circular sequence.

We will slightly modify the order type constructed in the proof of Theorem 3.6 by not using the Shor normal form directly. Instead we plug in the construction by Richter-Gebert, which we sketched as proof of the Shor normal form. This means we decompose the calculations in single computations. Each computation is done using one gadgets in a single scale. The order of the scales on the line ℓ is the following, see Figure 3.9. First we add a scale for each variable. Afterwards we add the scales for the computations in the order they appear in the decomposition of the formulas in single computations, i.e., from the leaves to the root in Figure 3.3. After the computation scales we add a

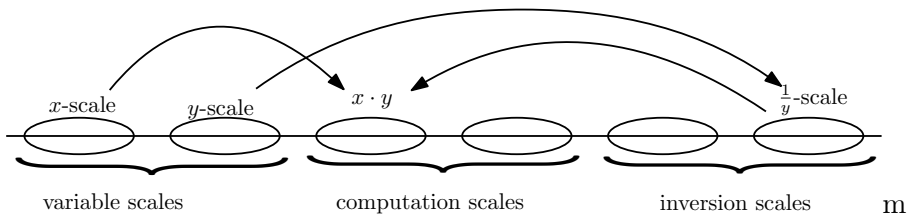

Figure 3.9.: The order of the scales.

scale for each inverse of a variable that is required in a multiplication gadget. We copy the original variables in these scale and invert them in this scale. The input variables are copied for an addition gadget and the second input variable of a multiplication gadget are copied from the variable scales on the left. The first input variable of a multiplication gadget $(1/x)$ is copied from the right.

The gadgets that realize the computations are placed as in the proof of Theorem 3.6, i.e., the slopes spanned by a points of the gadgets g_i are the smallest positive slopes that are spanned by points on ℓ and the points of the gadgets $g_1, \ldots, g_i$, and each tuple of points that lies on a common line with the point ∞ in the construction of Theorem 3.6 lies on a horizontal line. In Lemma 3.8 we show that the order type induced by the circular sequence is realizable. Thus, it remains to show that this non-simple circular sequence is realizable if and only if V is nonempty.

Therefore, note that all "horizontal" lines are in fact parallel if the circular sequence is realizable. Thus we can add the intersection point of all horizontal lines ∞ if the circular sequence is realizable. With the point ∞ the functionality of the gadgets is guaranteed and we obtain a point in the semialgebraic set V from the cross-ratios of the points on ℓ. On the other hand, if V is nonempty we can encode a point $x \in V$ by the cross-ratios on ℓ with the point ∞ as in Theorem 3.6. The gadgets do not have to be manipulated because the coordinates of the point on ℓ have the right cross-ratios, the lines ℓ^i_∞ intersect in the common point ∞ on ℓ. Mapping the point ∞ onto the line at infinity leads to a realization where the "horizontal" lines are parallel. The remaining slopes can be adjusted as in the proof of Theorem 3.6 by moving the points of the gadget g_i to the left on their horizontal lines. □

It remains to show that the order type induced by the circular sequence in Theorem 3.7 is realizable even if the circular sequence is not realizable. This is done in the following Lemma.

Lemma 3.8. *The order type described in Theorem 3.7 is realizable.*

Proof. We give a realization of the order type constructed in Theorem 3.7. An approximate placement of the points is sketched in Figure 3.10. The construction is the

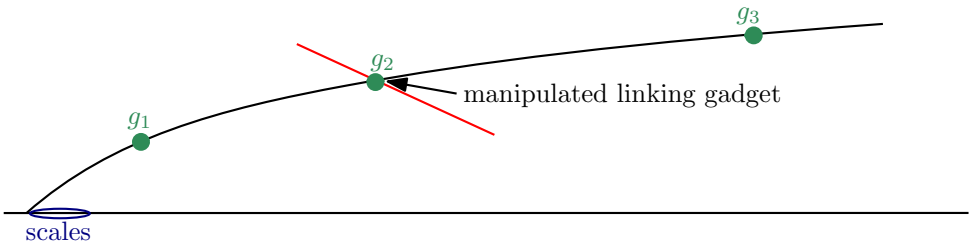

Figure 3.10.: The realization of the order type. In red: the "horizontal" line of a manipulated linking gadget.

following.

Each variable will obtain the value 9/2. While copying the variables into a different scale, we guarantee that the input variables in this scale have value 3/2 and 3 by using the manipulated linking gadget in Figure 3.12. Since the equations $3/2 \cdot 3 = 9/2$ and $3/2 + 3 = 9/2$ hold, we do not have to modify the calculation gadgets of the scales. We give coordinates for the points of the scales. The point 0_i, the point that represents the 0 of scale i is placed on the coordinate $(10 \cdot i, 0)$. If scale S_i is a relabeled addition scale, place the points on the coordinates $-x = (-3/2 + 10 \cdot i, 0)$, $y = (3 + 10 \cdot i, 0)$ and $x + y = (9/2 + 10 \cdot i, 0)$. If scale i is a relabeled multiplication scale we place the points on the coordinates $1/x = (2/3 + 10 \cdot i, 0)$, $1 = (1 + 10 \cdot i, 0)$, $y = (3 + 10 \cdot i, 0)$ and $x \cdot y = (9/2 + 10 \cdot i, 0)$. Finally, in an inversion scale, the variables are placed on $x = (3 + 10 \cdot i, 0)$ and $1/x = (1/3 + 10 \cdot i, 0)$. After placing the points on ℓ, we determine

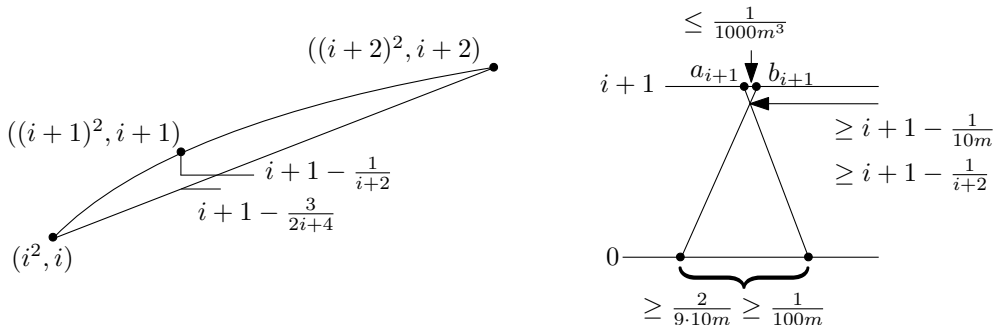

Figure 3.11.: Left: The placement of three consecutive a points of gadgets: realizing the points of the gadget g_{i+1} with y-coordinate larger than $i + 1 - \frac{1}{i+2}$

.

suitable positions for the gadgets. Therefore, we first scale the representation to assume that the points on ℓ lie in the interval $[0, 1]$. We place the points of the gadget g_i in a small region around the point (i^2, i). If all the points of the gadget g_i have a y-coordinate larger than $i - 1/(g + 2)$, where g is the number of gadgets, this realization realizes the order type. This can be seen from Figure 3.11 left. For this it is enough to place a_i and

b_i in a distance smaller than $1/(1000m^3)$, where m is the number of scales, as indicated in Figure 3.11 right using the interception theorem. Now we only have to show that

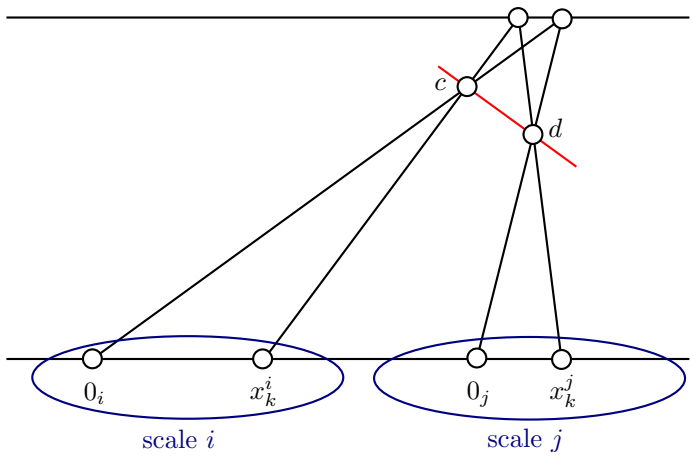

Figure 3.12.: A manipulated linking gadget where the value in the left scale is larger than the value in the right scale.

the orientation of triples with two points from one gadget and one point from another gadget are correct.

Here, we make the following observation: Each linking gadget links a larger variable on the left to a smaller variable on the right. This follows from the fact that the variables of value $9/2$ are linked from the left to points of distance $3/2$ or 3 from 0_i to the right, and the inverse variables are linked from $2/9$ on the right to $1/3$ on the left. This has the effect that the points c_i and d_i of a *manipulated* linking gadget g_i lie on a line which is rotated clockwise from the original horizontal line as shown in Figure 3.12. This line of negative slope still separates the points of the later gadgets g_j with $j > i$ from the gadgets with smaller index as shown by the red line in Figure 3.10. □

We have shown that the realizability problem for non-simple circular sequences that induce a realizable order type is hard in $\exists\mathbb{R}$. We now show that this result even holds for simple circular sequences.

Theorem 3.9. *The realizability problem for simple circular sequences with given realization of the induced order type is $\exists\mathbb{R}$-hard.*

For the proof of the theorem above we need the following definition. An abstract order type P is *constructible* if there exists a ordering of the points $(p_1, \ldots, p_n)$, such that

1. no three points among $p_1, \ldots, p_4$ are collinear, and
2. each p_i lies on at most two of the lines spanned by the points $p_1, \ldots, p_{i-1}$.

We describe the basic idea of how to construct a simple order type from a constructible order type, such that the simple order type is realizable if and only if the constructible one is realizable. As shown in Figure 3.13, a simple order type is constructed by re-

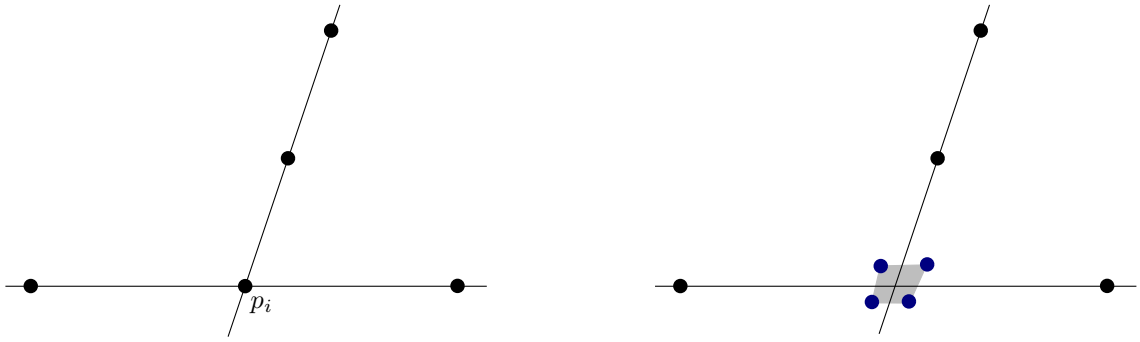

Figure 3.13.: Replacing p_i, that lies on two lines, by four points that reserve a convex region around the intersection point of the two lines.

placing each point p_i (starting from $i = n$), that lies on two lines spanned by points in $\{p_1, \ldots, p_{i-1}\}$, with points in convex position, which enclose the original point p_i and behave "similar" to the point p_i in the order type. From the new order type we can reconstruct the old one by placing the point p_i on the intersection points of the two spanned lines. The order type with the new points encodes, that the intersection point of the two lines lies inside the convex region C_i. The new order type constructed this way is realizable if and only if the old order type that contained p_i is realizable: The convex region C_i in the new order type is empty and thus the point p_i can be placed on the intersection point of the lines p_i lies on. This is the step where the constructability of the order type is necessary: The new order type encodes that the pairwise intersection of the lines lie in the convex region C_i. Since the order type is constructible there are at most two lines that determine the position of p_i, which allow to place p_i in the convex region. A non-constructible order type forces one p_i to be placed on the pairwise intersection points of three lines in C_i. These three points might not coincide in each realization.

Applying this replacement iteratively to the point p_i for $i = n, \ldots, 5$ leads to a sequence of constructible order types $P = P_n, \ldots, P_4 = \tilde{P}$, where P_t is the order type before replacing point p_t by points that span a convex region. In order type P_t all collinearities of at least three points appear in the first t points of the sequence of points shows the constructability. Since the first four points of a constructible order type are in general position the order type $\tilde{P} = P_4$ is simple.

With the following observation we obtain the $\exists\mathbb{R}$-hardness for the realizability of simple order types.

Proposition 3.10 ([Mat14]). *The order type constructed in the proof of Theorem 3.6 is constructible.*

Proof. We give an order that certifies that the order type in Theorem 3.6 is constructible. Therefore, let $g_1, \ldots, g_k$ be an order of the gadgets, that respects the order of the calculations from Figure 3.3. We give an iterative construction by adding the points of the gadgets iteratively to the sequence P. Assume the input points of the gadget g_i are already constructed. We append the points of g_i in the order of the labels shown in Figure 3.14 to the sequence of points. We observe that none of the points of the gadget lies on more than two lines when the point is constructed, thus Item 2 is satisfied by the sequence. □

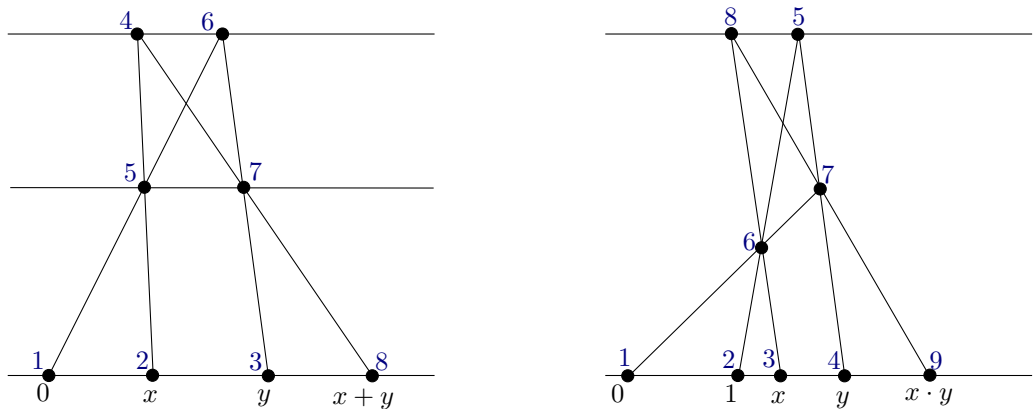

Figure 3.14.: The constructability sequence of points within one gadget.

As a corollary from the above proposition, we obtain the $\exists\mathbb{R}$-completeness of simple order type realizability.

Corollary 3.11. *Deciding the realizability of simple order types is $\exists\mathbb{R}$-complete.*

To apply a similar construction to the non-simple circular sequence, we need a slightly stronger condition than the constructability of the order type to overcome the following problem: If the points p_i and p_j lie on same parallel class of lines (which is encoded in the non-simple circular sequence) as another line that is already constructed, then the line through p_i and p_j is also already constructed. In other words, the common point of the parallel class on the line at infinity is already constructed. Since the only parallel class of lines in the circular sequence we constructed in Theorem 3.6 are the horizontal lines and the point ∞ is constructed in the beginning, this is not an obstacle for us. Hence we are ready to prove Theorem 3.9.

Proof of Theorem 3.9. We reduce the realizability problem of the circular sequences constructed in the proof of Theorem 3.6 to the realizability for simple circular sequences.

Let $(p_1, \ldots, p_n)$ be a sequence of the points that certifies the constructibility of the order type/circular sequence. We iteratively construct a simple order type $\tilde{O}$ and a circular sequence $\tilde{C}$ that is realizable if and only if O is realizable with circular sequence C. We set $O^{n+1} = O$ and $C^{n+1} = C$, where C is the circular sequence constructed in Theorem 3.6. For $t = n, \ldots, 5$ we apply one of the following steps.

If p_t does not lie in a line spanned by the points in $\{p_1, \ldots, p_{t-1}\}$, we keep p_t.

If p_t lies on two lines spanned by the points in $\{p_1, \ldots, p_{t-1}\}$ we do the following. We replace the point p_t by the set of four points P_t in convex position around p_t as indicated in Figure 3.15. The positions of these points relative to the position p_t and the lines

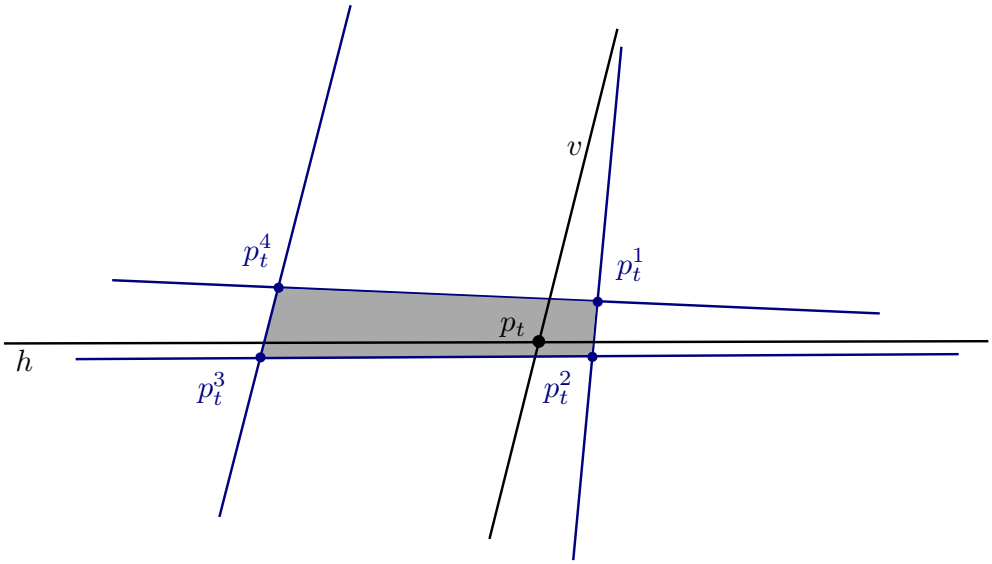

Figure 3.15.: Replacing a point on two lines by point in convex position: The lines defined by the new points are almost parallel to v or h

v and h are indicated according to the following rules, which indicate the order of the slopes through the new points.

- The points below h are very close to h compared to the points above h.
- The points to the right of v lie very close to v compared to the points on the left side of v.

The first item has the effect that the absolute value of the slope of a line through a point on h and a point of P_t that lies above h is larger than any slope of a line through any point on h and a point of P_t that lies below h if h is assumed to be horizontal. For

example the slope of the orange line in Figure 3.15 is closer to the slope of h than the slope of the brown line is. By drawing the region very small, the slopes of lines through the points on h and the points of P_t the slopes through a point of P_t and a point on h are the slopes closest to h. With this we can give the total order of the slopes of all lines of a point on h and a point of P_t which are larger than the slopes of h (and by symmetry of all lines that have a slope smaller than h): A point on h that lies to the left of p_t has smaller absolute value of the slopes with the points of P_t if it lies further away from p_t, and the points on the left produce smaller positive slopes than the points on the right. The slopes of the line $\ell(q, p_t)$, where q lies in one of the quadrants formed by v and h is replaced by the slopes of the lines through P_t. The order of the slopes of the lines $\ell(q, p)$ with $p \in P_t$ depends on the quadrant of q. The orders depending on the quadrant are indicated in Figure 3.16.

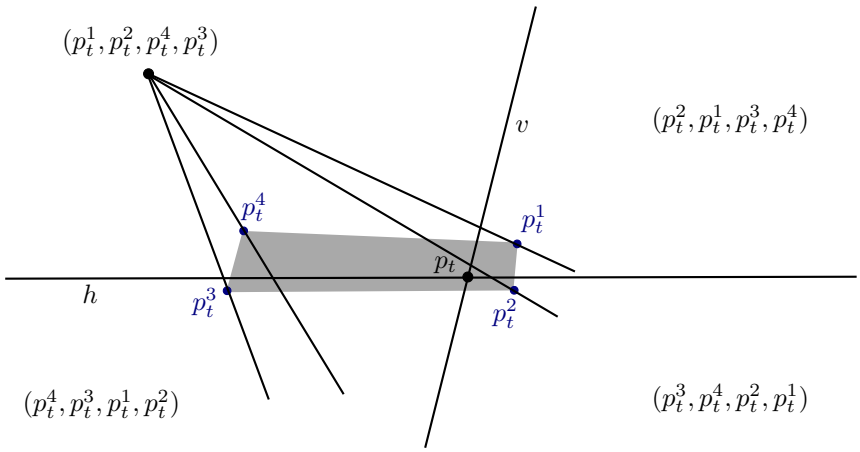

Figure 3.16.: The tuples indicate the order of the decreasing slopes of the lines $\ell(p, p_t^i)$ depending on the quadrant of p.

Finally, the slopes of lines on the boundary of the convex region are the closest to the slope of h for the upper and lower boundary of the convex region (resp. the slopes closest to v for the left and right boundary).

The slopes of the diagonals of the convex region lie between the slopes of lines through points in a quadrant and a point of P_t and the slopes of lines through a point on h and a point of P_t.

If p_t lies on one line spanned by points in $\{p_1, \ldots, p_{t-1}\}$, we apply essentially the same steps as in the case for two lines: If p lies on the line h we pick another point p that does not lie on h, which exists be the definition of constructible order types. We use the line $\ell(p, p_t)$ as the second line v to apply the same replacement of p_t by four points as in the step before. The resulting circular sequence $\tilde{C} := C^5$ is realizable if and only if the original circular sequence C is realizable: Assume that $\tilde{C} = C^5$ is realizable. We

show inductively that if C^t is realizable then C^{t+1} is realizable.

In the first case, where we did not replace the point p_t, the statement is true. In the second case, let R be a realization of the circular sequence C^t. We delete the points of P_t from R and add the point p_t at the intersection point of h and v, which lies in the convex region spanned by the points of P_t. We replace the points of P_t by the point p_t, intersection point of h and v. The point p_t has the same orientation in each triple as any of the points in P_t.

Since the slopes of the lines through a point $p \notin P_t$, which does not lie on v or h, and the points of P_t appear consecutively among all slopes, and the slope of $\ell(p, p_t)$ lies in this interval, the new realization is a realization of C_{t+1} as constructed.

In the third case we use the same arguments: We replace points of P_t by one point p_t that lies on its original line h inside the convex hull of points of P_t, which results in a realization of C^{t+1}.

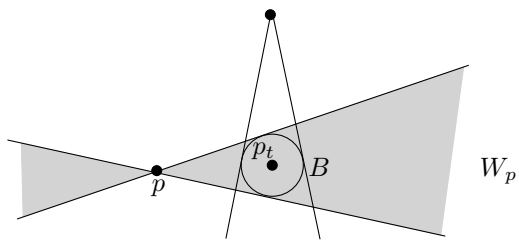

Figure 3.17.: The double wedge W_p defined by a ball around p_t.

On the other hand, we have to show that $\tilde{C}$ is realizable if C is realizable. Therefore, we construct a realization of C^t from C^{t+1}. We first determine ball B around p_t. Let W_p be the smallest double wedge originating at point p that contains the ball B as shown in Figure 3.17. We choose the radius of B, such that no point is contained in another double wedge and the intersection of two cones is bounded, unless the origins of the double wedges are collinear with p_t. This has the effect that placing the points of P_t inside the ball B leads to a realization, where the lines through a point p and the points of P_t lie inside of the double wedge W_p. Since the double wedges are disjoint outside of a bounded region the slopes of the lines through points of P_t form an interval, which lies inside of the wedge. Afterwards, we determine a small angle, such that the only slope in this angle is the slope of h. We construct a parallelogram around p_t inside B, with the sides parallel to v and h, such that the slopes of the diagonals lie in this angle. We place the points of P_t on the vertices of the parallelogram. Finally, we perturb the points slightly, such that the sides of the parallelogram are not parallel to v and h, without changing the circular sequence otherwise. This results in a realization of the circular sequence C^t.

The circular sequence C^t does not contain any parallel lines spanned by points including

$p_t, \ldots, p_n$. Thus C^5 is a simple circular sequence. In this way, we have reduced the realizability of constructible circular sequences to the realizability of simple circular sequences, which shows that realizability of simple circular sequences is hard in $\exists\mathbb{R}$. $\square$

Note that Property 2 of the constructible order type is necessary in the construction: The point p_t is replaced by a convex region. This region reserves space for the unique intersection point p_t of some lines. If p_t is forced to be the intersection point of more than two lines, then the order type guarantees that the pairwise intersection point of two of the lines lies in the convex region, but not that all three lines have a common intersection point, which is forced by the original point p_t.

Via projective duality we obtain the following result for line arrangements, which we use in Chapter 5 for point visibility graphs.

Theorem 3.12. *Deciding the realizability of simple allowable sequences of simple line arrangements is complete in* $\exists\mathbb{R}$.

Proof. A simple allowable sequence of a line arrangement corresponds to a simple circular sequence of an abstract order type via projective duality. Thus, a given abstract order O type with circular sequence C is realizable if and only if the dual pseudoline arrangement L is stretchable with allowable sequence C. This shows that deciding the realizability of a pseudoline arrangement with given allowable sequence is complete in $\exists\mathbb{R}$. $\square$

4. Slopes of segment intersection graphs revisited

In Chapter 2 we proved that minimizing the slope number of a segment intersection graph is NP-hard. Now, we show that the problem is even hard in the existential theory of the reals.

The proof is composed of the following steps. First we show that the problem of minimizing the slope number of a *partial, simple line arrangements* is hard in $\exists\mathbb{R}$. By a *partial line arrangement* we mean the arrangement of segments in a connected region, where each of the segments has both ends on the boundary of the region. For example the gray part of the lines in Figure 4.1 that intersect the gray region are a partial line arrangement.

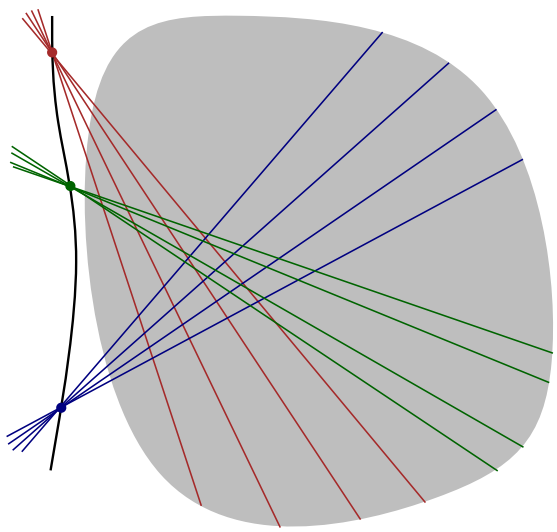

Figure 4.1.: The part of the lines intersecting the gray shaded region is a partial line arrangement.

We can use the *order forcing lemma* of Kratochvíl and Matoušek [KM94] to fix the partial, simple arrangement in a segment intersection graph to show that the slope minimization problem for segment intersection graphs is hard $\exists\mathbb{R}$.

Theorem 4.1. *Minimizing the slope number of a partial, simple line arrangements is hard in the existential theory of the reals.*

Proof. We start to describe the basic idea. Again, we use duality to prove the corresponding results for order types instead of line arrangements. Here we basically[1] use the standard dual map $\mathcal{D}$ given by

$$\mathcal{D} : \mathcal{L} \to \mathcal{P}, y = Ax - B \mapsto (B, A).$$

The problem of minimizing the number of different slopes of the lines translates into the problem of realizing the order type O using as few as possible different y-coordinates of the points. We consider the order type constructed in the proof of Theorem 3.9. In this order type we removed the point ∞ to make the order type realizable. However, it is hard in $\exists\mathbb{R}$ to check whether the point ∞ can be added, which is forced by the circular sequence. In this proof we force the point ∞, the common intersection point of all "horizontal lines", by using as few horizontal lines as possible to position the points of the order type on. Therefore, let s be the number of lines through ∞ in the simple order type that points lie on. Before removing ∞ we add points on each of these lines, such that each line contains at least $s + 1$ points. The resulting order type has a realization using at most s different y-coordinates if and only if the modelled inequality system describes a non-empty set: Since we have at least $s + 1$ points on each line which contained the point ∞ before the modification, we know that each of those lines is horizontal in a realization using at most s different y-coordinates. This shows that we can add the point ∞ again to obtain a realization of an order type that contains the original order type O. □

As in [KM94], we use the order forcing lemma to force a copy of the partial line arrangement in a segment intersection graph to transfer the properties of the line arrangement on the segment intersection graph.

Theorem 4.2. *Determining the slope number of a segment intersection graph is hard in* $\exists\mathbb{R}$.

Proof. We reduce the slope number problem of simple, partial line arrangements to the slope number problem of segment intersection graphs. Let L be a partial, simple line arrangement. For this simple segment representation we construct the graph G_L from Lemma 0.6. Then G_L has a segment representation using s slopes if and only if L can

[1] The standard dual map is given by $y = Ax - B \mapsto (A, B)$. To keep the orientation of our figures aligned to the figures used in the last section we reverse x and y-coordinate of the order type. The different choice of the duality map reverses all orientations of triples, which does not influence the realizability.

be realized using at most s slopes. A segment representation of G_L can be constructed from L in polynomial time. This concludes the proof that minimizing the slope number of segment intersection graphs is hard in $\exists\mathbb{R}$. $\square$

5. Recognition of point visibility graphs

In this chapter, we consider the recognition problem for point visibility graphs. In Section 5.1, we provide background on previous work on point visibility graphs and visibility graphs in general. In Section 5.2, we show simple visibility graph constructions, the *fan* and the *generalized fan*, all geometric realizations of which are guaranteed to preserve a specified collection of subsets of collinear points. The proofs are elementary and only require a series of basic observations. The main result, the $\exists\mathbb{R}$-hardness of point visibility graph recognition, is given in Section 5.3. The proof is based on the *uniqueness* of the representation of a fan and the $\exists\mathbb{R}$-hardness of the realizability of simple allowable sequences of line arrangements (Theorem 3.12). In Section 5.4, we give two applications of the fan construction. In the first, we show that there exists a point visibility graph that does not have any geometric realization on the integer grid. In other words, all geometric realizations of this point visibility graph are such that at least one of the points has an irrational coordinate. Another application of the fan construction follows, where we show that there are point visibility graphs, such that each grid realization requires coordinates of values $2^{2^{\sqrt[3]{n}}}$ where n denotes the number of vertices of the point visibility graph. We conclude the section by discussing upper bounds on a grid representation of a graph. This question is closely related to the decidability of the existence of a grid representation of a PVG. We show that this problem is decidable if and only if the *existential theory of the rationals* ($\exists\mathbb{Q}$) is decidable. The decidability of $\exists\mathbb{Q}$ is a longstanding open problem.

5.1. Background

The recognition problem for point visibility graphs has been explicitly stated as an important open problem by various authors [KPW05], and is listed as the first open problem in a recent survey from Ghosh and Goswami [GG13].

For *planar* point visibility graphs, a linear-time recognition algorithm has been proposed by Ghosh and Roy [GR14]. For general point visibility graphs they showed that recognition problem lies in $\exists\mathbb{R}$. More recently, Roy [Roy14] published an ingenious and rather involved NP-hardness proof for recognition of arbitrary point visibility graphs. Since $\exists\mathbb{R}$ contains NP the $\exists\mathbb{R}$-hardness implies the NP-hardness.

Structural aspects of point visibility graphs have been studied by Kára, Pór, and Wood [KPW05], Pór and Wood [PW10], and Payne et al. [PPVW12]. Many fascinating open questions revolve around the *big-line-big-clique* conjecture, stating that for all $k, \ell \geq 2$, there exists an n such that every finite set of at least n points in the plane contains either k pairwise visible points or ℓ collinear points.

Visibility graphs of polygons are defined over the vertices of an arbitrary simple polygon in the plane, and connect pairs of vertices such that the open segment between them is completely contained in the interior of the polygon. This definition has also attracted a lot of interest in the past. Ghosh gave simple properties of visibility graphs of polygons and conjectured that they were sufficient to characterize visibility graphs [Gho88, Gho97]. These conjectures have been disproved by Streinu [Str05] via the notion of *pseudo-visibility* graphs, or visibility graphs of *pseudo-polygons* [OS97]. A similar definition is given by Abello and Kumar [AK02]. Roughly speaking, the relation between visibility and pseudo-visibility graphs is of the same nature as that between arrangements of straight lines and pseudolines. Recently, Gibson, Krohn and Wang [GKW15] gave a characterization for visibility graphs of pseudo-polygons. Their characterization is based on the characterization of *vertex-edge pseudo-visibility graphs* by O'Rourke and Streinu [OS97]. Although, as Abello and Kumar note, these results somehow suggest that the difficulty in the recognition task is due to a stretchability problem, the complexity of recognizing visibility graphs of polygons remains open.

This chapter is based on [CH15]. A version that is extended by Subsection 5.4.3 is submitted [CH16].

5.2. Point visibility graphs preserving collinearities

We first describe constructions of point visibility graphs, such that all their geometric realizations preserve some fixed subsets of collinear points.

To do so, we introduce the following notations. The open segment between p and q is denoted by $\overline{pq}$. We will often call $\overline{pq}$ the *sight line* between p and q, since p and q see each other if and only if $\overline{pq} \cap P = \emptyset$. We call two sight lines $\overline{p_1q_1}$ and $\overline{p_2q_2}$ non-crossing if $\overline{p_1q_1} \cap \overline{p_2q_2} = \emptyset$. For each point p all other points of G lie on $\deg(p)$ many rays $R_1^p, \ldots, R_{\deg(p)}^p$ originating from p.

Preliminary observations

In the realization of a PVG, the point p sees exactly $\deg(p)$ many vertices, hence all other points lie on $\deg(p)$ rays of origin p.

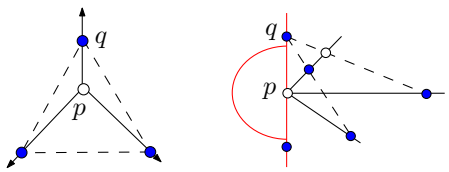

Figure 5.1.: (Lemma 5.1) Left: a point sees points on consecutive rays with small angle. Right: a vertex of $\deg(q) = 1$ in $G[N(p)]$ lies on the boundary of an empty halfspace.

Lemma 5.1. *Let $q \in N(p)$ be a degree-one vertex in $G[N(p)]$. Then all points of G lie in one of the closed half-spaces defined by the line $\ell(p, q)$. Furthermore, the neighbor of q lies on the ray that forms the smallest angle with $\overline{qp}$.*

Proof. If the angle between two consecutive rays is smaller than π, then every vertex on one ray sees every vertex on the other ray. Hence one of the angles of a ray next to the ray of q in the circular order is at least π and the neighbour of q lies on the other neighbouring ray. □

Corollary 5.2. *If $G[N(p)]$ is an induced path, then the order of the path and the order of the rays the points lie on coincide.*

Proof. By Lemma 5.1 the two endpoints of the path lie on rays on the boundary of empty half-spaces. Thus, all other rays form angles which are smaller than π, and thus they see their two neighbors of the path on their neighboring rays. □

Observation 5.3. *Let q, $q \neq p$, be a point that sees all points of $N(p)$. Then q is the second point (not including p) on one of the rays emerging from p.*

Proof. Assume q is not the second point on one of the rays. Then q cannot see the first point on its ray which is a neighbor of p. □

Observation 5.4. *Let q, $q \neq p$, be a point that is not the second point on one of the rays from p and sees all but one of the neighbors of p. Then q lies on the ray of the neighbor it does not see.*

Fans and generalized fans

We have enough tools to show the uniqueness of a PVG obtained from the following construction, which is depicted in Figure 5.2. Consider a set S of segments between two lines ℓ and ℓ' intersecting in a point p. For each intersection of a pair of segments, construct a ray of origin p and going through this intersection point. Add two segments s_1

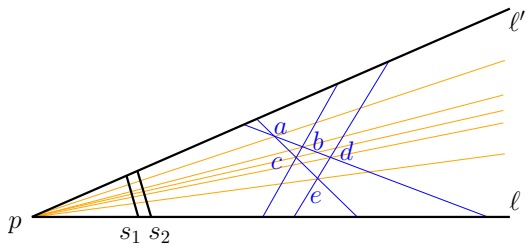

Figure 5.2.: A fan: a vertex is placed on each intersection of two lines/segments.

and s_2 between ℓ and ℓ', such the first intersection point on ℓ and ℓ' lie on s_1 and the second on s_2.

We now put a point on each intersection of the segments and rays and construct the PVG of this set of points. We call this graph the *fan* of S and denote it by $\text{fan}(S)$. Since we have the choice of the position of the segments s_1 and s_2, we can avoid any collinearity between a point on s_1 or s_2 and points on other segments, except for the obvious collinearities on one ray. Thus, every point sees all points on s_1 except for the one of the ray it lies on.

Lemma 5.5. *All realizations of a fan preserve collinearities between points that lie on one segment and between points that lie on one ray.*

Proof. We first show that the distribution of the points onto the rays of p is unique. By construction the points on s_2 see all the points on s_1, which are exactly the neighbors of p. Thus, by Observation 5.3, the points from s_2 are the second points of a ray. Since there is exactly one point for each ray on s_2, all the other points are not second points on a ray. By construction each of the remaining points sees all but one point of s_1. Observation 5.4 gives a unique ray a point lies on. The order of the rays is unique by Corollary 5.2. On each ray the order of the points is as constructed, since the PVG of points on one ray is an induced path. Now, we have to show that the points originating from one segment are still collinear. Consider three consecutive rays R_1, R_2, R_3. We consider a visibility between a point p_1 on R_1 and one point p_3 on R_3 that has to be blocked by a point on R_2. Let p_2 be the original blocker from the construction. For each point on R_2 that lies closer to p, there is a sight line blocked by this point, and for each point that lies further away from p there is a sight line blocked by this point. For each of these points we pick one sight line that corresponds to an original segment and $\overline{p_1p_3}$. This set of sight lines is non-crossing, since the segments only intersect on rays by assumption. So we have a set of non-crossing sight lines and the same number of blockers available. Since the order on each ray is fixed, and the sight lines intersect R_2 in a certain order, the blocker for each sight line is uniquely determined and has to be the original blocker. By transitivity of collinearity all points from the segments remain

collinear. □

The fan is the tool we need to show that PVG recognition is complete in $\exists\mathbb{R}$. In the proof for this fact, which is published in [CH15], we used a *generalized fan* for this reduction.

5.3. $\exists\mathbb{R}$-completeness of PVG recognition

In this section we show that the recognition of PVGs is complete in $\exists\mathbb{R}$.

Theorem 5.6. *The recognition of point visibility graphs is $\exists\mathbb{R}$-complete.*

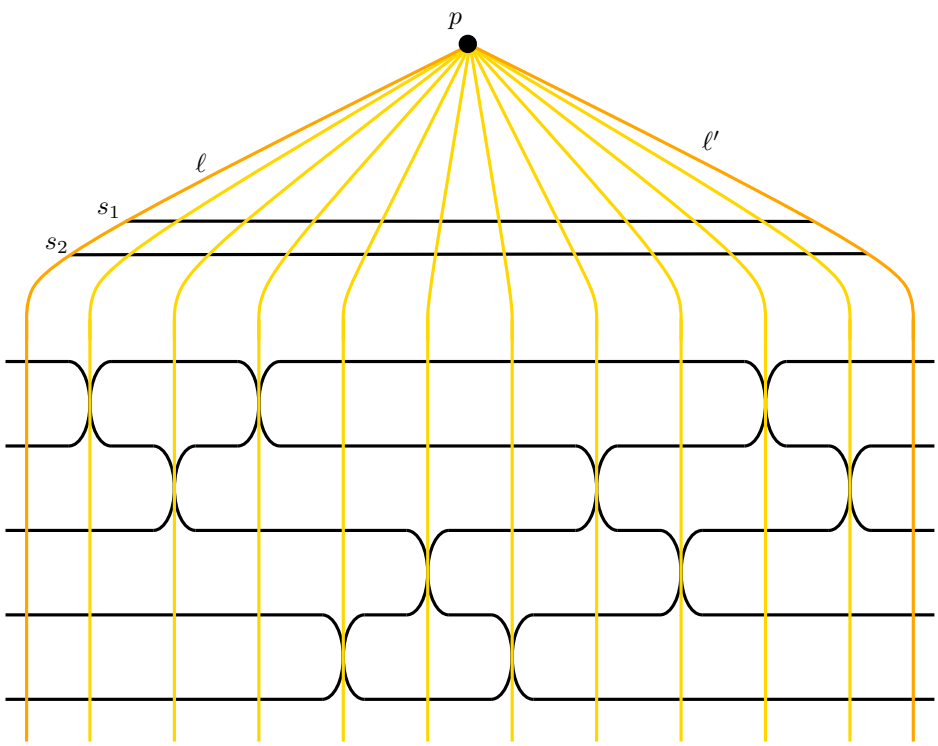

Figure 5.3.: Construction of a fan from a pseudoline arrangement A (black) with a given allowable sequence.

Proof. The idea of the proof is to reduce the stretchability of a pseudoline arrangement with given allowable sequence. Therefore, let L be a simple pseudoline arrangement with allowable sequence A. We construct the following graph G using the fan construction: We draw the wiring diagram of the pseudoline arrangement in the plane using x-monotone curves as shown in black in Figure 5.3, such that the order of x-coordinates

of the crossings agrees with the allowable sequence. We place the point p, where the rays of the fan are originating from, on the intersection point of a vertical line and the line at infinity. The lines ℓ and ℓ' that form the boundary of the fan are vertical lines that are placed to the left and to the right of all intersection points of the pseudoline arrangement.

We define G as the fan of this construction: We add two horizontal segments s_1 and s_2 that are spanned between ℓ and ℓ' above the pseudolines. We add a vertical ray from p through each of the intersection points of the pseudoline arrangement. We denote the set of vertical rays by V. On each intersection point of a ray with any other line we place a point. The only non-visibilities appear between non-consecutive points on one ray or pseudoline.

We show that G is a PVG if and only if L is stretchable with A as allowable sequence. If L is stretchable with allowable sequence A, we can repeat the construction of the fan as described above with the realization R of L to obtain a PVG realization. The described non-visibilities (the collinearities) are preserved in this realization by Lemma 5.5. Thus, it remains to show that there is a realization, such that all visibilities between each pair of points that do not lie on the same ray of the fan and not on the same line of the arrangement is not blocked. We show that this is possible by perturbing the realization of the fan. We do this again by dualizing the arrangement of lines and rays. Since

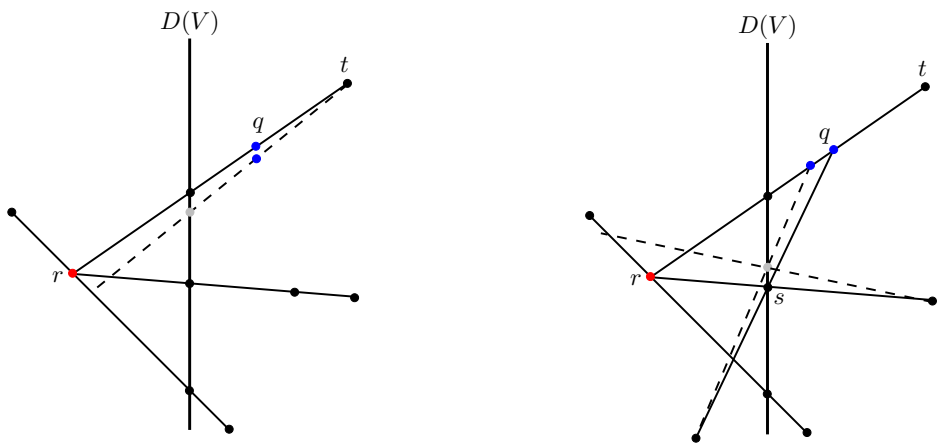

Figure 5.4.: Perturbing q. Left: Moving q away from $\ell(q,t)$. Right: Moving q along $\ell(q,t)$ to perturb another point on $D(V)$.

the points of the PVG lie on the intersection points in this arrangement we have to guarantee that three intersection points that do not lie on one line are not collinear. This dualizes to three lines spanned by three pairs of points of the order type that intersect in one point. We use the mirrored dual map $D : \mathcal{L} \to \mathcal{P}, x = Ay + B \mapsto (A, B)$ to map the realization R of L to an order type O. The vertical rays, which we add in

the construction of the fan, dualize to points on the y-axis; one for each intersection of the y-axis with a line spanned by two points of the order type. Now, assume that in the order type $D(L \cup V)$ there are three independent pairs that span lines intersecting in one point r. We consider a point q of in $D(L)$ that is contained in a pair that spans a line, which intersects two other spanned lines in one common point. Note that we can move q in an ε-neighborhood without creating new non-simple points in the spanned arrangement, and without changing the circular sequence, by continuity arguments.

When we move q, we also move some points of $D(V)$ to obtain the order type as indicated in Figure 5.4. If none of the independent pairs that create r contains a point on $D(V)$ whose position depends on q, then we can move q away from the line $\ell(q, t)$, where t is the point collinear with q and r, such that one line less intersects r. Otherwise, if there is a point $s \in D(V)$ whose position depends on q, then we move q along the line $\ell(q, t)$. This has the effect that the line spanned by s and one other point does not intersect r any more. This step reduces the number of lines that intersect in the point r. This perturbation reduces the number of triples of independent pairs whose spanned lines intersect in a common point. Thus, repeating this step leads to an order type without such triples. Placing points on the intersection points of the dual line arrangement leads to a realization of the PVG G. □

5.4. Point visibility graphs on a grid

In this section we show that there are PVGs that cannot be represented on a grid, i.e., that cannot be represented with rational coordinates. In addition we show that there are PVGs that can be represented on a grid, but each representation on a grid requires a doubly exponential grid size. Afterwards we show that deciding whether a PVG has a representation on a grid is decidable if and only if the existential theory of the rationals ($\exists\mathbb{Q}$) is decidable. The decidability of $\exists\mathbb{Q}$ is a longstanding open problem. We point out that the undecidability of $\exists\mathbb{Q}$ would imply that the grid size for PVGs with n vertices cannot be bounded from above by a computable function in n.

5.4.1. Irrational coordinates

First, we present a point visibility graph that has no representation, such that all points use rational coordinates.

Theorem 5.7. *There exists a point visibility graph, such that every PVG realization has at least one point with an irrational coordinate.*

Proof. We use the so-called *Perles configuration* of 9 points on 9 lines illustrated in Figure 5.5. It is known that for every geometric realization of this configuration in the

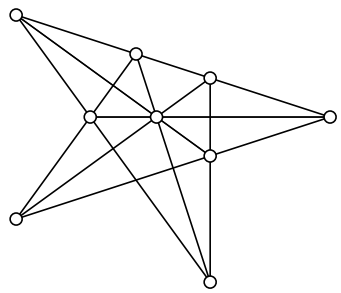

Figure 5.5.: The Perles configuration.

Euclidean plane, one of the points has an irrational coordinate [Grü03]. We combine this construction with the fan construction described in Section 5.2. Hence, we pick two lines ℓ and ℓ' intersecting in a point p, such that all lines of the configuration intersect both ℓ and ℓ' in the same wedge. Note that up to a projective transformation, the point p may be considered to be on the line at infinity and ℓ and ℓ' are parallel. We add two non-intersecting segments s_1 and s_2 close to p, that do not intersect any line of the configuration. Then, we construct a ray from p through each of the points, and construct the visibility graph of the original points together with all the intersections of the rays with the lines and the two segments s_1, s_2. From Lemma 5.5, all the collinearities of the original configuration are preserved, and every realization of the graph contains a copy of the Perles configuration. □

5.4.2. Large grid size

Here we give a simple construction for point visibility graphs where all its integer realizations require a superexponential grid size, thus a (binary) encoding of the coordinates has superpolynomial size.

With the following simple construction we encode a line arrangement in a fan as shown in Figure 5.3. Consider a line arrangement $\mathcal{A}$, and add a point p in an unbounded face of the arrangement, such that all intersections of lines are visible in an angle around p that is smaller than π. Construct rays ℓ and ℓ' through the extremal intersection points of the arrangement and p. By Lemma 5.5, the fan of this construction gives a PVG that fixes $\mathcal{A}$. Since there are line arrangements that require integer coordinates of values $2^{2^{\Theta(|\mathcal{A}|)}}$ [GPS90] and the fan has $\Theta(|\mathcal{A}|^3)$ points we get the following lower bound on the coordinates of points in any realization of this PVG.

Corollary 5.8. *There exists a point visibility graph with n vertices every realization of which requires coordinates of values $2^{2^{\Theta(\sqrt[3]{n})}}$.*

5.4.3. Recognition of point visibility graphs on a grid

We now prove that the recognition problem for visibility graphs on a grid is decidable if and only if the existential theory of the rationals is decidable . The computational complexity of answering the question *"Does this object have a realization on a grid?"* (*rational realization problem*) is unknown for various types of objects. Most prominently, it is unknown for polytopes and oriented matroids and (non-simple) order types and thus (non-simple) line arrangements. Matiyasevich [Mat70] showed that the existential theory of the integers is undecidable by giving a negative solution to Hilbert's tenth problem: Deciding whether a diophantine equation has a solution is undecidable. This cannot be applied to a grid realization of a PVG, since a realization of a PVG with rational coordinates, which can be obtained by a rational solution of the inequality system, leads to a grid realization by scaling. Hence for those geometric realizations on the grid the decidability of Hilbert's tenth problem over the rationals is of interest. Grünbaum [Grü72] conjectured in 1972 that there is no algorithm that enumerates all arrangements in the rational projective plane, which is equivalent to the recognition problem of order types that can be represented on a grid. This conjecture is still open. Poonen [Poo09] tried to define the integers using rational logic, i.e., to give a formula $F(w)$ over the rationals that is satisfiable if and only if w is an integer. These formulas are known, but they involve existential and universal quantifiers. The existence of such a formula using only existential quantifiers would show the undecidability of the existential theory of the rationals.

Similar to work a of Sturmfels [Stu87] for oriented matroids and polytopes, we show the following theorem.

Theorem 5.9. *The realization problem for visibility graphs of points on a grid is decidable if and only if the existential theory of the rationals is decidable.*

Before proving this theorem, we point out a connection of this question to finding an upper bound on the grid size of a PVG that is realizable on a grid.

Corollary 5.10. *Assume the recognition problem for PVG on a grid is undecidable. Then there is no computable function $f : \mathbb{N} \to \mathbb{N}$ such that each PVG with n points that is realizable on a grid can be drawn on a grid of size $f(n) \times f(n)$.*

Proof. We suppose that a computable function $f : \mathbb{N} \to \mathbb{N}$ exists, such that every PVG that is realizable on a grid with n vertices can be represented on a grid of size $f(n) \times f(n)$. Using this function, we can give an algorithm that decides whether a graph G with $|V(G)| = n$ has a realization as a PVG on a grid.

We first compute $f(n)$, then for each $x \in [f(n)]^{2n}$ we check whether G is the PVG of the point set $(x(v_1), y(v_1), \ldots, x(v_n), y(v_n)) = x$. If there is such an x, the algorithm returns the realization, otherwise no realization exists.

This algorithm is clearly an effective decision procedure, and thus the recognition problem for PVG on a grid is decidable – a contradiction to the assumption. □

If we try to use the reduction of Theorem 5.6 in order to prove Theorem 5.9, the main obstacle is that we cannot reduce to a strict inequality system. This leads to an open realization space, and we can always find a rational point in an non-empty open set. If we do not have an open realization space we do not know if we can perturb the lines/points as in the proof of Theorem 5.6 to get rid of collinearities of points of the PVG that do not lie on a common line of the arrangement. However, since we are only concerned with the decidability of the problem, we use a (non-polynomial) computable reduction.

Proof of Theorem 5.9. We give a computable reduction from the rational realization problem for (non-simple) line arrangements to the rational realization problem for PVGs. Given a line arrangement L of n lines, we construct a finite set of graphs $\mathcal{G}_L$, such that L has a rational realization if and only if at least one graph in $\mathcal{G}_L$ has a rational PVG realization.

We construct the set of graphs $\mathcal{G}_L$ as follows. We want to encode L in a fan. Thus, we pick one unbounded face f_p of L as the face to place the point p, the origin of all rays of the fan. Since we do not know the circular order of the vertices of L around p we create a set of graphs $\mathcal{G}_A$ for each possible circular order A of vertices of L around p, namely for each possible allowable sequence A with p in the fixed unbounded face f_p. The set of graphs $\mathcal{G}_A$ contains possible fans with circular order A of the vertices of L around p: Depending on a realization of L (with A) the visibility of two points of the fan that do not lie on the same line of L or the same ray of the fan can be blocked by another point. For each combination of additional blocked edges we create a single graph. Therefore, note that the only edges we require to keep the unique representation (all lines of L are preserved) of a fan (Lemma 5.5) are the edges of one vertex to the vertices on different rays on the segment s_1 (the segment that is added closest to p in the construction of the fan), and the edges of the path on one ray and the path on the segment s_1. All other *optional edges* can be removed (which may lead to a non-realizable graph) to keep the points on the lines of L collinear in each PVG representation of the fan. For each subset S of optional edges, the set of graphs $\mathcal{G}_A$ contains the fans forcing the circular order A around p, where exactly the optional edges contained in S are also contained in the fan.

Now, L has a rational realization if and only if there is a graph $G \in \mathcal{G}_L$ that has a rational PVG realization:

First assume L has a rational realization R. From R we construct a fan as before by placing p in the same unbounded face. The PVG realization obtained by this fan

construction is a rational realization of some graph G contained in $\mathcal{G}_A$, where A is the circular order around p in this realization.

On the other hand, a rational PVG realization of one of the graphs in $\mathcal{G}_L$ leads to a rational realization of L, since each graph in $\mathcal{G}_L$ preserves the collinearities of points on L in each PVG realization.

The set $\mathcal{G}_L$ can be computed from L. Thus, an effective decision algorithm for the rational realization problem for PVGs leads to an effective decision algorithm for the rational realization problem for line arrangements by applying the algorithm on each graph in $\mathcal{G}_L$. This shows that if the rational realization problem for PVGs is decidable, then the rational realization problem for line arrangements is decidable. On the other hand, we can encode the realization problem for PVGs in an existential formula as done in [GR14]. A rational solution of this formula leads to a rational realization of the PVG, because the variables encode the coordinates of the points. □

6. The planar slope number

The *slope number* of a graph G is defined to be the minimum number of distinct edge slopes in a non-degenerate straight line drawing of G. Similarly, the *planar slope number* of a graph is the minimum number of distinct edge slopes in a planar straight line drawing of G.

In this section, we consider the computational complexity of computing the slope number. We show that determining the *planar slope number* of a graph is hard in $\exists\mathbb{R}$.

6.1. Background

A simple lower bound for the slope number of a graph G is $\lceil \Delta(G)/2 \rceil$, where $\Delta(G)$ denotes the maximum degree of G, since at most two edges of the same slope are incident to one vertex. The main work in this area deals with the question whether the slope number of a planar graph is also bounded from above by a function in the maximum degree. This was answered negatively [BMW06, PP06, DSW07] by examples of families of graphs of maximum degree 5 with arbitrarily large slope number. In contrast, Keszegh, Pach, and Pálvölgyi have shown that the planar slope number is bounded by an exponential function in the maximum degree [KPP13]. For partial planar 3-trees [JJK+13] this has been improved to a polynomial upper bound of $O(\Delta^5)$ and for outerplanar graphs [KMW14] to a linear upper bound of $\Delta - 1$ (for $\Delta \geq 4$), even for outerplanar drawings.

It is NP-complete to decide whether a graph has slope number 2 [FHH+93], and it is NP-complete to decide whether a graph has planar slope number 2 [GT01]. Thus both problems, computing the slope number and the planar slope number are NP-hard. In the following subsection we show that computing the planar slope number is even hard in $\exists\mathbb{R}$.

6.2. Computational complexity

Theorem 6.1. *Deciding if the planar slope number of a planar graph with maximum degree Δ is $\lceil \Delta/2 \rceil$ is complete in $\exists\mathbb{R}$.*

Proof. We prove the theorem by reducing the stretchability of a pseudoline arrangement to the realizability of a planar graph using $\lceil \Delta/2 \rceil$ slopes. Let L be an arrangement of $n-3$ pseudolines. From L we construct a arrangement L' by adding three lines to L, such that all intersections of L lie in the triangle formed by these new lines. This can be done in a way, such that L' is stretchable if and only L is stretchable, for example as shown in Figure 6.1. This has the effect that all intersections of L are inner intersection

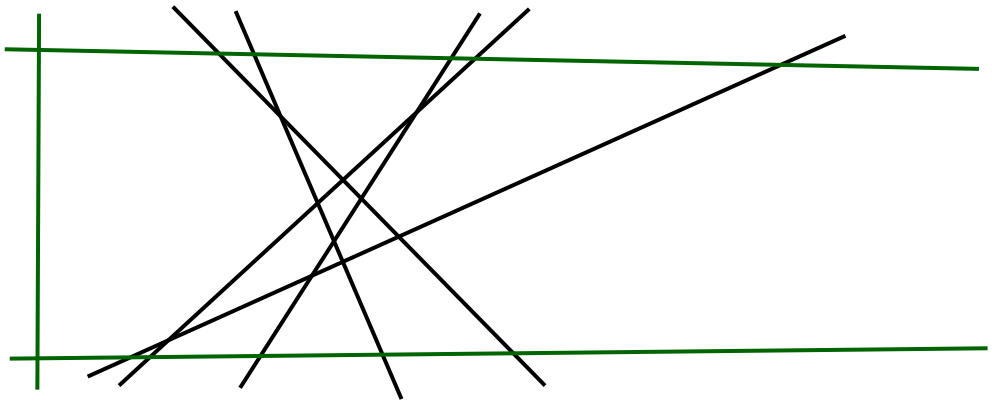

Figure 6.1.: Adding three lines (green) to a (pseudo)line arrangement L, such that all vertices of L are inner vertices.

points of L'. We construct the following planar graph $G_{L'}$: First take the intersection points of L' vertices of $G_{L'}$ and the line segments between the vertices as edges G. To this graph, we add a cycle C_f in each bounded face f. We determine the length of this cycle in the following way: Note that the (cyclic) order of the slopes of lines of a line arrangement is already encoded in the line arrangement. Consider a vertex v that is incident to a face f of the arrangement L'. Let ℓ and ℓ' be the pseudolines in clockwise order that form the boundary of the face f close to v. We define $w(v, f)$ as the number of (pseudo)lines of L', whose slopes lie clockwise between ℓ and ℓ'. The length of the cycle C_f, which we add in face f, is

$$1 + \sum_{v \text{ incident to } f} (2w(v, f) + 1)\,.$$

We connect each v, that is incident to f, to $2w(v, f) + 1$ consecutive vertices of the cycle. We do this in a way, such that $G_{L'}$ stays planar.

The graph $G_{L'}$ we just constructed has a straight line drawing with $\Delta(G_{L'})/2 = 2n$ slopes if and only if L is stretchable.

We first show that L is stretchable if $G_{L'}$ has a planar drawing with $2n$ slopes. Note that the graph $G_{L'}$ is, after contracting the two edges incident to the degree two vertex in each face, 3-connected and thus has, up to the choice of the outer face, a unique

embedding. Thus, the cyclic order of the neighbours around a vertex is the order which we constructed in the planar drawing. The inner vertices of the arrangement have degree $4n$. Thus, in a realization using $2n$ slopes, the opposite edges have the same slope. Since the edges coming from a pseudoline are opposite we obtain a realization of L by drawing each line on the path of the edges originating from its pseudoline.

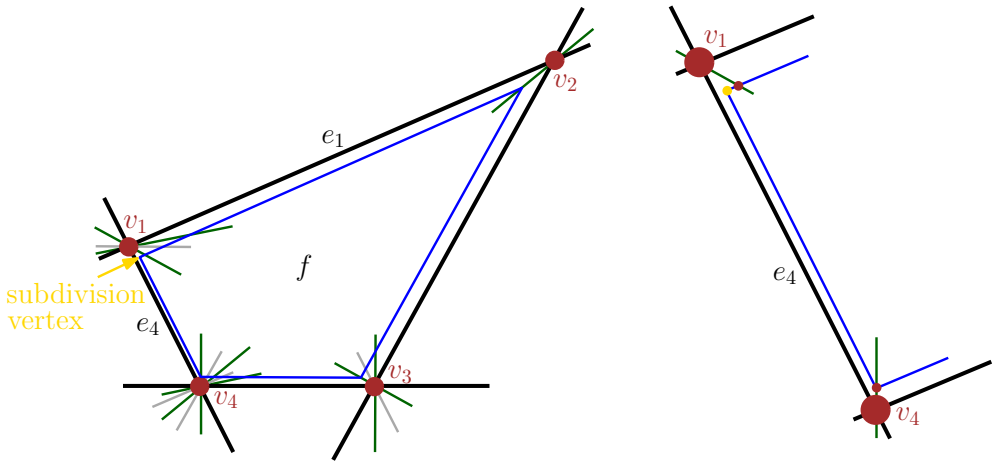

Figure 6.2.: Left: Drawing the cycle in the face. Green segments indicate slopes of the arrangement, gray segments intermediate slopes. Right: Using the subdivision vertex to close the polygon.

On the other hand, $G_{L'}$ admits a straight line drawing with $2n$ slopes if L is stretchable: Consider a realization R of L', which exists if L is stretchable. We construct a planar drawing of the graph $G_{L'}$ on R. We draw the vertices of L' on this realization.

Between two consecutive slopes used by the realization of L' we fix one additional slope. Through each vertex of L' we draw a line of each of the slopes. In each of the bounded faces we construct a polygon in the face that almost fills the whole face. The vertices of the cycle in the face are basically placed on the intersection points of the lines through the vertices of the face and the polygon. Starting from the vertex that follows the degree two vertex of the cycle clockwise, we enumerate the vertices and edges clockwise by $v_1, \ldots, v_{\deg(f)}$ and $e_1, \ldots, e_{\deg(f)}$. We pick a point close to v_1 on the most counterclockwise line through v_1 in f. From this point we draw the first edge of the polygon parallel to the edge e_1. We end on the clockwise last line through v_2 that lies in the face f. Iteratively, we proceed to draw edges of the polygon from the endpoint of the last edge close to v_i, parallel to the edge e_i, to the clockwise last line through v_{i+1} until we reach $v_{\deg(f)}$. If the vertices of the polygon close to v_1 and $v_{\deg(f)}$ can be connected by a segment parallel to e_n we do so and add the subdivision vertex on this edge. Otherwise, we use the degree two vertex to extend the edge parallel to e_1

or $e_{\deg(f)-1}$ to close the polygon as depicted in Figure 6.2, by placing the subdivision vertex on the vertex of the polygon that is not intersected by a line through a vertex. By construction the placement of the vertices on the polygons and the arrangement leads to a drawing of G_L using $2n$ slopes.

The graph $G_{L'}$ can be constructed from L in polynomial time. Consequently, deciding if $G_{L'}$ can be drawn using $2n$ slopes is hard in $\exists\mathbb{R}$. □

As a consequence of this reduction, we can carry over the result from non-simple line arrangements, as we already did for point visibility graphs in the previous section.

Corollary 6.2. *There are planar graphs, such that each planar drawing that minimizes the number of slopes has at least one vertex with an irrational coordinate.*

Corollary 6.3. *The problem of deciding whether a graph G has a planar straight line drawing on the grid with $\lceil \Delta(G)/2 \rceil$ slopes is decidable if and only if $\exists\mathbb{Q}$ is decidable.*

Determining the computational complexity of computing the non-planar slope number remains an open problem. We conjecture that it is also hard in $\exists\mathbb{R}$.

7. Open problems, questions, conjectures

In this chapter we list some open problems that occured in this thesis.

Chapter 1

The main open problems of Chapter 1 are the recognition problems for many subclasses of grid intersection graphs.

Problem 7.1. *Determine the complexity of the recognition problem for the subclasses of grid intersection graphs mentioned in Chapter 1.*

The following table lists the current state of these problems.

Class	recognition complexity	reference
GIG	NP-complete	[Kra94]
UGIG	NP-complete	[MP13]
3-dim BipG	NP-complete	[FMP15]
3-dim GIG	Open	
StabGIG	Open	
SegRay	Open	
BipHook	Open	
Stick	Open	
4-DORG	Open	
3-DORG	Open	
2-DORG	Polynomial	[STU10],[Cog82]
bipartite permutation	Polynomial	[DM41]

Chapter 2 and Chapter 4

Problem 7.2. *Determine the complexity of approximating the slope number.*

Chapter 3

Problems that are complete in $\exists\mathbb{R}$ have instances, where the obvious certificate has superpolynomial size. However, this does not imply that the problems cannot be solved in NP. For example, there are string graphs that have an exponential number of crossings in each repesentation, but string graph recognition is in NP [SSŠ03].

Open question 7.3. *Is the existential theory of the reals contained in NP?*

Open question 7.4. *Is the existential theory of the rationals decidable?*

Chapter 5

The recognition problem for many classes of visibility graph is open. We refer to Ghosh [GG13] for an overview.

Problem 7.5. *Determine the complexity of the recognition of segment visibility graphs in the plane.*

Open question 7.6. *Gibson, Krohn, and Wang recently gave a characterization of visibility graphs of pseudo-polygons [GKW15]. Does this characterization lead to a polynomial algorithm?*

Problem 7.7. *Determine the complexity of polygon visibility graph recognition.*

Beside this recognition questions, the following question has also been considered [Mat09].

Open question 7.8. *Consider a PVG G with an independent set of size k that can be represented by a point set in general position. How many vertices does G have at least?*

Chapter 6

Open question 7.9. *Is determining the slope number of a graph hard in $\exists\mathbb{R}$?*

The planar slope number of a graph is bounded in its maximum degree [KPP13]. The best known bound is exponential in the maximum degree.

Open question 7.10. *Is the planar slope number of a planar graph polynomially bounded in its maximum degree?*

Bibliography

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[ABB+11] Christopher Auer, Christian Bachmaier, Franz Josef Brandenburg, Wolfgang Brunner, and Andreas Gleißner. Plane drawings of queue and deque graphs. In *International Symposium on Graph Drawing (GD)*, volume 6502 of *LNCS*, pages 68–79. Springer, 2011.

[ABP06] Eyal Ackerman, Gill Barequet, and Ron Y. Pinter. On the number of rectangulations of a planar point set. *Journal of Combinatorial Theory, Series A*, 113:1072–1091, 2006.

[AG11] Christopher Auer and Andreas Gleißner. Characterizations of deque and queue graphs. In *Graph-Theoretic Concepts in Computer Science (WG)*, volume 6986 of *LNCS*, pages 35–46. Springer, 2011.

[AK02] James Abello and Krishna Kumar. Visibility graphs and oriented matroids. *Discrete & Computational Geometry*, 28:449–465, 2002.

[ANS80] Takanori Akiyama, Takao Nishizeki, and Nobuji Saito. NP-completeness of the Hamiltonian cycle problem for bipartite graphs. *Journal of Information Processing*, 3:73–76, 1980.

[APT14] Karim A. Adiprasito, Arnau Padrol, and Louis Theran. Universality theorems for inscribed polytopes and delaunay triangulations. *arXiv:1406.7831*, 2014.

[Bie91] Daniel Bienstock. Some provably hard crossing number problems. *Discrete & Computational Geometry*, 6:443–459, 1991.

[BK79] Frank R. Bernhart and Paul C. Kainen. The book thickness of a graph. *Journal of Combinatorial Theory, Series B*, 27:320–331, 1979.

[BLVS+99] Anders Björner, Michel Las Vergnas, Bernd Sturmfels, Neil White, and Günter M. Ziegler. *Oriented Matroids*. Cambridge University Press, second edition, 1999.

[BMW06] János Barát, Jirı Matoušek, and David R. Wood. Bounded-degree graphs have arbitrarily large geometric thickness. *European Journal of Combinatorics*, 13:R3, 2006.

[Can88] John Canny. Some algebraic and geometric computations in PSPACE. In *Symposium on Theory of Computing (STOC)*, pages 460–467. ACM, 1988.

[CCF+15] Daniele Catanzaro, Steven Chaplick, Stefan Felsner, Bjarni V. Halldórsson, Magnus. M. Halldórsson, Thomas Hixon, and Juraj Stacho. Max-point-tolerance graphs. *Discrete Applied Mathematics*, to appear, 2015.

[CCL13] Sergio Cabello, acceptedJean Cardinal, and Stefan Langerman. The clique problem in ray intersection graphs. *Discrete & Computational Geometry*, 50:771–783, 2013.

[CCM13] Steven Chaplick, Elad Cohen, and Gila Morgenstern. Stabbing polygonal chains with rays is hard to approximate. In *Canadian Conference on Computational Geometry (CCCG)*, 2013.

[CF13] Victor Chepoi and Stefan Felsner. Approximating hitting sets of axis-parallel rectangles intersecting a monotone curve. *Computational Geometry*, 46:1036–1041, 2013.

[CFHW15] Steven Chaplick, Stefan Felsner, Udo Hoffmann, and Veit Wiechert. Grid intersection graphs and order dimension. *arXiv:1512.02482*, 2015.

[CFS14] José R. Correa, Laurent Feuilloley, and José Soto. Independent and hitting sets of rectangles intersecting a diagonal line. In *Latin American Theoretical Informatics Symposium (LATIN)*, volume 8392 of *LNCS*, pages 35–46. Springer, 2014.

[CG14] Timothy M. Chan and Elyot Grant. Exact algorithms and APX-hardness results for geometric packing and covering problems. *Computational Geometry*, 47:112–124, 2014.

[CH15] Jean Cardinal and Udo Hoffmann. Recognition and complexity of point visibility graphs. In *Symposium on Computational Geometry (SoCG)*, volume 34 of *LIPIcs*, pages 171–185. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2015.

[CH16] Jean Cardinal and Udo Hoffmann. Recognition and complexity of point visibility graphs. *Discrete & Computational Geometry*, 2016. accepted.

[CHO+14] Steven Chaplick, Pavol Hell, Yota Otachi, Toshiki Saitoh, and Ryuhei Uehara. Intersection dimension of bipartite graphs. In *Theory and Applications of Models of Computation*, volume 8402 of *LNCS*, pages 323–340. Springer, 2014.

[Chv73] Vašek Chvátal. Tough graphs and Hamiltonian circuits. *Discrete Mathematics*, 5:215–228, 1973.

[Chv85] Vašek Chvátal. Hamiltonian cycles. In Eugene L. Lawler, Jan K. Lenstra, Alexander H.G. Rinnoy Kan, and David B. Shmoys, editors, *The traveling salesman problem: a guided tour of combinatorial optimization*, pages 403–429. Wiley New York, 1985.

[ČKNP02] Jakub Černỳ, Daniel Král, Helena Nyklová, and Ondřej Pangrác. On intersection graphs of segments with prescribed slopes. In *International Symposium on Graph Drawing (GD)*, volume 2265 of *LNCS*, pages 261–271. Springer, 2002.

[Cog82] Olivier Cogis. On the Ferrers dimension of a digraph. *Discrete Mathematics*, 38:47–52, 1982.

[DM41] Ben Dushnik and Edwin W. Miller. Partially ordered sets. *American Journal of Mathematics*, 63:600–610, 1941.

[DSW07] Vida Dujmović, Matthew Suderman, and David R. Wood. Graph drawings with few slopes. *Computational Geometry*, 38:181–193, 2007.

[EET76] Gideon Ehrlich, Shimon Even, and Robert E. Tarjan. Intersection graphs of curves in the plane. *Journal of Combinatorial Theory, Series B*, 21:8–20, 1976.

[Fel13] Stefan Felsner. Exploiting air-pressure to map floorplans on point sets. In *International Symposium on Graph Drawing (GD)*, volume 8242 of *LNCS*, pages 196–207. Springer, 2013.

[Fel14] Stefan Felsner. The order dimension of planar maps revisited. *SIAM Journal on Discrete Mathematics*, 28:1093–1101, 2014.

[FHH+93] Michael Formann, Torben Hagerup, James Haralambides, Michael Kaufmann, Frank T. Leighton, Antonios Symvonis, Emo Welzl, and Gerhard J. Woeginger. Drawing graphs in the plane with high resolution. *SIAM Journal on Computing*, 22:1035–1052, 1993.

[FMP15] Stefan Felsner, Irina Mustaţă, and Martin Pergel. The complexity of the partial order dimension problem – closing the gap. *arXiv:1306.1855*, 2015.

[FN11] Stefan Felsner and Johan Nilsson. On the order dimension of outerplanar maps. *Order*, 28:415–435, 2011.

[FT05] Stefan Felsner and William T. Trotter. Posets and planar graphs. *Journal of Graph Theory*, 49(4):273–284, 2005.

[GG13] Subir K. Ghosh and Partha P. Goswami. Unsolved problems in visibility graphs of points, segments, and polygons. *ACM Computing Surveys*, 46:22, 2013.

[Gho88] Subir K. Ghosh. On recognizing and characterizing visibility graphs of simple polygons. In *Scandinavian Workshop on Algorithm Theory (SWAT)*, volume 8503 of *LNCS*, pages 96–104. Springer, 1988.

[Gho97] Subir K. Ghosh. On recognizing and characterizing visibility graphs of simple polygons. *Discrete & Computational Geometry*, 17:143–162, 1997.

[GJT76] Michael R. Garey, David S. Johnson, and Robert E. Tarjan. The planar Hamiltonian circuit problem is NP-complete. *SIAM Journal on Computing*, 5:704–714, 1976.

[GKW15] Matt Gibson, Erik Krohn, and Qing Wang. A characterization of visibility graphs for pseudo-polygons. In *European Symposium on Algorithms (ESA)*, volume 9294 of *LNCS*, pages 607–618. Springer, 2015.

[GP80] Jacob E. Goodman and Richard Pollack. On the combinatorial classification of nondegenerate configurations in the plane. *Journal of Combinatorial Theory, Series A*, 29:220–235, 1980.

[GPS90] Jacob E. Goodman, Richard Pollack, and Bernd Sturmfels. The intrinsic spread of a configuration in $\mathbb{R}^d$. *Journal of the American Mathematical Society*, 3:639–651, 1990.

[GR14] Subir K. Ghosh and Bodhayan Roy. Some results on point visibility graphs. In *Workshop on Algorithms and Computing (WALCOM)*, volume 8344 of *LNCS*, pages 163–175. Springer, 2014.

[Grü72] Branko Grünbaum. *Arrangements and Spreads*, volume 10 of *Regional Conference Series in Mathematics*. AMS, 1972.

[Grü03] Branko Grünbaum. *Convex Polytopes*, volume 221 of *Graduate Texts in Mathematics*. Springer-Verlag, 2003.

[GT01] Ashim Garg and Roberto Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM Journal on Computing*, 31:601–625, 2001.

[HATI11] Bjarni V. Halldórsson, Derek Aguiar, Ryan Tarpine, and Sorin Istrail. The Clark phaseable sample size problem: Long-range phasing and loss of heterozygosity in GWAS. *Journal of Computational Biology*, 18:323–333, 2011.

[Hix13] Thomas Hixon. Hook graphs and more: some contributions to geometric graph theory. Master's thesis, TU Berlin, 2013.

[HNZ91] Irith Ben-Arroyo Hartman, Ilan Newman, and Ran Ziv. On grid intersection graphs. *Discrete Mathematics*, 87:41–52, 1991.

[Hof15] Udo Hoffmann. The slope number of segment intersection graphs. In *European Workshop on Computational Geometry (EuroCG)*, pages 109–112, 2015.

[JJK+13] Vít Jelínek, Eva Jelínková, Jan Kratochvíl, Bernard Lidickỳ, Marek Tesař, and Tomáš Vyskočil. The planar slope number of planar partial 3-trees of bounded degree. *Graphs and Combinatorics*, 29:981–1005, 2013.

[KM89] Jan Kratochvíl and Jirí Matoušek. NP-hardness results for intersection graphs. *Commentationes Mathematicae Universitatis Carolinae*, 30:761–773, 1989.

[KM94] Jan Kratochvíl and Jirí Matoušek. Intersection graphs of segments. *Journal of Combinatorial Theory, Series B*, 62:289–315, 1994.

[KMN05] Matthew J. Katz, Joseph S.B. Mitchell, and Yuval Nir. Orthogonal segment stabbing. *Computational Geometry*, 30:197–205, 2005.

[KMW14] Kolja Knauer, Piotr Micek, and Bartosz Walczak. Outerplanar graph drawings with few slopes. *Computational Geometry*, 47:614–624, 2014.

[KN98] Alexandr V. Kostochka and Jaroslav Nešetřil. Coloring relatives of intervals on the plane, I: chromatic number versus girth. *European Journal of Combinatorics*, 19:103–110, 1998.

[KPP13] Balázs Keszegh, János Pach, and Dömötör Pálvölgyi. Drawing planar graphs of bounded degree with few slopes. *SIAM Journal on Discrete Mathematics*, 27:1171–1183, 2013.

[KPW05] Jan Kára, Attila Pór, and David R. Wood. On the chromatic number of the visibility graph of a set of points in the plane. *Discrete & Computational Geometry*, 34:497–506, 2005.

[Kra94] Jan Kratochvíl. A special planar satisfiability problem and a consequence of its NP-completeness. *Discrete Applied Mathematics*, 52:233–252, 1994.

[Kyn11] Jan Kynčl. Simple realizability of complete abstract topological graphs in P. *Discrete & Computational Geometry*, 45:383–399, 2011.

[Mat70] Yuri V. Matiyasevich. Enumerable sets are diophantine. *Doklady Akademii Nauk SSSR*, 191:279–282, 1970.

[Mat09] Jiří Matoušek. Blocking visibility for points in general position. *Discrete & Computational Geometry*, 42:219–223, 2009.

[Mat14] Jiří Matoušek. Intersection graphs of segments and $\exists\mathbb{R}$. *arXiv:1406.2636*, 2014.

[Mnë88] Nicolai E. Mnëv. The universality theorems on the classification problem of configuration varieties and convex polytopes varieties. In *Topology and Geometry – Rohlin Seminar*, LNM, pages 527–543. Springer, 1988.

[MP13] Irina Mustaţă and Martin Pergel. Unit grid intersection graphs: Recognition and properties. *arXiv:1306.1855*, 2013.

[OS97] Joseph O'Rourke and Ileana Streinu. Vertex-edge pseudo-visibility graphs: Characterization and recognition. In *Symposium on Computational Geometry (SoCG)*, pages 119–128. ACM, 1997.

[Per81] Raoul Perrin. Sur le problème des aspects. *Bulletin de la Société Mathématique de France*, 10:103–127, 1881.

[Pil14] Alexander Pilz. *On the complexity of problems on order types and geometric graphs.* PhD thesis, TU Graz, 2014.

[Poo09] Bjorn Poonen. Characterizing integers among rational numbers with a universal-existential formula. *American Journal of Mathematics*, 131:675–682, 2009.

[PP06] János Pach and Dömötör Pálvölgyi. Bounded-degree graphs can have arbitrarily large slope numbers. *European Journal of Combinatorics*, 13:N1, 2006.

[PPVW12] Michael S. Payne, Attila Pór, Pavel Valtr, and David R. Wood. On the connectivity of visibility graphs. *Discrete & Computational Geometry*, 48:669–681, 2012.

[PW10] Attila Pór and David R. Wood. On visibility and blockers. *Journal of Coputational Geometry*, 1:29–40, 2010.

[RG95] Jürgen Richter-Gebert. Mnëv's universality theorem revisited. In *Séminaire Lotharingien de Combinatoire*, volume 34, pages 211–225, 1995.

[RG11] Jürgen Richter-Gebert. *Perspectives on projective geometry: a guided tour through real and complex geometry.* Springer, 2011.

[RGZ95] Jürgen Richter-Gebert and Günter M. Ziegler. Realization spaces of 4-polytopes are universal. *Bulletin of the American Mathematical Society*, 32:403–412, 1995.

[Roy14] Bodhayan Roy. Point visibility graph recognition is NP-hard. *arXiv:1406.2428*, 2014.

[Sch89] Walter Schnyder. Planar graphs and poset dimension. *Order*, 5:323–343, 1989.

[Sch09] Marcus Schaefer. Complexity of some geometric and topological problems. In *International Symposium on Graph Drawing (GD)*, volume 5849 of *LNCS*, pages 334–344. Springer, 2009.

[Sch12] Marcus Schaefer. Realizability of graphs and linkages. In János Pach, editor, *Thirty Essays on Geometric Graph Theory.* Springer, 2012.

[Sho91] Peter W. Shor. Stretchability of pseudolines is NP-hard. *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, 4:531–554, 1991.

[Sin66] Frank W. Sinden. Topology of thin film RC circuits. *Bell System Technical Journal*, 45:1639–1662, 1966.

[SSŠ03] Marcus Schaefer, Eric Sedgwick, and Daniel Štefankovič. Recognizing string graphs in NP. *Journal of Computer and System Sciences*, 67:365–380, 2003.

[ST15] Maurice Soto and Christopher Thraves. p-box: A new graph model. *Discrete Mathematics and Theoretical Computer Science*, 17:179–186, 2015.

[Sta47] Karl Georg Christian Staudt. *Geometrie der Lage.* 1847.

[Str05] Ileana Streinu. Non-stretchable pseudo-visibility graphs. *Computational Geometry*, 31:195–206, 2005.

[STS11] Anish Man Singh Shrestha, Asahi Takaoka, and Tayu Satoshi. On two problems of nano-PLA design. *IEICE transactions on information and systems*, 94:35–41, 2011.

[Stu87] Bernd Sturmfels. On the decidability of diophantine problems in combinatorial geometry. *Bulletin of the American Mathematical Society*, 17:121–124, 1987.

[STU10] Anish Man Singh Shrestha, Satoshi Tayu, and Shuichi Ueno. On orthogonal ray graphs. *Discrete Applied Mathematics*, 158:1650–1659, 2010.

[TMS76] William T. Trotter, John I. Moore, and David P. Sumner. The dimension of a comparability graph. *Proceedings of the American Mathematical Society*, 60:35–38, 1976.

[Tro92] William T. Trotter. *Combinatorics and Partially Ordered Sets: Dimension Theory*. Johns Hopkins Series in the Mathematical Sciences. The Johns Hopkins University Press, 1992.

[TS11] Claudio Telha and José Soto. Jump number of two-directional orthogonal ray graphs. In *Integer Programming and Combinatorial Optimization (IPCO)*, volume 6655 of *LNCS*, pages 389–403. Springer, 2011.

[Wid82] Avi Widgerson. The complexity of the Hamiltonian circuit problem for maximal planar graphs. Technical report, Princeton University, Electrical Engineering and Computer Science Department, 1982.

[Yan89] Mihalis Yannakakis. Embedding planar graphs in four pages. *Journal of Computer and System Sciences*, 38:36–67, 1989.

Abstract

In this thesis, we consider several aspects of representations of graphs in the plane. We consider mainly intersection and visibility representations of graphs. In both kind of representations, the vertices are represented by sets in the plane. Intersection representations represent the edges by the intersection of two sets. In visibility graphs, an edge corresponds to a visibility between the sets.

In the first part, we consider connections between representations of bipartite segment intersection graphs and their order dimension. In Chapter 1, we show that the order dimension of grid intersection graphs, the intersection graphs of horizontal and vertical segments, is at most four. We use this observation to study the containment relation of many subclasses of grid intersection graphs. We generalize the observation on the order dimension of grid intersection graphs, by showing that the order dimension of bipartite segment intersection graphs is at most linear in the number of slopes that is used in a representation. This leads to the study of the slope number of segment intersection graphs, the minimal number of slopes that is sufficient to represent an intersection graph. In Chapter 2, we show that the slope number of segment intersection graphs is NP-hard to compute, and that it behaves "non-continuously", i.e., it may drop, from a linear number in the number of segments down to two, upon the removal of a single vertex.

The proofs in the first part are based on combinatorial properties of the graphs. In the second part, we deal with the realizability problem and the complexity class *existential theory of the reals* ($\exists\mathbb{R}$). Many geometric representation problems for graphs are complete in $\exists\mathbb{R}$, for example the recognition of segment intersection graphs. We show that computing the slope number of segment intersection graphs (Chapter 4) and the recognition of point visibility graphs (Chapter 5) is complete in $\exists\mathbb{R}$. Both proofs are based on the $\exists\mathbb{R}$-hardness of the realizability of circular sequences, the order of slopes of lines that are spanned by a point sets, which we present in Chapter 3. We finish in Chapter 6, by showing that determining the minimum number of slopes that is sufficient for a planar straight line drawing of a graph, the planar slope number of a graph, is also complete in $\exists\mathbb{R}$. We discuss consequences the $\exists\mathbb{R}$-hardness for properties of the representation.

Zusammenfassung

In dieser Arbeit beschäftigen wir uns mit verschiedenen Darstellungsarten von Graphen in der Ebene. Hauptsächlich behandeln wir dabei Schnitt- und Sichtbarkeitsdarstellungen. Bei beiden Darstellungsarten werden die Knoten durch Mengen in der Ebene dargestellt. Bei Schnittgraphen existiert eine Kante zwischen zwei Knoten wenn die zugehörigen Mengen sich schneiden. Kanten in Sichtbarkeitsgraphen existieren, wenn Mengen zweier Knoten sich *sehen* können, also eine Verbindungsstrecke zwischen zwei Punkten der Mengen existiert, die nicht von einer anderen Menge geschnitten wird.

Im ersten Teil dieser Arbeit untersuchen wir einen Zusammenhang zwischen bipartiten Schnittgraphen von Segmenten und der Ordnungsdimension der zugehörigen partiellen Ordnung. In Kapitel 1 zeigen wir, dass die Ordnungsdimension von Gitterschnittgraphen, den Schnittgraphen horizontaler und vertikaler Segmente, höchstens vier ist. Wir verwenden diese Beobachtung um das Verhältnis vieler Unterklassen von Gitterschnittgraphen zu bestimmen. Wir verallgemeinern die Beobachtung über die Dimension von Gitterschnittgraphen und zeigen, dass die Ordnungsdimension von bipartiten Segmentschnittgraphen unbegrenzt ist, sie jedoch höchstens linear in der Anzahl der Steigungen ist. Dies führt zu einer Untersuchung der Steigungszahl von Segmentschnittgraphen, der minimalen Anzahl von Steigungen, die für die Darstellung eines Segmentschnittgraphen benötigt wird. In Kapitel 2 zeigen wir, dass das Berechnen der Steigungszahl NP-schwer ist. Außerdem zeigen wir, dass die Steigungszahl sich "unstetig" verhält: Das Entfernen eines Knoten von einem Graphen mit linearer Steigungszahl kann zu einem Gitterschnittgraphen führen.

Während die Beweise im ersten Teil eher kombinatorisch sind, betrachten wir im zweiten Teil Realisierbarkeitsprobleme und die Komplexitätsklasse *existential theory of the reals* ($\exists\mathbb{R}$). Viele geometrische Probleme sind $\exists\mathbb{R}$-vollständig, wie zum Beispiel die Erkennung von Segmentschnittgraphen. Wir zeigen, dass die Berechnung der Steigungszahl von Segmentschnittgraphen (Kapitel 4) und die Erkennung von Sichtbarkeitsgraphen von Punktmengen (Kapitel 5) $\exists\mathbb{R}$-schwer ist. Beide Beweise basieren auf der $\exists\mathbb{R}$-Schwere der Realisierbarkeit von *circular sequences*, der Reihenfolge der Steigungen der durch eine Punktmenge aufgespannter Menge von Geraden. Dieser Beweis basiert auf *Mnëvs universality theorem* und wird in Kapitel 3 präsentiert. Zum Abschluss zeigen wir, dass die Berechnung der minimalen Anzahl von Steigungen in einer planaren, geradlinigen Zeichnung, der *planaren Steigunszahl* eines planaren Graphen, $\exists\mathbb{R}$-schwer ist.